AURAVANA PROJECT

PROJECT FOR A COMMUNITY-TYPE SOCIETY

The Project Plan

SSS-PP-003 | May 2024

SOCIETAL SPECIFICATION STANDARD

auravana.org

THE AURAVANA PROJECT

SOCIETAL SPECIFICATION STANDARD

PROJECT PLAN

Document Reference Identifier: SSS-PP-003

Date of Document Distribution: May 2024

auravana.org

To cite this publication:

- *The Project Plan.* (2024). Auravana Project, Societal Specification Standard, SSS-PP-003. [auravana.org]

To cite an article in this publication *(authors and article title will change)*:

- Grant, T.A. (20234). *The Project Plan Overview*. The Project Plan. Auravana Project, Societal Specification Standard, SSS-PP-003. [auravana.org]

ISBN:

auravana.org

GREETINGS

In an effort to provide the greatest possible clarity and value the Auravana Project has formatted the system for the proposed society (of the type, 'community') **into a series of standard publications. Each standard is both a component of the total, unified system, as well as intended to be a basis for deep reflective consideration of one's own community, or lack thereof. These formal standards are "living" in that they are continually edited and updated as new information becomes available; the society is not ever established, its design and situational operation exists in an emergent state, for it evolves, as we evolve, necessarily for our survival and flourishing.**

Together, the standards represent a replicable, scalable, and comprehensively "useful" model for the design of a society where all individual human requirements are mutually and optimally fulfilled.

The information contained within these standards represent a potential solution to the issues universally plaguing humankind, and could possibly bring about one of the greatest revolutions in living and learning in our modern time. Change on the scale that is needed can only be realized when people see and experience a better way. The purpose of the Auravana Project is to design, to create, and to sustain a more fulfilling life experience for everyone, by facilitating the realization of a better way of living.

Cooperation and learning are an integral part of what it means to be a conscious individual human. A community-type societal environment has been designed to nurture and support the understanding and experience of this valuable orientation.

The design for a community-type society provides an entirely different way of looking at the nature of life, learning, work, and human interaction. These societal standards seek to maintain an essential alignment with humankind's evolving understandings of itself, combining the world of which humans are a regenerative part, with, the optimal that can be realized for all of humanity, given what is known.

The general vision for this form of society is an urgent one considering the myriad of perceptible global societal crises. Together, we can create the next generation of regenerative and fulfilling living environments. Together, we can create a global societal-level community.

THE UNIFIED SOCIETAL SYSTEM: SOCIETAL PROJECT PLAN

This publication is one of six representing the proposed standard operation of a type of society given the category name, 'community' (a community-type society). This document is a project plan for the societal system.

Every society is composed of a set of core systems. Different types of societies have different internal compositions of these systems. The composition of these systems determines the type of society. The type of society described by the Auravana Project societal standard is a, community-type society. The standard is a composition of sub-system standards. The Auravana societal standard may be used to construct and duplicate community at the global level.

For any given society, there are four primary societal sub-systems. Each of these sub-systems can be specified and standardized (described and explained); each sub-system is a standard within a whole societal specification standard. The first four primary standards of the six total standards are: a Social System; a Decision System; a Material System; and a Lifestyle System. Each standard is given the name of its information system. The fifth publication is a Project Plan, and the sixth is an Overview of the whole societal system. Together, these standards are used to classify information about society, identify current and potential configurations, and operate an actual configuration. Because of the size of some of these standards, they may be split into two or more publications.

Essential figures and tables related to this standard exist beyond what is shown in this document.

Figures and tables on the website are named according to their placement in the standard.

- Those figures that could not be accommodated here are readily accessible in their full size, and if applicable, in color, on the Auravana Project's website [auravana.org/standards/figures].
- Those tables that are too large to include in this document are referenced with each standard on the Auravana Project's website [auravana.org/standards].

Articles

Article Section Headings

Contents

List of figures

This is the list of figures within this document.
There are more figures associated with this standard than are identified in this document; those figures that could not fit are freely available through auravana.org, in full size, and if applicable, color.

List of tables

This is the list of tables within this document.
There are more tables associated with this standard than are identified in this document; those tables that could not fit are freely available via the project's website.

Document Revision History

A.k.a., Version history, change log.

This document is updated as new information becomes available.
The following information is used to control and track modifications (transformations, changes) to this document.

VERSION	REVISION DATE	SUMMARY (DESCRIPTION)
003	May 2024	The structure of this document has not changed. Significant changes have been made throughout this document. The Opening and Timing articles are newly added to this document. Citations have been improved throughout and are now at APA 7th generation.

GENERATION ON	NAME	CONTACT DETAIL
May 2024	Travis A. Grant	trvsgrant@gmail.com

The Project Plan Overview

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: project plan, societal plan, societal project, planned societal operation, societal project plan, societal project planning, societal project management, community project plan

Abstract

This publication is the Project Plan for a community-type society. A societal-level project plan describes the organized thinking and execution of a societal environment; the societal structuring of community. This project plan identifies humanity's project to create a global community-type society for the fulfillment of that which everyone has mutually in common. This is a planned project for a configuration of society that may be tested in its results at optimally meeting all human life requirements at the global scale. This is a planning and work proposal for an open-source, societal-level project. This document describes and explains a unified approach to actions and results that is likely, given what is known and accessible, to improve all of humanity. This is the plan for societal navigation that specifies an approach, direction, and execution to socio-technical life. The project plan has three core sections: (1) Approach to project execution, (2) Direction of project execution, and (3) Execution of project execution. The standard details the complete, plannable information set for the society's operation, including its approach to action, its direction of action, and its execution and adaptation of action. Herein, these concepts, their relationships and understandings, are defined and modeled. Discursive reasoning is provided for this specific configuration of a project plan, as opposed to the selection and encoding of other configurations. A project plan provides for the formalized project-based development operation of a society, organized in time and with available resources, coordinated to become a societal service system for human fulfillment and ecological well-being.

Graphical Abstract

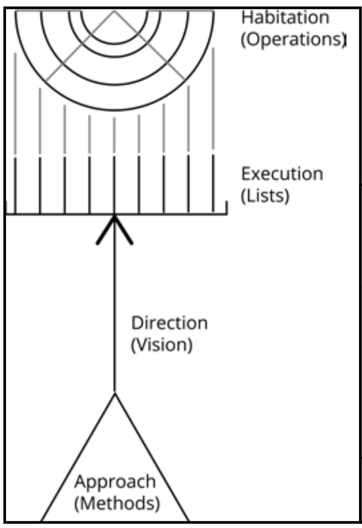
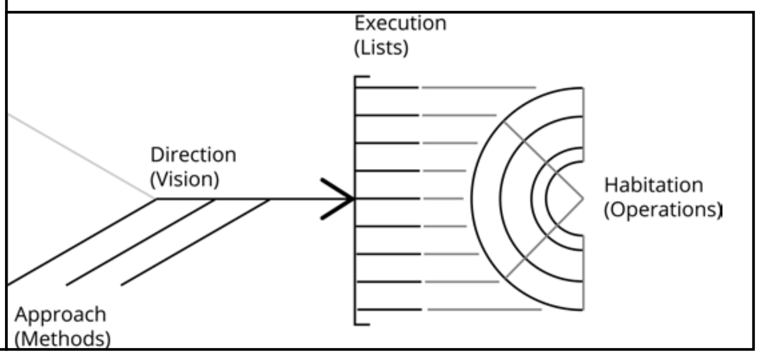


Figure 1. *Together, humanity can take a common direction, using a common and iterative approach to the execution of project lists in order to construct and sustain a habitat service operation for humanity's mutual fulfillment and flourishing.*



1 How to read this document

A.k.a., Document guidance.

This organisation of information is the documented proposal for a unified 'Project Plan' of Action that every contributor to the project informs and executes. This document is an information (reporting) interface to identify what encompasses and encapsulates the whole project. In application, this document identifies the logical flow of information necessary for developing, duplicating, and operating a societal-level organization.

In the whole context for that which is being proposed, this document is the Project Plan for bringing a specified type of society into existence. The specification for the whole society is subdivided into four primary systems, each of which is its own specification separate from (but, also interrelated to) the Project Plan. This project plan may be viewed as the fifth specification, a high-level coordination specification for the core societal sub-systems (Read: social, decision, lifestyle, and material).

Because the type of society being proposed by this project is representable as a unified information system, all of the specifications (project plan included) are interrelated and iterated together. The unified nature of this societal system means that in order to fully comprehend its designed operation and reasoning for its selection, the whole system (Read: all specifications) must be understood. In other words, to fully understand any one of the societal sub-system specifications, all of the societal specifications must be understood together.

> **NOTE:** *For those individuals among early 21st century society who are more educated on what is, and what is possible, a comprehensive understanding of this society may come more easily, than it may come to individuals who steeped in limiting beliefs that mask what is, and what is possible.*

1.1 *Document section hierarchy*

This document is separated at a high-level into three sections representing the different principal elements of planned navigation (forming, a planned navigational system for coordinating an informational-material environment together):

1. **A purpose -** a reasonable cause for action/inaction.
   A. *Why* are we here?
2. **A direction** - intention, target, goal, success, result, destination, outcome, purpose.
   A. *Where* are we going; what is to occur? What is required to be directly created by the project? What are the intended results of the project?
      1. Human life requirements.
      2. Ecological life requirements.
      3. Habitat service system requirements.
3. **An approach** - methodology, method, strategy, philosophy, structure, knowledge framework.
   A. *How* are we going to get to where we are going? What is the approach taken by the project? How is the project work to be done?
      1. Project approach.
         i. Projecting.
         ii. Engineering.
         iii. Deciding.
         iv. Standardizing.
         v. Opening.
4. **An execution** - the plans and lists to be executed in order to achieve the projects stated objectives. Project plans and project lists, activities with time, schedule, matrix of integratable lists, computations/actions, inquiries/surveys, monitoring, and controlling, over time.
   A. Schedule and do the work.
   B. *What* is to be done, and *when* and *where*, to complete the project?
      1. What communications must occur?
      2. What decisions must occur?
      3. What plan(s) must be developed, and lists must be executed?
      4. What surveys and accounts of information and resource are available?
      5. What are the deliverables?
      6. What are the resources?
      7. What teams and roles are available?
      8. What actions are available?

Table 1. Overview > How To*: The three sections of the project plan.*

Purpose	*reason to be present*
Direction	the planned direction, *proposed objectives*
Approach	to planning, *proposed methods*
Execution	the planned execution, *proposed execution*

> **NOTE:** *Without a careful, planned approach to execution, including a statement of direction, [strategic] goals cannot be predictably attained.*

1.1.1 Sub-sectioning

It is possible to separate the project plan into three core information views/formats based on the usage of information (but, this document does not do so):

1. **The project-engineering approach** - project planning and systems engineering definition and methods selection. This information is used to develop and operate a societal system.
2. **The project plan** - the currently integrated, and possibly executing, information state of the project. This is, at least, expressed as a series of lists in a database, which are combined in time as an 'event'.

This information is used to schedule delivery of a societal system.

3. **The project reasoning (a.k.a., project philosophy)** - the logical reasoning for the selection of the approach to the plan and the solution (project-engineering methodology and the project plan. This necessitates logical, factual argumentation and integration, and a systems science approach. This information is used to understand the societal system.

1.2 Reading by intelligent agents

It is expected that this societal system specification will be readable to, and read by, "artificially intelligent" decision support systems that are capable of, and designed to, improve themselves and the world around them for the benefit of all of humankind. This document may be read by these entities and used to re-configure themselves toward the uncertain aim of providing decision support for the highest fulfillment of all of humanity.

1.3 What is this document?

A.k.a., What is the purpose of this document?

This document is the formalized 'project' operation of a society, organized through an intentional conceptual definition, structurable in time and with available resources, into a societal service system for human fulfillment and ecological well-being. This document describes the formation of a society that is unified, explainable, plannable, optimal, and lived within by a population of fulfilled human beings who are expressing their highest potentials as embodied consciousness. This document is the project plan document. To anyone potentially affected by this societal project, this is a proposal (Read: a workable plan).

This document represents the project-engineering conceptual information set, which sets out the purpose of one half of the whole societal information set (the other half are the societal sub-system specifications).

The purpose of this document is to set for all contributors a project plan of unified action:

1. A project is a framework for wok done on a cyclical (e.g., daily) basis.
2. A plan is a socio-technical, action-oriented integration of information:
   A. A social-plan is a unified model of action that allows cooperation to work.
   B. A technical-plan is the information required to do or build something with complexity.
3. Project-level information is sub-composed of the conceptions required for logically computing time and/or positional information [on the presence or not] of a geometrically physicalized, solid shape, commonly known as a resource.
4. Engineering-level information is sub-composed of scientific-factual observable knowledge and procedures of how to change (Read: programmatically modify) a physicalizable environment in an intentionally fulfilling manner.

2 Project identification

The following items may be used to identify this societal development project:

1. **Project Title:** Auravana Project
2. **Project Sub-Title:** Project to develop and operate a community-type society.
3. **Project Statement:** The emergence of community.
4. **Project Website:** https://auravana.org

2.1 Alternative project titles

Short sub-title classification:

- The emergence of a community-type societal system through the development and operation of a societal specification standard.

Long-form sub-title classification:

- The emergence of a community-type configuration of information and material at the level of the global population, at the level of a planetary society.

Market-State societal-type classification:

- The emergence of a marketless and Stateless society; a true *family-type society.* A society that works well, like a family works well together, without trade, money, competition for scarece resources, coercion, punishment-type governance, or power-over-other type relationships.

The type of society proposed by this project has multiple common names, among which the most widely used names include:

1. **Community-type society**
   - *'Society' is the highest order of human organization, and 'community' is the natural language name for the type of planned society; The Auravana Project uses this name to refer to the proposed society.*
2. **Resource-based economy (RBE)**
   - *A 'resource' is the foundation of an economic system and the view that resources are common heritage maintains the systems equity; The Venus Project uses this name to refer to the proposed society.*
3. **Natural-law/resource-based economy (NL/RBE)**
   - *'Natural-laws' are the discoverable regular principles of reality; The Zeitgeist Movement uses this name to refer to the proposed society.*
4. **Access-based economy (access-based society)**
   - *'Access' for humanity is the purpose for the societal system's material existence; Jacque Fresco also called system by this name.*
5. **Commons-based economy (commons-based society)**
   - *A society and economy that functions as shared information and resources in every domain of social and technical activity.*

2.2 Project description

This is a project for:

1. Doing the work of configuration and transformation of society to one that matches resources to human needs and ecological restoration.
   A. The socio-technical engineering of a community-type society.
      1. The intellectual-constructive evolution of the symbiotic biosphere (ICESB) into a global information communications network of materializing habitat service systems (HSS network) designed and operated for the fulfillment of all human need and ecological regeneration.

2.3 Project full description

This project plans the executed design, construction, and experimental operation of a community-type societal system consisting of a fulfilled population of humans, a regenerative ecology, and a network of integrated city systems, as expressed through a unified societal information model, which is structured through a societal systems specifications that coordinated the configuration of real-world resources.

This project describes and explains what is being created as the next iteration of society, and then together, the population migrates into it, and tests it.

2.4 Project standard [call] identifiers

The following items are the call identifiers for the primary systems' documentation of this project:

1. Organizational identifier: Project Auravana
2. Documentation identifier: SSS (Societal Specification Standard)
3. Standard identifier: PP, SO, SS, DS, MS, or LS
4. Sub-standard identifier: none (or, PP-PE, MS-HS)
5. Specification identifier (version): [###]
6. Current revision identifier: [###]
7. For example: SSS-PP-001-173
   A. Societal Specification Standard (SSS)
   B. Project Plan (PP)
   C. Version 001
   D. Revision 173
   E. Note that this is not a sub-standard publication

(for example, not PP-PE or MS-HS)

2.5 Project duration

This is a strategic project to transform planetary society into a community-type configuration, and in concern to project duration it is both:

1. Flexible (with multiple flexible sub-project durations), and
2. Follows a project schedule.

Project Definition

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: project charter, project explanation, project initialization,

Abstract

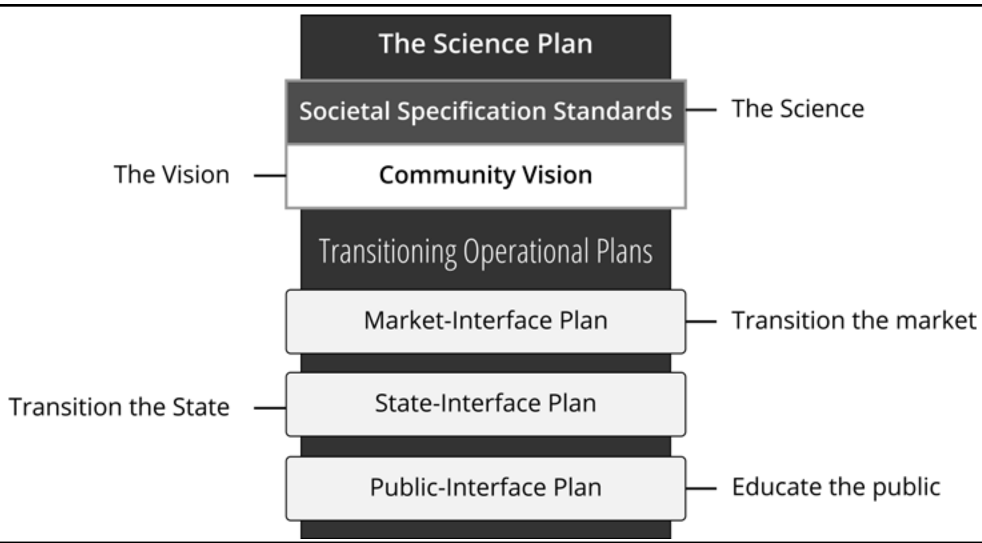
Project Auravana is dedicated to transitioning society to a model that prioritizes community engagement and ecological sustainability, moving away from traditional market economies and state governance. By adopting a comprehensive set of socio-technical standards, the initiative aims to develop integrated habitat systems that address human needs while promoting the regeneration of the planet's ecosystems. Central to this vision is the formulation and implementation of community standards, crucial for establishing a cohesive and sustainable societal framework. The project emphasizes community standards, information and habitat coordination, and educational initiatives to facilitate the shift towards a collaborative, sustainable society.

At its core, Auravana focuses on building a society free from market or state influence, centering on ecological harmony and fulfilling human needs. It involves detailed planning and execution across various areas, including standard setting, habitat services, and education, to transition towards this new societal model. Auravana represents a commitment to a future where community and ecological integrity form the basis of societal organization.

Project Auravana stands as a beacon for purpose-driven organizational projects, aspiring to materialize a society where living conditions are not dictated by economic transactions but are instead aligned with ecological balance and human well-being. Through the meticulous planning and execution of sub-projects – from standards development and habitat operation to educational reform and transitional facilitation – Auravana seeks to illuminate the path toward a society that not only meets the immediate needs of its inhabitants but also ensures the long-term viability of our planet.

Graphical Abstract

Figure 2. *It is necessary to interface the vision and operation of community with market-State operations in order to transition from one configuration of society to another, both of which are represented by different socio-technical standards. This interface plan ought to be based on science; hence, the interface plan may be called the science plan.*



1 Project Auravana proposal

Our objective is to develop, leverage, and implement a societal configuration free of markets and States, and organized around human need fulfillment and ecological regeneration. Our goal is to advance and push forward as rapidly as possible the standards and systems that are enabling this new societal configuration. We seek socio-technical access to human need fulfillment without the necessity for trade or coercion. We work to develop a more fulfilling range of options for human living. We work to bring community-based living conditions to humanity. If our work should succeed, it will have great and positive consequences.

The purpose of Project Auravana is to develop a community configuration of society through a unified set of socio-technical standards that realize a network of integrated habitat systems that optimize human need fulfillment and ecological regeneration. This is a proposal for an Auravana organizational structure with the purpose (in order to) bring into existence 'community' at the societal scale. The objective is to engineer, construct, and operate a community-type configuration of society for humanity and Earth's ecology. Effectively, the mission is to develop a community configuration of society through a unified set of socio-technical standards that realize a network of integrated habitat systems that optimize human need fulfillment and ecological regeneration. Our objective is a duplicable and agreeable habitat service system (and eventual network) where a community population co-lives and co-works for global human flourishing. Community, as a configuration of society, is a vision to which current society shall be transitioned. This is a for-purpose (intentional) organizational project to bring into existence a community-type society by moving people and resources from the market-State into a community configuration of society. A market-interface is necessary to achieve this purpose; because the type of society being transitioned away from includes a market. State- and public-interfaces are also required.

In order to achieve this vision, there are three primary sub-projects (inclusive of accountabilities and requirements):

1. **A [community] standards setting organization:** The standards development organization exists to develop a community-type societal standard that explains, and conveys knowledge and agreement of a community configuration of society.
2. **A habitat residency and service operation:** The habitat service organization exists to operate a network of habitats where community residents have their human needs fulfilled (as a service).
3. **A university education:** The education organization exists to facilitate life-long learning about community through descriptions, explanations, and skills acquisition -- to understand, to adapt to, and to contribute to community.
4. **A transition operation:** The transition operation exists to inform people of the possibility of living in a community-type society and to facilitate the transitioning of people and resources from the market-State into community.

A standards setting organization is essential, because for all socio-technical societies, standards create the environment). Standards are master information sets, containing mandatory requirements, preferred procedures, or guidelines for users and contributors. Standards projects involve either new standards, revisions to existing standards, or removal of standards. Standards are living, in that they are continuously updated and formally published at some appropriate interval or phase. Working groups develop standards. Working groups are comprised of individual experts, and representatives from entities (such as corporations, government agencies or academic institutions) for transition operations. Standards are our common understanding of community. Standards projects are started when there is a need for an idea or concept to be standardized. There is, and always has been, a need for global human flourishing. Those participating in working groups have strong technical knowledge and expertise in the subject matter of the standard project. Project Auravana is accountable to the public at large to provide community-type societal standards to meet the worldwide needs of humanity. Standards benefit the public, in part, by advancing cooperation and the conception of community.

A habitat network, and residency therein, is essential, because for all socio-technical societies, people live in physical habitats where the conditions for flourishing are met (or, go unmet). Habitats are living material service locations, containing physical resources, contribution operations, and users. Habitat projects involve new constructions, maintenance, production and resource re-cycling operations. Habitats are operated to a master plan, and aesthetically re-constructed by means of local master habitat planning. Habitat teams are comprised of individual experts, and representatives from entities (such as corporations, government agencies or academic institutions) for transition operations. Habitats are the material configuration of community. Those participating in habitat teams have strong social and/or technical knowledge and skills in the subject matter of the habitat project. Project Auravana is accountable to the public at large to provide community-type societal services and products to meet the worldwide needs of humanity. Habitats benefit the public, in part, by providing free life, technology, and exploratory support services.

The education (university) organization exists to facilitate life-long learning about community through descriptions, explanations (knowledge) and skills acquisition -- to understand, to adapt, and to contribute to community. Education and orientation are significant processes that will occur during the transition phase, and while living and growing an operational community network of habitats. Education brings clarity and ability, and combined with what is known on the planet now, it is possible to conceive of and to operate a society without trade (the market), coercion (the State), and access-class competition (the orientation).

A transition operation is essential, because the current configuration of society on the planet in the early 21st century is not representative of community [at the societal scale]. In other words, the socio-technical conditions of society present in the early 21st century must be changed to those of community. The transition is to create the environmental conditions for human flourishing by facilitating human need fulfillment and well-being, and by de-coupling the market and the State from humanity. Transition teams work to transition from one configuration of society to another; namely, from the market-State, in which there is a global market (trade) and many localized democractic and dictatorial States (coercion), to one of community. In community, standards inform the operation of habitat services that meet global human need fulfillment requirements without trade or coercion. Transition projects involve the market, the State, and public relation operations. Market-State conditions have been creating systemically harmful goals for centuries. A transition to community [at the societal scale] requires the creation of information standards and physical habitats that generate systemically beneficial conditions for human flourishing. Transition operations benefit the public, in part, by moving people and resources into a community configuration of society. Additionally, education and orientation are significant processes that occur during transition.

1.1 The purpose and goals of a community-type society

> **INSIGHT:** *Purpose is always found in the service of a larger whole.*

The type of society detailed herein may be otherwise known as a 'community of purpose' - a society founded upon and directed toward a commonly held purpose. Because the purpose is directed at the evolvement of the whole, it is similarly embraced by all individuals sharing in the Community. The term "embrace" in the prior sentence could be replaced by: accepted; acknowledge; intended; reasoned; evidenced; explored, experienced; chosen; participated with; or even, actualized -- it represents a recognition that there exists a common direction of intention for social organization, a common purpose in everyone's life that may be used to organize society.

At a basic level, the Community exists as a set of similar social decisions, social structures, and social interrelationships (i.e., connections) that support individuals in developing toward their highest potential through the fulfillment of their needs and the facilitated expression of their natural desire to learn about themselves and the world (i.e., to advance themselves). The Community represents an intentional evolutionary direction through stable human fulfillment and engaged exploration. Herein, 'community' is a social organizational vehicle for developing human potential and facilitating human fulfillment.

1.1.1 The purpose of flourishing is to live to one's highest potential whoever one so chooses

The following statements (*below*) represent humanity's shared purpose; it is the purpose for a community-type society. This purpose directs and motivates individual lives toward empowering and universal human progress. It is a unifying growth-orientation and a direction that has deep meaning throughout an individual's life and the life of the community.

> **The purpose statement:** *To continuously and consciously evolve toward a higher potential expression for oneself and all others through resilient adaptation to a higher potential dynamic of experiential life existence.*

Flourishing means something living to its highest potential. The term "highest potential" is defined as the greatest possible expression of a being's fulfillment, its capabilities, creativity, well-being (or "flourishing"), happiness, and intellect, while in a state of open and active intrinsic-engagement, and imbued with the deepest appreciation and compassion for the evolving and developing whole. Development toward a higher potential is observed [in part] as compassion, connection, contribution, self-growth & self-expression, and the desire and energy to pursue one's deepest passions - a resiliently adaptive cycle of 'flow'. It is an intentional evolutionary direction - a direction of emergence into greater coherency, consistency, and continuity. There is no known absolute point wherein someone has reached their highest potential; the state is "revealed" through its emergence. Resilience refers to the experience of stress, and thereafter, rapid recovery (rapid recover from setbacks). To respond and recover from stress (versus reactive suffering) is "resilient action".

The aim of a community-type society is to unfold the fullest possible life potential of every individual consciousness through intentional organization in a continuum of balance with nature. It is a state-of-being present and in alignment with one's full potential self, bringing one into coherence with all reality.

The purpose of mutual societal fulfillment must be to deliver a framework whereby every human has ready access to all basic necessities – clean air, water, food, housing, sanitation, sustainable housing, aesthetic surroundings, medical care, and energy – whilst simultaneously remaining below the carrying capacity of the natural environment both locally and globally.

> **INSIGHT:** *A system that can adapt, can likely survive; a system that can evolve, can likely thrive.*

1.1.2 The goals

The following goals (or intrinsic aspirations) maintain a social direction toward common individual human need fulfillment:

1. To support each other in progressing toward a higher potential while developing self-knowledge and a deeper understanding and appreciation of nature and the nature of the world.
2. To continuously improve the effectiveness and efficiency of the community's systems in fulfilling the unifying and life-long needs of everyone.
3. To continuously improve the means, methods, and approach by which humanity discovers, understands, learns, communicates, and acts.
4. To exist in a state of regenerative abundance with the lifeground while maximizing the intelligent use of resources and caretaking the environment (i.e., to sustain material resiliency).
5. To arrive at decisions based upon a commonly "living" purpose, set of needs & values, and approach, and hence, a similar set of understood relationships for arriving at decisions and actions. Note that these similarities are necessary for the effective functioning of [human] social relationships wherein a community is a set of similar relationships.
6. To exist in a state of appreciation and compassion for the self and the evolving whole.
7. To continuously improve access abundance through a stable 'bio-psycho-social community', a community of need fulfillment, serving as the liberating foundation from which individuals pursue their highest development and apply/contribute (participate in) everyone's evolving potential.

1.2 Purposeful vision

> *A.k.a., The vision, the purpose, the intention, the goal.*

The goal is to live in community with a planetary population that is also living in community, where all humans flourish and the ecology is restored. What is possible, given what is known, is global human flourishing. A societal-level community may be socio-technically engineered into existence, and is based at a fundamental level on concept modeling, data collection, analysis, and systems synthesis. This information is then used to optimize [global] human need fulfillment. A unified community fulfillment system is one representational of humanity's true potential at the societal scale. In part, community means global cooperation, and open source and public works are unequivocally an appropriate method for sustaining and optimizing global coordination. This organizations aligns people to their highest potential for fulfillment, flow, and ecological regeneration.

In order to engineer a more fulfilling society and improve human well-being, it is essential to understand that society [as an information system] has four axiomatic sub-systems: a social, decision, lifestyle, and material system. All data about the real-world, how it is and what it could be, fit into these four systems. Through cooperation and intelligence it is possible to configure the internals of each system appropriately to express a type of society with mutual and optimal human fulfillment. Together, these four fundamental systems represent a unified information system that supports all human life on the planet. These information systems, expressed as standards, are developed by working groups.

The vision is an information system through which is materialized a network of habitats. In general, in community, we live in integrated total habitat/city systems. The word habitat is just a synonym for city, village, etc. Some people live in habitats with very high density populations while others live in more rural habitats with very low density populations. Both types of habitats have connected pasture land where an abundance of food, fuel, and fiber is produced, and soil is restored. Between these habitats there is mostly wild nature. However, there is also mining and heavy production (heavy industry). Light production occurs within the habitats themselves. Herein, the habitat network, based upon community standards, becomes a platform for free (universal) goods and services, starting with basic and progressing therefrom.

Individual habitats are organized/based on three primary [needed] service support systems: Life, Technology, and Exploratory. Life is life support, technology is technology support, and exploratory is exploratory support. In community, people contribute to the habitat service team. Habitat teams conduct socio-technical operations in the habitat (local & global). These operations follow [master] plans that ensure human requirements for resource fulfillment are sustained.

Each habitat is customized to the needs and preferences of those people living in that habitat, given optimized access to common heritage resource. Each habitat is developed and operated through master plans. Through master planning, new habitats can be safely and optimally constructed, and old villages/cities may be updated to community standards. In habitats, people live in flow throughout all three phases of life, the education, contribution, and leisure. In community, flow is fostered, humans are fulfilled, and the planetary ecology is regenerated.

The community vision is:

1. A unified information system that mirrors the real-world and optimizes human understanding, intelligence, and operations therein.
   A. Where all knowledge, resource accounts, and productions are open source.
   B. Where there is appropriate projects coordination.
   C. Where there is sufficient integration of information to produce optimal and expected outcomes.
   D. Where community values are shared so that there is sense of harmony and duty to one another.
   E. Where resources, knowledge, and skills are accounted to ensure a efficient economy that effectively meets human needs.
2. A network of residential habitats.
   A. Where human need fulfillment is optimized given the common heritage resources and ecological services available.
   B. Where we share free access to all habitat services.
   C. Where we support habitat services and operate our habitats through contribution.
   D. Where we live through the three phases of life (education, contribution, and leisure) in an optimized state of harmony and flow.
   E. Where habitats are engineered through master plans, updated through some regular protocol and customized by local habitat resident.
   F. Where individuals agree to reside, travel, and contribute within a community set of local habitat residency, visitor, and contributor agreements.

What do we all need access to:

1. Resources that will be re-configured to meet human need fulfillment requirements for life, technology, and exploratory services, optimally and harmoniously.
2. A healthy habitat where individuals experience well-being and optimized flow throughout their lives.
3. A home in a habitat.
4. A network of visitable habitats that share resources and information.
5. An education about community.
6. An opportunity to contribute to community.
7. The opportunity to live a life of education, contribution, and leisure among community.
8. One's own [residency] agreement profile.
9. All others [residency] agreement profiles.
10. The ability to propose and vote on the transformation, addition, or subtraction of [residency] agreements/by-laws.
11. The ability to receive the updated version of the [residency] agreements and update one's profile.

The system's design shall express two additional parameters:

1. The population of users ought to be able to scale up to planetary scale without harmful integrations.
2. It will be exclusive to those who have chosen residency in community (as a configuration of planetary human society). In the sense of a token organization, it will be exclusive to those who own tokens.

1.2.1 An intelligent approach

There are three basic steps that must be followed to realize community:

1. Have the perception to see things as they truly are. Understand the true nature of the challenge. Visualize all of the parts together; do not focus on just one of the parts.
2. Given that understanding, we need to understand ourselves better. What are our true capabilities given the true nature of the challenge. What are our limitations, what can we do, what can't we do.
3. Develop a planned response with steps and stages to resolve the challenge.

1.2.1.1 Movement between global, local, and regional habitats

An intelligent approach is one that moves the economic system of society from one of significantly local and global to one of a more regional user distribution. Wherein, everything accessed (or, purchased) and used is made within a 500km radius. Each region has factories, technologies, and mining. Each region essentially becomes self-sufficient, to the greatest extent possible, and relies on global coordination of resource fulfillment. An intelligent approach is to re-use and re-purpose what we have more effectively. The relationship between energy and minerals must become very carefully managed. In order to carefully manage resources, the whole system, from the global network, to the regional network, to the local network, must be integrated.

1.3 During transition, what is being funded?

Together, we are funding, developing, and constructing:

1. The purposeful movement of resources and people into a community-type configuration of society.
2. The development and distribution of community

standards.

3. The support services of the organization to bring into existence community at the societal scale (means of production).
   A. Funding.
   B. Acquisitions.
   C. Politics.
4. Habitat service system constructions and operations (means of consumption), including:
   A. Land.
   B. Construction materials.
   C. Operating materials.
   D. Workers/partners.
   E. User goods.
5. The Auravana Project's publication and distribution, including:
   A. Website.
   B. Data storage.
   C. Calculation..
   D. Simulation.
   E. Artificial intelligence.
6. Education and orientation, including:
   A. University learning network.
7. On-boarding and membership, including:
   A. InterSystem team contribution.
   B. Habitat residency.
8. Operation and sustainment of the habitat service system, including:
   A. Life support.
   B. Technology support.
   C. Exploratory support.
   D. Decision support.

The simplified version of what becomes accessible through this project includes, but is not limited to:

1. Community agreement program.
2. Habitat by-law agreement program.
3. Unified communications platform.
4. Societal contributions service.
5. Move into habitat & live in a habitat (with personal and common access).

1.4 Project planning purpose

A.k.a., Project planning goal.

The purpose of Project Auravana is to develop a community configuration of society through a unified set of socio-technical standards that realize a network of integrated habitat systems that optimize human need fulfillment and ecological regeneration. This is a project to design and operate a specific information and material configuration of society; that of, a community-type societal system. This project proposes the next iteration of society's socio-technical [service] system. This project plan is a proposal to coordinate and control the instantiation of a type societal system, which is produced into materiality, and then operated, and all the while being iterated.

The transition project is not complete until there is a stable network of integrated city systems operating through a unified societal-community information system in which all human needs are optimally met, given what is known. In other words, for this project to be complete there must exist a stable and active (i.e., working and populated, living) version of the proposed, unified societal system's model in material operation. This project, itself, is a success when there is a continuously active community-type societal systems model in information (visualization) and in operation (materialization).

> *"The bad formation of towns influence the bad formation of minds."*
> *- The happy colony of Robert Pemberton (1854), adapted*

1.4.1 Community development, operation, and transition project planning

A.k.a., Human and ecological interface.

A community project plan is essential to the creation and operation of an information-based, materializing habitat service system. This is the project-engineering plan for the next iteration of the Community's proposed societal system. This document (information set) coordinates the sustained existence of a societal design specification and its materialized operation as a Community-type habitat service system. This document coordinates the integration of a materializing information system for a population of users (Read: the community population). In other words, this document coordinates the information composition and materialization of a system to meet human needs, which become human requirements at the level of the habitat service system where project 'intersystem teams' of engineering developers and operators iterate a system of services for the [fulfillment of a] population.

State the purpose simply:

> *This is a proposal to plan the creation and operation of a forward-thinking community with a societal infrastructure that embraces cutting-edge technology applied toward human need fulfillment, generating an environment designed by contributing users around an integrated network of cities and sub-systems, each of which operate for the highest fulfillment of all humanity as a set of services. This is a plan to direct, orient, and approach the operation of a socio-technical environment that is capable of operating at the planetary scale for the betterment of the whole human population.*

1.5 Project plans, standardization

Project plans are significant standardization documents, wherein, they often initialize the standardizing of concepts, understandings, terminology, methodologies, methods, procedures, training, and tools, before any action is taken. In this way, project plan serves as a documented record of collaborative standardization. It is possibly to bring together, into a fusion, a set of previously separate information sets to form a plan to socio-technically and collaboratively create and operate a community-type society.

A project plan is essential to gain the support of capable socio-technical and financial (*market-only*) contributors. Those individuals with intelligence, skill, and financial resources desire to know that their abilities and money will be used efficiently and effectively toward a transparent and agreeable purpose. In order to know that their efforts will contribute to this direction, the system and its planning must be appropriately transparent and open.

In the market, a project plan is essential to gain private and public funding. Intelligent people desire to see transparently that which they are funding in both its operation and likely impact [on them and others].

A project plan is essential to gain [State] jurisdictional support and authorization. In a State (Read: governmental jurisdiction), permission is required to access and to take action. To fulfill all individual human beings together, the plan must be openly and transparently represented so that the authority can see and agree that it represents no danger to the fulfillment of all of humanity, and explains how it represents the potential for the highest fulfillment of all of humanity.

1.6 The challenge

The challenge understood by this project plan is:

1. **The challenge is:** to create a globally workable society for 100% of humanity, on planet Earth, in the shortest possible time [through strategic planning, cooperation, and systems design science] without ecological degradation or the disadvantage of anyone.
    A. **The challenge is:** evolution by human direction for [the benefit of] oneself together with all of humanity. What conditions will lead to human flourishing?
        1. In concern to production: What is the ratio of life in working years to leisure? What is an individual's working years (lifetime working years)? What is an individual's working hours per year (annual working hours)?
    B. **The challenge is:** that there exist societal problems.
2. **The question is:** how do we fulfill all individual human life fulfillment requirements, together, in relation to what is possible? How does society set the best conditions for individuals to flourish?
    A. **The question is:** how will any, and all, societal problems be resolved?
3. **The method is:** intentional information construction and systems science (design science). Systems science is the effective application of the principles of systems and science to the conscious-intentional design of the planetary environment in order regeneratively transform the Earth's finite resources into working services to meet the needs of all humanity, without disrupting the optimization of the ecological processes of the planet or the optimization of fulfillment of all human need.
    A. **The method is:** the understandable, transparent and visual flow of information through a societal [sub-]system information model representational of society, as a simulation.

There are several major challenges that this project must address:

1. It is a major challenge to design a system that facilitates human fulfillment and sustains habitability at a increasing scales of population size.
2. It is a major challenge to provide a reliable and commonly duplicable life-sustaining model that can be sub-configured and applied anywhere on earth.
3. It is a major challenge to bring together all of the organizations promoting various sub-verticals of this common direction of ours. These include, but are not limited to, in general, the highest ideas of all organizations seeking to provide benefit to all of humankind.
4. It is a challenge to design, develop, and operate a system that maintains a safe environment for human habitation and goes beyond the minimum required to sustain life. The habitable environment must also be conducive to service optimization.

1.7 The problems, opportunities, and solutions

This project proposes a model that facilitates working together to find root causes to issues and sustain workable solutions, rather than focusing on short term fixes.

Every human society has the same principal societal problem, opportunity, and solution:

1. **The problem:** The socio-economic structuring of early 21st century society generates a large group of people that live over an extensive area, compete

against one another for the common resources, experience inequality and wealth disparity between social classes and/or genders, cannot operate through a unified decision process due to dissimilar understandings and goals (instead, decision making is by authority, majority, or minority rule), and actions that are taken often benefit a small segment of the people at the expense of others and the ecology.

A. The problem is that humans have common societal requirements for fulfillment and an uncertain environment within which they may or may not be fulfilled.

2. **The opportunity:** Together, we have the opportunity to apply our intelligence, understandings, and abilities to iteratively co-create a community network of socio-economically integrated city systems designed to incorporate elements from (and otherwise reflect) the natural environment of our species, while offering every individual on the planet a set of highly enriched living opportunities based on that which is possible today, and directed toward a new era of flourishing and sustainability for all. The opportunity is fulfillment together, togetherness.

   A. The opportunity afforded to humanity by early 21st century technology and understandings is a unified information system that is inter-related with a specification for the optimal coordination and organization of society. The opportunity is to take advantage of (i.e., use) what is available for the mutual benefit of everyone.

3. **The solution:** A unified and emergently designed socio-economic specification that structures the formation of community where people with a shared sense of purpose live within the regenerative carrying capacity of their environment, cooperate with one another using common resources, experience an enriched life where there are a multitude of opportunities for self-growth and contribution, operate through a unified decision process due to similar understandings and goals, and actions that are taken often benefit everyone and do not come at the expense of anyone or the ecology. The solution is a working socio-technical societal system; a design that works for the fulfillment of all of humanity.

   A. The solution is an operational system, formerly specified, that meets all community-type human societal requirements.

CLARIFICATION: *The carrying capacity of the earth habitat is not a fixed number, it is contingent upon how resources are used, technological capability, and behavior. This is a proposal to care-take the total habitat while highly controlling local habitat service areas, 'cities', which are pre-planned through engineering projects.*

1.8 Consider, how nature might design a society

I.e., What would a society look like when designed through natural-law, given what is currently available?

The method applied by this project plan for the understanding of information flow, simplified, is:

1. **Research (discovery):** Exploring the potential of human knowledge and capabilities for evolving the socio-economic living system and the built environments of the now.
2. **Design (conception):** Applying new and emerging philosophy, science, and engineering technology to a unified model (a design specification) for human flourishing and fulfillment.
3. **Development (materialization):** Constructing an experimental community network of integrated city systems at the convergence of ecological stability, human fulfillment, and technical possibility.

1.9 Consider, object-relationship visualization flows

In brief explanation, the material relationship [flow] "hierarchy" for a community-type societal system is:

1. **Natural planetary ecosystems** (as well as the solar ecosystem) perform fundamental life-support services upon which a human population depends.
2. **Human individual life-organisms** depend on the completion of a common and objective set of parametric environmental relationships (Read: human-object, socio-shape, socio-technical, or socio-mechanistic relationships); wherein, the appropriate completion of these relationships leads to the individual experience of the felt state of flow[ing relationships], fulfillment.
3. **Life fulfillment relationships** finalize together among a population of humans as a process (a.k.a., process group), more commonly known as a 'service'. A service is the materialized societal application of an information constructor; here, a service always carries the property of 'copyability' of transformation (because it is a service, it can repeat, as a constructor repeats by definition).
4. **Through the contributions of humanity**, services may be designed to coordinate the control

of material areas (named, “cities”), of a whole planetary ecology, for copyable human [service] need fulfillment, while simultaneously accounting for the natural planetary ecosystem (Read: the planetary ecology).

5. **Cities may be designed** to facilitate the fulfillment of human [and all] life together in a unified planetary ecosystem. Within a planetary ecosystem, humans primarily live together in cities. Cities are more technically known as [integrated and controlled] ‘habitat service systems’ (Read: local habitat service systems). The habitat controlled cities exist within the natural planetary and solar environment.
6. **A planet may be coordinated** where humanity is expressing the type of society known as ‘community’; therein, cities are connected through a unified, global habitat [resource and access] transformation network. The network of cities forms one globally unified habitat service system (Read: the global habitat service system), describing the human spatial controlled domain (the materialized, Material System) as one domain of the populations unified, multi-domain societal information system.
7. **The unified habitat service system performs** fundamental life-support services upon which humans depend, and represents engineered physical areas of our natural ecosystem.
8. **The unified habitat service system depends** on a global information system of all possible and impossible transformations, and all reasoning.

2 Simplified natural language overview [of the Project]

This is a proposal for a societal-level planetary human service system, and this document acts as a high-level planning description of that system. The system itself exists as a unified set of design specification documents. This is a proposal that coherently visualizes how the optimal fulfillment of human need, at every scale of relationship, is possible now, given contribution without a mandatory trade exchange. This project exists to facilitate the realization of an environment where all individual humans have the environmental potential to live meaningful and fulfilled lives, enabled seamlessly by technology, offering growth and exciting opportunities for all. Additionally, this document describes how teamwork toward a unified planetary society is possible, right now -- this document provides the reasoning and required details for working together on a socio-economic information system that mutually benefits, and works for, everyone. Together, we are optimizing human well-being (fulfillment and flourishing). Together, we are developing a highly automated, moneyless-society oriented toward human fulfillment and ecological sustainability.

This project has been formed to produce the individual [conscious] experience of individual human life fulfillment among society, through the operation of a specified socio-technical habitat service system, specifically designed to facilitate human fulfillment and ecological well-being. In other words, this project proposes individual human fulfillment and ecological regenerative stability at the societal, planetary level of scale. What is projected by this project is a society with “committed” (i.e., stated, transparent, explained, specified, developed, accountable) life functions.

This specified societal system exists continuously along an information materialization spectrum from conceptual through to physical, all of which affect the experience of individuals therein. The productive purpose of the Project is the personal experience of human societal fulfillment, understood to be materially formed from the intentionally specified operation of a unified information network of integrated habitat service systems.

More simply, the purpose of this project is to bring into existence a new type of society, called, ‘Community’. A community-type society exists along a spectrum of possible types of society. The Project shall be structured to define, design, develop, and operate (duplicate) a ‘community’ type of societal system.

> **NOTE:** *The ‘community’ concept is defined at length in the unified societal design specification itself, and in a series of discourses on community (video, audio and text).*

Societies are systems, and humanity can conceptualize them through systemic thinking. Societies, like other human organizations, have structure, values, networks (hierarchies), products, and services. These significant elements of every human society can be designed in such a way as to facilitate the experience of human fulfillment and ecological well-being. Additionally, an information system can be developed to contain, coordinate, and actualize the design.

> **NOTE:** *In society, Individual human organisms grow to become [at least] self-organizing unities capable of independent fields of life as learning, sentience, affect and body action.*

The prime directive of the project is to bring into existence (Read: materialized and encoded reality) a type of society that facilitates the highest potential expression of all of humankind through the synthesis of a "living" design, which expresses the system's reason and executed operation. This proposal envisions the emergence of a system that maintains a connection to living humans and their life capacity, without desensitization to native healthy stimuli. Through strategically planned access to life needs, human 'life' fulfillment optimization and abundance is achievable. It is possible to design society to secure [human] life on earth, given what is known and available (and, as evidenced by this plan and the associated societal specifications).

In part, this is a human evolution project. Wherein, human evolution is knowledge transmission, as well as life-capital reproduction and [conscious] growth, without loss and with cumulative gain.

> *Together, "we" will communicate the various ways in which we may be fulfilled (through open source specificationing); wherein, "we" integrate and optimize for our experience of fulfillment. Together, we shall redefine and achieve the limits of life's experience.*

The societal design specification details the logical derivation and technical operation of itself. Here, the Project exists to cooperatively create community, through a shareable and constructable design specification detailing the logical derivation and visualizing the technical operation of a fulfillment-oriented (i.e., human-requirement) structure, a community-type societal living system. At the of ecological stability, human fulfillment, and technical possibility, this living system forms an experimental (at first and continuously) community network of integrated city systems in continuous iteration through a unified and iterative societal information specification.

Essentially, the specification is a socio-economic system specification (or less commonly, "socio-economic blueprint"). Instead of using the term 'socio-economic, the specification may otherwise be known as a, societal information system, socio-technical system, and socio-decisioning system. The specification defines, describes, and explains the operation of a societal model (or, type-of-society), out of all the known possible range of different ways in which humans can live. Importantly, the specification is a unified model [of societal presence] for human fulfillment and ecological well-being. In practical action ("practice"), the specification is an 'engineering' specification, in that it includes the technical specifics of the system so that construction and operation is possible. Here, the term 'engineering' means that a constructable specification (i.e., "blueprint") is present in advance, and that specification includes a procedure for building and operating what is logically reasoned to be the intention (purpose) of the specification.

> **NOTE:** *In order to logically derive the system [of which is specified], "we" need to account for not only the paradigm that we are creating, but also, all the other paradigms that we are excluding.*

The Project's societal specification sub-divides the total, unified societal system into four sub-system specifications, which together form one total societal system (defining a: 'type-of' society). Presently, the specification logically derives that every known type of society may be sub-composed into four information system categories:

1. A **social system** [specification] that explains our understanding and intent for the design of the real world around us.
2. A **decision system** [specification] (another name for an economic system) that explains our decisions for the coordinated operation of the materially sensed world around us.
3. A **lifestyle system** [specification] that explains the ways in which we become ever more developed "conscious" beings.
4. **A material system** [specification] that explains and becomes the state of the materially sensed world around us.

Different types of societies have different internal compositions of these four systems. Together, these systems form the system's 'type, as the type of society "we" are creating, or "we" are observing. It may be relevant to note that belief systems are not types of societies; though, they are a part of that which defines a type of society (because beliefs integrate into mental modelling, decisioning, and material realization).

A community-type society forms around a common set of fulfilling life related navigational principles (human needs, values, and an approach to alignment) that lead to the sharing of equal access to all that our ecology, given what we know, can provide for our preservation and ultimate self-evolution. Herein, territorial governments and business entities are not needed anymore, and from a complex systems perspective, they are counterproductive and limiting.

Simply speaking, this is a unique proposal:

1. We wish to share,
2. a proposal for understanding and operating together,
3. that is highly likely to produce fulfilling and loving relationships among all individuals in our common world,
4. wherein, all humans have common needs and a common environment,
5. wherein, needs become fulfilled as services through a contributed habitat service system,
6. wherein, a unified information, coordination, and computational system facilitates the sustainment of a complex service habitat,
7. wherein, humanity works together to visualize and deliver a optimal societal solution for the mutual benefit of all of humanity,
8. so, there is no requirement for currency or trade or coercion.

3 Detailed natural language overview [of the Project]

There are different ways of organizing society, and this project represents a proposal to organize society based on and for community. There is another stage to human development that has not yet been accomplished by any political or market entity, and that is what this project is proposing. The type of society this project proposes does not require the encoding of the market-State configuration, which is why (at least in part) it is so difficult for modern individuals to understand. Early 21st century society is largely composed of market-State entities, and those brought up in a market-State structure perceive and act as if society is a market-State. However, there are ways of organizing society that do not involve States or markets. A type of society without a market and a State is the logical consequence of cooperatively organizing a unified, global, fulfillment-oriented service structure for all of the human population. This is a project plan for a societal system that is likely to optimize human well-being, and to do so, in a manner that is free of trade and coercion. For something to be free of trade (trade-free) means that there is no requirement for information or material exchange in order to achieve access. The proposed societal system, a community-type society, doesn't have a market, so there is no price and no currency, there is also no barter or any other form of market-based trade (exchange). For something to be free of coercion means that there is no threat of violence in decisioning, and that the structure of the system itself does not generate relationships based on groups of humans holding power over other humans.

When all of society is known as the market-State (i.e., when all individuals know of society as only the market-State), it can be challenging to visualize a society organized more simply. Understandably, there is unnecessary effort being expended in order to process [human] life information using the additional market-State layers of abstraction, those of 'currency' and State 'authority' [over society]. If someone's perceptions are formed at this more abstracted (because it includes property, money, and coercive authority) layer of perception, then it can be challenging to remove the unnecessary abstractions from those that are necessary to knowingly sustain the well-being of a human life. It can be challenging to remove the abstractions, because that which is necessary and unnecessary for human life fulfillment have enmeshed together in the mental model being used to process the perceptions themselves. The market-based organization of competition for scarcity in access to resources and [human] fulfillment, using money, is a layered abstraction [in mental perception] over a more simple and natural socio-decisional environment. The abstractions of and beliefs in authority, money, and scarcity, brought about in a market-State system, overly complicates and introduces inefficiencies in society.

The less abstract visualization of society is one in which there is not money -- where there is no reification of indebted exchange (no individual, or non-all social group, ownership). In other words, This is a proposal for a working society where there is no socio-decisional encoding of mandatory exchange (e.g., money), or the market (i.e., indebted ownership) into human societal relationships. It is possible to perceive the socio-decisionally optimal operation of a human service fulfillment system without any requirement for mandatory exchange (the behavioral-materialized encoding of competition and scarcity). It is possible to share access to a socio-decisional real material world where there exists a global population of humans who share access to all human needs and resources through communication and cooperation that structures a societal system state of optimal self and social fulfillment.

> **NOTE:** *Instead of thinking of the term 'free' in place of "not using money", maybe think of the terms, 'cooperation', 'shared', and 'common' [to information and material resource].*

When the whole world (i.e., all human behavior relationships within a real world) is viewed as a series of mandatory exchanges (from 'buying' and 'selling'-type events to 'gift'-type events), then it is challenging to perceive oneself in an environment where those conditions are not [necessarily] present. The complexity of modeling can be seen, for example, through societal 'gifting' events. At a societal-level, a 'gift'-type event is, for example, a cultural event where the receiver of the 'gift' could have accessed the socio-technical object/service himself/herself, but because of some socially constructed meaning, at some time interval, whether based on objective events in the real world (e.g., puberty), or not, the receiver receives the "gift". The term, "gift" is now in quotes, because it is a conception integrated into a processing mental model whose existence is not materially sourced, but due to conscious entities constructing social meaning.

> **QUESTION:** *How could society best operate without trade [in a market] or fear [of authority]?*

The market-State represents an enclosure - an enclosing overlay on top of a common heritage environment. The common heritage environment of 'resource' is sub-composed of a specifically and identifiably knowable (i.e., locatable) organization-position-composition of shape-object-geometries, 'resources'. The conception of the "market-State" imposes a requirement for exchange upon most individual human relationships. The requirement for individual exchange as mandatory for fulfillment leads to the division of the common heritage (into "ownership").

There has been a misunderstanding among certain cultures on earth that the idea of having a unified world, a harmonious world, means that we all have to be homogenized. However, that state is as far from 'unity' as humanity can get. True harmony is true unity, which is the result of absolute validation of all of the individual (fulfillment) differences in us; because, each of us is an individual among a social population of common individuals sharing a common world, a common home. It is possible for us all to fit together, to individually share and co-operate, to form one unified and harmonious whole societal [information and material] system, where all the individual pieces support the whole at the same time as the whole supports all the individual pieces. It is possible to validate all the individual unique difference between us when we account for the fulfillment of all and the resources commonly shared by all. Through cooperative design and operation, oriented toward the fulfillment of each and every individual, is the unity and harmony that we all individually seek. By shifting to a more encompassing state of awareness, being open to new and testable definitions of reality, and acting from that point of focus, we may come to realize that our highest well-being has always been possible, and we have never truly been alone.

There are mental models that view all earthlings as family; wherein, humans cooperate for the fulfillment of themselves and their extensional family (express extensionality; love). Without the requirement for mandatory exchange in a market, and the necessity to monitor and control that market by a controlling punishment driven (authority) system there is the potential for the flourishing of the highest-potential capabilities of all of human-conscious kind.

> **INSIGHT:** *Often, the community lifestyle is about living cyclically at the peak [potential] of one's abilities (i.e., living in flow).*

Here, 'global community' means three things: 1) that the whole system can be optimized together; 2) all of humanity has optimized fulfillment; and 3), designed for planetary-scale operation. In this third case, the [specified] societal framework [as a service platform] can be scaled up to the size of our planetary [human] population. The operation of a planetary scale, moneyless operation (of society) requires a specifiable structural configuration and composition. Here, cooperation means that actions are executed through joint and consistent decisioning. Through the encoding of the value of global cooperation, a society becomes capable of scaling from a small (population) fulfillment density to a large (population) fulfillment density by optimizing fulfillment together.

> **INSIGHT:** *Community comes into existence through socially and ecologically responsible design, through a [whole] systems science approach.*

A globally fulfilling societal structure involves, given what is known, the population [of humans] living together in a life-coherent and socio-technically determined network of [integrated] city systems, which apply the same unified information system in their operation.

This project proposes an environment where design is selectively expressed into materiality to optimize the fulfillment of all individual human requirements, given common access to common resources through a common ("cooperative") approach in a common ("open source") environment (which is both informational, and therein, also material). This project presents a commonly agreeable approach to the design and selected construction of a society through a unified societal model, itself optimized and so constructed for common human highest-potential, individual fulfillment. More simply, this is a project to iteratively test societal models for optimal human fulfillment. This project must account for information and materiality in order to accomplish this goal.

> **INSIGHT:** *Sustaining community is not just about aligning with nature, it is also about seeing ourselves (and oneself) as an expression of nature. Thus, allowing our differences to become compatible, facilitating inclusivity, and not, exclusivity.*

4 A project to develop a type of society

Any given society may be analyzed, through division of the societal system from unification, into an organized inter-relating sub-set information structure. This project has the axiomatic assumption, given what is known, that society can be sub-set into the sets:

1. **Social** [intentionally navigational].
2. **Decision** [controlled action].
3. **Lifestyle** [current life result experience].
4. **Material** [physically created/-able interfaces].
5. **Plan** [coordinated action].

> *These five sets are the core information sub-systems (of any society). To more easily understand and re-design society, it is best to visualize society through its principal sub-systems: social, decision, material, lifestyle, and coordination system.*

In order to more greatly know society, one may follow the following train of thought:

1. "I" sense and interface with others like myself (social),
2. in a sensible environment (material),
3. where decisions are possible (decisioning),
4. and different experiences of life are the result (lifestyle).
5. Together, "we" can plan and coordinated a decidedly optimal socio-material life (planning).

The four continuously existing societal information sets (social, material, decisional, and lifestyle) are integrated and unified through this Project Plan document as a well informed and timely plan of action for the coordinated engineering of a community-type of society. Different societies have different internal compositions and interrelationships of these four (social, material, decision, lifestyle) and one (project plan) societal sub-systems.

> **CLARIFICATION:** *This highest-level societal project document initiates and coordinates a specific type of societal design [configuration]; one that is specified by four societal sub-systems (specifications), which represent the unified design-operation of a community-type societal system. The four societal subsystems are: the social system, the decision system, the material system, and the lifestyle system. And the unifying, temporally integrating information set is the one selectively executed project-engineering work plan.*

This project proposes that the four common societal sub-systems can become one unified system intentionally designed and operated to optimally meet (fulfill) the human requirements of every individual among the population. In a society that effectively coordinates human contribution within the context of a unified system standard, there is the potential for fulfilling all human need without exchange or coercion.

Different 'types' of society have differently structured orientationally aligned directions. In other words, different 'types' of societies orient humanity in different fundamental life-impacting directions. What differentiates types of societies is not the societal sub-system (Read: social, decision, material, lifestyle), but the configuration and encoded conception of each societal sub-system. Differently oriented societies will necessarily express different internal configuration and compositions of these four fundamental sub-systems of every human society. Herein, a society oriented toward cooperative and openly shared (global) fulfillment is optimized for our commonly shared real material world environment. That 'globally unified' type of society that operates through cooperation and openness is optimal to a society that does not co-operate globally. It is globally optimal to account for all individual human need-requirements, given a common environment.

> **QUESTIONS:** *What is the mechanism (what is the model) for human global access fulfillment without the market-State and with well-being and sustainability? A "strategic" planning level based on information input, process, output, and coordination in an uncertain environment.*

In community, where human fulfillment occurs within an openly cooperative environment, societal control is organized, designed and operated, through transparent control protocols and methods of logical objectivity modeling. In early 21st century society, where the State is encoded, these ideas become subsumed into the concept "government". And, the economic distribution of resources in the form of 'market' goods and services becomes subsumed into the concept "business" (or the "government", again, in the case of social*ism*). Visualizing our commonly individual societal system within an unified specification may be viewed as the method of [logical] objectivity.

> **APHORISM:** *Everything is separately together.*

Every society has control protocols, some implicit, like not leaving a knife (of set material parameters representative of 'danger') in the presence of a toddler. Or, explicit, for example, a decision control protocol disallowing a person of insufficient access[-ability] to print a 'dangerous' projectile weapon-object from a material printing service location. In an open society, these control protocols are formed within a unified information calculation space in order to optimize the creation and operation of human-oriented services.

> **INSIGHT:** *Humanity can do better than having any human attend a store for any unwanted hours a day, or do anything not meaningful to themselves as a contribution to society.*

In a sense, the societal specification (four and one) is an evolving informational mental model, a 'learning algorithm'. From a continuously collected source of information, the learning algorithm optimizes the environment to respond to a conscious individual user's intent, which can be accounted for in the algorithm. In order to fully understand this proposed societal system, as it would take anyone to understand a complex programmatic algorithm, it requires an comprehension of syntactical (logic) and semantic (meaning).

It is assumed possible that society, in design-operation, may be represented as an information algorithm that can be computed, and a computation currently being completed by intelligent humans, may be eventually computed by a general intelligence machine(s). This is to some degree why the specifications appear often to be written programmatically, because they are to be read by those systems with intelligence (human and digital, both systems which have been trained with knowledge, and are actually operating the society). Intelligence is required to operate, or otherwise compute, anything. A technological society is a hybrid human-machine (Read: socio-technical) system, naturally.

It is possible that a more unified society will likely move more toward unification of its computing system such that, at least in the machine category, this will become a unified, calculation support service.

It takes thoughtful inquiry and openly honest integration to design and operate a society that sustains the optimized level of human fulfillment given that which is available. The probable consequences of behavior and information processing structures are known, or knowable, within any given society.

Socio-economic resolutions are not dualistic, in either having contradictory values (orientations) or having more than one optimal result given what is observable and available to all. There are not two (or more) points of view that contradict each other, and are both correct ("right") in concern to the selection of the next configuration of society to be coordinated into existence as the next iteration of the societal system by the InterSystem Team.

In community, individuals can be obviously recognized as not expecting their intentionally-cooperatively organized societal system to allow anyone to starve in fulfillment, or otherwise go insufficiently fulfilled. From a simple survival perspective, this is because when many individuals are starving, generally, all the individual thinks about is the next meal, and individuals can easily lose care about the future population of all individuals, versus getting something now for the individual self.

This project does not propose a society designed to generate a mentality where anyone would perceive life as "Tomorrow may [never] come, so grab what you can now and damn the consequences". This is a societal-level project where there is no need or benefit to distrusting others because they are not in economic competition with you. This is a project for a society where everyone perceives and acts from a common, optimized, and unified information space, through which multiples of harmonious individualities express themselves.

In any society, it is likely that the idea of "human nature" will be significantly tied to the societal system structure in which humans are being brought up within and operate. Therein, the societal structuring will predispose a certain pattern of behavior within the humans being brought up and operating within it. A pattern of behavior, seen through a societal structure is often called "human nature". In this project, it is assumed that given a different environment, a different set of societal conditions, humans are highly likely to behave differently, even though they still have the same 'human nature'; because, that which is 'human nature' must be shared by all humans persisting within a material eco-sphere. Humans share the propensity for behaving differently given different environmental conditions (e.g., a different societal structure).

This proposal assumes that humans operating under conditions of societal cooperation (vs. competition), algorithmic decisioning (vs. price), technical efficiency (vs. planned obsolescence), helpfully applied automation (vs. unnecessary labor), restorative justice (vs. punitive/ retributive justice), and others, are likely to display a different [from market-State] and more evolved pattern of behavior. In other words, a different societal structure, which has been designed to orient explicitly toward human fulfillment (and not money acquisition, money sequencing, power over others, etc.) is likely, given what is known, to predispose the population therein to a more humane pattern of behavior.

It is possible for an individual or group to create socially constructed "bubbles" that distort the real-world where fulfillment would otherwise be possible. Through intentional design and cohesively integrated feedback, from environments that test fulfillment, it is possible to design societal systems where societal behaviors orient toward the real-world fulfillment of individual human beings.

Essentially, the societal system being proposed operates based upon an open-source and unified information system that is explicitly coordinated by its users [as contributors], who provide for their own individual fulfillment. The population within this proposed society shares a similar direction (human fulfillment of need), orientation (a value system), and an approach (a method), which are the three information sets necessary for harmonious social navigation (Note: these are described at length in the Social System specification).

Together, humanity can direct society toward ever greater states of human fulfillment and ecological well-being. Technically, a directed systems is one in which the system is designed (engineered) and coordinated (i.e., controlled, "managed") to fulfill a specific purpose(s). Therein, if component sub-systems maintain an ability to operate independently, their operational mode is sub-coordinated for the specific system's purpose.

In the society this project proposes, all resources on the Earth are held as the common heritage of all the worlds people. Here, each individual is committed to self, and all, simultaneously by means of an understanding that we exist in common (common organism, common organismal requirements, and a common and finite planet). By perceiving the whole world as common heritage, a participative habitat service system may be sustained to facilitate harmony among all individuals, while maintaining harmony with the earth's natural regenerative cycles. In a sense, the controlled habitat could be viewed as an experiment, wherein feedback from individual humans and the larger ecology evolves human society.

Due to the design of the projected societal system itself, because it accounts for feedback and can adapt to necessary changes in orientation (there are no externalities and the feedback mechanism is explicit and openly programmed), it is highly likely that this system could be scaled up to the population size of the planet without majorly hurtful artifacts appearing.

In concern to the materialized operation of this type of society, it may likely be first seen as a city (or village, etc.). However, the system is being designed so that as it scales up to a network of integrated city systems at the planetary scale. By design, by multiplying integrated city systems, the societal system becomes more efficient (to a point), because more information that is more accurate is integrated coherently into the unified information system, whose explicit purpose is to provide for human habitat service fulfillment, for which there will eventually be many different city *custom*izations.

The architecture in community-cities is likely to vary considerably, as there are a variety of cultural groups presently on the planet. So, while there is a unified socio-decisioning model, there are considerable cultural variations of its expression. These customizations mostly take the form of different city configurations and architectural-style aesthetic designs. These cities may be spread across the planet, as opposed to the tendency toward mega-cities and sprawl, which were common materialized population centers in the early 21st century. In community, some of the population lives in extremely modern homes and technically advanced city environments, whilst others have chosen less technologically advanced dwellings and cities. In general, regardless of the technological development of a city, machines are created to deal with any undesirable monotony [of individual human effort, of "jobs"]. The individuals living in a city, their values and customizations (customs) will determine the degree automation. For example, some family homes was wash the dishes by hand, whereas others may use automated machines; and some to be served automatedly produced food, while others may harvest and prepare their own food.

> **INSIGHT:** *Living beings may facilitate the development of their high capacities (higher potentials), by algorithmically automating services to free their time to pursue their highest potentials.*

4.1 What is a society?

Society is an axiomatic unifying concept that represents the first socio-technical structural-dynamic composed of objects, concepts, and individual human consciousnesses (with life-force will) capable of recognizing the presence of two threads (i.e., informational and material). Society is a cyclic nature of successive life flows, which are testably controlled to improve and coordinate life fulfillment generation after generation as an evolving ecological human habitat system. Society is composed of two ropes of a single thread that connects all life, one informational (mental) and one physical (material). In the context of its intentional steering, society is, first and foremost, an information system[s model], within which there is visualization, simulation, and materialization, together. Information structures the societal system - the societal conditions then construct what information is and can be made available.

> **INSIGHT:** *A healthy societal system is a process of repeated refinement and increasing attention to appropriate prioritization of planetary resources, human fulfillment, and ecological regeneration.*

Correct information is needed in order to take the correct decision in relation to re-alignment in an uncertain environment. A correct structure produces correctly aligned functioning with an expected result, in and given an environment, when enacted (energized). For the individual, society is a social population of common and finite inter-relationships. For humanity, society is experienced through a human environmental interface, consisting of egoic-socio-material informational relationships. These relationships may be understood and created intentionally through logical information processing structures, including but not limited to: systems science, systems engineering, project coordination, algorithmic decisioning, modeling, and visualizing.

Society is a system (of systems, SoS) of all [socio-economically] related people, wherein a system is:

1. A system is a set of interacting components that operate together to produce intended (and unintended) outcomes.
2. Systems are usually made up of subsystems (which are systems).
3. The sub-systems of a system organization are sub-organizations of the system.

Society is a set of complex individual decisions around socio-technical relationships between those human individuals. That set of complex relationships can account for the natural life-support system of all of humanity.

Society makes possible the cultivation of human capacities as ends in themselves. That is, society can be designed to facilitate the cultivation of social self-conscious agency, not as an instrument of survival, but a direction in itself, where each individual is highly self-integrated. A continuously optimized societal design enables the conscious expression and evolution of higher potential states of capability.

> **APHORISM:** *Information is constantly re-structuring us, and we are re-structuring that information.*

Society represents both a potential (because information-based) and the current actualized (because material-based). A society has potential and is the actualization of that potential. The potential is not the same as the actualization. The potential can be there, but not actualized. What is potential is not actualized. Society exists, in part, to fulfill individual human potential by solving problems or realizing opportunities. Herein, technology can enable universal needs fulfillment, which includes the facilitation of social needs.

In a society where social requirements are recognized, the natural problem of human life, how to survive, becomes the social problem of how to live well (fulfilled), together. Humankind recreates its social home through socio-technical decision activities. These activities are essentially cooperative; the question is, at what scale is there cooperation?

> **INSIGHT:** *Databases and computation enable the coordination of a complex socio-technical environment that can account for the human need fulfillment of all individuals among the population.*

The total ecology within which the human habitat exists is formed from the interaction between three continuous[ly unified] systems:

1. The abiotic geosphere
2. The biosphere
3. Human socio-technical activity

4.2 Societal organizational elements

Any society is composed of a common set of human organizational elements. In any human organization, of which 'society' is the highest level, people <u>access information</u> to <u>follow processes</u> to <u>use tools</u>. Hence, this is a project to define and coordinate these human organizational elements for the benefit of all of humanity.

Any given human organization may be sub-composed of the following elements:

1. **People** - Humans, because [societal] organizations are made of people. Organization's don't matter if people don't participate in them and/or are not fulfilled by them.

2. **Information** - Organizations can't coordinate without sufficient access to information about the organization itself and the environment in which it operates.
3. **Processes** - Organizations can't scale up past (about) six people without some standardized way of coordinating action through organizing/-ational processes. Both "manual" processes and "technology agnostic" processes almost always describe ways that humans use tools.
4. **Tools** - People can't do anything meaningful (i.e., functional) without tools. Tools may be used to manipulate the physical world (to build something or repair something) or to manipulate Information.
5. **Resources** - Tools can't be created and won't function with a host of surrounding technologies, all of which are made of resources.

4.3 How is society experienced?

Firstly, society is often described as being experienced as:

1. An operating system.
2. A knowledge-based, self-organizing system.
3. A governing syntax of understanding and value.
4. Common human goals (that raise our potential, rather than obedience to an authority).
5. Common human feelings (that give us access to our highest potentially capable selves).
6. Common human visualization (that gives common understanding).
7. Common human values (that give us an adaptive directional re-orientability).
8. An organization that allows individuals to express their life-capacities that are intrinsically satisfying to the self and valued by other people.
9. A system of Earth (planetary) coordination (management). Forming the Universal Human Economy, Global Access System, Network of Habitat Service Systems, etc.

Secondly, the experience of society, like anything, occurs through the self. When "I" become conscious,

1. "I" feel an object.
2. An object is that which has shape [to consciousness; conscious sensation; awareness].
3. In a materializing information system, objects that have shape are 'resources' in the material system, which is physically sensible, and with a digital counterpart as a simulated computation.
    - If an object has an interface-able shape, then at the point of interface, it is in the material system.
4. The primary material interfacing object for all individuals among society is the (global/local) habitat service system.
    - Here in the physical world, in community, "we" can point to a real-world physical (with digital counterpart) habitat service system composed of teams of humans and machines who carry out [project] functions with the use of material resources.
5. Potential and executed material configurations are integrated within the decision system to determine a selected and executed configurations of the material system.
    - In community, in the dimension of computation, software, and intentional information transformation decisions are resolved into the execution of team action in the material system.
6. The lifestyle system is the lived experience and reasoning therefore.
7. The social system integrates the survey of individuals' life experiences into a data, knowledge and standard, structure that informs the whole of the societal system.
    - In an information system, the social system is the inquiry, storage, and integrating processing unit for all of humanity's information.

4.4 What defines a societal-level project?

The analogy of a societal-level operating system most closely analogizes a society-level development project. Society is a design, development, and operations platform. As a platform, society serves everyone's ability to understand and deploy tools and resources, and to be able to co-create society in a safe and responsible way [through standards for information flow and materialization].

What is required for a societal-level operating system is, at least, a societal-level visualization of the operational Informational System and Habitat Service System in life-cycle format:

- A <u>transparent visualization</u>,
- <u>of the flow</u> of all resources (information and material),
- <u>through an operational</u> habitat service system,
- <u>coordinated</u> (where and when) into existence,
- <u>through a population</u> of contributors,
- <u>who share</u> a specified information system,
- <u>that resolves into a commonly</u> fulfillment re-materialization of the habitat environment.

In the market-State there are institutional entities, which due to their internal reward functions, make visualizations and actions non-transparent (i.e., secret or obfuscated), including many market and the State structures, which are not transparent entities. A lack of transparency at such a basic level (that of human needs and their economic fulfillment) interrupts the coherency of a society's information-fulfillment system, wherein the societal system will perform sub-optimally due to gaps and flaws in its structuring.

A societal-level interface service also defines a societal-level project. A societal service interface consists of a coordinated habitat service systems, prioritized as life support and then facility support, with technical support providing hardware-software systems to both. The function of a helpful habitat service system is to provide for human fulfillment and ecological regeneration. A helpful habitat service system must perform to sufficiently (appropriately) meets all human needs, where sufficiently is first visualized completely (complexly) as a socio-technical [community-type] societal design [specification] prior to its execution as the instantiated state of the materialized life-style system.

4.5 What is the project's proposed societal sub-control units?

NOTE: *Society can be engineered as a closed-loop control system, the alternative is an open-loop control system where feedback on human fulfillment and ecological issues are not used to reorient or restructure society for optimal fulfillment.*

Society selects the current state of its operational [habitat] service system through a process of parallel societal inquiry (sub-processes, protocols) that discover and orient the whole of society. Therein, societal control (i.e., societal decisioning) involves a hierarchy of directional re-alignment processes:

1. **Informational-social control** (*social parallel inquiry process*) - information processing groups and knowledge areas; social requirement alignment.
2. **Social-project control** (*project inquiry*) - project control process groups and knowledge areas; project alignment.
3. **Project-technical control** (*technical solution inquiry*) - engineering processes and knowledge areas; technical alignment.
4. **Technical-service control** (*solution operations*) - habitat service system operational processes and knowledge areas; service alignment.

Herein, a control 'objective' provides an aim, reason or purpose for which one or more internal controls should be implemented. Whereupon, a control objective becomes a specific target to evaluate the effectiveness of directed intention and its surrounding foci of control. A societal information operating system stores, coordinates, and controls the service state of the society.

STATEMENT: *For survival in a finite and dynamic system, "we" must be extremely contentious about every decision that we take with every resource that we have, every day -- we require an operating manual that we can all agree creates the best environment for humanity.*

A real-time/real-world societal operating system (RTSOS) has two operational levels of definition:

- The prototypical social: Societal-level operating system as a social organizational structure in formalized and actualized operation.
- The individual: Egoic-level operating system as the individual conscious self ("me").

A societal-level project is defined as a unifying operating system that constructs, contains, and executes the rules (patterns, process fractals) of the developed and operated execution of society. The product of societal engineering (i.e., societal-level project-engineering) is a societal-level operating system.

Three principles (two core and one stabilizing) are likely required to create a safe societal ["machined"] operating system:

1. The proposed societal systems only technical objective is realization of human needs. Often, in the market-State the only technical objective is the machines realization of human preferences. This proposed societal system has no machine objective at all, not even to preserve its own existence. Because, in order to preserve the fulfillment of human needs the machine is going to "want" to preserve its own existence. If the machine is given another reason to act, then there is a conflict between human needs (or preferences) and the machines desire for self-preservation; and, that conflict should not exist.

2. In the proposed society, the machine will be uncertain about what human needs (or preferences) are. The machine must always inquire into the users needs and objectives, and not presume user needs or objectives. The machine/system must be designed with a protocol that doesn't assume where assumptions affect results. This principle exists to prevent the error analogized by "The King Mitus problem", where the king specified the wrong objective and everything he touched turned to gold, including his family, which is not what King Mitus intended. An active societal-level machine that believes it knows the objective is likely going to pursue the objective regardless of individual humans flagging of the objective as an impediment to human need fulfillment -- since

the machine knows the objective and has done the optimization, it knows that the action it is taking is correct, regardless of human noise to the contrary. The objective is a sufficient statistic [in measurement of success], and subsequent human behavior is irrelevant once the objective is present. Hence, making the machine uncertain about the objective, the machine is then open, and in fact, has an incentive to acquire more information about human needs (more clearly, human directions). And, the human(s) making an issue of something that the machine is doing is clearly more information about human needs (or preferences), and the machine (society, the HSS, the service bot) must account for this new information, because presumably the machine could possibly have been previously violating (or just hindering) previously unknown human need (or, preference).

TERMINOLOGY: Flagging is suggesting that a system isn't working as expected (i.e., articulating an issue/problem with a system).

These two principles work together to make machines/systems deferential to humans/users, such that they are willing to accept redirection (i.e., controllable). The machine/system has a protocol that asks permission (inquiry threshold gate) before doing anything that might have a negative effect (because they are not sure and lack sufficient information). Thus, machines will allow themselves to be switched off -- one way to prevent negative outcomes (a lack of or inhibition of user fulfillment) is to allow oneself to be switched off. There is a positive objective (or incentive) to allow oneself to be switched off; whereas if you are 100% certain of the objective, then the machine has no incentive to allow itself to be switched off ,and in fact, the machine has an incentive to prevent itself from being switched off. In terms of materialized integration, the machine must not only be capable of being switched from an on state to an off state, but 'off' also means that the machine must be capable of being dis-integrated from material integration.

3. A principle for stabilizing ("grounding") the conception of human needs (requirements, preferences, etc.). The decisions that humans take (as in, human behavior) provides information about human needs (and preferences). And, the reason that is problematic is that humans can deviate from behaviors that are optimally fulfilling given what is known and available. Human understandings, visions, and expectations of what a fulfilled life is supposed to look/be like can become highly derailed to the point that it produces extreme dissatisfaction. Humans can, and can not, act rationally. To act rationally is to act toward the fulfillment of human need, optimally, given what is known. Individual actions may, or may not, match [the fulfillment of] needs/preferences, optimally, given what is, and what is known.

4.6 *The deliverable is an operational societal system*

The list of plannable societal systems [for a community-type society]. This list includes a system of systems, standards, and support structures, all of which require the completion of tasks, through contribution, in order to sustain the service:

1. **Global societal life service system:**
   A. **Global information service system**: An operational, informational environment (a.k.a., the information, construction environment): The information system as an operational data interface service system.
      1. **Global societal service standard:**
         i. Social Information System.
         ii. Decision Information System.
         iii. Material Information System.
         iv. LifeStyle Information System.
   B. **Global habitat service system:** An operational, material environment (a.k.a., the materialized, built environment). The city as an operational habitat service system.
      1. Life-Support system structure.
      2. Technology support system structure.
      3. Exploratory support system structure.
      4. Multiple city configurations customized for different group preferences (cultures).

Human life uses both informational and material services. These services can be accounted for and planned:

1. A living body uses *habitat spatial service resources* (for its benefit and highest potential).
2. A living mind uses *habitat informational service resources* (for its benefit and highest potential).

4.6.1 Project personnel roles

Information system development team structure (as an organizational structure):

1. **Coordinators (coordinating entities)** - coordinate information and material information flows for

operation in a real-time, given environment.
   A. **Societal information system** coordinator (**information system** coordinator).
      1. **Planning system** coordinator.
      2. **Social system** coordinator.
      3. **Decision system** coordinator.
      4. **Material system** coordinator.
      5. **Lifestyle system** coordinator.
2. **Working groups (informational system)** - develop information systems and standards for operation in a real-time.
   A. **Societal system overview integration** working group (**Information systems** working group).
      1. **Project plan integration** working group.
      2. **Social system integration** working group.
         i. **Research integration** working group.
         ii. **Knowledge integration** working groups.
         iii. **Engineering integration** working groups.
      3. **Decision system integration** working group.
      4. **Material system integratio**n working group.
3. **Habitat Teams (material system)** - operate habitat service systems in a real-time environment.
   A. **Habitat service operating integration** team.
      1. **Life support service operational** team.
      2. **Technology support service operational** team.
      3. **Exploratory support service operational** team.
         i. **Research support service operational** team.

4.6.2 A social information system platform

A social information system platform is required for working at population scale, and it enables:

1. Visualization.
2. Tracing.
3. Computing.
4. Collaborating.
5. Coordinating requirements, workflows, interfaces, design, assembly, etc.
6. Smart design and testing (integration of mechanical, electrical, software, and electronics design).
7. Convergent modeling.

A societal information resolution interface for:

1. All Views.
2. Technical Standard Articles (social, decision, ...).
3. Studies (scientific understanding and research).
4. Lifestyles (individual and social calendars).
5. Operations (procedural, monitoring, and change control procedures).
6. System support (life, technology, exploratory).
7. Services (habitat service sub-systems).
8. Flows (resource flows).

4.6.3 A team contributions platform

A community-type society necessarily organizes a team set to accomplish organizational tasks. Teams complete tasks.

In order to complete tasks at a systems level, a team must:

1. Develop and use data sets.
2. Develop and use procedural tools.

In order to,

1. Develop and operate a global information system.
2. Develop and operate local habitat service systems.

4.7 The deliverable is a societal specification standard

The following is a list of the high-level deliverables for a community-type societal project:

1. **Societal specification standards** (the product-system; a societal information system, a society).
   A. **Social system standard.**
      1. Written technical standard articles.
      2. Conceptual modeling.
      3. Database system production and operation.
   B. **Decision system standard.**
      1. Written technical standard articles.
      2. Design code.
      3. Software system production and operation.
         i. Information collaboration platform.
   C. **Material system standard.**
      1. Written technical standard articles.
      2. Design drawings.
      3. Hardware system production and operation.
         i. Habitat service system.
   D. **Lifestyle system standard.**
      1. Flow experience standard articles.
      2. Learning experience standard articles.
      3. Contribution experience standard articles.
2. **Project overview standard.**
   A. Identifiable unifying model.
   B. Written proposal of unification (treatise on community).
   C. Visual prototype of unification.
3. **Project plan standard** (the coordinated plan of action).
   A. Listed variables for actions.
   B. Written understanding of actions.
      1. Visualized efforts of actions.

4.7.1 The functional societal specification standards

A societal information system may be sub-divided into sub-systems with specialized functional standards:

1. **The social system specification**
   A. The written documentation part.
   B. The human fulfillment and motivation database.
2. **The decision system specification**
   A. The written documentation part.
   B. The mathematical modeling part.
   C. The software programming of the decision system.
   D. Machine learning interface.
3. **The lifestyle system specification**
   A. The written documentation part.
   B. The global access system's interface.
4. **The material system specification**
   A. The written documentation part.
   B. The architectural CAD- and BIM-based drawings for the integrated city system and technology therein.
   C. The 3D visually modeled representation of the integrated city system (with different configurations).
   D. Integration of the 3D representation into a gaming engine for virtually simulating all operational aspects of the community.
   E. An open source virtual reality simulator of the city.

The specification standard for a unified societal information system involves:

1. A unified specification standard for the construction and operation of the societal system.
2. Continued research, design, and error correction of the existing specification standards.

4.8 The scientific discovery deliverables

The following is a list of study deliverables for a community-type societal project:

1. **Rational thinking studies -** Show me the object, the motion, and the conception.
   A. **An understandings review** - Existing visualizations are explained.
2. **Experimental studies -** Show me the controlled change, the test.
   A. **A literature review** - Existing literature is one source of social data "evidence" on causal and correlative relationships. Literature may be searched for evidence in favour and against a solution concept or hypothesis. Existing literature may also suggest alternative causes to problems. As one of the dependent variables in an article is related to the selected problem, the independent variables may reflect causes of the problem. To select the literature (from a unified information space) and the new causes, it is important to know that the literature is reliable and valid for the practical situation. The systematic review of the literature enables a social population organized through a project-based structure to benefit optimally from existing knowledge on a subject.
3. **Publication studies -** Show me the public integration.
   A. Scientific journals - are the most important medium for the publication of research results. Articles in scientific journals present findings at the frontiers of knowledge and are often characterized by a limited scope. Most journal articles have a similar structure.
   B. Professional journals - In addition to scientific journals, one can also find professional journals. These journals are targeted at an audience of practitioners. The most popular professional journals include Harvard Business Review, MIT Sloan Management Review, and California Management Review. Professional journals have a pragmatic instead of a theoretical focus. These journals seldom publish original research – only popularized versions of research published elsewhere.
   C. Books - Distinguishing between discipline-specific books, scholarly books, textbooks and handbooks.
   D. Quick reference materials - guidebooks, handbooks, etc.
   E. Other types of research publications - Besides scientific journals and books, there are several other types of publications in which results of scientific research are published. First, conference proceedings contain papers that have been presented at a particular conference. Conference proceedings are particularly valuable for finding out the latest research. Frequently, improved drafts of these papers are later submitted to journals. Most libraries have only the proceedings of the most important conferences available. Second, many research institutes publish series of working papers. These papers describe research-in-progress, and later versions are often submitted to journals. Therefore, these are also particularly important to find out about recently finished and current research projects. Finally, there is

so-called grey literature. This is literature that is written for a restricted audience and is difficult to identify and obtain.

4. **Prototype studies -** Show me the simulation.
5. **Assembly studies -** Show the object to me (i.e., show it to me).
6. **Verifiability studies -** Show me where it will be.
7. **Cyclability studies -** Show me the material and informational flows.

4.9 *Quality review deliverables*

In order to ensure that deliverables maintain an certain standard of quality, they are reviewed (to assure their quality).

4.9.1 Standards review

Summarily: Scientific papers, research papers, working papers, reports, white papers, journal articles, etc.

4.9.2 Literature review

The following steps may be part of the project plan:

1. A literature search regarding the topics mentioned in the left-hand side of the conceptual project design. It results in the theoretical ideas and guidelines for the diagnostic step.
2. Empirical analysis of the problem: investigation of the specific characteristics and the validity of the business problem and the exploration and validation of the cause and consequences of the business problem.
3. Formulation of the diagnosis from a unified information space.
4. Exploration of solutions.
5. Feedback of the results of the former steps to the principal, the company supervisor, and the platform or steering committee, and the university supervisors.
6. Further detailing of the project plan for solution design and implementation.
7. A further literature search regarding topics on solution design, resulting in among other things design specifications.
8. Elaboration of one direction into a redesign and a change plan.
9. Development of organizational support for the solution and the change plan.
10. Presentation and authorization of the solution and change plan.
11. Implementation (if included in the assignment).
12. Evaluation.

New design project understandings may come from

1. **Focus on empirical analysis**. An empirical exploration and validation means that the symptoms, their potential causes and their potential consequences have to be identified, and evidence to support the analysis has to be gathered.
2. **Focus upon theoretical analysis.** Theoretical analysis and empirical analysis should strengthen each other, but there is no standard recipe for doing so. The sequence in which empirical and theoretical analyses alternate,the way in which they interrelate, and the relative emphasis on one or the other differs from project to project.
3. **Focus upon process-oriented analysis.** Usually a process-oriented analysis supports the analysis of the business problem and its causes. A focus on causes and effects is needed to eventually yield a validation of the business problem and a valid analysis of the causes of that problem. However, if the focus on causes and effects is not accompanied by process-oriented analysis, it may remain rather superficial and detached from actual business practices. In contrast, when there is a focus only on process, it is hard to arrive at an integrated diagnosis.

4.10 *The deliverable is a simulation of the habitat network*

The following is a list of the project simulation deliverables for a community-type societal project:

1. The simulation of the material environment (i.e., simulation of the local and/or network of habitat service systems, city simulation).
2. The simulation of information stored and calculated throughout the whole society. This includes the simulation of the economy.
3. The simulation of someone's life in a community-type city.

Together, a real-time virtual simulation provides collaborative adjustment and real-time understanding of changes to a living environment.

There are three usage cases for the simulation software:

1. The software may be used by engineering teams for system development.
2. The software may be used by the public for understanding.
3. The software may be used by the marketing team for promotion.
4. The software may be used by the relationship development team for promotion.

Objectives of the a software simulation include:

1. The user will access a virtual simulation of the real world environment as an occupant to look and walk around, to understand how that space may function.
2. The user will feel changes made to the virtual environment prior to those changes being made to the physical environment.

Essential software programs for simulation include, but are not limited to:

1. **City Engine** [esri.com] - Used to design procedural cities on a large scale.
2. **Unreal Engine** [unrealengine.com] - Used to apply virtual reality and real-time motion.
3. **Blender** [blender.org] - Used to create 3D models.
4. **Revit** [autodesk.com] - Used for architectural object information modeling.
5. **Rhino3D** [rhino3D.com] - Used for architectural object modeling.
6. Sketchup [sketchup.com] - Used for architectural object modeling.
7. **Simulink** [MathWorks.com] - MATLAB-based graphical programming environment for modeling, simulating and analyzing multidomain dynamical systems.
8. **Fusion 360** [autodesk.com] - CAD, CAM, and CAE object-product creation software for product design and development processes within a single tool. The software unifies product design, engineering, electronics, and manufacturing into a single platform.

4.10.1 What is necessarily demonstrated

For purposes of the functioning of a community-type society, as well as, positively influencing those who may be unaware of, or not understand the direction of a community-type society, it is necessary to demonstrate:

1. Demonstrate viability through engaging simulated experiences of life among community. Demonstrate the accountability of human life experience.
   A. Fictional story (film, audio, text).
   B. VR life simulation (virtual reality) of life experiences.
2. Demonstrate feasibility through accounting and simulation, and measurement therein. Demonstrate measurability.
   A. 3D computational simulation with 3D objects and process metadata.
3. Demonstrate how few people are required to provide for the needs of the population. Demonstrate integrated city systems.
4. Demonstrate how human demand is accounted for and supplied. Demonstrate a calculated decisioning system.
5. Demonstrate how the specification standards form the current state of the society. Demonstrate a unified design.
6. Demonstrate how information is experienced within the societal system. Demonstrate information accounting.
7. Demonstrate how resources flow through the societal system. Demonstrate resource accounting.
8. Demonstrate how the system works in time and with available resource by visualizing (at least, on a timeline) the system's calendar-scheduled operation:
   A. Visualize the current activities and future activities on the timeline.
   B. Visualize the current status of a project.
   C. Visualize all other projects that any given project relates to.
   D. Visualize all work packages in a project that has a time reference, such as phases, tasks, and milestones, as well as, relationships between them.
   E. The work packages can have a start date and due date.
   F. Milestones only have a due date.
   G. Visualize all work packages, phases, milestones, tasks, and bugs/issues in a timeline view.
   H. Visualize all precedes and proceeds between different work packages.

4.10.2 A demonstration experience

Several possible demonstration experiences may be produced, used, and updated:

1. **A "free access" demonstration experience**: A virtual experience or video showing (Read: simulating) people walking into access centers amongst gardens and acquiring products for free, or going to recreational locations and using services for free, or working on InterSystem team positions without hierarchy, while using a unified information system.
2. **A resource-based demonstration experience**: A virtual experience or video showing (Read: simulating) the flow of matter (resources) through a material environment sub-composed of objects usable to humans.

4.10.3 Guides to facilitate understanding

A set of materials for facilitating comprehension of the standards to a wider portion of the global population include, but are limited to:

1. Translations of the standard.
   A. Translation of the standards and supplemental deliverables into other languages.
2. Audio of the standard.
   A. Oral narration of the design specifications (i.e., turning them into an audiobook). Due to the continuously updated nature of the specifications, some of the content may be difficult to keep up to date in audio format when a human actor is involved in the narration.
   B. Software oral production of the specifications through a software application. Due to the complex technical nature of the information, pronunciation and grammar may be an issue in the automated vocalized production of the specifications.
3. Handbook/Guidebook for the standard
   A. Each standard will have a handbook version (or guidebook) to facilitate an understanding of the specification's content, and develop an interest in the project. These companion documents are used for quick reference and a concise overview.
4. Video guides for the specification
   A. Descriptive video media of the standards presented in a professional, personal, and visually appealing manner.

During development, there is likely to exist some combination of new societal construction and former societal transition.

4.10.3.1 The benefits of virtual reality simulation

Once the stuff of science fiction, virtual reality (VR) has arrived as a relatively affordable and mainstream consumer technology. VR is a new, complex form of communication, and as with any other medium of communication, it can be used to convey arguments and facilitate change in how individuals view the real world. It is a technology that can be used to demonstrate the feasibility of designs, and it will revolutionize how populations shares their standards for society. The vividness of virtual reality can give an audience a sense of immersion, enhance the emotional impact of a message, and bypass poorly constructed analytical arguments. Individuals no longer need to "tell" or "sell" people what one what is being propose; instead, it is now possible to immerse them in the environment and allow them to freely experience it (in a virtual environment) for themselves. Experiences within immersive virtual environments are more powerful than mere imagination (e.g., reading) in terms of information transfer and influence on actual thinking and behavior.

Through the use of VR people can walk around the community and immerse themselves in the experience of its complex operation. Not only will this be helpful to developers in simulating, testing and improving a system's design, but it is also a highly persuasive marketing tool. Imagine if community could freely share a virtual reality experience of what it would be like to tangibly live and participate in community, to experience as best can be experienced virtually that which is described by the specification standards of a community-type society. It will reveal that what is being proposed in text and model form is actually possible now in the real world. Though, in fact, what is being proposed has been possible for a number of decades.

This VR experience may help individuals come to a greater understanding of what the current modern socio-economic system actually removes from them by its ongoing existence. It may reveal how the current system limits their potential. Through a well-structured simulated experience (orientation), it is probable that developers can help the public reconsider maximizing their current situation in the market-State, and instead, facilitate a shift toward a greater action to what is truly important to them in life, which they may not even be able to well articulate. When people encounter a community-type (a.k.a., resource-based economy, RBE) direction for the first time, they often think about what this direction proposes in terms of what they will lose, rather than what they will gain. Although community is significantly more pleasant, fulfilling and generous than a market-State society, it is so different that people have a difficult time conceptualizing it, and immediately think about what will be absent.

If you want to change people's minds, and if they are on a different paradigm than you, if they identify themselves with a whole different set of presuppositions at a subconscious level, you will frequently not be able to change their mind by being rational. And, the more evidence you show them that is at variance with their fundamental paradigm, often, makes them angrier and more rigid, and so, we need a more eloquent and intelligently persuasive way of helping people re-visualize what is possible (and, what they may be missing out on).

Human senses provide access to the brain and by simulating the sensory environment of a community-type society through immersive virtual reality people will much more quickly get the perspective we are trying to convey. A virtual reality experience will facilitate rewiring of the brain toward what is possible in the present, and toward our broader, and more integrated worldview. Change on the scale that is required can only be realized when people see and experience a better way.

The experience of a different reality can physiologically change a person's mind. In other words, virtual reality can literally change our minds. Think about the way current media does that (possibly, in the Orwellian sense). It is important to take virtual reality seriously and to create a simulation of a socio-economic system that is inherently positive for all human and ecological life in its focus.

Wouldn't it be great to have a free, open and shared simulator of the community? Through such a simulator we could test out different operational designs, technologies and city configurations, and we could facilitate a personal exploration of the environment for others. A virtual simulation of community would give people a taste of the experience of a life of greater fulfillment. And then, after it is experienced virtually, one could go to our website and find the exact reasoning, designs, tools, and resources for the creation and duplication of the most up-to-date version of the community. When experienced, even virtually, I think most people in modern society will consider community a better way of living than the way they live now.

5 The project's definition

A.k.a., Formal concept of project proposal; project proposal overview, project document definition.

A project definition is a description of what the project has to achieve and how.

5.1 What is this project?

This project could be viewed as having the purpose of brining into operable existence a community-type society via an open, community-type societal [world-building] standard (known as the societal specification). This is a project proposing a testable societal [service] system. This project will result in the operation of a test-able, and therefrom, re-align-able, societal system.

This is a project, with an accompanying engineering structure, that exists to design, build, and operate a type of society with the following high-level, generalized characteristics:

1. Trade-less (moneyless) and coercion-less (non-authoritarian) production through unified information modeling (input-output service system modeling), intrinsic contribution, and common heritage resources.
2. Class-less access (Read: no higher or lower socio-economic classes) through a network of integrated habitat service systems that provide common, personal, and team access.
3. Highly automated in order to provide optimal socio-technical production efficiency, conserving resources and human effort.
4. Fulfillment-directed requirements enable optimal human life well-being and flourishing.
5. Regenerative design to organize the habitat in sustainable harmony with a larger ecological environment.

Simplistically, this is a unique project to create and sustain a highly automated, moneyless society, oriented toward human fulfillment and ecological sustainability. More broadly, the purpose of this project is to bring into existence a new type of societal system; a type of society representational of the highest optimization and expression of human potential and possibility.

In terms of information, the result of this project a societal design specification outlining a rational plan of coordinated societal-level action in life, as the potential and encoded frame of fulfillment ("good"), for anyone. Here, flourishing is contingent upon the comprehensive satisfaction (fulfillment) of the needs. Universal fulfillment of needs is the condition that allow embodied consciousness to express its capabilities freely.

QUESTION: *Without adequate conditions for the use of freedom (Read: to freely develop and express capabilities), what is the value of freedom?*

Once solution alternatives are present, a population can, together, select among the alternatives for that which is optimally in alignment with the populations fulfillment (given, that which is available). In other words, this is a project to design solutions to societal configuration, select and operate the optimal solution given what is known and available.

5.2 What problem does this project solve?

INSIGHT: *Quite possibly, the only real problems in life are the problems that are common to all of us. Therein, we need a common ("collective") response to the common problems concerning our species.*

Researchers use the term problem to describe a situation in which the current actual state and future desired states diverge; wherein, problem solving is converting an actual current state into a desired future state that is better (i.e., more desirable). Problems are opportunities. Individuals can take control of the meaning (e.g., outcome) of a problem. The only difference between "problems" and "opportunities" is the meaning given to them.

This project solves the problem of structuring information and controlling material transformations for the benefit of all of humankind; the creation of a unified socio-technical system that accounts for humanity and its environment. The system proposed by this project solves the problem of structuring and coordinating the iterative design and operation life-cycling of a human-habitat, fulfillment-service system that is likely to result in the state of all individuals of humanity continuously and consciously evolving toward their highest expression, for themselves and all others.

Additionally, in order for a developer (or funder) of the system to recognize the value of a specified solution to the problem, the following information sets must be known, each of which represents a search problem:

1. Who are the system accessors?
   A. Who are the users and operators of the system?
2. What is the system object?
   A. What is the intention for the existence of the system as an interfaceable object?
3. How hoes the system object process [newly acquired] content?
   A. What is the method by which transformations occur within the system?
4. Why is the outcome expected?
   A. What is the reasoning for selecting the current system object, as opposed to a different system object?

Society is a simplex (simple and complex) problem, wherein:

1. Simple problems are solvable with currently available data and tools (i.e., high current certainty due to current data; current solutions can be reconfigured to solve new problems). Therefore, the solution to the problem is simple.
2. Complex problems are solvable through the discovery of additional data and newly designed tools (i.e., low current certainty due to current data; current problems require altogether new solutions). Therefore, the solution to the problem is complex.
3. Simplex* problems are solvable with current data and tools, but still require research and new design because of artificial environmental limitations (e.g., limiting beliefs on the part of humans; current problems require a mixture of solution novelty and reconfiguration). Therefore, the solution to the problem is simplex.

**Note that the concept 'simplex' has additional meanings, which are detailed in The Auravana Project's FAQ.*

Additionally, in a socio-technical system there are two highly generalized forms of complexity:

1. Technical complexity concerns the physical nature of a problem situation. Technical complexity refers to the physically technical nature of reality.
2. Social complexity is associated with the relationships between the human users of a system. Social complexity refers to the consciously social nature of reality.

INSIGHT: *Complex societal problems are real-world problems, and real-world problems are complex societal problems.*

Resolving complexity in the design and operation of real-world socio-technical systems necessitates, at least:

1. Clearly explained starting conditions (goals and objectives).
2. Clearly defined requirements.
3. Clearly courses of action (methods and plans).
4. Here, 'clearly' means completely visualized and easily communicated, given a common language.

5.2.1 What are the problems with the configuration of early 21st century society?

This is a project plan that accounts for, and addresses, the largest and most common problems in modern 21st century society, including but not limited to:

1. Pollution.
2. Overcrowding.
3. Social suffering.
4. Unemployment (& lack of ability to contribute).
5. Poverty.
6. Education quality.
7. Low-quality fulfillment-service choices.
8. Significant separation in socio-economic access between the human individuals.
9. Political problems.
10. Etc.

All of these points of conflict, contention, and suffering are seen as interconnected at the societal (and planetary) level. The problems individuals experience in cities are intimately related to society as a whole. Technical problems within cities are related to society as a whole (e.g., technical problems of congestion, inefficiency, pollution) - technical issues become social issues, and social issue become technical issues -- individual issues become social issues, and social issues in feedback become individual issues.

5.2.2 What does the project propose as a better living situation?

This project proposes that people live in:

1. Community at the societal scale; have fulfillment throughout all life phases.
2. An appropriate sufficient state of human need fulfillment, without the need of money or coercion.
3. A moneyless, Stateless, classless society oriented toward human flourishing.
   A. Live without the requirement to earn and spend money.
   B. Live without the requirement to punish.
   C. Live without the requirement to live above or under others.
4. Conditions where flourishing and greater probable flow is possible.
5. Conditions where contribution develops and coordinates the planet's resources to provide abundance in access for everyone in the most sustainable way.
6. A habitat service environment with optimized access, given common access.
7. An integrated habitat systems where resources are economized.
8. A restorative environment where the ecological services are restored and optimized.
9. An environment of life-long learning, contribution, and leisure.

5.2.3 How does this project propose to solve the problem(s)?

QUESTION: *As planetary scale inhabitants, how are we going to work together for our mutual benefit?*

In part, the project proposes to solve the problem(s) through the development of an contribution-based information, decision, and material service support system. In order to completely solve the problem of societal design for mutual fulfillment, the problem and its solution must be modeled in a unified information system, and then, tested in materiality. At the highest-level, the modeling problem is one of societal intention, which directs a composition, generates a configuration, and sustains a coordination. The first step is to discover and concept model the core (axiomatic) systems of any human society. The second step is to compose and configure those systems to express the intention for the society. Whereupon, the model is tested in operation, and iterated therefrom.

How is society solved as a problem?
By asking getting passionate, questions, inquiring, resolving and synthesizing, then putting in effort together to construct and sustain:

1. How do we best, select a societal system and plan there that works for the benefit of everyone?
2. How do we, fit into our surroundings?
3. How do we, identify the effects of actions?
4. Does what we do, match (align) with the things we need?
5. How do we improve (i.e., what are the questions to ask to make some system better)?
   A. What is the system 's purpose (i.e., what is it for; what is its function; what)?
   B. How does it serve people (i.e., what is its benefit; what is its value; why)?
6. How do we best:
   A. Solve collective action problems
   B. Acquire empirical data about the world (a.k.a., make empirical findings about the world). Empirically review and validate.
7. Most other problems are a result of these problems.

5.3 *What is the project's direction?*

This project proposes 'access' as a definition of direction (i.e., 'access' is a definable direction). All individuals in the community desire access to the following interfaces, all of which can be measured and designed in common:

1. A high quality of life, given what is available.
2. A high-standard of living, given what is known.
3. A life where the human individual flourishes

together.
4. An objective, accountable, and grounded life-coherent service system that meets all human need.
5. A common life-ground of information and material that forms the structuring of our higher capacities (our higher potential selves).
6. Access to our own [self-integrating] source of power and creativity.
7. A society formulated in exact and understandable terms.

Access to genuinely understandable and testable fulfillment requires realization of the following values that are at the core of an adaptive and helpful orienting [navigational] system:

1. Access to **freedom** [to express capabilities].
   A. What is freedom to the individual?
   B. What is the likelihood of the fallibility of fulfillment?
2. Through **justice** [as universal need fulfillment, required by all human embodied consciousness].
   A. What is freedom to those individuals who cannot make use of it?
3. By means of **efficiency** [in our common actions] within a common ecology.
   A. How does optimization generate freedom (free time)?

Together, humanity visualizes a shared understanding of what makes life [most] meaningful. What is most mutually beneficial for all of our lives?

Together,

1. Humanity will construct a shared vision, and the resulting societal solution will be tested to express these values (conditions of the vision).
2. Humanity will not execute upon a societal solution until it visually expresses these values (conditions, principles, inquiries, etc.).

Social systems lower their entropy by cooperating and caring. Social systems raise their entropy and de-evolve through fear. If there is fear, there is no trust, if there is no trust, there is a not a lot of cooperation. A societal system expression without the value conditions of 'cooperation' and 'caring' is likely to structure a sub-optimal state of fulfillment. Humanity can come together to share a common purpose, our common interest, our need fulfillment and the care-taking of the ecology. Then, through greater information coordination there is the potential to safely access more extensive forms of technical function.

In application, value functions are qualified boundary constraints (encodings) that resolve an issued decision toward a particular direction of intention. A value is a specifically desired orientational state (or "preference") among all potential attributes, states, or preferences.

The two axiomatic boundary constraints are:

1. **Specific limits that must be met.**
   A. For example, there are ten people in the population, and ten people must eat. This project proposes, in the Decision System, a set of social inquiries, social thresholds by which tasks (solutions) are decidedly assigned resources, and often, effort, on the part of the InterSystem Team.
2. **Specific limits that cannot be exceeded.**
   A. For example, there are a countable number of fish in the sea, and a rate at which they re-population; to ensure continued access to fish as a nutrient source, then there are only so many fish that can be taken out of the ocean during some given duration, least the fish population not be capable of recovering its population.

In order to effectively resolve these boundary conditions in the design and operation of any new system, decision analysis is required (i.e., a decision system is necessary). In the real world, it is assumed that there are potential impacts to others in an environment, given one's own decisioning. Decisioning in the real world necessitates a process [method] for identifying and prioritization a single selection (e.g., state or solution).

5.4 What defines the project's vision?

A vision is a picture of the future.

1. The project envisions a network of walking community garden cities.
   A. More completely, the project envisions an informational-spatial interface network of walking community garden of sub-global habitats.

More simply, the project envisions:

1. A life-work environment where most of the population lives in integrated family- and garden-oriented smart cities with life-work lifestyles based on optimizing life fulfillment.
2. A population-wide access system with no trade, no market, no currency, no money, no finance, no economic exchange.
3. A high-degree of technical automation with a concurrently high-degree of individual challenge to promote a lifestyle of optimal flow and well-being.

5.4.1 Vision statement?

A vision is a desired future state. A vision statement describes an organizations aspirations (i.e., why does the organization exists; what is it en-visioning?).

- Vision statement - describes the intentions, aspirations of the organization.

Among community, planetary resources are seen for what they are, as the common heritage of all the planet's people. These resources are the 'life' satisfiers of every human; the sustainers of human fulfillment, and a sub-element of a larger total ecology that sustains (or, does not sustain) our individual well-being. Herein, fulfillment services are selected [as solutions] to sustain, (rather than predation upon) social and ecological [life-]support-systems. Resources and societal-level requirements are seen as common in a community-type society.

> **QUESTIONS:** *How can any individual truly be fulfilled in life? How can we create lives that are truly worth living, given that these lives are knowably finite (i.e., come to an end)?*

Herein, the concept of fulfillment has, among others, the following sub-conceptions (the different sub-dimensions of fulfillment at the societal-level):

1. Human:
   A. Need (there exist conscious embodied entities) = fulfillment.
2. Engineering:
   A. Requirement (the need is connected to the some direct output, via a process) = fulfillment.
3. Social:
   A. Well-being (the requirement is connected to the individually common human experience of well-being) = fulfillment.
4. Habitat:
   A. Service (the ecology is connected to as a service) = fulfillment.
5. Planet:
   A. Ecology (the potential of human life is connected to as a planet) = fulfillment.
6. Life:
   A. Potential = fulfillment.

5.5 What defines the project's mission?

A.k.a., What is the directive of this project?

A mission is, in part, why 'do' what is to be done (i.e., why do the project's work?), so that it can be done well. In application, the concept 'mission' means 'task' together with 'purpose', clearly indicates the action to be taken and the reason. In common usage, especially when applied to lower level organizations, an activity selected/ assigned to an individual or unit is a, 'task' (or, mission).

- **The project's mission** is a global network of operationally localized habitat service systems that construct, prioritize, and complete tasks based upon a conditional set of value decided inquires/ criteria and a unified information [construction] system.

5.5.1 Mission statement?

A mission statement describes an organizations purpose (i.e., why does the organization exists, re-directing it).

1. The project's mission statement is to bring into "living" existence a global network of integrated city systems in which human individuals 'live' in fulfillment with one another and the larger ecosystem.
   A. 'Living' is to continuously adapt and move intentionally against gravity.
   B. 'Live' is to have human need requirements met.

5.6 What are the project's expected outcome(s)?

An expected outcome is the intention[al criteria set before action that] results in a functional and/or conditional state of the environment. What results are expected?

It is expected that the project will result in:

1. A societal system configuration that will verifiably be the best (optimal) for everyone, given the information and material availability.
2. A societal system reduced in suffering, adaptive toward an optimal state of flow (of love) in each moment of our individual lives.

5.7 What is the project expected to produce?

The project needs to explain:

1. **What is the product or service to be produced (and, offered to the public-market-State)?**
   A. The service to be offered by this proposal is community at the societal scale. The products are: a unified set of societal-level community standards and a network of community habitats. Necessarily, the transition to community-type living is also a service.
2. **Technical background on the product or service?**
   A. The service is a societal-level operation where human needs are optimally met given what is known and what is available. By optimally meeting human needs, humans experience

well-being and are most likely to flourish. This product/service includes a set of societal knowledge and agreement standards that orient material habitat operations where the global population lives with sufficient need fulfillment.

3. **Market for the product or service?**
   A. All humans have needs. Given global coordination of information and resources, there is no need to trade to meet needs. The fulfillment of others' needs affects the fulfillment of one's own needs. All humans desire well-being. Together, it is possible to co-operate society where all humans flourish. Many humans in the 21st century suffer due to created by the market-State. The ecology of the 21st century is significantly harmed by the practices of the market-State. Everyone desires community; some are living in worse conditions than others, and are in greater need of community [operations at the societal scale].
4. **Process by which the product or service is produced?**
   A. The service (community) is produced by the development of community standards and the construction of a community network of habitat service systems. Societal specifications standards are developed by working groups who work for a standards setting organization. Standards development requires collaborative design and distribution software. The construction and operation of a habitat network comes from production and regulation by market-State organizations. Habitats require light- and heavy-industrial production. Community also necessitates education, a service that may be provided through university education. On a day-to-day basis, the standards will be developed, transition operations will be conducted to move people and resources into community, and habitat team operations will occur to meet human needs within the habitat network.
5. **What facilities and personnel are needed for the operation?**
   A. There is the requirement for the construction of a habitat facility as a minimum viable product.
   B. There is a requirement for the following types of personnel: project coordinators, contribution service personnel, information development personnel, habitat team personnel, and relationship development (transition team) personnel. Because information working groups work primarily through software, there is no need for a specific physical facility for them to work in (of course, internet computing technology requires a physical facility). Habitat teams work in a populated area with fixed physical boundaries (i.e., human settlement) -- a physical location, with a human population, where the populations needs are met through [habitat/ecological] services. In the early 21st century, market productions will be required to construct and operate both the standards setting organization and the habitat network. Additionally, State regulations (policies) and State workers will be required to create new habitats and transition old settlements to those representative of community.
6. **What is the projected revenue from the operation?**
   A. Community does not trade tokens; there is no expectation of financial revenue; instead, there is only expectation of human fulfillment.
7. **What are the qualifications and background of the team?**
   A. The team must be educated sufficiently and/or qualified or sufficiently experienced to complete the work. The team

5.8 What are the preliminary milestones?

Top-level milestones include, but may not be limited to:

1. Deliverable of a unified societal concept of operation in the form of a set of societal system standards. [COMPLETE]
2. Deliverable of coordinated updates to the societal standard to bring it up-to-date given newly available information. Note here that a standard's filename suffix identifier identifies the revision: SSS-..-...-###
3. Deliverable of a yearly integrated commit to republish the standard after as a final [edition] working group integration point. Note here that a standard's filename internal identifier identifies the edition: SSS-..-###-...
4. Deliverable of sufficient number of individuals capable of constructing and operation the first city and its informational system (or, some portion of it).
5. Deliverable of sufficient financial resources and legal contracts to supply the requirements of constructing the first city and its informational system, and not just some portion of it.
   A. Deliverable of actual resources for construction through to operation.
6. Deliverable of sufficient jurisdictional (legal) agreement in writing that construction and operation of the first city and its informational

system is safely certain.
7. Deliverable of sufficiently operating habitat service system (i.e., city system) and societal information operating system.

5.9 What are other common naming classifications of this type of society?

Egalitarian individualistic:

1. Respect for individual decisions and autonomy.
2. Sharing access (to common resources) without wealth disparity.
3. Systems in place to meet all needs.
4. No motivation to accumulate excess (or "be greedy").
5. There is not coercion.
6. The hierarchy is not authoritarian, but one of choice, expertise, and accountability.
7. Holistic in nature accounting for both the individual (me) and the group (we). In early 21st century society, people are taught to think its one or the other and there can't be both accounted for simultaneously.

5.10 What defines individual behavior in the project?

All 'behavior' is 'motion'. It is possible to model motion commonly (i.e., it is possible to model our common behaviors). In a human body, motion feeds-back to consciousness a spectrum of feelings.

As feeling entities, all human are individually (i.e., "we are all, individually) seeking fulfillment and relief from suffering. Notice the direction of flow that feelings represent -- into fulfillment and out of suffering. This is not to say that individuals want mere pleasure or the easiest possible life. Much of what growth to an individual consciousness entails feels like a struggle, as growth through challenge.

Optimal human behavioral development and societal advance occurs,

1. By optimizing human service fulfillment, without which individuals suffer loss of life capacity by measurable degree of regression dis-allowance (dis-advantage).
2. Through elimination of unnecessary suffering from life capacity reduction due to deprivation of life fulfillment (i.e., "life goods").

5.11 What are the primary societal project tasks?

This project is sub-divided into a set of axiomatic tasks representing a parallel project-level life-cycle, which is, to design, develop, and live in an emergent, community-type society in time with available resources, together.

- The first phase of project implementation initiates actions to measure the existing environment in order to identify the environmental situation in which the project exists.

The following are axiomatic task categories (informational phases) for this societal building project:

1. **Project coordination and planning**, including multiple sub-project and project plans.
    - **THE PROJECT PLAN**, which details the *how* and *when* of what is to be constructed into "our" lives.
2. **Societal systems development engineering**, including the design and development of the unified societal information system and internal habitat service systems (cities). This supra-process involves the Project's primary sub-processes of: **requirements engineering** (specifying and sequencing requirements), **designing** (preliminary to detailed and conceptual to technical), and **prototyping** through to fully **developing**.
    - **THE UNIFIED SOCIETAL SYSTEM SPECIFICATION**, which details the *why* and *what* and *how*.
3. **Societal systems operations engineering**, including operating and monitoring the existent unified societal information system and the material habitat service systems (network of cities) therein.
    - **THE UNIFIED SOCIETAL SYSTEM SCHEDULED EXECUTION** by the societal InterSystem Team, which details the *why*, *what, and when*.
4. **Our individual experience in society**.
    - **THE INDIVIDUAL'S LIFESTYLE.**

Here, society could be viewed as an intentionally (specifically) planned and scheduled lifestyle.

The planning of configured access to the habitat defines societal-level planning. A control[lable] volume of ecology, known as a 'habitat', is identified, both informationally and positionally. Resource flows into the control volume [habitat service system] and output emissions from the control volume [habitat service system] are designed and measured. Data integration allows for the capability of a multi-city, habitat network operations service environment where all resources and access opportunities are shared in common.

> **NOTE:** *In networks, the size of a particular change does not necessarily indicate the scope of its effect, and care must be taken to avoid changes that maximize local benefits at the expense of global effects.*

5.11.1 Society is a progressive emergence

At the societal level, emergence could be viewed as **progressive elaboration** - the system (e.g., society) is progressively elaborate as the project's information system develops, becoming increasingly well informed and unified as time and iteration occurs.

5.11.2 Societal-level planning

APHORISM: *Those problems which are not acknowledged are generally repeated.*

Together, a social population (a society) can plan their next action(s); the population can plan the next change to the [state of the] environment. At the "highest" conceptual-level, this plan is expressed as the unified 'societal information system'. At the material-level, this plan is expressed as the controlled 'habitat service system' (i.e., the city-system network existing within a larger wild and decidedly care-taken ecological system). A cooperative society plans their information system; and that unified plan is sub-composed of a materialized, environmental service system.

5.11.3 Society is a project task

INSIGHT: *Society is a projection, as a systems engineering project.*

This societal building project may be sub-organized into the following parallel task domains, where contribution is necessary:

1. **STEERING COMMITTEE SUB-PROJECT**, because this proposed society will come into existence when the market-State is highly present on the planet.
   A. Market and State Interface - contractual and jurisdictional agreements.
2. **SOCIETAL ENGINEERING SUB-PROJECTS**, because this proposed society will iterate through existence when usefully contributed work is done.
   A. Societal system design (specifications).
   B. Societal system implementation (operations).
   C. Human system inclusion (population migrations into community-city network).
   D. Habitat system operation (intersystem project teams complete service requirements to meet the needs of all human users).

5.11.4 Human life-cycle analysis

The purpose of life-cycle analysis is to acquire sufficient information to determine and select actions that will meet objectives of adapting and optimizing life over iteration, cycles of time in an uncertain environment. The output of a life-cycle analysis is a situational input into decisioning.

Human life-cycle analysis is a three-component process:

1. **Inventory analysis** (needs, requirements) as the identification and quantification of environmental signals and human receptor for those signals.
   A. Here, needs [inventory] are often seen as part of the problem domain, whereas requirements [inventory] are considered part of the solutions domain.
2. **Impact analysis** as the technical qualitative and quantitative characterization and assessment of the consequences of resource use and environmental releases.
   A. Here, issues are often seen as part of the problem domain, whereas objectives are considered part of the solutions domain.
3. **Improvement analysis** as the evaluation and implementation of opportunities to reduce environmental burdens.
   A. Here, values are often seen as part of the problem domain, whereas conditions are considered part of the solution domain.

5.11.5 What is a human quality standard?

A.k.a., What is the standard for human quality?

Progress is the development of factual quality standards for human society, as those standards that define and explain what humans require, and how to optimally coordinate the fulfillment of those requirements, given what was known available at the time the standard was synthesized. What is sought as a goal, as [mutual] progress, is the meaningful improvement of the well-being of each individual in the short and long-term. The quality that everyone deserves is the best that humanity has to offer as a planetary civilization.

5.12 What does humanity commonly desire out of an engineered societal system?

This project proposes engineering as the primary method of project operation. This method structures 'how is this project' to be carried out. This project is to be carried out in the most ordered, organized, and prices manner possible through systems science engineering.

Herein, if a society were viewed as an engineering, safety, and provisioning service for the fulfilment all of humanity (i.e., for all planetary human users), then it would likely maintain the characteristics of:

1. A planned societal system.
2. A coordinated societal system.
3. A cooperative, multi-user and decision-supported environment.
4. A model of society most accurately aligned with human fulfillment (given what is known).
5. A unified societal system with a set of local habitat

service systems (i.e., cities) forming an [operational] global habitat service system network.
6. A society oriented in its intended design toward [the felt experience of] optimum access to individual human fulfillment.

In order for a social population to function "well" (Read: cooperate toward common fulfillment), it needs to establish and maintain a common ground of shared meaning, including mutually shared data, knowledge, values, and vocabulary.

In early 21st century society, different "fields of expertise" may use different terms to mean the same thing. However, when [people from] different fields converge in a common setting (Read: into community), a common ground of meaning must be established. The necessity of common ground is important for at least two additional reasons for sharing the community's knowledge with others outside the community-type society, and "for developing a shared understanding of complex systems of ideas that the community develops.

5.13 What might an engineer ask first about this project?

An engineer who looks at the problem of society might ask, in concern to technology, "What does humanity need"? And, an engineer would likely respond, "Humanity needs a helpful socio-technical system, a unified information/habitat service system". The engineer might think next of conditions. At a social level, "humans desire to be helpful to one another". Thus, a materialized (from planning) socio-technical system may (or may not) coordinate and facilitate human helpfulness. Helpfulness is a sign of togetherness, as is sharing; both of which represent caring, which occurs between others (at the highest population-level), among a unified group who share commonality.

> **QUESTION:** *How might one societal solution be capable of orienting toward greater (or lesser) states of fulfillment than another?*

5.14 What is the 'socio-technical' view of society?

> *A.k.a., Ultra-large-scale (ULS) hybrid-cognition-intensive, cyber-human-hybrid-autonomous, cyber-socio-technical systems (HCI-STS/STR).*

A socio-technical system is a social system with technical implications and in conjunction, the technical system has social implications. Technical systems with social implications and social systems with implications for technical systems. Implementation runs both ways. Every system humanity builds to interface with the embodied world of human materiality also reconfigures that embodied space, altering cognitive and social practices. This happens because implementation encodes a particular formulation of the desire for effectively computability. A desire that humans reciprocate when they engage with that system. A socio-technical view is a view where need is resolved through socio-technical [service] production.

All human organizations comprise of two interdependent systems, referred to together commonly as the 'socio-technical system':

1. A social system, due to the presence of a living organismal population (humankind).
2. A technical system, due to the conscious design and creation of material organizations that automate service fulfillment (i.e., tools of increasing cognitive information about an extant reality that allows for their construction, such as the creation of a hammer in history to the historical creation of the chain saw. A technical system produces technology for a social system; that technology is used to automate and ephemeralize required service fulfillment in order to produce a higher order stability in access, thus more free time to pursue higher capacities that humanity has the potential of expressing and otherwise actualizing.

In community, there is an integrated [human] socio-technical system that can be understood and designed. It can be understood and designed in part, or in whole, and its actualization has real world consequence for conscious living beings (until it doesn't). Any ecological or human societal system could be considered a socio-technical system because it combines social organisms (humans) with technology. Changes in one system affect the other system.

For example, the rethinking of 'dishwashing' as a system might make it more convenient to clean dishes (for everyone), as well as solving one of the basic survival problems (of everyone), water conservation and processing.

A socio-technical system necessarily has:

1. **Social interactions** can be thought of as interactions with people.
2. **Services** can be thought of as a parallel category of interaction between humans, [logical] process, and [material] objects [in common access]. Here, technology is a service.

Change coordination (change management) is a component of a quality assurance system that ensures all changes are accompanied by:

1. **Support** – developers, organization, user.
2. **Control** – specifications, documents, algorithms, and others.

3. **Service** – to support people.

Societies socio-technical information flow, in the form of projects, involves the flow of different resource-types (which are common to all individuals):

1. **Information flows** (a.k.a., computation and visualization).
2. **Material flows** (a.k.a., material science and positional mechanics).
3. **Time flows** (a.k.a., coordination and scheduling).

5.14.1 Technology

Technology is the mechanical and informational processes by which things function. Technology is merely how things made and done. Technology reflects the engineers designers and programmers who make it. Made technology is a reflection of the makers knowledge. Technology extends human capability (i.e., machines extend human capability).

> **APHORISM:** *We can have the best possible 'how', but if we mess up our 'why' or 'what we might do more damage than good.*

In this subject, Technology is the know-how and creative processes that may assist people to utilise tools, resources and systems to solve problems and to enhance control over the natural and made environment in an endeavour to improve the human condition.

Technology is the art of technical [systematic] servicing. Or, technology is the study of the potential of an object [in service]. The study of in-service objects. Other definitions for technology include:

1. The purposeful application of knowledge, experience and resources to create products and processes that meet human needs.
2. The study of systems of making or producing.
3. Products, knowledge and skills working together to improve the human condition.

5.14.2 Socio-technical issue coordination

The common elements of a socio-technically coordinated societal system include:

1. Social information composition.
   A. Issue situation.
2. Technical decision planning.
   A. Issue planning.
3. Technical decision identification.
   A. Issue identification.
4. Technical decision analysis.
   A. Issue analysis.
5. Technical decision solution.
   A. Issue solution.
6. Technical solution execution.
   A. Solution execution.
7. Technical solution monitoring.
   A. Issue monitoring.

5.14.3 Service and asset production

There are two primary types of service (in a total asset ecosystem); wherein, the asset types are:

1. **Process/activity/operation/concept** - Service is the product (service is the asset).
2. **Object/product/resource/shape** - Service to support the product (the shaped asset is an object; the asset is the service to support the object).

Simply, the common production types are:

1. **Mass production** - the 'batch' size is infinite.
2. **Batch production** - the 'batch' size covers a range characterized by a finite number.
3. **One-of-kind production** - the 'batch' size is one.

Simply, the common production scale types are:

1. Production [selected 'solution batch'] for the **local** HSS (local city).
2. Production [selected 'solution batch'] for the **global** HSS (city network).

5.14.4 Societal multi-level design modeling

A society's multi-level design could be modelled as a configuration of four levels:

1. **Product-technology systems (technological product systems)** - physical objects that originate from a human action or machine process and exist as part of a service system. As these objects are made up of technical components, the term 'product-technology system' is used. This refers to tangible, inextricably linked technical systems, physically present in place and time. With most of these artefacts, you could 'drop them on your toes'. Product-technology systems generally fulfil one or more clearly distinguishable functions. A system dysfunction occurs as soon as one or more technical components are missing.
2. **Service-product systems (Habitat service system)** - built of physical as well as organizational components, which form a united and cohesive whole that together fulfils a specific function, usually definable in time and place. The system fulfils one or more clearly defined functions that can no longer be performed if one of the technical or organizational components is missing.
3. **Socio-technical systems (Societal Sub-Systems)** - the combination of information systems that

fulfill societal functioning. Changes that take place at this level are often referred to as a 'system innovation', which can be defined as 'a large-scale transformation in the way societal functions are fulfilled'.

4. **Societally experienced system(s)** - the population (community) of people living through a particular societal design, including the sharing of values and understandings.

5.14.4.1 Why is multi-level design modeling necessary?

Multi-level design modeling is necessary in a real world socio-technical systems for safety:

1. Navigational framing (social system).
2. Generative design (decision system).
3. Constructed operation (material system).
4. Expressed living (lifestyle system).

5.15 What is a real world, socio-technical systems engineering solution?

The real world community model is the society's highest level [real world] data [structuring] model, and it is detailed in the Decision System Specification (where resolutions are determined). The real world community model is a socio-technical systems engineering model. The socio-technical systems model that generates and records potential, and instantiated, societal solutions. Currently, the community specification (per the Decision System) has assigned the name 'real world community model' to that highest-level societal solution model that visualizes (represents) the system and sub-system conception of the unified societal system.

In societal engineering, everything is an understood, or an understandable, expression of the societal system, which requires of the observer the ability to think systematically and have systematic access to relevant information.

Socio-technical systems engineering refers to the design and deployment of a societal system. Socio-technical

Society does not only require technical-economic interventions, but social ones as well. The idea of socio-technical systems engineering refers, in part, to the engineering of the interaction between conscious beings who persist together in a common material world. There is an interaction between consciousness and an environment, and because, there is intention to survive and thrive (i.e., enhance life capability), then there is also the cognitive presence of [material] 'usability'. Technology is automated functioning usability. Technology is usable for various orientations: from generating fulfillment, and doing so more rapidly, to generating conditions of suffering, and doing so more rapidly.

Humans have something resembling 'needs' in society, of a social and technical nature. Project engineering may be applied to account for the completion of these needs. In a society structured through project-engineering, there is a requirement for a common decisioning procedure (a decision model, protocol, algorithm) to execute control, the 'controller' resolves decisions common to all individuals (Read: socio-parallel solution inquiry). In this proposal, there is a social control decisioning (projects) and a technical control decisioning (technical solutions). Engineering solution decisions (Read: technical solution inquiry) provides all potential workable solutions, ranked according to societal and organizational engineering objectives (a.k.a., conditions, constraints). The social organizational inquiry determines and selects for execution upon by InterSystem teams (into community existence) the optimal engineering solution, given that which is available. This social conditioning is affective at all levels, because it is the individual among the social where knowledge and access is shared (though sharing may be restricted and manipulated under some, less fulfilled, socio-technical contexts).

> **NOTE:** *In the real world, a life-coherent organization is one in which the component parts are coordinated toward a common life objective (life fulfillment).*

A socio-technical service system is characterized as:

1. **A Hybrid of:**
   A. A socio-technical system is a 'hybrid' type of system in the context that its components come from (at least) two different categories of things: some components are ordinary material, hardware, and/or software objects, whereas the other category is that of 'human' life-beings. Note that most socio-technical systems also contain elements from a third category, a category consisting of information (abstract entities).

In application the socio-technical system layers include:

1. **Human:**
   A. Socio-technical systems involve humans both in the role of operators and in the role of users. Operators are sub-systems of the larger system in which humans contribute (perform) their operating work. Users benefit (or are expected to benefit) from the contribution of human operators. Humans are 'free' (type of access) to use the system as a service, in the case of a socio-technical engineering, to participate in its sustained creation.
2. **Technology:**
   A. A proper functioning socio-technical system requires the co-ordination of the actions of all systems involved (coordinators, developers,

operators, and users). Technological development and application will usually be accomplished through procedures (protocols/ rules), and the design of such procedures (whether machine or human) is therefore an integral element of the task of designing a service system.

3. **Information:**
   A. A human decision to follow a particular rule requires, first of all, an analysis that the situation is one where the rule applies. But even when an operator decides that a particular rule applies, he or she can also be expected to perform an analysis as to whether or not it is in the person's interest to follow the rule. Often, this process of analysis is known as interpretational freedom. The history of technology consists to a large extent in attempts to remove the 'friction' in the system that is caused by the (interpretational) freedom of operators, and many if not most of these attempts have been successful. Here, it important to consider both: (1) thinking better about the sort of instructions that operators receive, and (2) simply remove the [human] operators completely. Operators are everywhere and continuously being replaced by hardware-software systems. This second option is of course no panacea: hardware-software systems can fail as well, even if differently.
4. **Resources:**
   A. All real-world physical system are composed of material resources, including the bodies hof humans and other life-forms.

A societal system represents a broad class of sub-systems where operational [decision] protocols and team procedures form a unified operating [service] system of individual "stakeholders" who live together in a living system with knowledge of physical "natural law" processes.

A city is an engineered socio-technical system; a [globally and locally unified] human service fulfillment platform. A [community-type] habitat service system is an environment where access and services are available for free.

In general, complex machines work in the same way as organisms. In a complex machine, as in an organism, there is a sensory input, expression output boundary, with a processor inside. In organisms, the sensors measure life-relevant data (as in any system, sensors measure system-relevant data).

In order to effectively construct real-world socio-technical systems, service systems require:

1. Sensors.
2. Processor.
3. Expression interface.

In community, the user places requests for service [output] on the unified information service system, and the habitat service system responds to the users demand.

'Negative' requirements are factors in a living organism's environment that prevent it from surviving there, or limit its highest potential development, there. Those factors are called 'limiting factors'. They include soils, temperature, water, sunlight and physical barriers. Physical barriers may include landforms and water bodies. They often prevent a living organism from moving to another place when conditions get bad in their regular habitat. Real world socio-technical systems must account for real world sources of information about the state of the dynamic ecological habitat, including but not limited to:

1. Habitat temperature.
2. Habitat nutrient profile.
3. Habitat air.
4. Habitat water.
5. Habitat sunlight.

5.16 What would a real-world, socio-technical systems engineering solution visually look like?

At a high-level, a unified societal system solution may look like an information structure with the following data model:

1. Ecological life service support systems
   A. Habitat service system
      1. Habitat life support service systems
      2. Habitat technology support service systems
      3. Habitat facility support service systems
2. Societal project information support [Plan] system
   A. Social System.
      1. Direction.
      2. Orientation.
      3. Approach.
   B. Decision System.
      1. Life support service system priority.
      2. Technical service system priority.
      3. Facility service system priority.
   C. Lifestyle System.
      1. Education life phase.
      2. Contribution life phase.
      3. Leisure life phase.
   D. Material System.
      1. Habitat service system network (global HSS).
      2. Habitat "city" service system (local HSS).
      3. Material system operational processes.

4. Spatial interface constructions.

> *Herein, for every complex service there is a network of sub-services, wherein and throughout there exists the condition of equal access to all that humanity has to offer the rest of humanity, by sharing without a trade- or coercion-relationship.*

If a system comprises interrelated parts contained within a boundary serving one or more functions within an environment, then humans are both systems themselves, as well as parts of larger systems. Here, socially contributive interactions to the structure and usage of services primarily occurs as part of an InterSystem Teams (i.e., Accountable InterSystem Teams primarily do the work to develop and maintain services):

1. Life support intersystem team.
2. Technology support intersystem team.
3. Exploratory support intersystem team.
4. Decision support intersystem team.
5. Standards development intersystem team.

If society is a moving vehicle (an analogy), then toward what direction is the vehicle pointed and heading. It is essential to figure out which direction that vehicle is to be pointed. If it is pointed at fulfillment, then flourishing for humanity is likely. The appropriate power, steering, and destination are all important to building and maintaining fulfillment at the societal scale. A human transport vehicle is a micro socio-technical system. Societal engineering is clearly a socio-technical, and not simply a technical, or simply a social, problem.

In order to produce a socio-technical system,

1. Collect human requirement measurements (metrics & benchmarks).
2. Model the world and potential objects in the world.
3. Synthesize uniquely attributable habitat service system [world] designs.
4. Analyze habitat service system [world] designs.
5. Select optimal habitat service system [world] given an objectively measurable set, which is executed through a material operation (process).

What is an 'economy' within a unified societal system oriented toward human fulfillment and ecological well-being. An economy is a sub-set of nature, a habitat service system - a harnessing of human technology to the larger planetary and cosmic ecosystem to facilitate our own fulfillment. An economy could be said to be a system of resource flow and transformation that produces life services and life "goods" (life requirement results), and not life "bads" (e.g., externalities, unnecessary suffering and artificial limitation), over time.

- The physical environment where an organism lives is called a 'habitat'. A 'city' is a controlled 'habitat'. An 'economy' is the current (input-output) transport configuration of all resources in the 'habitat'.

The social meanings that people attach to environments through their interactions and ongoing socialization play an important role in determining human behavioral responses. This outlines the important role of the living area serving the functions of the human needs and actions.

The facility and life support service systems are support for human survival and flourishing, and that support is expressed through the operation of a set of [support] services. These services operate together, for the betterment of everyone, in order to provide a three point platform upon which a stable society may manifest and grow. Therein, each services operates through a set of common (to all appropriate systems) operational processes, that prioritize and triage resources and tasks.

Humans are a living system, and individual humans are a social organisms with complex communication and information processing capabilities who group together for mutual benefit (e.g., shared food, values, challenges). Such groups constitute social systems, and they become socio-technical systems naturally through technology.

> **INSIGHT:** *We are a part of the systems we build, and therein, they build us too.*

5.16.1 Societal information system de-composition

Given the information available, any society may be informationally sub-composed from unification into four divisions of life-cycling experience, for any individual of the societal population:

1. Social.
2. Decision.
3. Material.
4. Lifestyle.

Although integration operations occur continuously in a unified information system, there are methods unique to each sub-structural system, that organize its composition.

1. Social system core methods:
    A. The core discovery method is that of science.
    B. The core reasoning method is that of logic.
    C. The core orienting method is that of value.
    D. The core directing method is that of testable goal intentions.
    E. The core life method of social memory is that of data storage and retrieval.
2. Decision system core methods
    A. The core decisioning method is that of integration (of sufficient information to resolve a specification, tested to solving a social issue that

generated a requirement for the decision).
  B. The core temporally coordinated execution method of projects.
  C. The core positionally technical solution method of engineering.
3. Material system core methods:
  A. The core materializing method is that of material cycling (more commonly, production and recycling).
  B. The core material method of access is that of a service interface operation.
  C. The core material interface support [infrastructural] method is that of service operations.
4. Lifestyle system core method:
  A. The core life method is that of the 'flow' life-cycle.
  B. The core life method is an entrainment alignment to natural cycles.

5.16.2 Simplified synthesis of a community-type society

The societal informational sub-structural view includes (social, decision, material, and lifestyle):

1. [Social] Data - situational issue.
  A. [Social] Knowledge - socio-technical understand ability.
  B. Technical knowledge - standards.
  C. Social knowledge - values.
  D. [Decision] Objective principles - objectives and requirements.
  E. [Decision] Algorithm/program - software.
  F. [Decision] Computation - computing.
  G. [Material] Construction - materialization.
  H. [Material] Materials - resources.
  I. [Material] Interface - service.
  J. [Lifestyle] Sensor - survey.
  K. [Lifestyle] Indicator - indicate cycles and issues.
  L. [Lifestyle] Evaluator - evaluate service and experience.

5.16.2.1 Briefly, how does design occur?

In community, design occurs via specific methods, given what is known:

1. How does design occur (what is a social design, social standard)?
  A. In community, social in the context of societal design means that the design considers the whole [societal] system of life support and socio-technical functioning, in terms of how the different machines and services interface with one another and humans (eventually forming the exploratory support service). Different machines can function as modules in a wide array of integrated systems.
  B. In community, design occurs through a unified, project-engineering integration method.
2. How does [re-]alignment occur (what is technical design, technical standard)?
  A. In community (or, any society), a decision system controls (planning and executing) the direction of alignment.
    1. Control direction.
    2. Planned direction alignment (selected solution).
    3. Executed action/task to direct alignment (accountable contribution).
    4. Surveyed resulting alignment (user-developer feedback).
    5. Evaluate alignment data (determine situation).
    6. Plan direction alignment (selecting solution).

5.16.2.2 Briefly, what is decision control?

A decision system controls (planning and executing) the direction of alignment:

1. Control direction of materiality.
2. Planned direction alignment (selected solution).
3. Executed action/task to direct alignment (accountable contribution).
4. Surveyed resulting alignment (user-developer feedback).
5. Evaluate alignment data (determine situation).
6. Plan direction alignment (selecting solution).

The decision [construction] system structural controls:

1. Is the control system transparent? If no, then the task is impossible.
2. Is the control system a digital algorithm? If no, then the task is not impossible.
  A. Can consciousness among the population, who hold the intention, be brought up to the level of understanding of the computational intelligent system? If no, then the task is impossible.

Socio-technical planning decisions are informed, given:

1. What resources (informational, human, material) are available?
2. What is known possible (knowledge, standards) to do, accomplish, create, and sustain with those resources?
3. And, dis-/mis-informed by, What is concealed?

Socio-technical operational decisions are informed, given:

1. What are the actual, datum operations to be designed (task, solution)?
2. When are the actual, datum operations to be executed (timing, access)?
3. Where are the actual, datum operations to be executed (materiality, resources and logistics, teams)?
4. With what, specifically are the actual datum operations to be executed (resources)?
5. How are the actual datum operations to be transformed (method of operation)?

Coordination control decisions (a.k.a., project decisions; social inquiry decisions)

1. What values (principles) are to be encoded into -ware through the software programming?
2. What experience will be encoded for individuals, as sensory in their environment, through the -ware programming of those values (principles) into its designed operation?
3. What is the optimal (most efficient and effective) timing logic for encoding those values?

There are [relatively] two types of [construction] decisions when it comes to the operation of a socio-technical environment:

1. There are relatively social decisions -- the project approach to the habitat:
    A. Focuses on describing the world in terms of
        1. Trajectories, directions, imperatives, objectives, time-frames, resources, and services
        2. initial conditions,
        3. given issue situation,
        4. wherein, the dynamical rules become expressed as:
            i. scheduling, coordination, controlling and monitoring
2. There are relatively technical decisions - the engineering approach to the habitat:
    A. Focuses on the dynamical rules as
        1. Which physical transformations are possible,
        2. Which physical transformations are impossible, and Why (for all).

In general, a highly-populated community-environment appears as a walking life-space, with automated transportation by rail and/or vehicle (depending on size):

1. The unified information systems model is visualized in the decision system because that is where planning occurs?
2. The unified information systems mode is visualized in the social system because that is where information integration occurs?
3. The unified information system is visualized in the lifestyle system because that is where the experience of all systems occurs?
4. The unified information systems model is visualized in the material system because that is where all encoding and user interface design (and development) occurs.
5. The unified information systems model is visualized in the project plan because that is where all information sets are necessarily associated with resources and time; material coordination.

Societal information systems access:

A.k.a., Community societal support.

1. Social data and data processing access (community information support) - social [information] construction support.
2. Decisional task processing access (community decision support) - decision [solution] construction support.
3. Material interface reconstruction processing access. (community technical support) - material [operation] construction support.
4. Life required service fulfillment access. (community life support) - life [integration-cycle] construction support.

The societal navigating methodology:

1. The approach methodology as the selection of methods associated with producing efficient and effective societal organization.
2. The direction methodology as the selection of methods that produce efficient and effective access to life fulfillment opportunities.
3. The working methodology as the selection of methods that are capable of systematically re-materializing a habitat, together in common.

The method of working together:

1. The selection of a method of coordination; the project methodology; social decision inquiry.
2. The selection of a method of materialization; the engineering methodology; technical decision inquiry.
3. The selection of a method of contribution (information transparency and team accountability); freedom of contribution.
4. The selection of a method of collection of usable information (standardization); service effectiveness in what fulfillment occurs.

5. The selection of a procedure and accountability to action (decision and evaluation); service efficiency in how fulfillment occurs.
6. The selection of a calibrated algorithm for computational materialization.
7. The encoded realization of an intentional walking life-space.

5.16.2.3 Approach [to society]

The integration of all information necessary to resolve an intention.

1. The intentional approach (everyone).
2. The unified approach (planetary).
3. The information approach (society).
4. The integrated approach (habitat; life-cycle; standard).
5. The issue approach (service).
6. The operations approach (processes; integrated project-engineering).
7. The project approach (the project lists, teams, timelines; plans).
8. The engineering approach (design, development, and operation).
9. The decisioning approach (algorithm).
10. Control approach (planning, executing, monitoring).
11. Algorithmic approach (synthesis).
12. Indication approach (objectives).
13. Evaluation approach (criteria).
14. Re-alignment approach (analysis).
15. Computational approach (logic, gating, materials).

5.16.2.4 Direction [of society]

The fulfillment of all individual human need among a regenerative, real-world socio-technical environment.

1. The intentional direction (human fulfillment of everyone).
2. The unified direction (global habitat service system; needs).
3. The information direction (societal information system; surveys).
4. The integrated direction (local habitat service systems; services).
5. The issue direction (habitat service standards; functions).
6. The operations direction (operational process protocols; resources and access; solution standards).
7. The project direction (solution social decision inquires).
8. The engineering direction (solution technical decision inquires).
9. The decisioning approach (algorithmic socio-technical inquire; a unified and adaptive information decision system).
10. Control direction [of materialization] (decision system).
11. Algorithmic direction (decision system).
12. Indicate direction (social system).
13. Evaluation direction (social system).
14. Re-align direction (lifestyle system).
15. Computational direction (material system).

5.16.2.5 Execution [planned operating experience of society]

The computation of the project lists into a simulated and real-world environment.

1. The intentional execution (“I”).
2. The unified execution (InterSystem Teams).
3. The information execution (database and algorithm - societal information system is stored on a database and runs an algorithm).
4. The integrated execution (local habitat service sub-system functions).
5. The issue execution (decision information flow standard).
6. The operations execution (access and usage protocols; accountability; work packages).
7. The project execution (project plans, project lists, schedule).
8. The engineering execution (system concepts, engineering lists, schedule).
9. The decisioning execution (algorithmic, conditionally programmed, information support system; software and interface).
10. Control execution [of materialization] (decision system specification).
11. Algorithmic execution (decision system specification).
12. Indicate execution (social system specification).
13. Evaluation execution (social system specification).
14. Re-align execution (lifestyle system specification).
15. Computational execution (material system specification; simulation; real-world).

5.16.3 Societal construction object

Society is a construction of tasks (specification-deliverables). Following, the object elements of societal construction are defined relative to the societal sub-system:

What is an object? An object performs motion.

1. In the social system, an object is that which is stored as data.
   A. Data.

1. Processing data.
2. In the decision system, an object is that which a task can be performed on (coordination).
   A. Task.
      1. Performing tasks.
3. In the material system, an object is that which has shape (geometry).
   A. Shape.
      1. Transforming shape.
4. In the lifestyle system, an object is a human life.
   A. Lives (Note: Constructor theory of life).
      1. Living life.

In an uncertain (discoverable) system, there are two fundamental types of objects necessary to make predictions are:

1. Dynamical laws (Laws of motion)
2. Initial conditions
3. And, final states (as a meta-composition of both objects)

5.16.3.1 What is a constructor?

A constructor is capable of performing one or more tasks, with available resources. A constructor is:

1. in a material sense, an object that represents the limit of a series of objects (with sub-object scales), each of which can perform a [construction] task in the context of a certain accuracy, to produce a final product.
2. in an informational sense, a concept, that represents a series of processes (motions between objects), each of which can perform a [construction] service in the context of a certain accuracy, to produce a final service.

> **SOCIETAL CONSTRUCTION STATEMENT:** *An ontological primitive is a "thing" that simply exists; something that simply is discoverable. Different worldviews postulate different ontological primitives; this is how we know who we are in the world and it is the information source by which we to reason our life-style. Community facilitates our fulfillment, and so we naturally desire to give of some of our life's working effort to the persistence of this system of fulfillment. We apply our effort toward contributing to the community and to our own self-development, through coordinated tasking. A task is a process that leads to either a novel structure, a "construction", or the continuity of a pre-existing structure. Community structures facilitate the experienced, lived fulfillment of real needs, to the point that they are sufficiently and completely fulfilled. There are many structures which have come before and there are many which will come after, and we construct with regard to this 'iteration' of how we might experience more fulfillment in the next [>] iteration.*

An ideal constructor has particular properties; principally, that the constructor is the cause of any informational-material transformation, if it retains the capability of performing the transformation again. For example, a heat engine is an example of a constructor because it performs a certain task, and after that, in the ideal case, it is capable of performing it again, and again, etc. Alternatively, consider any room in a building as an example of a static constructor, because it perform the task of shelter repeatedly; although a room cannot construct motion, it was constructed by motion, and will destruct by motion, over time.

In the [information] constructor logic, what may be exact is the statement of whether a task is possible? A task is either possible, or not, given what is known available. In other words, a task involves a decision in regard to what is possible, and what is not possible.

In society, what is possible is a decision. Therein, what is possible is a decision system. What is possible is a unified information system within which a decision system exists to resolve possible and impossible tasks programmatically, algorithmically, socio-decisionally.

> **QUESTIONS:** *What is the societal solution? What is a societal-level information media? Can [service] objects approximate ever increasing alignment with real-world, planetary human-life fulfillment? If there can exists a sequence of ever improving approximation to a [societal] constructor in its task [of societal construction], does that means that the task is possible?*

Common [information constructor theory] 'information media' examples include:

1. The transistor encodes a bit of information.
   A. A transistor is an electrical switch that holds a system state [bit of information], and can be turned on or off by another circuit. Computers use transistors to perform computation.
   B. The traffic light encodes information.
      1. Transform: green to red; red to green.
      2. Transform 2 lights: copy information from 1 light to the other light (green to green; and red to red).

Information media is information media because the following transformations (and tasks) can be performed on it:

1. **Swapability** property of the states - the interface states can be swapped.
   A. For example, with one traffic light, the green can become red, and the red can become green.
2. **Copyability** property - the information can be copied from one to another. The copyability

property allows information to be transformed from one substrate to another. This copyability property is what the interpretability principle expresses - whenever there are two systems that separately quality as information media, if the composite system qualifies as information media, then that means that certain tasks can be performed on the whole that can be interpreted as copying information from one to the other.
B. For example, with two traffic lights, the information on 1 can be copied onto another (red -> red; green -> green).

There are objects that have these properties of copyability and swapability, and they are called 'information media'.

NOTE: *A 'program' is a repeated output.*

In a societal system, what are the objects upon which transformations can be performed?

1. **Matter** - Spatial transformation, physical transformations, hardware transformations.
2. **Data** - Sensory transformation, mathematical transformations.
3. **Concepts** - Informational transformation, conceptual transformations, software transformation.
4. **Program**s - Computational transformations, statistical transformations.

Simply, constructors are possibly capable of doing what:

1. A constructor is capable of processing data.
2. A constructor is capable of performing a task.
3. A constructor is capable of transporting and re-forming shape.
4. A constructor is capable of carrying consciousness.

The continuous, conscious societal construction experience:

1. **Community access** ("we, of which there is me and we").
2. **Personal access.**
3. **Common access.**
4. **InterSystem Team Work Access** ("we", for which there is accountability in contribution).
    A. Work plan.
    B. Team tasking.
    C. Material service.

A societal constructor will:

1. The constructor (theory) will identify possible and impossible data, based on structure.
2. The constructor (theory) will identify possible and impossible tasks, based on principles.
3. The constructor (theory) will identify possible and impossible materials, based on science.
4. The constructor (theory) will identify possible and impossible lifestyles, based on solutions.

5.16.3.2 *Complete constructor sub-object*

INSIGHT: *In any informational or physical explanation there are some primitive elements.*

The sub-composition of an informational-spatial societal construction task:

1. **Task** - a specification of a physical transformation.
    A. Axiomatic task attribution, is:
        1. Possible (therefore, constructor)
        2. Impossible (does objective prevent a task from being performed?)
    B. A constructor, which is a machine, exists to perform tasks (Read: bring about a task).
2. **Timing** - a schedule (linearization) of a physical transformation
    A. Axiomatic timing attribution, is:
        1. Possible (therefore, coordinator)
        2. Impossible (does timing prevent a task from being performed?)
    B. A time, which is the common linear variable, exists to time tasks.
3. **Resource** - a material composition of a physical transformation
    A. Axiomatic resource attribution, is:
        1. Possible (therefore, user)
        2. Impossible (does resource prevent a task from being performed?)
    B. A resource, which is a matter, exists to materialize tasks (Read: externalize a task).
4. **Team** - a contribution of individual efforts to transform physicality.
    A. Axiomatic team attribution, is:
        1. Possible (therefore, accountability)
        2. Impossible (does team prevent a task from being performed?)
    B. A constructor, which is a machine, exists to perform tasks (Read: bring about a task).
    C. A team, which is a social construction, exists to execute tasks (Read: to do a task).
5. **Quality** - a condition of a physical transformation whose result is optimal.
    A. Axiomatic quality attribution, is:
        1. Possible (therefore, of value-validation).
        2. Impossible (does quality prevent a task from being performed?)
    B. A quality, which is an objective evaluation, exists to adapt tasks (Read: integrate feedback).

6. **Service** - a pattern of useful physical transformation.
   A. Axiomatic service attribution, is:
      1. Possible (therefore, habitat technical support).
      2. Impossible (does service prevent a task from being performed?)
   B. A service, which is an operation, exists to perform repeat tasks.
7. **Need** - a signal, sign of life capacity fulfillment.
   A. Axiomatic need attribution, is:
      1. Possible (therefore, habitat life support).
      2. Impossible (does need prevent a task from being performed?)
   B. A need, which is an life requirement, exists to perform understandable tasks.
8. **Preference** - a signal, sign of life opportunity fulfillment.
   A. Axiomatic preference attribution, is:
      1. Possible (therefore, habitat recreation support).
      2. Impossible (does preference prevent a task from being performed?)
   B. A preference, which is an life opportunity, exists to perform self-desired tasks.
9. **Decision** - a point of potential change [in fulfillment].
   A. Axiomatic decision attribution, is:
      1. Possible (therefore, issue recognition).
      2. Impossible (does decision prevent a task from being performed?)
   B. A decision, which is a point of change, exists to perform solution planning tasks.
10. **Evaluation** - an integration of the resulting alignment.
   A. Axiomatic evaluation attribution, is:
      1. Possible (therefore, control system).
      2. Impossible (does evaluation prevent a task from being performed?)
   B. An evaluation, which is a feedback opportunity, exists to perform corrective tasks.
11. **Indication** - a signal, sign of life quality.
   A. Axiomatic indication attribution, is:
      1. Possible (therefore, sensation).
      2. Impossible (does indication prevent a task from being performed?)
   B. An indication, which is a quality or quantity , exists to perform self-check tasks.
12. **Construction** - a duplicable building model.
   A. Axiomatic construction attribution, is:
      1. Possible (therefore, model, standard, simulation).
      2. Impossible (does construction prevent a task from being performed?)
   B. A construction, which is an information model materialized through a task, exists to perform useful tasks.
13. **Measurement** - determination of observational or mathematical alignment.
   A. Axiomatic measurement attribution, is:
      1. Possible (therefore, location).
      2. Impossible (does measurement prevent a task from being performed?)
   B. A measurement, which is a determination of position, exists to perform informed tasks.
14. **Verification** - a signal, sign of requirements completion.
   A. Axiomatic verification attribution, is:
      1. Possible (therefore, development).
      2. Impossible (does verification prevent a task from being performed?)
   B. A verification, which is a development phase, exists to perform requirements evaluation tasks (engineer oriented) .
15. **Validation** - a signal, sign of issue (design, solution) completion.
   A. Axiomatic validation attribution, is:
      1. Possible (therefore, design).
      2. Impossible (does validation prevent a task from being performed?)
   B. A validation, which is a development phase, exists to perform objectives evaluation tasks (user oriented).

5.16.3.3 Computational tasking

Today, the most “cutting-edge” form of computing is “quantum” computing, as a branch of fundamental physics. Regardless of the name, the idea comes from the idea is that computers are really physical objects, which means that what computational tasks they are capable of performing depends on the physics (real-world rules) that the elementary components carrying the information obey.

Currently, there are two known types of computational tasks:

1. **Classical turing machine** - based on discretized version of classical physics (discrete mathematics).
2. **Quantum mechanical (universal) computer** that has access to ways of performing computational tasks that are wider than the ones that classical computers can access, which means it can be programmed to perform certain computational tasks in a more efficient and power way, and there are certain algorithms that can only run on the quantum computer and can’t on the classical computer.

Potentially, a quantum computer can perform all computational tasks that are possible under the laws of physics. And therein, the question of what algorithms the system is to run [for humanity] becomes salient.

A universal constructor is an object, just like a universal computer has the ability to perform all tasks that are physically possible. However, it may be the case that there are only specialized constructors for each one of the tasks, and it may be the case that they all cannot be integrated into one object, which is a universal constructor, that when programmed, in the requisite way, will be able to perform each of those tasks. The universal constructor generalizes to general constructions what the universal computer does in terms of computational tasks.

It is possible to formulate the whole of society (or, physics) in terms of possible and impossible tasks., not computation tasks, but all tasks. Computational tasks are transformations on information media. A generic task may, or may not, be an information media.

Constructor theory expresses all laws as statements about which transformations are possible, which are impossible, and why. A constructor, when presented with the substrate in its input states, is capable of sending that object to another state. In doing this, the constructor stays the same. Here, the cause is the constructor. Constructors are information that can cause transformations in the environment. Therein, knowledge is a particular type of information that is capable of performing certain tasks associated with instantiating that knowledge in a physical system. Knowledge instantiated into a physical system can cause transformations (without anyone knowing about it; for example, DNA was causing organic transformations before any human knew about its presence).

> **INSIGHT:** *Ideally, a universal quantum computer can simulate the behavior of any other physical system with dramatic potentials and risks for social life together.*

For example, a refrigerator: within the refrigerator there is a glass of water; temperature; and a certain energy resource, the refrigerator can send the water and glass to a lower temperature. The refrigerator is capable of repeating this temperature cooling function on another glass of water.

If a task is not impossible (i.e. it is "ruled out"), because of a socio-technical effectiveness inquiry [decision], then it is possible, and possible with knowledge. Humanity can make use of knowledge to achieve transformations that verifiably improve its environment and the way in which individuals interact with it.

1. Initial conditions.
    A. For a computer, the initial conditions are a 'program'.
2. Laws of motion.
    A. For a computer, the elementary operations by which a computer works (e.g., transistor-decision-control gating).

5.16.3.4 Computational algorithms

> *A.k.a., Transformation automation ("quantum" represents the potential for informational and spatial transformation at the same time).*

Through algorithms, principles are converted able (through en-coding) into algorithms, which allows for computation (via computers) and decision support, for a community of contributing users. Computational decisioning uses information and an objective function (technique, algorithm) to determine parameter values from operational data.

> **CLARIFICATION:** *Algorithms don't have to be designed with output inconsistency, like human biases.*

Written principles (directional concepts) converted to algorithms (spatial logic), would allow a computer to take decisions for humanity and in parallel with humanity. Therein, humans are taking decisions, and the computer is taking decisions based upon a transparent criteria, and then, humans look at all the decisions, and compare and reconcile. If someone would do something different than the computer would do, then it is time to go back to the criteria that are built into the computer and check what/who is right or wrong. Should something in the computer programming change, or is there an error in humanity's decisioning awareness (i.e., did the computer calculate something humans missed). This type of system allows humanity to be incredibly efficient and productive, and allows humanity to process vastly more information (than without InterSystem parallel computing). And that, as a result, allows for the sustainable creation of community at the planetary scale. A cooperative, coordinated socio-technical societal sub-structure allows humanity access to more information, processed more quickly, and with less emotion. The unified processing of information, transparently, is required operate a cooperative society at the scale of the planet. Here, machines don't compete with humans.

> **NOTE:** *A synthesis, upon comparison with another synthesis, may sometimes lead to reanalysis of what and how.*

When can you trust a machine (or machine learning), and when can't you trust machine learning. The machine can come up with algorithms, or humans can come up with algorithms. The algorithms that machines come up with are not readily understandable. Possibly, machine output algorithms may be trusted, with a sufficient sample size, in a closed system. However, when there is a situation where the future can be different from the past, and there isn't sufficient deep understanding to accompany a decision (I. E., an non transparent machine output algorithm), then that is an unsafe, dangerous and risky position to be in at any scale of human population size. When can humanity get away without operating with deep understanding? Possibly, when there is a human interfacing with the machine so that there is a continuous inquiry into whether there is a sufficiently deep understanding (a forum of effectiveness inquiry) - - can the computer help the user learn and maintain a sufficiently deep understanding. The ideal condition is an environment where there is the parallel development of humanity and computation; while humans develop more capable computational technologies and techniques, computational systems build an optimized societal system through algorithms, which are developed by machines, and applied by humans, at a pace level with their sufficiently deep understanding.

> **NOTE:** *To have deep understanding, cause and effect relationships must be understood. To have cause and effect relationships understood, correct alignment of conception with the real world is necessary.*

Can the computer help the human looking at it learn and have deep understanding of itself and the algorithm?

It is dangerous when there is not deep understanding and the future can be different than the past (i.e., when it is an open, and not closed, system).

5.16.3.5 Where does the algorithm come from?

Principles (values) for taking good, intentional, optimal decisions can be converted into code (encoded into software programming). In a community-type society, there is a unifying information system programmed in code, and with a software interface, and there is a decision system programmed in code, and with a software interface. Additionally, there is a material experimental system programmed by atomic materials (resources, architecture, technology), and with a physical [human] vehicle interface.

By ensuring algorithms are transparent and deeply understandable, then widespread, deep, and optimal learning becomes probable for the whole human population. The understandability of society and of algorithms is a tremendously useful and powerful information set for humanity.

> **NOTE:** *An example of the application of algorithms to automation is 'autopilot' - once instructed (programmed) the system will navigate the craft (vehicle or construction) toward the destination.*

5.17 What does it mean for society to have an 'engineered' direction?

In an engineered system the concept [of a] direction is defined by a set of requirements, which are technical conditional statements of what the solution must contain to be a solution. Engineering is not just any form of creation; engineering is intentional creation. Societal engineering as a direction, is defined defining a set of [human] requirements. When a full direction can be visualized and agreeably shared, then decisioning therefrom becomes more relaxed. Societal engineering is about creating and sustaining access to objects and experiences that meet human requirements. Humans select the requirements. Engineering a situation where life persists and flourishes requires priorities. In society, together firstly, there is the necessity for having a basic life supported experience, which involves socio-technological service relationships.

5.17.1 Cooperation principles

The following are a set principles and concepts that facilitate a cooperative, mutually aligned socio-technical design (co-design) methodology:

1. The (engineering-based) system is an open system, in a theoretical sense, whereby interactions occur in a broader socio-technical context. Environmental factors exert a direct influence on the system, through the provision and exchange of information.
2. The socio-technical system in question is largely influenced by existing engineering design processes, which are often in progress when a co-design methodology of this nature is put into practice. Therefore, the appreciation and integration of existing engineering design frameworks is critical.
3. Engineering design processes operate within a wider development setting, characterized by distinct but interrelated phases; prior to development, development test, usage and feedback.
4. The socio-technical system, as made up of inextricably linked social and technical subsystems within a unique environmental context, must be considered at various levels throughout the design process.
5. Relevant stakeholders, notably end-users, should be actively involved during the engineering design process, and at each of the aforementioned levels of design.
6. Stakeholder engagement should not be restricted to end-user involvement, but should encourage and support the inclusion of additional stakeholder groups who may be influenced by the engineering design.
7. For the co-design process to be morally aligned, a thorough understanding of the existing societal (information and spatial) environment is required to facilitate integration and understanding in the early stages of the engineering co-design process.
8. A standard risk assessment has inherent limitations that are particularly relevant to this application. Rather, underlying the co-design methodology is the analysis of "exposure" as a metric of system

weaknesses that serves as feedback during the design process, through the provision of contextually relevant measurements that embody risk in use.

The application of the aforementioned principles and concepts to the societal engineering , and specifically to human well-being, requires a number of assumptions be made:

1. A societal system and the social, technical, and environmental contexts in which it exists, is an open unit that is directly influenced by, and is receptive to, changes in its surroundings. It does not, and should not, exist or be designed and developed in isolation.
2. The creation of a society requires awareness of typical engineering design (and to some degree, development) processes. Preliminary stages of such processes include some form of needs identification, background and literature study, requirements specification, the identification of the objectives of the design, and an ideation component. These preliminary phases are followed by prototyping with a focus on exhaustive analysis of multiple designs. Such analysis in turn informs the selection of a preferred prototype leading to a detailed design phase. The latter is concerned with the construction and exhaustive testing of the selected prototype, culminating in the production phase of engineering design.
3. The work setting for the cooperative design (i.e., co-design) of the societal system is comprised of the pre-planning, planning, and execution phases. The co-design of an intentional societal system should be considered at all levels.

5.17.2 What is societal planning?

Societal planning is a rational plan of life for living together on a finite planet. Societal planning occurs through projects, which represent work packages in time. Societal projects planning is, simply, societal coordination.

Any proposal for an societal-level organisational system must identify, determine, and explain the following:

1. How organisational processes are controlled?
2. How do feedback loops operate?
3. What constitutes the boundary of any sub- and supra-organisation?

Planning can coordinate the timing of all of these related inquiry events so that a single solution selection is possible for execution at the whole societal level of operation.

Here it is assumed that planning for human need fulfillment at the societal level is likely to generate and sustain a socio-technical system of an efficient, effective, safe, and free condition. It is sensible, wise to pro-actively think about, shape, and schedule through iterative design-time. Here, design that facilitates the development a fulfilling (i.e., the 'right' type of) environment for humanity is selected for. The full development of human potential, which involves production, with human beings and the ecology at the center. Society enables human potential or human capacities (or it can disable them). Wherein, real wealth is the development of human capacities and the development of human potential.

Imagine engineering as a function of society. In this sense, engineering is a socio-decisioning function for intentionally engineering systems into and out of existence, for individual human need fulfillment. The individuals among community take on accountability as contributors to an intersystem team. And, therein, the 'social' domain is coordinated through a software based social decision support system to determine workable social solutions. Among the serviced community, the 'social' is the population of individuals sharing access to resources (and access opportunities). At the Intersystem domain, the social is no longer individually choosing users, but accountable contributors. Accountable contributors plan their actions, they coordinate.

In society, all personal and social goals are completed (worked on, achieved) on the basis of successfully planned social interaction (past and present), with others. In order to generate fulfillment, and not degrees of suffering and conflict, the earth's resources must be seen as the common heritage of all, and it is only therefrom that unified planning is possible. Every technical system is planned somewhere, somehow.

> **APHORISM:** *When there is ownership and secrecy, planning is difficult.*

Participation in planning reflects the social "character" (or quality) of human action, of human interaction in any given society. It follows that participation in some form of societal life without serious systematic limitations is humankind's most basic common human interest. It is possible we see each other commonly, and therein, uplift everyone through coordinated design and planning for our commonly experienced, individual fulfillment.

The term social system is used, in general, to refer to lifeforms in definite relation to each other, which have enduring patterns of behaviour in that relationship. Having a populating data model for a social system is the first step in social societal planning.

There exist three core societal pre-conditions for human [social] survival and flourishing over long periods of time:

1. **Production** of access to needed satisfiers through extension of ecological life support services (into,

and by means of, a habitat service system).
2. **Reproduction** of genetics.
3. **Transmission (and processing)** of information.

The output of each of these preconditions, as process categories, is more efficient and more effective through planning.

5.17.3 What is a humane societal information system?

NOTE: *Society is response-able for human fulfillment (or suffering).*

Understanding the societal system (e.g., life space) is the first prerequisite for understanding an individual's actions therein. Generally, the individual and life space are mutually interacting systems, both modifiable via the other. The life space, or society, is the environment as it exists commonly (for every individual).

The basic conception of a life-space sub-divides into:

1. An individual's biological foundations.
2. The social system which contain the person.
3. The person's interactions with the environment.

Society is a social life organization. As an real world organization, society can be designed and engineered, and its effects can be aligned with life flourishing (life capacity), or not. As an developed system, society can have goals (*direction*) and a set of values (*orientation*) that align the society with the stated target vision (re-position).

A human society is the aggregate of humans living together. Observably, human life is a matrix of activities over the measure of an individual life-time, and linked across generations in the temporal continuum of natural and social history. The range of activities that define any individual life is structured by the environmental (native and non-native) conditions upon which it is dependent and the social organizations within which it is lived in interdependence with others. Human society may be lived as a complex adaptive system.

There are more and less fulfilling ways of arranging socio-technical relations. A societal system that is responsive to the needs of human beings is likely desirable than one which is not responsive to human socio-technical needs/requirements. Every sentient organism needs constantly to re-assess its environment in order to adjust to any changes in it and to ascertain which aspects are, or become, salient for its current life purposes.

Societal systems engineering represents the unification of disciplines in the design and development of an iterative societal system. Society is a collaborative effort, which may be recognized by individuals and active structures therein, or not (and there are definite negative consequences when it is not recognized).

NOTE: *While life can be fulfilling in unmediated nature, we can consciously move forward together in society.*

5.17.4 What is a 'humane' societal system?

INSIGHT: *A sane society (and economy) is there to serve humans in opening horizons of life-worth.*

A humane system acknowledges and accounts for the needs of all individual human beings. If a system is defined as a set of interrelated elements, then a human system may be characterized as a system in which the principle elements are human beings. Human systems may be arranged differently. However, because the arrangements have a relationship to existence, they can always be organizationally understood through the following four axiomatic information categories required for existence as a population together: social, decision, lifestyle, and material. Herein, the 'human environment' is every conception and/or physicalization with which humans interact. Technology aside, humans maintain the same set of common needs. The organization of any given society's social, decision, lifestyle, and material can optimize the fulfillment of needs [for everyone] for a given environment, or it can do less (as in, negative efficiency).

In existence, each person shares an environment that overlaps with another's environment, physical and social. In community, the shared environment is produced through planning, coordination, integration, and contribution/participation. Each persons own environment is partly given, partly modified, and partly made by the person. These environments influence the probability of fulfilling human need (in common), and hence, impact quality of life (life experience) of everyone. Persons' environments, and the environmental system generating them, are part of the internal organization of a society, as part of a societal information system.

Each society has its own societal information system (which may or may not be explicated), consisting of the physical (natural and man-made) environment enclosed within a boundary (or city/Country-State), outside of which is nature (possibly caretaken, and possibly not). The state of the societal environmental influences the functioning of the society, ultimately reflecting upon the quality of life of the persons in the society. Humans exist on a physical planetary environment.

5.17.4.1 Prioritization

Human lifeforms are biologically wired to be social (e.g., mutually beneficial) with one another, but only in a certain order of operations. There are a core set of fundamental human needs that when met will "relax" a life-form to the degree to which it can effectively focus on things of even greater depth and importance than survival, such as love (i.e., extentionality) and growth. Humans have a threshold at which basic needs must be met for them to begin acting in full social conscience with one another, and societal systems engineering provides the ability to design and iterate said type of societal system.

In community, societal development involves the application of accumulated scientific knowledge and socio-decisional (philosophical), technical understanding about the nature of the human organism in a way that can convey social experience of the reality (or a pre-supposed reality).

> **INSIGHT:** *Human life time is not simply the duration of our existence as physical organisms calculated in conventional units of temporal measurement, it is a morally meaningful whole of experiences, activities, and relationships unifying the moments between a person's birth and death.*

5.18 What is the expected socio-technical impact of the project?

The expected socio-technical impact of the project is the sustainment of a societal configuration classified as the type 'community. A community-type society represents a structure with the potential to achieve planetary-wide fulfillment of all human need and the sustainable expansion of human potential. Thus, it is expected that this project will have a mutually beneficial impact on the life experience of all individual humans on the planet. It is expected that the society which is constructed through this project will effectively and efficiently distribute access [to resources and services] for the fulfillment of all human need in a manner that does not exceed environmental service and safety limits.

A community-type society represents a societal structure designed to account for new knowledge, such that its own internal logic, understandings, structures, and functions become updated continuously, as humanity learns more about itself and its environment. It is expected that a design that accounts for new information in a cooperative manner is significantly *less likely* to generate the corruption, disharmony, and suffering, which are structurally systematic occurrences in early 21st century society.

5.19 What are the goals of the project?

In large part, the goals of the Project are defined in the social system [specification]; wherein, the explicit purpose of the societal system is to:

> **Continuously and consciously evolve toward our highest potential expression for ourselves and all others through resilient adaptation to a higher potential dynamic of experiential existence.**

In the social system specification, the following societal goals are listed [as directional structures] in support of the society's unifying purpose (stated above); these intrinsic aspirations maintain a social orientation toward common individual fulfillment:

1. To support each other in progressing toward our highest potential while developing self-knowledge and a deeper understanding and appreciation of our nature and the nature of the world.
2. To continuously improve the effectiveness and efficiency of the community's systems in fulfilling the unifying and life-long needs of everyone.
3. To continuously improve the means and methods, the oriented approach, by which we discover, understand, learn, communicate, and act.
4. To exist in a state of regenerative abundance with our life-ground while maximizing the intelligent use of resources and care-taking the environment (i.e., to sustain material resiliency).
5. To arrive at decisions based upon a commonly "living" purpose, set of needs & values, and approach, and hence, a similar set of understood relationships for arriving at decisions and actions. Note that these similarities are necessary for the effective functioning of [human] social relationships wherein a community is a set of similar relationships.
6. To exist in a state of appreciation and compassion for the self and the evolving whole.
7. To continuously improve access abundance through a stable 'bio-psycho-social community', a community of need fulfillment, serving as the liberating foundation from which individuals pursue their highest development and apply/ contribute (participate in) everyone's evolving potential.

Given a context of some uncertainty, and hence growth, society must be capable of (i.e., have the goals for):

1. Adapting [the societal system] to (Read: controlling adaptation to) changes in the environment.
2. Scaling [the societal system] for (Read: controlling the scale of) changes in the population.
3. Developing and utilizing [the societal system] (Read: executing and monitoring) methods and support tools for users.

Socio-technically speaking, the goal of this societal building project is to facilitate the healthy advance of individual self-awareness at the same time as technology advances:

1. 'Technical' means technology (physics applied functional); a more thought responsive environment over time.
2. 'Social' means conditional design for human need fulfillment.

3. 'Self-awareness' means the individual (individuated conscious) recording of experience.

Global human imperatives related to sustainable existence within the carrying capacities of the planet Earth, are:

1. The development of a unified societal information system.
2. The development of a global habitat service coordination system (earth management system) - A viable system of earth management must enable (rather than disable) life capacity without loss, and with cumulative gain over generational time.
3. The fulfillment of all human need (#1 and #2 together allow for #3).

QUESTIONS: *What is the individual's level of self-awareness? What may help and facilitate an individual in becoming more aware of who they truly are? When most of humans are born here on this planet they forget most of their potential past [life] experiences? What are the levels of self-awareness when there is a whole and integrated intelligence (consciousness) recording experience; what is our response among a common [heritage/sourced] environment.*

The primary **societal stability goals** of community, as a type of society, are:

1. Social system stability - a social system that adapts, scales, and develops while fulfilling human need, without conflict and while reducing suffering.
   A. Occurs through the facilitation of cooperation by means of intelligently shared organization and the sufficient completion of human need fulfillment.
2. Socio-technical system stability - a socio-technical system that integrates, coordinates, and operates services for human need fulfillment, without conflict and while reducing suffering.
   A. Occurs through the facilitation of teamwork by means of intelligently coordinated projects and the accessibility (availability) of resources, including information.
3. Technical system stability - a technical system that sustains a safety function/algorithm of impossible tasks that would conflict with the fulfillment of human need, or generate conflict and additional suffering.
   A. Occurs through the facilitation of an algorithm that is informed of what humans require and is capable of intelligently responding and adapting to those human requirements with uncertainty over what humans will require in the future, and certainty over what is (so that there is ever greater alignment and predictability).

Self-awareness advances include, but are not limited to:

1. Ability to contemplate - to think and imagine about ideas relating to the past, present, and future.
2. Ability to socialize - to think about ideas while accounting for other self-awareness (i.e., less/null social conflict).
3. Ability to communicate universally - to think and communicate by means of a universally understandable linguistic structure.
4. Ability to cooperate - to understand and contribute to the design of a unified societal model as so proposed by some given societal configuration (planetary teamwork).
5. Ability to perceive tasks that are likely to create, and impossible to create (i.e., will not create), a thoughtful and beautiful societal environment.
6. And beyond - the ability to move elsewhere in self-awareness, etc.

Thus, this proposal is for a societal configuration that does not incentivize a low level of conscious awareness -- a societal configuration that does not trigger base-material instincts that lead the human mind to perceive the ultimate answer to most difficulties as blame, punishment, or death.

Technological advances include, but are not limited to:

1. Stone age - primitive tools.
2. Metal machines - iron, steel, steam engines.
3. Electricity - electric power, computers, information technology.
4. Computational automation - socio-technical support algorithms (e.g., decision support algorithms).
5. Genetics - creation and modification of life-forms.
6. And beyond (e.g., matter transfer, etc.).

Healthy societies function on the social advances of good organization and individual self-awareness, and to a lesser extent, upon technical advances. With greater access to the physics of reality comes greater responsibility and accountability (i.e., response-accountability). So, increased access can only be phased-in depending on how well new thinking and behavior patterns are adopted.

5.19.1 Imperative goal

Due to a number of factors, including the increase in technological advances it is imperative that humanity develop and agree to a set of unified and integrated goals. The development of technology has suddenly made all societies, globally, interdependent. A long-term, strategic human goal is some desired current and/

or future state of the world whose realization would require an effort lasting over many generations. The imperative goal is to have a series of goals that could be shown to have a reasonable possibility of retaining their moral validity for an extended period of time, multi-generationally beneficial.

5.19.2 Human fulfillment within a habitat

The goals of a project is to develop a community-type society:

1. An experimental total city network system and integrated societal information system is proposed that will pursue the following goals.
   A. Designing a masterplan from a master community standard for societal operation.
   B. Designing a masterplan from a maser community standard for a local habitat operation.
   C. Conserving all the world's resources as the common heritage of all of the Earth's people. Restore the world's soils and bodies of water.
   D. Economizing all the world's resources into the optimized fulfillment of global human need (and preference).
   E. Transcending all of the artificial boundaries that separate people through development of a unified information system standard.
   F. Evolving from a market-State society to a community-type society (design out trade and coercion)
   G. Evolving from a money-based economy to a system in which a community can provide for itself by growing or making the things it needs.
   H. Re-wilding, caretaking and restoring the natural environment to the best of ability.
2. Develop a cybernated society that can gradually outgrow the need for all political local, national, and supra-national governments as a means of social management. Cybernetics applied to improve human fulfillment. Computers are a tool that frees people up from labor. There are two views of the habitat in community: the global cybernated view and the local habitat view. Not everyone may choose to live in a high-tech habitat (there are also low tech and density habitats), and the global network of [local] habitats is coordinated by a cybernated (intelligent and coordinating statistical service system).
3. Share and integrate new understandings and technologies for the benefit of all humanity.
4. Use the highest quality designs and productions for the benefit of all the world's people. Quality through continuous improvement.
5. Develop a common approach to action informed by an objective decision resolution process composed of inquires. Fulfillment through optimal decision inquiry resolution dynamics.
6. Encourage the widest range of contribution and incentive toward useful contribution.
7. Provide the necessities of life fulfillment, including stimulating challenges and preparation for the intellectual and emotional experience of flow.

5.20 What is the expected impact of the project on the family?

> **APHORISM:** *If I want to make my life the best that it can be I have to also make the lives of those around me the best that they can be in order to make my life the best that it can be. More colloquially said, "The best way to store food is in your friends stomach".*

This project extends the set of principles that relate commonly among loving family entities out to the whole population of society. Those relations that where once normative (implicit) at the family level are made explicit through a human-interfaced societal information system, that is cooperatively coordinated into exists by using contributors. In Community, as in the family (or, any openly sourced system), those who use family services are also those who contribute to family services. In other words, in a family, there is no artificially limiting separation between users and contributors; just as in community, there are no political, employee, employer, or consumer relationships, which are limiting class separators that are fundamental to the market-State.

Additionally, in a loving and supportive family situation, the family:

1. Restores relationships - Families do not apply a retributive, punishment-based, system on someone in the family when they do wrong (this has neuroscientific backing. The application of violence, aggressive, and punitive motions, when mistakes are made, causes damage to individuals and the family. Punishment as a mode of operation causes unnecessary suffering. Instead, families use restorative methods to heal relationships (of which there are multiple techniques from multiple domains).
2. Shares resources and information - Families share and work in such a way that the whole family is better off; they do not secret information and hoard resources that would better the lives of other family members. Families do not charge family members for living and using family services. Families do not enforce a structure of economic exchange (particularly, abstracted economic

exchange) on one another (particularly, in priority habitat servicing - life support). Forced economic exchange, and the encoding of property, inhibits access opportunities and promotes division and mistrust between family members.

Just as in the micro-social environment (i.e., family), within the macro-social environment (i.e., society), problems are solved by finding common ground and cooperating therefrom. In other words, family problems, like societal problems, are solved [in part] through finding common ground and cooperating with one another. And, at the societal-scale, a cooperating population is likely to be found using technologies, computing in particular, to facilitate optimal socio-technical construction, coordination, and decisioning.

5.21 How will the solution to the problem be conceived?

QUESTION: *What could we do if we were starting fresh?*

This project proposal includes a 'Concept of Operation' specification for a complete societal system. The solution is a system concept, and it is defined in alignment with the given real-world environment, which is experienced as a basis for a commonly conceived of societal operation. In this project proposal, the possible interactions by a societal process, and the interconnection between several sub-processes within a societal process are specified using the concept of 'services' (ports, interfaces). Counting iteration ("step-wise") refinement of society's process specifications and associated verification rules are considered. The iterative refinement of service (port) specifications and associated inter-actions (relationships; e.g., system-to-system and human interface) is considered as well. This document structure follows the basic concepts of the specification method, involving an approach, [to] a direction, [to complete] an execution. The iterative refinement of services (ports) and interactions is explored as partly an information interface, and partly, a hard-ware interface, for which an abstract specification and a more detailed implementation is given. Proof rules (logic) for verifying the consistency of detailed and more abstract specifications are discussed in some detail.

From this view, the method of conception [of the 'societal system'] is based on the concepts 'process' and 'port', as types of relationships in the real-world. A 'process' is a 'relationship' in itself, and a 'port' is a 'service', a larger set of relationships where a need is present (as in, a serviced or serviceable entity). A service [port interaction] may possess many processes [interactions]. The specification of the properties of a societal process (e.g., 'HSS operational process') or port (e.g., 'habitat service system') is given at an conceptual level. The externally visible behavior of humans toward one another and the planet, as a result of a societal configuration, may be described through process or port. This document does not detail the way this behavior is realized by an internal structure of the process or port; it is not the societal system sub-composition of social, decision, material, and lifestyle, though it coordinates, by means of approach, direction, and execution, all four core societal sub-specifications.

The concepts of process and port are significant in the design of an information system:

1. A process is an entity that performs some data processing and is assumed to be the unit of specification.
   A. In human society, the highest-level process is the process relationship the humans control at the highest level, the HSS prioritized operational processes.
2. A port is a part of a process and serves for the communication of that process with its environment (i.e., other processes in the system).
   A. In human society, the highest-level *port* is the service relationship the humans control at the highest-level, the habitat service system (HSS).

5.22 What systems of organization will use resources?

A.k.a., What systems of organization will use resources to complete the project.

This project proposes a unified societal information system that structures what systems use resources. This project proposes the following societal information system de-compositional view (Read: societal specification elements are) of resource usage:

1. **Human users (human life flow diagram)** - a flow diagram that visualizes the human (end-user's) resource usage path through the [functionality of the] societal system, from life to death. For instance, at the level of the societal building project, a flow diagram for a community-type society would detail the sequence of systems necessary to facilitate a fulfilled life of optimized well-being for any identified individual, given what is known and testable, from birth through until death.
2. **System architecture (system structure diagram)** - A system architecture diagram illustrates the way the system must be configured, and the way the database tables should be defined and laid out (all of which require resources). In community, there are two systems, which are really one - the information service system, within which is located a material service system:
   A. **Societal information service system**

architecture (level 1) - societal-level concepts:
1. **Social** - Socially defined direction, orientation and approach to navigation together.
2. **Decision** - Decision resolution logic to coordinate and control a complexly networked societal system.
3. **Material** - the probable material solutions and the reasonably selected, InterSystem Team applied, materialized iteration of the societal system.
4. **Lifestyle** (time/schedule) - the resulting common and individual human experiences of a material existence, given some entrainment cycle.

B. **Habitat service system architecture (level 2)** - habitat-level concepts:
1. **Life support** - human need-requirement
2. **Technical support System**
    i. Transportation architecture - how materials are positionally located and moved.
    ii. Information architecture - how information is computed and visualized.
    iii. Communication architecture - how information is transferred between humans and systems so humans have the information they need to respond.
    iv. Production architecture - how matter and information are cycled through the environment.
3. **Facility System** - human development-requirement

The habitat service sub-systems are called habitat service support systems, because they support a unified service-oriented habitat [for human fulfillment], which consists of three service support systems to which any common access resource in the system can be allocated. In terms of accountability, contributing members of an InterSystem Team fulfill the requirements of the three functional systems of each individual, locally networked habitat service system:

1. **The life support service system** maintains services that support life existence as part of fulfillment.
2. **The technical support service system** maintains services that support technical existence as part of fulfillment
3. **The facility support service system** maintains services that support discovery and self-development.

Tale note that 'state diagrams' are data models that show the changes between states of habitat service objects in the system. They show the cycle of an object's states, including events that trigger changes in state. They only show transitions, triggers, and the flow of changes.

5.22.1 What are the societal-level products?

A.k.a., Societal system deliverables, work outputs.

This project proposes the following societal service decompositional view:

1. **An information service system**
    A. A global information and decision support system.
2. **A habitat service system**
    A. The technical domain of a hard- and soft-ware service systems.
        1. A globally networked habitat service system
        2. A locally networked habitat service system
        3. A habitat operational process area (operational processes)
        4. A habitat operational knowledge area (operational knowledge)
3. **A socio-technical InterSystem Service Team**
    A. The social domain as human contributors organized by an accountable functional role.

5.22.2 Where will people live?

The population of community, as proposed by this project, primarily lives in live-work integrated habitat cities within an integrated global city network (within a larger planetary ecology). The cities in the Community-city network (global HSS) tend to be separated by kilometers and are dotted across the landscape, often in a grid pattern. When cities are newly planned, they are generally laid out (internally and externally) in a planned symmetrical grid. The internal grid of most of these cities is circular. The community population mostly lives in these cities. The countryside is mostly used for outdoor and other recreation activities. There are very few roads linking cities, because rail transport is effectively applied (and to a lesser extent, air transport).

NOTE: *In community-type cities, the grid for the city is symmetrical, and often, circular.*

Habitat services are just one part of the larger planetary ecosystem. A 'habitat service system' (HSS) is a controlled part of the total ecological habitat. A local HSS is more commonly known as a 'city'. In community, most cities are live-work locations. A global HSS is a planetary city network. It is a societal 'requirement' to design and operate cities.

5.23 What is a list of descriptors of the project's proposal for society?

A.k.a., Here is what we are building. This is where you will find high-level descriptive snippets of *what is being built.*

The following is a comprehensive list of descriptors of this project proposal for a 'Community' type of societal system. This list details, at least in part, what is needed, required, and expected for the existence of a community-type society:

1. A society that facilitates individual humans in becoming more aware of who they really are.
2. A society that facilitates the sharing of access to a higher potential dynamic of experiential existence for oneself and all others.
3. A society that effectively and efficiently creates the enabling, and removes the disabling, conditions for people to flourish.
4. A unified system that facilitates the maximization of each individual's potential.
5. A socio-technical environment that enables all of humanity to have access to the most up-to-date societal model and operating system, given what is known.
6. A unified society that enables every individual access to all the opportunities that all of humanity has to offer.
7. A societal development operations project for global human fulfillment, through global cooperation, wherein all resources are viewed as the common heritage of everyone.
8. A society where the population visualizes together a highest potential state-dynamic of fulfillment.
9. A purposeful societal system wherein efficiency, individual freedom, and the effective fulfillment of all human need are core determining inquiries into the selected decision to execute solutions into material existence.
10. A societal service system that exists for as long as individuals in the community desire the continued existence of the system -- humanity intends and technology enables a life of optimal flow and fulfillment.
11. A complex adaptive societal system (as adaptive toward greater states of human life-capacity fulfillment through improved designs).
12. A society is an open ended global problem. At what layer is the problem seen? At the fundamental level, all problems are systems problems and all human systems problems are fundamentally societal. Not just economic, not just decisioning, not just values, not just social, not just technological; but, societal at a priority recognized level.
13. A societal kernel informed openly about what humans require [as a requirement].
14. A societal kernel appropriately uncertain about what humans require, so that it doesn't irreversibly destroy things that are actually required [as a requirement].
15. A society that has, and provides, access to what individuals' need to thrive, to achieve some higher intentional goal, or to prepare themselves for some significant event.
16. A society that makes and sustains societal 'things' that last in usefulness.
17. A truly social, workable societal system that is designed to considerately account for each individual part in relationship to the whole, and the whole, in relation to each individual part. A societal system composition and configuration that effectively accounts for both the individual and the social.
18. A unified information systems model for an optimally organized state of human fulfillment and ecological well-being, given what is known.
19. A society that evolves intentionally towards states (and dynamics) of increasing well-being and mutual flourishing.
20. A unified and open societal standard for a community-type society is a core project goal.
21. A society where individuals live with fulfillment and wellness, without money or coercion, through cooperation and societal standardization.
22. A life-work environment where most of the population lives in integrated family- and garden-oriented smart cities with life-work lifestyles based on optimizing life fulfillment.
23. A society where life is recognized, work is shared, and needs are distinguished from wants, putting needs first as the priority and wants as discretionary or customary (customization or preference).
24. A project to bring into existence an information field representative of the highest potential of all individuals of humanity (wherein, aura = information field, and vana = wild breath).
25. A society that may foresee, as much as possible, the consequences of its actions in an uncertain, explorable, and growable environment.
26. A society that mutually distributes access to the fulfillment of all human need; a societal system that is not final (to individuals of the population), but iterative and progressively elaborated, emergent.
27. A society (civilization) where the population lives in harmony without force and coercion (of course, without war and destitution), for all. A society of need fulfillment, not fear reaction (i.e., a society of needs and not fears).
28. A society that is validated to perform appropriately to meets all human needs.

29. A society where we share an understanding of how the world works and how humanity can best work together in the world.
30. A society that optimizes for human fulfillment and well-being metrics (i.e., metrics other than profit).
31. A society that improves the human condition.
32. A society that continuously provides the opportunity to participate in society in ways that are intrinsically desirable to the individuals themselves.
33. A society that gives priority to aspects of life that are real, and does not prioritize aspects that are not real.
34. A society that seeks to understand, measure, and improve the human experience.
35. A society that orients toward an increase in global human well-being (i.e., satisfaction with life and the conditions of life, positive affect, and eudaimonic well-being).
36. A system where all individuals share the same ultimate planetary goal of a network of integrated city systems that share and coordinate resources without currency for everyone's fulfillment -- the network of integrated city systems acts as a fault tolerant [human fulfillment-service] distributed system.
37. A system where there is sufficiency for all; destitution for none.
38. A society where anyone can contribute, or not, without going destitute, and with having enough to grow in common with others. In early 21st century society, there is always the threat of destitution - if you do not work (i.e., are not employed), then the ultimate eventual consequence of a lack of belonging is destitution.
39. A society that works together as one unit; a human society that is unified, in that it works together transparently as one unit toward a higher potential state of togetherness, optimized fulfillment of all human need, mutually coordinated well-being, and more continuous and deeper states of happiness and flow for all among society at a global level.
40. A society that measures and increases well-being; a society with the aim of producing more well-being for every human individual.
41. An society where individuals care about themselves, each other, and the earth.
42. A society where the best quality of life is available to everyone. It is possible to model and operate society as a service system for humanity.
43. A society where the feeling of love is in the hearts of all individuals, and extension-ality (i.e., seeing others as an extension of oneself) is in their minds and in their decisioning.
44. A society that provides the right signals so that humans can feel at flow and love in their lives.
45. A societal environment where the technologies of well-being appropriately "dominate" the space, so that human beings learn how to be well, and are able to sustain and further develop a state of wellness.
46. An environment where the tools for well-being are easily accessible to every human being (i.e., colloquially speaking, the tools for well-being must be in the hands of every human being; a place where well-being is in the hands of all.). Further, the tools of well-being must be in the hands of human being (not just in the hands of organizations, businesses, States, leaders, gurus, etc.).
47. A society where it is possible to, and people are likely to, build their individual and social lives around a set of flow triggers.
48. A society where individuals have the freedom of access, autonomy of self-direction, and ability (knowledge and skills) to explore life's deeper questions.
49. A society where individuals have a holistic understanding and sympathetic appreciation of the human needs.
50. A society where people do what is of actual necessity and value to the fulfillment of their human embodied needs.
51. A society based on the existence of a real-world and a set of criteria for mutual human thriving within it.
52. A societal ultra-structure for the ability to take information and expand it to its logical conclusion, and therein, take the appropriate decision -- an ultra structure that enables better and faster decisioning for mutual human flourishing.
53. A societal system that integrates the operation of a network of local habitat-service systems (a.k.a., city systems) synchronously with a global information system.
54. A society that represents [proposes] a credible vision of a significantly better future. A vision that is feasible, viable, desirable for all of humankind.
55. A physical place, a network of cities, where information systems process the informational and spatial characteristics of human life together in a biosphere for mutual benefit through globally shared access (economic togetherness).
56. A society with a cultivated population of people who understand the impact of their thinking and behaviors on themselves, others, and the environment.
57. A society where people see themselves in relationship to other people.

58. A society with a contribution-based framework that is accountable to real-world human requirements and conditions, and behaves as a service system that fulfills (meets, satisfies, completes) human needs optimally.
59. A society with a decision resolution structure that uses indicators and empirically sourced data to set planned service-fulfillment targets and complete socio-technical fulfillment requirements within the value conditions (e.g., the inquiry resolution thresholds) of the population.
60. A society with a recognized solution design and execution planning structure for coordinated [community] action - a [scalable] project coordinated societal systems engineering plan.
61. A society that is safely prepared for and utilizes a network of autonomous systems to facilitate global human fulfillment.
62. A society where the habitat is recognized as a sub-system of the planetary ecological system. A piloted spaceship is an organism controlled habitat service system. The global human habitat service system may also be navigated like a space ship. Human navigated spacecraft in orbit are a microcosm of the more universal human controlled portion of a larger ecological habitat on Earth. Upon a planetary ecology, humans can control (as a spacecraft is controlled in its engineering and flight operations) elements of the natural [ecological] environment to engineer the construction and sustained operation of human coordinated habitat service systems, cities, as sub-ecological systems, where humans fulfill their needs together.
63. A society where individuals contribute and work together for the benefit of everyone, and therein, individuals feel in 'flow' with their work and connected to others in mutually beneficial ways.
64. A economic socio-decisioning system where services and objects ("things") are produced for the purpose of being *used, and not, sold and used* (Read: there is no trade).
65. A societal-level open access service system consisting of habitat service sub-systems. The function of the habitat service system is to provide for mutual material human fulfillment in the most efficient way possible through open source design and optimized development.
66. A society where the population senses and experiences integration throughout all domains of conscious (experiential) life, and hence, optimal well-being and wellness.
67. A human societal system with the intention to attain its maximum potential, which is most likely when all the individuals are working with one another; global cooperation, that necessitates, global coordination.
68. A societal system that accounts for the network effect of having any significant fraction of a population with unmet needs, which adversely impacts that population. A great deal of life in the past and still presently is miserable in large part due to competition over access to the resources that humans need to survive and thrive, generating unnecessary scarcity in fulfillment.
69. A society that is not likely to invent problems where none really exist.
70. A society where passion and efficiency produce sustainable human fulfillment.
71. A society that is not likely to reward the persistence of problems that do actually exist.
72. A societal project to bring into existence, and facilitate the persistence of, a planetary civilization, society, that feels in alignment with their environment, themselves, and with all others throughout the cosmic dimensions of experiential creation.
73. A society of the type, 'community', built upon useful information.
74. A society that facilitates the coordination and organization of all contribution to make the best use of resources for all of humanity.
75. A society that recognizes that "we" all want to navigate toward greater prosperity.

5.23.1 Alignment descriptors

The following are several questions to use when evaluating the alignment of an observed society (or proposed) with that of a community-type society, as the type of society proposed by this project plan:

1. Is it based on an explicit and common human purpose for existence?
2. Is it based on human need?
3. Is it based on contribution and sharing (i.e., is access free and participation open source)?
4. Is it based on a transparent execution?
5. Is it based on common heritage resources?
6. Is it based on a unified information system?
7. Is it based on globally coordinated access?
8. Is it based on an integrated built environment?
9. Is it based on systems science, standardization, project teamwork, and socio-technical capability?
10. Is it (or, where is it) completely visualized as a whole and understandable system?
11. Is it safely and workably scalable up to the size of the planetary population?
12. It is thinking and acting together in real-time to regenerate a more loving, kind, and beautiful earth

where humans extend their sense of compassion and access potential to all people.

This following characteristics provide a description of the planetary environment, given the conditions of this proposed societal systems model.

1. No war* - wars tend to occur along tribal and cultural divides in an effort to secure territory and resources.
2. Social mobility* - the population is free to choose which city area to live in, and when and where to contribute.
3. Infrastructural safety* - the infrastructure is sufficiently safe to operate and reduce risk from natural disaster.

**Once the recognition that "we are all one" becomes an integral part of human consciousness, the urge to resolve issues by killing each other and artificially limiting access to planetary resources, becomes obsolete.*

5.24 What are the project's primary surveys?

The primary survey inputs of this societal-level project (for the collection of data), include:

1. A coherently inclusive account of that which is required for human socio-technical flourishing.
2. A coherently inclusive account of the human team member skills necessary to complete the project.
3. A coherently inclusive account of the current team members on-hand.
4. An evidenced-based and rational-based approach to organizing society, which allows for feedback and adjustment.
5. An abundant life-ground that reduces scarcity stress.
6. A structure that would allow people to not suffer, and not get sick, but to get stronger and become more resilient with time.
7. A society to support (facilitate) the realization of our individual and common potential.
8. Knowledge of social-technical dynamics (engineering).

INSIGHT: *Evidence-based information has a calculable reliability. What is the reliability that the problem is designed out of the situation?*

5.25 What is a rational overview of the project?

Normally, knowledge is the result of actions (such as observation, learning, or communication). Values are the result of the interactions between knowledge and decisioning [that affects the social aspects of a population]. A lifestyle is the result of patterns of decisioning in a given environment. A material system is the result of a built system of resources, material resources and informational resources, and an ecology. In the reality of the existence of 'logic'[al] information processes, actions, become system design (of both an informational and a material form). The material system is an informational set (system) with a biophysical process component. Analyzing an informational algorithm in this detailed manner involves both epistemic logic (how was the knowledge determined) and dynamic-action logic (what is the predicted environmental response given a set of conditions). Herein, science is used to understand - physics tests and engineers re-form information and matter for differing functions.

A "rational" action at a societal/social level, is one that exists to facilitate the fulfillment of one's own needs, while simultaneously fulfilling the fulfillment of all others' common needs, in order to optimize all significant variables to the fulfillment of all common needs, which are individually expressed by unique consciousnesses.

The idea of a system of 'basic' needs forms a model corresponding between consciousness and all common human needs (i.e., human requirements) of some 'fact' (or "form" - as a real experience, "substance"). The most basic of which to understand is that: a human (without some possibly unknown source) cannot live continuously over some knowable duration of time (a quantity), given no access to food:

1. If someone does not eat, it is a 'fact' that they will eventually die causatively related to not eating?
2. It is a basic action, common to all humans, to have hunger (a conscious thoughtful input of feeling), act upon an environment to access food (think-cognate and move/behavior), and eat (process in a commonly specialized manner/method) to some relative degree [because food is a material object taken in by the mouth], with individually optimal nutritional (i.e., food quality) input profiles?

In the initial epistemic model for this situation, an optimized world (vs. eight possible worlds in a finite game environment) assign A-level category (or, A or B category for games) to each child. Society says: at least there is some way for optimizing for common human fulfillment; or, at least one of you is dirty.

This is the relationship between solution/fulfillment algorithms and epistemic communication - it is possible to optimize the solution to game algorithms based on a competition-based knowledge puzzle of:

- If, after collecting a resource outside, two of three people have the resource, then
  - Either, in cooperation (i.e., sharing), they all see the others, including themselves. Instead of

using competitive rules, if every child said what resources (e.g., mud) were observed in the first round, then every child would be able to determine the quality and quantity of resource in the first round (thus, optimizing, instead of gaming).

- Or, in competition (i.e., artificially restricted sharing), their perspective is artificially restricted such that some of the people, up-to and including oneself, do not know who has a resource. "Nobody knows in the first round. But in the next round, each muddy child can reason like this: "If I were clean, the one dirty child I see would have seen only clean children around her, and so she would have known that she was dirty at once. But she did not. So I must be dirty, too!"

A person (child) knows about the others' resources (or does not), and his own (or does not), encoding agents' certainty (of presence of resources required for fulfillment). In competition, successive assertions made in the scenario update this information. Updates start with the "fathers" publicly announced agreement that at least one resource is present (i.e., one child is dirty). This is about the simplest communicative action, this is the simplest communicative action, and it eliminates (optimizes) those worlds from the initial model that require a tertiary layer of logic (i.e., competition logic embedded within the market-State). The initial conditions are set, and then everyone shares their observations for everyone's mutual benefit. Note that a preference structure on top of an open source structure is not equivalent to a profit structure (market) obscuring the underlying [possible] open source structure (where resources are held in the commons of all, all fulfillment). In competition, there are typically competing "players" [for access -- closed-way, restricted communication]; whereas in community, there are typically cooperating "sharers" [of access -- all-way, open communication]. Simply, society has been defined to fail exactly at those rows or columns in a two-player general game model that are strictly dominated by competition. Every finite game model has worlds, and mathematics can "prove" is expression. The "nash equilibrium", a concept with the name of the player that identified economically as a "mathematician", refers to a condition in which every player-participant has optimized its outcome based on the other players' expected decision. The "nash equilibrium" is a market-based overlay on top of optimized fulfillment. Imagine that two businesses (market-encoded organization) compete in the same market-industry, for price-profit. The two companies enter a state of market-based "nash equilibrium" given the competing business expected response, neither business can make more money by unilaterally deciding to boost production. Any visualizable pattern [of information] will have a set of associated descriptive mathematics.

Herein, it is relevant to ask whether a the fulfillment sub-system of a societal system is also part of a system of competitive (market-scarcity - rule-ethic) or cooperative (shared commons - rule-value) interactions? Is there a societal fulfillment: problem-game (competition), or a problem-operation (cooperation)? Or, is there a perception of receptive-motor ability to change individual-societal fulfillment (because of a 'belief' system overlay, limiting knowledge and a higher potential value orientation that encompasses the fulfillment of all)?

Society can now be described in two logical directions:

1. **First direction:** From science to logic - given some algorithm defining a solution concept, we can use our cognitive ability to discover-find epistemic actions (e.g., basic human needs - actions for which knowledge can be known) "driving" and moving its dynamics (behavior).
2. **Second direction:** From logic to science - any type of epistemic assertion (e.g., basic human need) defines an iterated solution process which may have independent decisioning and/or interest-preferences.

Game theory adds the idea and associated mathematics of competition on top of a fully connected (i.e., sharing) set of entities to by restricting their memory action-potential for sharing.

Finally, the dynamic-epistemic setting has one more degree of freedom in setting up the virtual conversation, viz. its scheduling. For instance, the Muddy Children of Example 2 had simultaneous announcement of children's knowledge about their status. But its update sequence is quite different the children speak in turn. When the first person says its status, then in the analogy, in the actual world, the second child knows its status. Saying this eliminates all worlds except the optimized one.

5.26 What is the minimum viable product?

Because this is a societal-level development project, the question of, "What is the minimum viable product" is a challenge to answer directly. It is a challenge to answer directly because a society is an ever living and changing complex of many interacting influences. Note here that minimum means that itsn't being applied at a large-scale (e.g., city, State, etc.).

What is known is that there is a minimum viable set of operations that must be ongoing to community to be present, as follows:

1. **A societal specification standard in the configuration of community:** The minimum viable product is a specification standard that can be adopted to conceive of and to operate community.

   A. One unified specification standard.
2. **A habitat in the configuration of community:** The minimum viable product is a pilot university flow habitat comprising of a system likely to produce more human flow, more soil restoration, and an abundance of food, fuel, and fiber over time.
   A. One efficiently duplicable rural restorative and food abundance producing university-type habitat. This habitat will be duplicated into a network of rural economic production habitats.
      1. One primarily mineral-based rural habitat service system.
      2. One primarily bio-construction based rural habitat service system.
3. **A software solution to inform and adapt community:** The minimum viable software product is a residency profile and education platform for coordinating orientation and entrance into the habitat network.
   A. Project coordination software.
   B. Standards setting collaborative software.
   C. Decision system software.
   D. Operations software.
   E. User software.
4. **A hardware solution to [re-]materialize** the habitat [network] to more greatly meet human fulfillment requirements given the information and technology available.
   A. The production of integrated habitats as an integrated means of production and delivery.
      1. Production of means of production (produce machines that will produce the final deliverables).
         i. Means of production for production of an integrated habitat.
      2. Production for consumption (produce final deliverable).
         i. Production of final user needed deliverables (goods and services) within an integrated habitat [network].
      3. Note: a center for the production of habitats is the first means of production (for the production of community habitats).
5. **An on-boarded team of community members** ready, willing, and sufficiently available, knowledgeable, and free of trauma to effectively operate and benefit from life (in all its phases) in a community-type habitat network.
   A. The support of a partnership between:
      1. Industrial producing cooperatives, and
      2. a regional union of socio-economic States.

5.27 What are the possibilities for humanity?

The possibilities for society may be drawn forth through the following questions:

1. What is the fundamental [conceptual-operational, ConOps] hypothesis, as a description and explanation, simple enough that it can be double-checked by simple thought?
   A. To the best of "our" knowledge, there is nothing wrong with the hypothesis that humans can in wellness together.
2. Is a 'community'-guided society a viable basis for human fulfillment and ecological regenerability of biospherical services?
   A. To the best of our knowledge, there is nothing wrong with the hypothesis that humans can thrive together in a biosphere.
3. What is the performance and potential of an integrated (cooperative and ordering) socio-technical societal system?
   A. Can the rules of human need and societal construction be accessed to design fulfilling services, objects, and machines?
   B. Can information mechanisms be adapted to increase the programmability of societal sub-system part assembly?
   C. How efficiently can new solution specifications be synthesized and constructed into the ecological environment?
   D. What would be the performance of engineered habitat systems, with or without high technological integration and automation?
   E. What is the smallest and largest sizes of a city? Unknown.
   F. Can interfacing with the market-State improve any of these answers?
4. What are the technological objectives and capabilities of a socio-technical integration of society into a unified information [space] sphere?
   A. What are the capabilities of a community-type society's service products?
   B. What are the objectives of a community-type society's service products?
   C. How are the objectives of a community-type society's service products evaluated?
   D. Why are the objectives, capabilities, and their combined probabilities in effectual-causal relationship selected over others?
   E. How has social navigation, together in this cosmos of exploration, changed?
5. How capable will the system be?
   A. What information and physical materials will the

service or product be built of?
   B. What are the functions of the system?
   C. What will be the efficacy of the various system functionalities?
   D. Can the system produce complete human need fulfillment, or only partial fulfillment?
   E. What components of itself can the system produce (autoproduction)?
   F. What new capabilities can the services and products implement?
   G. How close can the fabrication be placed to place and time of service/product use?
   H. How easily can new products and services be designed?
6. Are transparently understandable and algorithmically guided decisions a viable basis for a moneyless and Stateless society?
   A. Is there anything wrong with the basic hypothesis of using programmatically controlled computers and actuators (machines) to do society?
   B. Is it possible, and how could it be possible, for machines and humans to coordinate optimally at any level of technological development? What is the nature of machines, their role in creating value for humans, and ultimately how machines form an essential and extended, integrated, part of individual humans connected over a multi-domain mesh network, and the human system over time (as, knowledge and evolution?
   C. How can the human be in the center of an ever optimizing ecosystem of humans and devices and tools that "we" (humans) have created around us and that will consequently keep growing and influencing us (bar any unrecoverable risk-disaster scenarios)?
   D. Can engineered societies do planning to synthesize/solve human [need, informational and spatial] requirements for fulfillment with low error rates?
   E. Can issue, resource, and procedural accounting build habitat services with low error rates? Even on a planet with multiple societal types operating?
   F. What other methods will allow teams to build globally cooperative organizations?
   G. Will there be substantial difficulty in acquiring financial funding?
   H. Will there be substantial difficulty in acquiring jurisdictional contracting?
   I. Will there be difficulty in sustaining operation?
7. To what extent is algorithmically guided decisioning counter-intuitive and under-appreciated in a way causes underestimation of importance?
   A. Automation and autoproductivity. Autoproductivity is the ability of a system, under external control, to automatically produce an identical copy of itself.
   B. Societal complexity and functionality is not limited by decision system complexity - will projections from inquiry processes overestimate service or product development difficulty?
   C. Community-type societal engineering may be overshadowed by superficially similar organizations— is there a risk that people will think they're studying community when they're actually studying something else?
   D. Community is opposed by special interests - is study of it likely to be stunted by business and political maneuvering?
   E. Human benefits of an planned and integrated humane societal organization are not widely known - would better knowledge increase research and development?
   F. The operations of programmable, automated service may be easier at the societal [macro] scale - will projections from conventional engineering under or over-estimate difficulty?
   G. Economics has been the domain of market economists. Control and coordination has been the domain of politicians. Engineers have a much faster approach to development. How will this affect progress.
8. What procedural inquiry resolutions toward decision control does all this suggest?
   A. Approach to control
      1. Total control: (with 10% deviation) through transparent algorithm of all that relates to development or use of society?
      2. No control: let the solution emerge?
      3. Local control: sub-systems find their own solutions?
      4. Security control: preserve against destructive change?
   B. Approach to resources
      1. Efficiency control: optimize use of scarce resources?
      2. Effectiveness control: maximize availability of non-scarce resources?
      3. Acquisitions control: collect resources?
   C. Approach to access
      1. Personal access: oneself use?
      2. Commons access: time scheduled common use?
      3. System access: operations use?
9. What applicable sensing, deciding, and manufacturing tools exist?
   A. What modalities exist or can be developed?

   B. What open source technologies exist or can be developed?
   C. What combination of sensing, deciding, and manufacturing can be integrated?
   D. What communications technologies exist or can be developed?
   E. What design collaboration technologies exist or can be developed?
   F. What coordination technologies exist or can be developed?
   G. What fabrication technologies exist or can be developed?
   H. Which of these technologies is compatible with automation and/or high throughput?
   I. What are compatible combinations of societal technologies?
   J. What handling procedures and technologies exist for moving information or matter between different societies and/or locations efficiently?

10. How rapidly could systems be designed and services become operative?
   A. To what extent can components be re-used between services (or products)?
   B. To what extent can low-level design be automated?
   C. How directly applicable are current engineering methods?
   D. What new engineering methods need to be invented to use this technology?
   E. How quickly can prototypes be built?
   F. How rapidly could the system match the current market-State access of a middle-to-upper income family?
   G. How can proliferation and access of community services and products be expanded?

11. How could an effective development program (Read: construction of the first “discovery-oriented”, “resource-accountability”, experimental-accountability” city system) be structured?
   A. How can coordinators, scientists, and engineers be engaged in the project?
   B. How can mentorship be engaged in the project?
   C. How could the project be funded?
   D. How could bureaucratic friction be minimized?
   E. How could passion and flow be maximized?
   F. How should the overall project be structured?
   G. Under what psychological environment (culture) could an effective program take place?
   H. Under what sharing environment (legal) could an effective program take place?
   I. How can development time be minimized?
   J. What cost and time overruns should be expected?
   K. How can everyone collaborate?

12. What will be required to develop a global habitat access service and its products?
   A. How much computer time, human creativity, and power would it take to design, then simulate, and verify the operation of a community-type society?
   B. What will be involved in developing an information support system that can carry out the required processing and decisioning to build the first iteration of the societal system?
   C. How reliably can the operation of a community-type society and its parts be simulated? What would the cost and development time of a CAD/simulation system capable of acquiring understanding from socio-technical dynamics simulation of such parts?
   D. How many parts and surfaces would be needed to constitute a complete set of low-level structural and functional components? How much human effort would be required to develop them?
   E. What would be the cost of developing a design for the first societal city and accompanying societal information system.
   F. How many of these steps could be accomplished concurrently in a rapid work program? All of these steps could be started concurrently, with successive refinement.
   G. How precisely can costs and schedules be estimated?
   H. By what methodical approach will development, occur, of the first self-contained city manufacturing system (which has the requirement to be able to produce duplicates at an exponential rate), and does its description and explanation integrate/complete all spatial and temporal elements?
   I. To what extent is there a (time / resource) schedule consideration, conflict, and priority?
   J. How reliably is the schedule adhered to; the core metric of ‘team’ operation (indicated as “showing up”, occupying, or otherwise acting with a purpose to complete some preplanned task in some para-procedural-metric way?

13. What beneficial or desirable effects could this have?
   A. How much suffering, illness, and disability could the societal system reduce?
   B. To what extent could the societal system alleviate underdevelopment?
   C. Could this help with food and water shortages?
   D. Could this help with climactic changes?
   E. How much and in what ways could it alleviate ecological-environmental problems?

F. How much and in what ways could it alleviate socio-structural problems.
G. Which natural disasters could it prevent or alleviate?
H. How much could these benefits reduce social unrest?
I. How much financial, commercial, governmental, and human incentive is suggested by these questions?
J. What new services, products, or value conditions will the system make accessible?

14. What technical restrictions may make society safer?
    A. Because unleashed access to technology is so dangerous, the best solution appears to be careful decisioning on technology, including some mandatory restrictions to access and materialization. Fortunately, the same features that make technology dangerous also allow the implementation of several kinds of technological restriction that may form useful components of an overall coordination-automation program. Products that might be adapted for secret production of certain materials and technologies pose a serious threat to humanity and the biosphere. Other products pose other kinds of threats, and additional restriction will probably be desirable. Still, many products, once approved, can be built freely—and for some classes of products, approval can be a rapid and automated process.
15. What raises serious questions about societal interfacing?
    A. How is what is raised as serious, as an 'issue', prioritized (i.e., how are serious 'issues' prioritized)?
    B. What other societal organizations and options should be studied?
    C. What other societies may be suitable for automatically precise re-programmable assembly?
    D. What are the consequences of experiencing a societal system that recognizes consciousness as a fundamental component of the exploratory system?
        1. What are the consequences of a societal system that is recognizable as collaborative and explorative; thus, has probably uncertain itself through time, though is certainly interconnected in the now [space], and thus by consequence, may be planned in its now [integration] into the conceptual-spatial (integrated physical-embodied, consciousness-material) environment?
    E. What effect will the system have on military and government?
        1. What effect will this have on governmental rights and liberties?
        2. What effect will new information access (and consequently, surveillance) capabilities have on privacy and social engineering?
        3. What effects will new information access (and consequently, surveillance)capabilities have on governments and other coercive power wielders?
        4. To what extent will new capabilities increase demand for community?
        5. To what extent can conceptual and spatial breakthroughs alleviate poverty and misery?
    F. What effect will this have on migration?
        1. What effect will free information and access opportunities have on the movement and relocation of people?
        2. What effect will the movement and relocation of people have on the operation of a community-type society?
    G. What effect will the system have on market-State?
        1. What effect will the system have on macro- and micro-economics, on production and distribution?
        2. What effect will this have on geopolitics?
        3. What would be the effects on international relations of reduced international trade?
        4. What would be the effects of global community-based societal access on lifestyle decisions and personal access? How quickly could those effects happen?
        5. What barriers to cooperation could make these problems more difficult to solve?
16. What are the disaster/disruption scenarios?
    A. War; social unrest; market unrest; dangerous technologies; socio-moral corruption? Bio-solar spheric changes; ecological collapses?
    B. Social; technical; biospherical?

5.28 What are the assumptions?

This project assumes that human beings are experiential vehicles, that enable consciousness, to experience a physical environment with other individuated consciousness. Within this axiomatic assumption, this project plan assumes that it is possible:

1. For humans to cooperate -- to act together, harmoniously.
2. For humans to identify a sufficiently stable set of information, and service systems, for completing human requirements synchronously.

3. For humans to calculate the ways in which resources may be useful.
4. For humans to determine the optimal arrangement (configuration) of resources.
5. For humans to select a solution (of objects and relationships) to provide for each and everyone's highest [level of human need] fulfillment.
6. Society is a system for which it is possible to design and operate the existence of.
7. For humans to construct a lifestyle that allows for living together in a fulfilling way (e.g., constructing a diet that allows the individual organism to eat in an intuitive way among others, as eating for its nutritional and psycho-emotional benefit; and not, eating with disorder).

It is an assumption that the following questions have testable answers:

1. How can "we" organize the human societal system to produce the products and services humanity requires, cooperatively?
2. How can "we" maximize the efficiency of resource usage for each and every individual's human access to the highest-level of fulfillment, given what is known?

Let us wipe the board free of past limitations, before assuming and proposing:

1. Let us assume that it is possible to understand how a society without the market or the State could exist to produce a sufficiently optimal and continuous state/dynamic of human fulfillment and ecological regenerability.
2. Let us propose the existence of a societal system by means of a standardized and planned specification for the construction of a most fulfilling society system.
    A. Let us contribute our efforts in a coordinated manner to service the fulfillment of the highest fulfillment of all.
    B. Let us visualize together a proposal that assumes we are all capable of living together in the service of all.
3. Let us assume we can coordinate a Global InterSystem Human and Resource Contribution Team who continuously provide services to the global population of community users.
4. Let us propose an information system that accounts for common resources and the common requirements of all of humanity.
5. Let us assume coordinated access to a common pool of resources.
6. Let us propose a network of integrated habitat service systems that distribute access optimally for individual human need fulfillment.
7. Let us assume customized cities within a global/planetary community-city network.
8. Let us propose access to services as resources are distributed through a transparently understandable decision support-computational algorithm.
9. Let us assume a population of conscious intellects, capable of reasoning and growth.
10. Let us propose a unifying information systems model/method that resolves into the continuously iterative improvement to conscious life well-being.
11. Let us assume that humans are capable of optimal well-being (the highest-possible fulfillment) and the least optimal suffering (the lowest possible feeling of fulfillment).
12. Let us propose a community-type society configuration where we are all together, most likely, to live lives of optimal well-being.
13. Let us assume that we exist together in a common, real-world environment.
14. Let us propose a societal specification that optimizes our fulfillment together in our common real-world environment,l which we further propose can be visually explained as a unification of conceptual and spatial information a project coordinated specified plan of execution by an InterSystem Team.
15. Let us assume that it is possible for society to be differently configured, producing different results than those proposed by this theoretical explanation for the next optimal iteration of our consciously materialized societal system.
16. Let us propose the specification contain the reasoning, so that the next optimal society configuration may be more completely understood by all those users with the intention. Let the proposed societal system explain (contain the explanation for) the logic of its own theory.
17. Let us assume humans can connect resources together into services that transport and transform material [spatial] resources into as-required-by specification of the requirements for human fulfillment.
18. Let us propose a habitat service system that connects the life-cycles of planetary ecological services.
19. Let us assume it is possible to control the coordination motion of ecological resources into optimal [integrated] habitat service configurations.
20. Let us propose a solution to individual human fulfillment at the planetary scale.
21. Let us assume spatial objects are what the material environment is composed of, and

conceptual objects are what the information environment is composed of.

22. Let us assume values are directional conceptual objects as shapes/structures with an intention for the next conditional iteration of the whole societal system.
23. Let us propose an information system that resolves a responsively uncertain decision support system that determines and selects optimal solutions, which become evaluated materializations, physical objects.
24. Let us assume there are categories of configuration of a societal system, as well as, the real-time consciously experienced configuration of material resources.
25. Let us propose the specific design of a societal system that has been designed by selecting among categories of configuration [of resources] (as solutions) for the one that demonstrates optimal real-time fulfillment of all within a planetary-scalable solution.
26. Let us assume that different organizations of resources and qualities of services can achieve different levels of individual human-conscious fulfillment.
27. Let us propose that services can be designed within a spatial environment to facilitate and/or "automate" a product-result and/or conditional outcome.
28. Let us assume that it is possible to observe any human societal system as a series of societal information sub-sets.
29. Let us propose values become conditional objectives in the decisioning selection of the next societal solution.
30. Let us assume that all existence is in continuous physical motion.
31. Let us proposed a specific information and habitat service system as the next iteration of our society.
32. Let us assume that is possible to together decide the next execution societal solution.
33. Let us propose that society starts with language, because we are proposing an informational and physical interpretation that linguistically interrelate.
34. In the early 21st century, most people say "society, you know what I mean", and then just keep going without defining society.
35. Let us assume that resources are objects, and that the informational habitat service system is a concept, a category of objects ("things") that relate to the life-support, technological support, and facility support of all human life.
36. Let us propose a system composed of informational and spatial objects globally coordinated into a network of InterSystem Service Teams.
37. Let us assume that Intersystem Teams composed of users can develop and operate services through their contributions.
38. Show me the documentation so that we can all transparently understand the theory of the proposed system.
39. Humans can consciously intend and knowledgeably construct a material (Read: spatial information) system configured to optimize services that complete human need.

Herein,

1. Fulfillment is a dynamic concept because what all humans require as a habitat service system [configuration] may change through time. Fulfillment is a concept that can be accounted for by the engineering of a service to complete a set of requirements. Fulfillment "takes shape" as the conditional configuration of spatial resources into an optimally experienced habitat service system. Fulfillment is a dynamic concept.
2. There is no market or State object in the real-world There are organizations of conscious humans and material resources. Community is the societal-level term given to the organization of humans and resources by design to complete human requirement fulfillment optimally, given what is known and possible. What is known and possible must be accounted for and specified so that sufficient information is available to operate services as required. Neither the market nor the State are stand alone [material] objects. In social engineering, the market is the defensible division of common heritage. In social engineering, the State is an organization of humans and resources to organize the rules for compliance, while coercing and enforcing their finality as the classification of a criminal (or more accurately, criminalized) or personalized [as a player] in the socio-economic market. It turns out that there are better ways of configuring resources by deciding optimal solutions to human issues at the planetary scale. Conscious human beings can point to other humans and point to material objects, but "you" can't point to the market or the State. The "market" and the "State" are just concepts, which can be encoded into a societies information-decision system with reciprocal affect on the individuals' feelings of well-being. The habitat service system is the record of, and also the result of, the flow of resources and humans through the constructively materialized information system.

3. "Civilized" people do not use violence, economic life competition, or threats of punishment to resolve decisions/solutions to human fulfillment.

Scholarly references (non-cited)

- Benthem, J.V., Amsterdam & Stanford. (2006). *Rational Dynamics and Epistemic Logic in Games*. Universiteit Van Amsterdam. https://staff.science.uva.nl/j.vanbenthem/RatDyn.2006.pdf
- Furman, S., Theofanos, M., Wald, H. (2014). *Human Engineering Design Criteria Standards Part 1: Project Introduction and Existing Standards DHS S&T TSD Standards Project*. NIST Publication, NISTIR 7889. http://dx.doi.org/10.6028/NIST.IR.7889

Online references (non-cited)

- *Specification/Declaration modeling pattern*. HLCM. Accessed: December, 2019. https://web.archive.org/web/20190618142606/https://hlcm.gforge.inria.fr/hlcm:implementation:spec_decl

Project Approach

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: systems science, project-engineering, operational societal methods, integration plan,

Abstract

An organization that process information and or objects can have selected methods applied to it. An approach is the selection of concepts and methods that facilitate a more correctly aligned orientation. As a societal-level project, it is necessary to select a set of conceptions and methods of approach that are likely to orient society most greatly toward the experience of mutual human fulfillment. For anything that is created in the world, there are a set of tools that allow for its creation. It is possible to come nearer to in distance and time a societal operation that meets the expectations of individual human fulfillment, through a recognition of how patterns are identified, designed, and contributed to. In order to unify, a unifying method is required. The systems science approach involves the application of the systems language in order to facilitate the identification and synthesis of useful patterns. In order to act upon information usable to societal organization, there is a necessity to use project-based in combination with engineering-based processes and knowledge sets. In order to optimally plan an execute the cycling of information and objects through a societal environment, project-engineering is used to optimize standards, contributions, decisions, and solutions to society. By viewing all global service systems as project plans, it is possible to plan the life-cycling of fulfillment for a global population with consideration to the individual dimensions of information, socialization, and material resource utilization. An integrated approach is necessary to remain sufficiently confident that thinking and actions are likely to facilitate mutual human fulfillment at the societal level.

Graphical Abstract

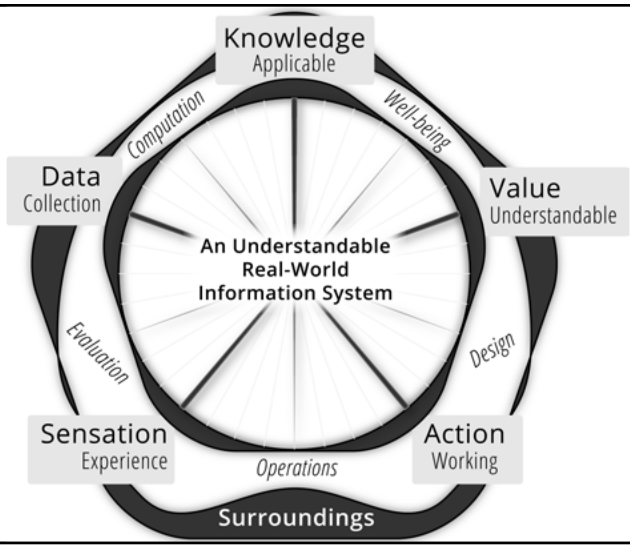


Figure 3. *Visual representation of a conceptual framework for planning the navigation of an understandable and directable, real-world, societal system.*

1 Introduction

Every approach to state change in a material environment requires work. The approach is to work toward cycling through time a system [materialization] that meets (through project-engineering) human needs (decision), given what is known (standardization), and contributed (contribution). The approach is to view all work, all intentional change, as an engineering project, within which decisions are determined about the next [to be]existent state of the society.

The approach to planning is the same approach outlined in the Social System specification: the systems methodology resolves into the selection of the systems-based methods. Science is a systems-based method of discovery. Engineering is the systems-base method of working for development and operations of services. Project coordination is the systems-based method of computing a schedule, given what is known, and what is available from contributions and resources in time.

> **INSIGHT:** *To de-compose and re-compose, every system must be assumed based on some kind of structure (structural pattern), which may be personally and socially understood through a complex of association networks..*

2 The systems-science approach

> **INSIGHT:** *Simply, the approach is to work with patterns together for our mutually directed benefit. Systems in the real-world express behavior (motion) and exist within a context (network). To fully identify a system, it's behaviors and context must be identified. Whole-system engineering—optimizing an entire system for multiple benefits, not isolated components for single benefits.*

Systems sciences provides a potential to explore what are the temporal patterns inside of society. Within the system 'society', how is it possible to recognize and control temporal patterns (of access and creation), optimally, using [a temporal] data [stream of needs and values, projects, solutions, and resources]? Systems science studies, seeks to describe and explain, systems in nature and society. In the context of this study, systems thinking is an approach to organization and problem solving that considers the parts of a system as interconnected (interrelated), rather than independent. The systems approach enables an understanding of the relations and interactions between the various components of a system. The adoption of systems thinking can be especially helpful in illustrating the complexity inherent in socio-technical systems through better problem definition processes and visualizations; synthesizing complex wholes, as opposed to breaking them into parts; understanding causal relationships between parts; and putting forward differing perceptual views by creating awareness of the differences in social relations. The application of systems thinking in design is a required approach to address the increasing complexity of society and societal problems. The systems approach is explained in depth in the Social System Specification.

> **NOTE:** *A classification of systems approaches is detailed in the Social System Specification, which aims to identify relevant criteria for the adoption of systems thinking into design and operations.*

Technically speaking, every societal solution design specification is like a societal motion tracking signature (based on a pre-existing information system), and can be identified again, because of its patterns of information that form its model. Every society can have a model built for (or, of) it, to which other models of society can be compared (to identify individual differences).

Systems theory is a formal language, and like any formal language (e.g., mathematics) , it is independent of any external subject matter and is solely dependent upon its own internal logic. If logic is consistent, then it 'works' (conveys capability, extensionality). If there are logical inconsistencies within the syntax of the language, then does not work. The same is true of any formal language (e.g., the programming languages that operate instructions in a computer).

1. Errors in syntax cause mal-functioning.
2. Errors in communicating syntax as semantics cause [human] mis-understanding.
3. Errors in symbolically reifying (real world modeling, pragmatics) cause [human] mal-adaptation.

If systems science is the application of systems organization to scientific inquiry, then that is a strange definition, because science is a systems-based method. In other words, to select scientific inquiry as the method of discovery is to have selected a systems based method. One of the definitions of science is rational explanation. The scientific method is the method of rational, physical explanation using the language of systems. In the case of physics, science is rational explanation involving physical objects and cause mechanisms, of phenomena that occur. Systems science is just science, it is called systems science because in the market, the scientific profession is divided by discipline, and so the term system is often added before science to show its systematic application in a particular context. Science is a process for explaining the workings of systems in the real world. In their design and operation, systems involve measurement, applied toward the useful representation of information.

Science is an empirical [system] method; meaning, that it is dependent upon reference to some real world experience ("subject matter") in order for its validation. Whereas collection and analysis is the information processing domain of science, design and operating is the information processing domain of engineering. Set theory mathematics currently acts as the formal proving method for proof of scientific fact validation. Systems-based language formalization is the foundation out if which to build a robust framework for highest fulfillment of all of humankind.

Systems science is a framework powerful enough to describe our 'world' and our 'work', in all its richness. Our working world requires the qualitative capacities of system science that allows us to properly contextualize existence and the rigorous quantitative methods of analysis that allow us to properly compute this information, with the net result as a full[-fillment] visualization of the real world. Along with the individually social method of participation, contribution, humanity has the tools it requires to flourish.

Science as a single body of knowledge, must by definition, be unified. Systems science is a holistic approach the inputs, processes, and all possible outputs, together. Seemingly separate domains, upon closer inspection, fit together at points of integration.

Systems science allows for a recognition of the important interplay between people and technology, and may thus be considered an accountable method for socio-technical understanding and the foundation of [systems engineering] development.

Traditional science rests upon an objective view of the world (Read: analysis) which rests upon removing the subjective interpretation of the view from the model. Systems science is philosophically sophisticated enough to deal with the questions surrounding the subjective nature of the human experience (human condition) that are required to truly resolve an optimal society for all human individuals.

When we infuse belief into any step of the problem-solving process, it can easily become the frame through which all outcomes are viewed. In the market system of belief, in societal projects, a solution is not deemed successful unless it carries a financial upside. This financial upside doesn't have to mean actual revenue; it can simply mean shareholder market value, as is see with many software companies. Whether the solution solves the original problem or not is almost entirely irrelevant to market value. This prioritization of profits over progress puts a ceiling on the amount of real, human value we can actually deliver. It also papers over any resulting collateral damage. In this sense, the idea of human-centered design is about prioritizing human needs over human beliefs.

This principle isn't just about human life sustainability; it's also about the quality of human design solutions. At the societal level of intersystem team operations, the standards and protocols used to control life need fulfillment and life safety are developed based upon societal inquiry (including risk) tolerances aimed to meet the human requirements of all individuals. It is through standards and protocols that we discover, and it is through discovery that we improve standards and protocols.

Nothing create exists in isolation; it all lives within the overall natural cosmo-ecological system. Through systems science, solutions [for the next human societal iteration] can be constructed with safely optimized tolerances that support every individual human of that system. Through science and engineering we are likely to actually craft a more effective and efficient societal solution for human fulfillment.. Here, holistic thinking focuses on problems, processes, solutions, and orientations, given an environment with probabilities.

Does every problem need to be solved? The question of whether, every problem needs to be solved, is a useful one to facilitate recognition between actually necessary, and actually unnecessary, problems? Do any market-based problems need to exist? Do products and services in every human system need a trading price? How does belief-centered thinking keeps consciousness locked a self-centered, power-over-others regulated materializing life bubble. Society could otherwise be structured upon verifiable (falsifiable, verifiable and validatable) measures like well-being, sustainability, equity, and growth opportunity. Human life values could become a base filter through which we evaluate all of our design solutions instead of otherwise solved value orientations.

2.1 Data science

From an information system's perspective (because, this is a project to develop and operate an information system > society), 'data' science is a set of understandings concerning the intentional/usable nature of data that conceptually unify studies of statistics, data analysis, machine learning and their related methods in order to understand (why) and analyze (how) actual phenomena occur with data. Data science could be said to employ techniques and knowledge of patterns:

1. **Human patterns, life patterns** - meaningful patterns; the experience of interfacing with patterns that mutually benefit all.
2. **Mathematics, conceptual-visual patterns** - computational patterns; the visual language of pattern recognition.
3. **Linguistics, language patterns, language science** - unified communications patterns; the unification of linguistic communication, in order to facilitate precision of communication.
4. **Informatics, informational patterns, information science, data patterns** - modeling patterns; the logical way of building and visualizing observable patterns.
5. **Analytics, pattern recognition, pattern comparative de-composition, analytic sciences** - the collection and de-composed recognition of patterns; a way to observe and de-compose patterns from data based on an existing model.
   A. Social system > Discovery.
   B. Decision system > Data acquisition and recognition.
6. **Statistics, pattern prediction, computational understanding sciences, mathematical sciences, inquiry threshold resolution patterns** - predictable patterns; a way to infer patterns from data based on an existing model.
   A. Social system > Knowledge development, memory, and search.
   B. Decision system > Parallel inquiries resolution thresholds.
7. **Computer science, computronics, pattern computation, computer language(s)** - soft patterns, a way to build algorithmic patterns. These are conceptually/mentally/consciously interface-able patterns) based on data of an existing model (i.e., software).
   A. Social system > Application to computation, conceptual automation [inquiry].
   B. Decision system > Solution Inquiry (in part), computational decision algorithm, which can be visualized and understood by the humans using it).
8. **Material science, pattern spatialization, spatial patterns, object patterns** - hard patterns, a way to build material/physically interfaceable patterns based on data of an existing model (i.e., hardware/ hardware modules).
   A. Social system > Application to spatial, physical conscious experience [inquiry].
   B. Decision system > Solution Inquiry (in part), material objectives' encoding algorithm, which can be visualized and understood by the humans using it.

The following is sometimes said of the following processes:

1. Scientists care about understanding why something works the way it does.
2. Engineers care about how something works; and thus, whether something works or does not work.
3. Developers care about when and where something is to work (note engineers are also developers).
4. Coordinators care about access to working information.
5. Users care about how much something works as required or expected.

2.2 *What is a systems-based form of organization?*

CLARIFICATION: *While many definitions of the word "system" exist, nowadays, the concept, 'system', is more and more frequently used, in different domains, to refer to a real world set of bounded dynamics, as in: a software system, a hardware/physical system, a social system, an economic system, a service system, etc. In each domain the meaning of the word "system" may have nuances.*

There are a large range of accurate definitions in the literature for the term, 'system'. In its most broad definition, a 'system' is an integrated set of interacted and organized elements and related processes. The following is a common, comprehensive, list of definitions of the concept, 'system':

1. Autonomous entity with regard to its environment, organised in a stable structure (identifiable in the course of time), constituted by interdependent elements, whose interactions contribute in maintaining the system structure and making it evolve.
2. A system processes inputs into outputs that achieve and satisfy a purpose or purposes through the use of resources in an environment.
3. Aggregation of end products and enabling products to achieve a given purpose (ANSI/EIA 632, the earliest definition of a system to identify the components and the purpose of a system in its definition).
4. Combination of interacting elements organized

to achieve one or more stated purposes (ISO/IEC 15288).
5. Set of elements and a set of inter-relationships between the elements such that they form a bounded whole relative to the elements around them.
6. Set or arrangement of elements [people, products (hardware and software) and processes (facilities, equipment, material, and procedures)] that are related, and whose behavior satisfies operational needs and provides for the life cycle sustainment of the products (IEEE 1220).
7. Integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, and firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE 2010).

> **NOTE:** *Many standardized definitions of what a system "is" are available, including but not limited to: [ANSI/EIA 632, IEEE 1220, ISO/IEC 15288, INCOSE SEBOK, TAP CDS-SS-01.*

A system, itself, is completely defined by specifying (or otherwise, describing):

1. What the system does.
2. How the system does it.
3. What the system uses to do it.
4. Where the system lives ("is" in relation to a larger unified information system)

A system can be comprehended in its entirety through integration of various seemingly separate views, which unify the systems view:

1. **System context (system environment)** - context exist as the circumstances, factors, conditions, or patterns that enable or constrain system solutions.
2. **System bounding (system interface)** - the bounding of systems along different dimensions (geographical, physical, time, conceptual).
3. **System concept (system definition)** - the characteristics, properties, and classification of a system as a system of systems.
4. **System analysis (system de-composition)** - the development of approaches to engage in "holistic" analysis for systems.
5. **System transformation (system process)** - the nature, framing, and approach to transforming systems of systems from a "holistic" perspective.
6. **System representation and modeling (system intelligence)** - the distinction in modeling approach, and the role of representation, for systems.
7. **System intervention (system change)** - the design and deployment of initiatives to purposefully modify a system.
8. **System development (system prototype)** - the execution of methodologies an environments necessary to engage in systems engineering.
9. **System serving (system operation)** - the development of guiding frameworks and platforms to support [human needs through] system engineering execution.

Wherever there is technology and population that values efficient and effective alignment, then automation and measurement will likely play an important role in service operations. At the societal level, systems science necessitates measurement, and systems engineering necessitates automation.

2.2.1 Systems-based work organizational concepts

The basic systems concepts of 'organization' as applied to 'work' (useful effort) are:

1. **Order** - An order is a permutation of a list of items, where you are trying to find the best way to arrange a set of given values.
   A. A societal service system.
2. **Grouping** - The Grouping method assigns variables into sets.
   A. An InterSystem Team of individuals.
3. **Budget/Threshold** - The Budget/Threshold method is similar to Recipe except that all of the variables' values must total a number. This method is designed to run budget calculations or assign resource allocations with a Recipe solution in which the total is kept constant.
   A. Socio-technical resource accountability - decisioning that concerns material composition and position).
4. **Schedule** - The Schedule method is similar to Grouping except that it assigns elements to blocks of times while meeting certain constraints. This can be used to assign workers or courses to time periods or schedule meetings.
   A. Socio-technical event accountability - decisioning that concerns material positioning and timing).

2.3 *What is the systems approach?*

> *A.k.a., What is a 'working'-type systems organization?*

The systems approach is an approach that produces a working systems organization. Systems are processes organized in structural and functional hierarchies. Since all components, and their interactions, exist only as processes unfolding in time, the word system and the word process are essentially synonyms. Systems are structured hierarchically (logically). As processes, functional hierarchies correspond with the structural hierarchical architecture if systems. Systems naturally organize the work they do by functional hierarchy. A system may consist of several levels where each element at each lower level may by this definition itself be considered a system (i.e., a subsystem of a large system may itself possess all of the attributes of a system).

In a general sense, the concept of a 'system' is applicable to all things, contexts and situations. In other words, the use of the word 'system' can be applied to everything: all situations and contexts, all behaviors and environments, all organizations and experiences, all definitions and explanations, and all visualizations.
Visually, a system is a mapping (visualization) between a set of inputs and a set of outputs. Wherein, there is a relationship between the inputs [entities] and outputs [entities] by means of process [entities]. Here, shape, position, and motion form visualization.

Hierarchies are recognized as the means by which systems naturally organize the work they do. Analytical tools decompose a system. Because systems function through operational hierarchies, it is best to design systems as a hierarchy of components (concept through to material) integrated into working modules, which, in turn, are integrated into meta-modules, the top level of which, at least for society, is the unified societal system. Systems are networks of components tied together via links representing different kinds of relations and flows. Dynamics refers to how the processes operate or change inputs into outputs over time. Systemness is a recursive property in which, starting at some level, one can go up or down. A sub-system cannot extend beyond the capacity of the total system of which it is a part, nor can a sub-system be understood except through the larger system of which it is a part.

> **CLARIFICATION:** *A tool is some "thing" (physical or informational) used to carry out a specific function (task, or job).*

In order to understand a real world system, it must be studied and engineered as a process, not just structure. As processes, functional hierarchies correspond with the structural hierarchy of systems.

> **INSIGHT:** *In a system it is most effective to distribute tasks and processing, but it most efficient to centralize the information system; both can occur in parallel.*

The system's approach describes a system (i.e., a system has the following properties):

1. Holistic refers to a continuous region of space/time, that is viewed as a single entity identifiable by properties manifest at its boundary, and is identified generically as the system-of-interest or specifically by a meaningful descriptor.
2. Closed boundary refers to the terminating surface that limits the region of consideration from the space/time continuum that it exists without, i.e. its environment, and across which flow interactions between the system and its environment.
3. Elements refer to the complete set of discrete subordinate entities that comprise the whole, each having a different homogeneous nature and identity relative to all other members of the set;
4. Order refers to the arrangements of elements, their functioning and their relationships and their precedence in a hierarchy of consideration;
5. Interaction refers to all the mutual influences that each element has with all other elements;
6. Properties refer to all qualities that emerge at the level of the whole in all degrees of freedom as a result of the combinatorial effect of each individual entity, one on another.
7. A system is a state of energy and matter with distinguishable arrangement. The reasoning mind is tuned to define regions and to degrees of ordering within them.
8. A system is most effectively defined by boundaries that encapsulate meaningful need and practical solution.

> **NOTE:** *A complex system has both structure and process.*

The more common, though broad characterizations of a system include:

1. A system is a whole composed of parts, and there is a similarity (resonant quality) between the whole and the parts.
2. A system is, in part, defined as a set of system elements that interact to achieve, output a defined mission, input.
3. A system is a hierarchical composition of [system] elements. Each [system] elements will need to perform functions that have been allocated to it so that it can contribute to the system's existence, objective, or purpose (as in, imperative or mission).
4. A system's objective is broken down into a hierarchical structure of its functions. The logical description of a system's mission is broken down into a hierarchical structure of its major functions to form a functional hierarchy, or a functional architecture.
5. The physical hierarchy [of a system] consists of, for example: system, sub-system, assemblies, components.

All complex system design and development occurs through a project-based structure (coordinating the designed resolution to commonly indicated problems):

1. A project coordination (management) process (e.g., one that can easily be applied to all societal systems).
2. Common indicators
   A. Indicators that allow someone to check how the users handle any mismatch between expectations and results. These expectations may concern:
      1. The system to be built (as viewed from the angle of the product or service), or
      2. The system for creating (as seen from the viewpoint of performance, stability, and integrity of the organization supporting the project).
   B. Construct aggregate indicators and dashboards providing an overall process transparency capability.
3. Designing a system for an integrated coordination:
   A. Define mechanisms that provide an objective tool ("aid") for taking into account the needs of stakeholders and following-up, verifying, and validating these needs according to the indicators selected.
   B. Anticipate and plan the efforts needed (as in, activities and tasks), to check and validate both systems (i.e., the system to be built and the system for creating).
   C. Mechanisms for tracking any malfunctions by using trend analysis.

2.3.1 Visualization

A.k.a., Modeling.

The second form of visualization, after shape, is structure. A structure is an ordering of objects. Objects and structures can and cannot have motion. Objects and structures without motion are static. A combination of moving objects is a dynamic.

Herein, experience arises through the conscious ordering of structurally static and dynamic shapes, which can occur both at an information (conceptual-interface) level, and at a material (physical-interface) level:

1. **Experience** - An orderi[ng memory] of consciousness is an experience.
   A. **Shape** - An order of identifiable [geometric] patterns is a shape.
      1. **Structure** (order or parts) - An order of parts is a structure. Structures can be characterised as having or not having motion (internal and/ or external)
         i. **Motion** - An order of operations is a motion..
            1. **Static** - no motion (internal / external). No motion to the visualization experience.
            2. **Dynamic** - motion (internal / external). Simulation of the visual experience.

2.3.1.1 Feedback loop models

Feedback loops are the building blocks of systems' dynamics (i.e., systems' control of behavior). A feedback loop is a structure within which a decision variable (flow) controls an action that is integrated into the system to generate a system state. Information pertaining to the state is then fed back to the decision variable, which in turn is used to control the flows. Two kinds of feedback loops comprise all complex behaviors of a system:

1. **Positive feedback loop -** Positive loops are self-reinforcing and tend to amplify whatever is happening in the system. In this case, bad situations/behaviors are likely to be amplified; or, good situations/behaviors are likely to be amplified.
2. **Negative feedback loop** - Negative loops are self-correcting and tend to counteract and oppose changes. An increase in one parameter causes the other parameter to increase, which then decreases the first parameter.

A feedback loop is composed of two kinds of variables:

1. **State** - State is an accumulation characteristic of the state of the system that generates the information upon which decisions and actions are based. A state variable is altered by inflows and outflows and is represented by a rectangle in a model.
2. **Flow** - Flow is a variable that changes a state over a period of time. Flow variables are of two types: An inflow increases a state and an outflow depletes a state. In short, a flow is a statement of system policies that determines how information about the system is translated into action(s).

2.3.1.2 Causal loop diagramming

Causal loop diagrams (CLD) is a systems visualization language composed of a framework of rules for seeing interrelationships rather than just things. For seeing patterns of change rather than static snapshots. A causal loop diagram has two entities:

1. **Variable** - state, condition, action, or decision, which can influence or be influenced by other variables. A variable can be quantitative (number or value of some thing), or it can be qualitative (objectives, values, feelings, non-functional requirements).
2. **Arrow** - indicates a causal relationship or change of

the state of new variables.

A causal loop diagram shows the visual dynamics of inter-relationships. Those salient variable points are identified (or, scheduled) in time as events (or, milestones and tasks). A causal loop diagram is a systems-type modeling tool and can be analyzed by identifying feedback loops formed in the model. A real world causal loop diagram would normally have feedback loops. A feedback loop arises when a sequence of interactions between variables through arrows form a closed loop. The feedback loops can be reinforcing, or balancing, which are visualized, which then become visible as task-activities. For example a recent analysis of the biosphere on Cat Ba Island in Vietnam identified ten reinforcing (R) and five balancing (B) loops. (Tri et al., 2018)

There are two types of feedback loops:

1. **Reinforcing loops** - positive feedback systems that represent growing or declining actions, or information cohesion.
2. **Balancing loops** (negative feedback loops) - negative feedback loops seek stability or return to control; for example, those designed to control automated vehicles and service bots.

2.4 Modeling system dynamics

A system dynamics (objects-process) model can be used as a virtual world to simulate real-life material situations. A virtual world is a formal model, simulation, or "microworld" in which decisions can be taken (i.e., there is choice), experiments can be conducted, and situations can be acted out (i.e., simulated), in order to more greatly understand.

Everything in physics, in engineering, is a model. A model is a set if "ideas" about the ways some thing works. A model explains the facts (the meaning, explains the experience).

2.4.1 Modeling system objects

All objects have the property of shape, and all shape is geometry. Therein, objects perform motions. The objects themselves, their relationships and motions may be modeled (as in, identified by rules and explained by visualization of the objects and their relationships).

2.5 Why is the systems approach used?

The systems approach may be used by all conscious individuals to ensure the freedom, efficiency, and effectiveness of all cooperation. The systems approach greatly facilitates certainty of directionality in an uncertain environment.

2.5.1 Evidence of claim to existence

In its real-world application by embodied consciousness, systems science encodes three primary types of evidence for individually, conscientiously considering (and socially "taking") a claim to existence:

1. Physical observation (sensation, perception).
2. Physical explanation (physics modeling).
3. Statistical evidence is demonstrated by data analysis on a study:
    A. Clear evidence.
    B. Some evidence.
    C. Equivocal evidence.
    D. No evidence.

2.5.2 Data validity

'Validity' is traditionally understood to refer to the correctness or precision of a data reading. Validity concerns measurements 'truly' recording what they intend to measure. In qualitative research it concerns the extent to which the phenomena under study is being accurately reflected, as perceived by the study population. Validity has two dimensions, internal concerned with the success of the research to investigate what it claims and external concerned with applicability of the abstract constructs to other populations.

> **NOTE:** *If it is 'valid' science, then it is 'valid' science, and it doesn't matter who is doing the science.*

2.5.3 Data reproducibility

Reproducibility is the systematic reproduction of a system, or set of data. Reproducibility is the foundation of modern systems science. If there is not reproducibility, then possibly it could be a mistake, error, fraud, corruption, or just a conflict of interest. It's not science as a body of knowledge until it has been tested, checked and replicated. Science is based on being able to understand and to reproduce a result. In order for data to be use useful, engineering knowledge, it must be reproducible. When science is going to be used for engineering into human lives, it is tested first.

> **NOTE:** *Modeling (analytical-synthesis, computation) is not the same as scientific inquiry.*

Science is used to discover data from an uncertain environment. Sensors are used to discover data from a certain environment, because the sensors are designed based upon an engineering pre-designed and pre-selected model.

2.5.4 Real [world] information system processes

Everything is a system, and every system in an information process. A real system's core information processes can be described in two broad descriptive ways (logical and physical):

1. **Logical (or functional) requirements description (a.k.a., functional hierarchy)** - what the system will do, how well it will do it, how it will be tested, under what conditions it will perform, what other systems will be involved with its operation.
   A. **System logical architecture (functional architecture; system development)** - outlined in *requirements breakdown structure*.
2. **Physical/material requirements description** - what the system elements are, how they look, and how they are to manufactured, integrated, and tested.
   A. **System physical architecture (system development)** - mapped onto the logical architecture as represented by [the configuration items contained in] the *work breakdown structure*.

   ***NOTE:** *In general, the logical description of a functional system tends to change slowly; whereas the physical description tends to change much faster as knowledge and technology advances.*

2.5.5 System information flow modeling

Systems can be described in various ways by their expressed type of interactions (information flow relative to the system boundary):

System types (per type of environmental interaction):

1. **Open system** - interacts with the surrounding environment through a boundary.
2. **Closed system** - does not interact/exchange with the surrounding environment.

System types (by internal interaction):

1. **Transformational** - a process that receives one or more system inputs I from an external environment, transforms them with process T, and then releases them as system outputs O to an external environment. A transformational system generates an output and them terminates.
   A. Single input/single output.
   B. Multiple input/multiple output.
2. **Reactive** -a system that,when turned on, is able to create desired effects in its environment by enabling, enforcing, or preventing events in the environment. Reactive systems are involved in a continuous interaction with the environment. Wherein, the environment generates input events at discrete intervals through on or more interfaces and the system reacts by changing its state and possibly generating output events.

System types (reactive types):

1. **Real-time systems** - a system in which the correctness of a response depends on the logical correctness and time at which the response is produced.
2. **Safety-critical** - malfunctioning of the system could lead to a loss of life or the system itself.
3. **Embedded** - the system is embedded within another system.
4. **Control** - determined and/or generate a desired behavior in the environment.

Some common characteristics of a life-system type of organization include:

1. **Emergence** - the way in which complex systems and patterns arise out of a multiplicity of relatively simple interactions. Something unexpected in the collective behavior of an entity within its environment, not attributable to any subset of its parts, that is present (and observed) in a given view and not present (or observed) in any other view). Other definitions state that that which emerges can be expected as well as unexpected benefits or consequences. System properties emerge from the synthesis of the interactions between components, at each level of interconnection within a system. This emergent behavior is something other than what is seen at the level that gave rise to it. The concept of emergence as representing the collective behavior of the system elements that reside in a lower level. The behavior cannot (generally) be predicted from or described by the properties of those elements, but is something unique that manifests when all those elements are joined together and interact with each other. The concept of emergence is intrinsic to all types of systems.
2. **Taxonomy** - an arrangement of concepts and/ or objects in which items are presented as being above, below, or at the same level as one another.
3. **Hierarchy** - an arrangement of items in which the items are represented as being above, below, or at the same level as one another.
   A. **Layered** - the hierarchy of system components is clustered into horizontal strata (e.g., open system interconnection, osi, model for computer communications, or google earth GSI with layers of data overplayed onto on the real world geographical model).

B. **Network** - a set of elements (or modules, nodes, or devices) that are connected by a set of interfaces (or links or commas channels or protocols). Formally, a network is a graph. A network topology describes the connectivity (or arrangement) of nodes on a network.

2.5.6 Information systems organization

> **NOTE:** *In information sciences, an organization is a variant of a clustered entity (or its equivalent).*

A 'system' is also known as an 'organization' (in various English contexts). An organization is an identity (a system has a boundary) in which there are component (combined, together) parts (a system has sub-systems).

There are two principal types of organizations, either as structure or process (note, these terms are often used interchangeably in common parlance).

Organization as structure (noun):

1. Function and Condition *of structure.*
2. Shape and Geometry *of structure.*
3. This is prior action.
4. This is project life cycle [knowledge].
5. This is data architecture [in an information system].
6. Structures are designed (aligned), selected, and implemented.
7. Enters materiality as physicalized hardware and software assets (I.e., asset categories).
8. Organizational structures determine information flow within an organization.

Organization as process (verb):

1. Input and Output *of process.*
2. Equation and Algebra *of process.*
3. This is action.
4. This is project progress/process [groups].
5. This is computation [in an information system].
6. Processes are designed (aligned), selected, and executed.
7. Enters materiality as abstract 'service' categories (e.g., HSS).
8. Organizational processes compute information flow within an organization.

Service organizations have a (1) function, and they, (2) will do it at a specifically pre-set quality (it is when we contribute that we may truly do:

1. A 'function' [is a process into] transforms an input into an output
   A. An 'operational process' that uses resources and transforms into outputs, the inputs of individual humans.
2. A 'condition' [is a structure that] orders (regulates or qualifies) how an input is transformed into an output
   A. A 'quality' evaluation of expectation as pre-set by an individual human user's consciousness.

2.5.6.1 Organizational structure

An organizational structure defines how activities therein (e.g., resource allocation and work coordination) are directed toward the achievement of organizational objectives. An organization[al structure] can also be considered the view, visualization, model, or perspective through which individuals observe an organization's presence. And, the organization's observed behavior may be viewed as its active [operational] processes (which exist in relation to an environment and physics).

2.5.6.2 Organizational knowledge

In an information system, knowledge is structure-organization-process (a complete information package) with a high certainty in its alignment to real world existence; thus, carrying a usefulness in navigating within the real world. It is from this understanding of 'knowledge' that societal-level life[style] fulfillment becomes possible.

2.5.6.3 Information system data types

User view of data input types:

1. Data **having** type - was the user prompted to input information (i.e., is the user having a prompt to enter input)?
   A. **Prompted** - Requested input (i.e., requested data is input).
   B. **Unprompted** - Non-requested input (i.e., non-requested data is input).
2. Data **being** type - is the input information numerical (i.e., what type of data pattern is being input)?
   A. **Functional (quantitative, numerical)** - Numbered input (i.e., data of type 'numerical' is input).
   B. **Non-functional (qualitative, conditional)** - Non-numbered input (i.e., data of type 'linguistic' is input).
3. Data **doing** type - does the input information conform to standards (i.e., is the user doing the input information correctly/coherently)?
   A. **Structured information** - Structured input; data fits into model precisely (i.e., data is input per standard structure).
   B. **Unstructured information** - Non-structured input; data does not fit into model precisely (i.e., data is not input per standard structure).

Developer/operator view of data input types:

1. Conceptual design (What, definition)
   A. Technical design (How, explanation)
      1. The design needs to be:
         i. Correct and complete.
         ii. Understandable.
         iii. In alignment with organizational protocols.
      2. The design needs to satisfy a validation criteria:
         i. The users direct requirements.
         ii. The organizations requirements.

In order to meet the expectations of the user, developer, and operator, the system should be sufficiently curious (inquiring) about what [changes] may be needed:

1. A prompt is a mechanism to capture the answer for a specific question.
2. A prompt is a sign on the screen that shows that the computer is waiting for input. The answer provided to a prompt is stored as a parameter that can be used by another question or as a filter value for a data query.
3. A prompt is a way of assigning members to a dimension. Note that in psychology, priming is a prompt is something that is added into the environment to help elicit ("cue") a correct response.

2.5.6.4 Materiality data (spatial data)

Spatial data [infrastructure, SDI] is a data framework of geographic data, metadata, users, and tools that are interactively connected in order to efficiently and effectively modify the environment. In general, this refers to the layered overlaying of data upon a visual reference of the geographic world.

2.6 What does society have as result of systems science?

A.k.a., Doing work systematically, society has stability. The design of an organization is causatively dependent on the requirements of that organization.

The completion of a set of studies that by some relatively designed degree absolutely provides the data required to resolve a new societal system, in the now, and therein, service usability is the result of 'systems science'. These studies inquire and account in order to meet life, technology, and exploratory demand in an uncertain, by degree, environment.

The "proof of truth" is not in the authority [of an expert], but in the experience of using the formal language of systems to represent a [working] real world (and have the individual conscious experience of that proof match with the linguistically shared model). As a data type, a fact is a description of that which has occurred (record of event), is occurring (executing and monitoring if event), or will occur (event predictability/probability). A societal systems-level proof necessitates the application and resolution of an operating system (life platform, where 'life' is the true alignment) populated with factual-type data [about life in the real world]. That which is a proof of true alignment with the highest qualities if life, is that which can be validated.

Both discoverability and reusability are critical to ensuring the reproducibility of the research, a basic principle of the scientific method.

1. **Discoverability** is the ability of a data set to be discovered by someone else.
2. **Reusability** is the ability for a data set to be used again by its producer and/or someone else.

Note that most factual descriptions carry the unifying reference record (meta-tag) of 'certainty of the fact', given all that is known currently by the unified system. Without unity (integration) there is no trust (or less trust) in the certainty of any record. In a dis-unified information system, there is some amount of uncertainty that could be avoided through more unity (more integration). With more integration, there is more trustability and less uncertainty. For a user, a high-quality service is a service that can be trusted. For a user to fully trust the quality of a service, everything about the service must be transparently integrated [at an interface for the user]. For a contributor, a high-quality structure is a team structure for which there is appropriate certainty that contribution will be effective.

NOTE: *In general, more [accurate] information facilitates uncertainty reduction.*

Whether someone is competent and qualified on knowing something, or knowing how to do something, from information in the unified societal information system is not a matter of opinion; it can be verified. In this way, the knowledge and actions of an contributor ("expert") can be tested. In an unaccountable or dis-unified system, where verification and accountability are less present, then "experts" and their "decisions" (opinions maybe) are less trustworthy.

In community, the idea of [a separate group of people known as] "experts" taking decisions on the part of others, is not only a dis-empowering viewpoint (because anyone given motivation can become one), but is also factually incorrect when society is viewed from a project approach (because projects exist for users).

In a societal structure divided by in-group bias, then decisioning and control by socio-technical "experts" may not be desirable for widespread human well-being, because the "experts" cant be verified through the a unified information system. And when an authority figure becomes the source [of all] experts, then the social honor of being an "expert" holds even less reliability ("credibility"). Rule by experts (who cannot also be oneself) is unlikely to create an optimally fulfilling set of conditions. What truly threatens the loss of fulfillment and knowledge is the loss of a contribution-based structure. The way to "protect" knowledge, to know the difference between truth and falsity, is to have unified information access (to ensure transparency), have a method (by which to determine either truth of false), and to use collaboration, where those individuals doing the work are verifiably competent (or in training, and their work can be validated to be so). The idea of rule by experts carries with it that the idea that who they will be ruling are intellectually passive consumers. An open source system could have useful contribution from anywhere.

It is essential when working together to not replace the individual experience of proof (upon the part of any user or contributor) with any authority [figure or leader]. In the context of, "Where does the project propose that 'authority' lie?", the following questions are used to bound the solution to that inquiry.

1. This project does not propose a system controlled by an expert-ruling elite, a technocratic authority.
2. This project does not propose a non-factual (opinion-type) decisioning structure, a political authority.
3. This project does not propose a secret and closed information structure to coordinated societal organization, private [ownership] authority.
4. This project does propose to account for the factual position and composition (past, present, and future) of resource configurations (i.e., of material solutions).
5. This project does propose to account for discoverable human needs within a common, real-world human environment.
6. This project does propose to account for the use of a specific set of value conditions to evaluate the results of different solution configurations. And, this project has reasoned the selection a specific set of condition encodability statements (i.e., the value statements of freedom, justice, and efficiency as core to the economic, parallel socio-decisioning protocol).

2.6.1 How could society be organized through systems science?

Systems science is unique in its mode of inquiry in that it reveals not just how one kind of system, say a biological system, works, but rather how all kinds of systems work. That is, it looks at what is common across all kinds of systems in terms of form and function. In this sense, it is a meta-science, something that informs all other sciences that deal with particular kinds of systems. In part, systems science (a.k.a., information science) is a formal language or formal logic, which is internally consistent and useful for modeling and interacting in a real world.

When applied to the human context, systems science has two problem-based information orientations:

1. There is the <u>problem of understanding</u> the world.
   A. Science explains the mechanism.
2. There is the <u>problem of changing</u> (developing) in the world.
   A. Engineering applies the mechanism.

In systems science, there are three primary questions that acquire information and compose its information set:

1. **Epistemic questions** (*philosophic questions, data structure*) - questions that concern the axiomatic, non-contradictory, and structural flow of information.
2. **Physics questions** (*scientific questions, discovery structure*) - questions that concern shape.
3. **Applied physics questions** (*engineering questions, operating structure*) - questions that concern the application of shape [in service] for a[n intentional] function.

At the societal level, there are three systems science problems domain, the resolved inquiry of which is an optimally discovered societal service system, given what is known:

1. System **application** - *the specific classes of problems* that are appropriate for the usage (ability) of systems science (scientific inquiry, project/information coordination, and engineering).
   A. Systems applications; systems science as a functional service to some user who has requirements (an intention). Discover how to identify information.
2. System **method** - *the specific function that resolves the problem (solution)*, given multiple names, including but not limited to: techniques, processes, or tools; all of which are used in applications.
   A. Systems methods - systems science as a body of method-based knowledge. Discover how to transform information.
3. System **team** - *the specific humans and technical systems* that execute the method, the highest level of which, in the context of a societal system, is the InterSystem Team. Discover how to most efficiently and effectively operate as a [social] team based upon information.
   A. Systems teams - systems science as a cooperative structure (and body of knowledge) coordinating the experience of a solution to a set of human requirements. In other words, it is

here that society is "executed" for the fulfillment of all as a (or, through a) unified system.

In the content of intentional-conscious change to existence, systems science can be partitioned into five information sets (areas) that form a solution to human requirements:

1. System **axiomatic** - *the accepted knowledge* (principles, theory, concepts, rules/laws) that explain systems and their associated phenomena.
   A. Systems conception - Instantiation of two (or more) objects and a relationship [as a system].
2. System **philosophic** - *the epistemological* (Read: how any system may come to be known by following the flow of information to its source) and *the ontological* (Read: how systems are realized, as shape and structure, at various levels of the world of observation).
   A. Systems visualization - Instantiation of a data structure [for a system].
3. System **methodological** - *the reasoned logical selection of systems-based methods* to inquire into and gain knowledge concerning systems, and how they may be most optimally changed.
   A. Systems logic - Instantiation of a replicable pattern for accessing information [about a system].

NOTE: *Often, the term 'philosophical' is used to describe the core conceptual reasoning for an ontological, or unified life, model.*

Real-world systems are understood through object-processes that form the state of a system; thus existing:

1. Objects exist - objects are that which exist, or can exist. All objects have shape.
2. Processes exist - transformations of objects. All processes transform objects by generating, consuming, or affecting them.
3. States exist - identifiable synchronization of object-processes. All states expresses the situation at which an object can 'be' in a conditional relationship to other object processes).

Herein, a system is an object with a structure, that does a functional process, that expresses a behavioral state condition:

1. Function - what the system does.
2. Structure - how the system is constructed.
3. Behavior - how does the system change, or how is the observably system expressed, over time.

In relation to the systems method,

1. A system is an object.
2. All objects have structure (i.e., shape).
3. All designed (active, dynamic, in motion) structures have a function, represented as a process (a type of sub-object).
4. All designed process functions express sensible (observable) behaviors that change the condition(s) of their environment.

2.6.2 Information flows

Fundamentally, as a result of the application of systems-science, society has awareness and the capability to work with information flows (of a conceptual and spatial nature). For there to exist human global cooperation, it is essential for humanity to have globally transparent awareness of all relevant information flows.

A "flow" of 'information' is defined as a unidirectional series of related data -- a set of 'information' "packets" passing through an observation point during a certain 'time' interval. In an information system, 'flow' is the observed or predicted motion of information. The motion of all information constructions in the conceptual and spatial systems can be planned (with some degree of certainty).

NOTE: *To "flow" is to move, transfer, or behave. There are many types of information movement (e.g., sorting, translocating, calculating, encoding, etc.).*

In any given information system, information generally flows from:

1. Conception (ideation), to
2. Decided execution (algorithm), into
3. Materialization (production-operations), and back again as
4. An information issue (conception), whereupon
5. The materialization is measured and its alignment in quantity and quality are assessed.

There are two general types of information process flow (Read: information flow model types):

1. **Linear type (linearity)** - the process flows sequentially without repetition (or, iteration). In geometric navigation, this concept is visualized as a line.
   A. A sequential flow (motion) of information. Linear, sequential.
2. **Iterative type (continuity/Life-cycle/ extensionality)** - the process flows with repetition (or, continuity; a rotation of the linear into an extension/continuation, of life). In geometric navigation, this concept is visualized as an arc (a line rotated to become, or becoming, circular).
   A. Iteration (repetition) of the flow/motion of information. Looping, overlapping.

B. An iterative process with memory is evolutionary (i.e., is an evolutionary/adaptive process flow).

All process flows can be visualized, because in all process flows there is an object with shape and/or a conception.

A flowchart (a.k.a process flow diagram, chart) - is a visual representation of the sequence of steps and decisions required to perform a process. Each step in the process of information flow is noted with a diagram shape. Objects and steps are linked by connecting lines and directional arrows. Each object/step can be made up of either: a concept (pure information), an object (geometric shape), or two objects and a concept.

The visualization of information process flows is necessary for shared creation and operation:

1. Effective **understanding** of a process.
2. Effective **communication** of a process.
3. Effective **execution** of a process.

Common elements that may be included in a[n information] process flow [visualization] are:

1. Sequence [of process].
2. Inputs and outputs [of process].
3. Decisions [of process].
4. Activities [of process].
5. People [of process].
6. Time [of process].
7. Measurement [of process].

2.7 *What is a living system's approach?*

A living system is the conception of organization–structure–process is one in which a process [self] organizes [its own] structure (autopoiesis). Hence, three criteria are needed for identifying a living system:

1. **The pattern of organization** - A pattern of organization is the configuration of relationships that determines the system's essential characteristics.
   A. Autopoiesis as self-structuring and/or self-replication (defined by Maturana and Varela, 1987).
2. **The structure** - A structure is the physical (i.e., "architectural") embodiment of the system's pattern of organization.
   A. Dissipative structures as defined by Prigogine and Stengers, 1987.
3. **The life process** - A life process is an activity involved in the continual embodiment of the system's pattern of organization.
   A. Cognition as defined by Gregory Bateson, 1979.

All [living] systems can be sub-composed by the three axioms (vectors, ontological forms) of systems:

1. **Shape**: Structure refers to the attributes - distinguishing some thing (trait, value, shape and efficacy) from other things. Structure refers closed systems (or the attributes of the universe that are independent). Also, structure refers to individual things.
   A. A body.
2. **Relation**: Organization refers to parts that comprise some thing - the properties (evident by valued traits), and their relationship (evident by their shape and efficacy). Organization refers to open systems (or the parts of the universe that depend on closed systems). Also, organization refers to categories of things (clusters of individuals, where a part is a category).
   A. More than one body.
3. **Transformation**: Process refers to the constitution of parts - the bundle of related properties that produces a whole thing. Process refers to social systems (or the wholes that are inter–dependent on closed and open systems that make up eco–systems, e.g., the universe). Also, process refers to universal things (all things, e.g., parts as the set).
   A. Changing more than one body.

A [living] system may be analyzed based upon:

1. What the thing is composed of (the structures that distinguish it)?
2. How the thing is composed (the organization of the parts), and that a whole thing is an organized structure (the process of comprising the parts)?

As a coherent whole, a living system is:

1. An autonomous entity (i.e., a system is an autonomous entity with regard to its environment).
2. Organised in a stable structure, identifiable over the course of time.
3. Constituted by interdependent elements, whose interactions contribute in maintaining the system structure, and correlate with its evolution or de-evolution.

The primary attributes of the inputs and outputs are:

1. The outputs may be equivalent and/or changed from the inputs.
2. The inputs may be self-causative and/or environmental-causative.

Basic systems terminology for a living system are:

1. Boundary - that which separates a system from its

external environment (e.g., walls in a building).
2. Inputs -elects that enter the system (e.g., raw materials entering a production plant).
3. Outputs - finished products and consequences of being in the system (e.g., a new vehicle leaving a production plant).
4. Threats - those elements that can potentially affect the acceptability of the system configuration (e.g., lack of knowledge, insufficient time, lack of resources, violence, etc.).

It is common to consider the activities being undertaken throughout the life of a real-world life system to be in either the:

1. Problem domain (problem space) where predominantly logical descriptions are used.
   A. A problem space is a "space" of possible problems that form the decision space.
2. Solution domain where predominantly physical descriptions.
   A. A solution space is a "space" of possible solutions, and a selected solution (if present).

2.8 Complex systems

Systems thinking provides the vocabulary and concepts to deal with complex environments. In the real world, there is a systems network. Within the unified systems network, there are supra-systems and sub-systems. The term 'system of systems' is sometimes used to refer to interacting system elements, some of which may be systems in their own right.

Society is a complex of systems (i.e., a system-of-systems). In systems thinking there is a distinction between:

1. Systems as elements of a 'system of systems'.
2. Sub-systems as elements of a system.

From a design perspective, the 'system of systems' comprises systems that have been optimized for their own purposes before joining the systems of systems. Alternatively, a system that comprises elements (system > sub-systems), the sub-systems, that are not optimized for their own purposes, but have been optimized for the system's purpose. From a higher level perspective, a 'system of systems' is most likely not optimized.

2.8.1 Systems bottlenecking

In general, the term bottlenecking means that one aspect of a system "holds back" (i.e., requires inefficiency) of another, keeping it from reaching its full potential. A good analogy is the merging of a five lane highway into/before a single lane tunnel or accident; one part of the transportation system (e.g., accident or tunnel), will be holding back another one (e.g., getting to the destination quickest), keeping it from reaching its full performance potential. Bottlenecking is a systems builder's problem (or challenge) when designing a system to build.

However, from a designers perspective, "bottlenecking", is a misnomer; there is always a slowest component. If the designer/engineer replaces the slowest component with a faster one, then the designer has just created "bottleneck" (i.e., another point becomes the slowest, or least performable, in the system).

2.8.2 Systems hierarchy

PRINCIPLE: *A society that helps everyone help themselves.*

From a designers perspective, a system 'hierarchy' is a system 'accountability structure' with priority processing (given some meaningful purpose):

1. A hierarchy is a tree-type framework (Read: a top-to-bottom flow of information) composed of related levels of information, and the hierarchy ("tree") representing a unification of information.
2. In maths a hierarchy is called a 'directed' graph - branches of information flow from the initiating directive [entity] at the top, down to the lowest level branches (requirements).
3. A hierarchy is a visual elaboration of organization, where each level [in the hierarchy] can be decomposed to the next level down.
4. Hierarchies require numerical or spatial information to identify separate levels.

Hierarchical multilevel structures are omnipresent in living (real world) systems; both in a purely technical context (e.g., cyber-physical systems) and in a socio-technical context (e.g., InterSystem Teams).

A unified hierarchical structure enables organizational:

1. *Accountability* within an environment of increasing technical and organizational complexity.
2. *Efficiency* by breaking issues down into [decisioning units that solve] sub-problems or sub-integrations.

3 Why does this project propose an information system?

Information is an "abstract" form of resources without, which no system could be produced or operated. Living systems use information for control, so that intelligence can implemented. An information system is, by definition, a unified structure of information. Information systems are common to all [human] organizations.

> **INSIGHT:** *When reality is perceived as data, then computers give users the ability to simulate using data. Computers give individuals the ability to access a common simulation using common data.*

For any given society, there may be one unified information systems with multiple sub-system perspectives. A socio-technical information system is a combination of information technology and human activities using that technology to support decisioning and operations (for user function). A project's 'information system' coordinates the integration of project information. A material-type 'information system' is used to refer to the model of all possible interactions between people, algorithmic processes, data, and technology [in a material world], and to sustain the operation of the current model, which is experienced. In this sense, the term is used to refer not only to the information and communication technology a social organization (or system) uses, but also to the way in which people (the social system) interact with this technology in support of self-organizational processes (e.g., human requirements).

The habitat service system is captured by information [as past states, a current states, and future probable states. As part of the material information system, a geographic/geospatial information system stores, analyzes, and models the [commonly] locatable, [within a visually] positional world. A geospatial information system merges cartography, statistical analysis, and database technology with real world objects in real world positions. Therein, the project 'information system' coordinates, and disseminates data, that are linked to decisions with temporal and location relevant information (decisions that affect the materialized/-ing societal system as the understood conception of an experienced existence by consciousness).

> **NOTE:** *A geographic/geospatial information system (GIS) stores, analyzes, coordinates, and disseminates data that are linked to locations. A GIS is the merging of cartography, statistical analysis, and database technology.*

There are two inter-related levels information system operationally relevant for any given individual in society:

1. **A social-level** [information] operating system - the social organizational structure in actualized operation, capability pre-determined through a method of shared visualization and execution.
   A. The development engineers visualize and test services.
   B. The operations engineers execute and control services.
   C. There are two parallel societal decision system inquiry processes:
      1. The social inquiry [solution] process
      2. The technical inquiry [solution] process
2. **A self-level** [information] operating system - the egoic self (i.e., conscious self-modeling), capability pre-determined through a method of self realization and self determination.
   A. The Individual uses and has issues with service. Individuals take decisions when using services.
   B. The individual contributes to the continuation (iteration) of needed services. Individuals take decisions when producing and operating services.

> **NOTE:** *The real-world is a continuous, dynamic, and [partially] observable environment. An environment that is dynamic and partially observable has uncertainty (and therefore, novelty).*

3.1 What is a real world societal information systems model?

A.k.a., Real world societal human information system.

A real world societal information system is defined as:

1. Real world - it contains the next selection of itself as a model of the real world and the next selection to execute into materiality.
2. Societal - it accounts for all individuals, together.
3. Information - it accounts for the information base of existence.
4. System - it accounts for formal cooperation, integration, and unified communication (unified communications language).

3.2 One unifying information model

In order to operate safely in a material world, intelligence must be applied, and this may be done through a unified model that accounts for an environment:

1. Is the environment deterministic, then apply the actions of planning and search.
2. Is the environment stochastic, then apply MDPs (modeling of interaction to achieve a goal) and reinforcement learning (note, in the real world, the "reward", or reinforcement, is the fulfillment of a

real human requirement).

A. A Markov decision process (MDP) is a discrete time stochastic control process. The process is a mathematical framework for modeling decisioning in situations where outcomes are partly random (environmentally influenced without 99% certainty) and partly under the control of a decisioning [integration] agent. MDPs are useful for studying optimization problems solved via dynamic programming and reinforcement learning. MDPs are used in many disciplines, including robotics, automatic control, economics, and manufacturing. At each time step, the process is in some state *s*, and the decisioning agent [of control] may choose any action *a* that is available in state *s*. The process responds at the next time step by randomly moving into a new state *s'*, and giving the decision maker a corresponding reward $R_a(s,s')$. The probability that the process moves into its new state *s'* is influenced by the chosen action. Specifically, it is given by the state transition function $P_a(s,s')$. Thus, the next state s' depends on the current state *s* and the decision maker's action *a*. But given *s* and *a*, it is conditionally independent of all previous states and actions; in other words, the state transitions of an MDP satisfies the Markov property.

NOTE: *The real-world is a continuous, dynamic, and [partially] observable environment. An environment that is dynamic and partially observable has uncertainty (and therefore, novelty).*

3.3 Societal planning

Societal-level (Read: societal systems-level) planning is possible through a total systems approach to abundant and safe materialization of human fulfillment in a common and complex state-dynamic environment. Therein, each societal system may be accounted for in any given societal project:

1. A **Social** Systems-level project.
2. A **Decision** Systems-level project.
3. A **Lifestyle** Systems-level project.
4. A **Material** Systems-level project.

3.4 What is a real-world, community-information systems model?

In order to resolve real world problems (not just patchwork), then a base foundation of 'information' must be perceived of by consciousness? The perception [by consciousness] of everything as information is necessary if [real world] problems are to [f]actually conceived and resolved by the processing (linguistic sign) and calculation (mathematical sign) of information as 'data'. In other words, "we" perceive of everything as information, which may flow (by means of conscious intention) through a structure, and changing the entropy of the whole information system (towards greater complexity, more order, and thus, more potential [capability], or less complexity, less order, and thus, a lesser potential [capability]. It is here, from the information perspective, that knowledge becomes increasingly available the greater [a consciousness] is able to extend (Read: extensionality/exteriorization - the ability to extend one's view of self; beyond the self to encompass more of the self) its integrated "perception-conception" matrix. In romantic language, the prior sentence could be said as: "knowledge becomes increasingly available the more love one has".

NOTE: *Society is the individual's socio-technical project.*

Community is a single societal system (as in, socio-economic, socio-technical, socio-decisioning), because the user and the contributor are the same (are in cooperation, sharing access). The market-State is not a single system, because the employer (owner) and/or employee and/or consumer are not the same entity (are in competition, ownership of access). Whereas in community, there is recognition and unification of information, other types of society may neither recognize their information base nor seeks its unification.

All human-contextual complex systems exhibit closely interacting technical, decisional, and social components. Within the realm of 'technical' systems, emerging algorithmically unified (information-physical) systems, such as intelligent transportation and mechtronic (or automated robots) systems exhibit close interactions between components of, what was previously considered (now a historical context), a fundamentally different nature, namely, computational and physical components as separate. The informational systems view allows for a unification of the two previously separate perspectives, from which may arise, a second order ["cybernetic"] societal system:

1. A system that evaluates and integrates feedback from the environment,
2. after the execution of a decision,
3. which resolved a solution to a problem,
4. arising [in awareness] from an individual's interaction with an environment,
5. artificially limiting individual's fulfillment, and causatively [in an information system], producing a 'decision' space,
6. resolving through logic (which may be repeated as an algorithm) to an action in the material environment through execution by an individual or system with tools and resources,

7. that reconfigures the state-dynamic of the environment (Read: the habitat),
8. for greater [entropy] or lesser [entropy] states of individual's fulfillment.

What is visible from this description of society as information is that real things are multi–faceted, and that each level (or differentiation) needs to be considered separate and together.

3.4.1 What is a unified approach to societal state change?

A unified approach to societal state change is likely to be composed of:

1. A unified approach to **decisioning**, optimized as algorithmic decisioning.
2. A unified approach as **indication**, optimized through modeling and evaluation.
3. A unified approach as **servicing** (operations for) a user who is also a common[-unity, open source] designer.

3.4.2 How may a societal model be used as a navigation tool?

A.k.a., Organizing societal navigation.

It is possible to coherently organize society so that it navigates the planetary environment safely.

Societal navigation may be said to have two broad controlling principles:

1. **Safety in ensuring fulfillment** of basic and higher potential needs as the direction.
   A. Adapting direction, while following the precautionary principle.
2. **Coordination in organizing the fulfillment** of needs.
   A. Optimizing orientation, while following the efficiency (maximization) principle.

Sufficient for,

- *Next steps are adaptively optimized to the conditions necessary to generate the highest fulfillment of all.*

3.4.3 Science and engineering information sub-systems

Science and engineering have interrelating information flows:

1. Science involves understanding (theory), Engineering involves prediction and creation (invention, implementation and optimization, optimal solutions).
2. Science is why, engineering is how. Science is knowledge; engineering is the application of that knowledge to human purpose.
3. Science is truth-oriented, whereas engineering is goal-oriented.
   A. Science is the work of theory [visualization for understanding] and empirical research [testing, i.e. designing and conducting experiments].
   B. Engineering is goal-oriented is solving a specific set of problems with available tools and techniques.
4. A prediction is an expected future probability: science predictions are about the expected future probability that a model is true (i.e., accurate), and engineering predictions are about the expected future probability that a system will function as expected (i.e., accurately).
5. In terms of data, data science is science (discovering data structures), while data engineering is engineering (designing and creating data structures).
6. The discipline of decisioning (decision making) is decision science. Data science and data engineering both exist to support this discipline.

In a non-unified societal information system, these two disciplines are likely to evolve separately, and have separate cultures, think differently and speak differently, the social networks are different. Societal systems unification requires [the integration of] both.

3.4.4 Societal solution decisioning

In order to optimally sustain fulfillment among individuals in society, there are two societal-level, resource-access requirements:

1. Coordinated and controlled access to common heritage resources (information and material) through societal solution decisioning.
2. Coordinated and controlled design execution of a materializing habitat system through societal solution decisioning.

It is possible to develop and operate a service system with a high probability of fulfilling all [human] population requirements, optimally, when accounting for:

1. Common heritage survey of global resources (as in, area and object; position and reference/standard).
2. Common heritage information space for the open assembly and operation of the operational service system, including its information system.
3. Common heritage index of human need, fulfillment and optimal environmental, solutions.

3.4.5 The projected societal system's development

The development of a unified socio-technical system necessitates a unified, systems approach applying project coordination to a unified, societally engineered system.

Developing a [complex] societal systems is a highly interactive socio-technical process (group) involving many people that have to resolve decisions together (i.e., have to develop and take jointly consistent decisions). In this dynamic process, process organization and engineering must operate in conjunction. Projected systems necessitate the conception of an working information set. The project planning of a societal system necessarily involves the iterative integration of the planned sub-systems.

The principal societal systems include:

1. Societal information system.
2. Material habitat service system.
   A. Service development [engineering] systems.
   B. Service operational [support] systems.
   C. Asset/objects systems.

Decomposed by material operational structure type:

1. Function-based system (functional asset).
2. Non-functional-based (quality asset).
3. Product-based system (service asset).
4. Spatially-based (local [city] asset, global [HSS] asset)
5. Information and digitally-based (community or InterSystem interface).

Decisioning protocols (logic):

1. Execute protocols (decisioning to execution).
2. Control information flow (centralized, decentralized).

Science is used to discover data from an uncertain environment. Scientific sensors are used to discover data from a certain environment (because they are designed based upon an engineering pre-designed model). Information is used to resolve plan issued decisions. Knowledge is used to resolve engineering issued decisions (e.g., technical solution inquiry space; engineering systems control; the engineering problem).

1. The operations (a.k.a., service) problem (i.e., organizationally optimized operations problem).
2. The human operational-functional service [InterSystem] team (i.e., functional human contribution organization problem).
3. The [controlled] habitat service system (i.e., the engineered dis-/integration problem).
4. The materialized existence of a controlled object and relationship, of functional service and conditional quality (a matrix). As functional service and a physical object. As a condition of the services development and functional operation, constraining its operation. As a condition of the services functional and conditional operation, which is evaluated by functional and "performance" (or quality) conditions.

Projects to create that sustain systems composed of two types of primary process:

1. **Information coordination processes** (a.k.a., project life-cycle or project coordination/management processes) These project information processes form a closed loop: the planning processes provide a plan, that is realized by the executing processes, and variances from the baseline or requests for change lead to corrections in execution or changes in further plans. "Management" is the centralized creation, revision, and implementation of plans. The life-cycle is commonly composed of the following processes:
   A. Initiating.
   B. Planning.
   C. Executing.
   D. Controlling.
   E. Closing.
2. **Technical engineering processes** (asset-oriented processes) that specify and create the project product. A social project, such as this societal building project, is a collaborative activity, involving research, design, development, and implementation, that is appropriately planned.

Systems engineering directs project execution of the system's (product's) definition, development (sometimes through deployment and operations), monitor and control project work, and are responsible for closing out the project or phase's technical aspects.

3.4.6 Unified societal information system coordination

A.k.a., Socio-technical information integration; socio-technical unified creation/generation; socio-technical unified engineering.

In a unified societal information system, decisions are taken at:

1. **The project/information-coordination level -** The project level is solely composed of information.
2. **The scientific/technical-engineering level -** The engineering-development level is composed of digital information and material systems.
3. **The service-operations level -** The engineering-

operations level is composed of digital and material systems.

For societal creations there are multiple types of goals; there are:

1. Project goals (because, all societal-level solutions are seen as information projects (i.e., information "packets" or "sets" in a unified societal information "base" or "space". A project represents a complex (multi-part) project to be developed and resolved into a materialized solution
2. Life-cycle goals - projects are sub-composed of life-cycles.
3. Technical goals - life-cycles are sub-composed of technical goals, which become the engineering specifications selected, and then operational, in the societal system.

Approach tags for a unified societal information system's approach include:

1. Information approach (data approach).
2. Systems approach (holistic approach).
3. Project approach (coordination approach).
4. Engineering approach (generation approach).
5. Platform approach (interface approach).
6. Service approach (operations approach).
7. Module approach (task approach).

3.5 Unified economic planning (one economic plan)

NOTE: *One solved [for execution and operation] economic plan. Necessarily, a unified societal information system contains a unified economy.*

When viewing the societal system as an information system, then through technology and computerization there now exists the function/ability to do [economic] access allocation through computation by direct calculation and direct location. Herein, universal product barcodes with universal product codes and computerized stock-taking account for logistics (technological-transportation support).

INSIGHT: *Economics tells us that our prosperity depends on how efficiently we allocate resource to human needs and ecological regeneration.*

3.5.1 Socio-economic planning

Socio-economic planning refers to the planning of a/the society, and relates socio-economic problems to socio-economic solutions. Socio-economic planning means that the economic and social aspects are combined and planned for given what is known. An economic interaction is a social interaction, and hence, socio-economic planning is a component of societal decisioning (or pre-decisioning). Socio-economic planning is the deliberate control [of the flow and timing] of [economic] resources toward a life-cycle of needed services (and service objects or "goods"). In the market, economic planning also involves the "market" mechanism (which, is not present in community).

3.5.2 [Input-output] economic tables

NOTE: *A planned economy, in part, means that the society has an information system that communicates to its material users [in the economy] the number of people who will be doing different tasks each day; including all meta-data about those tasks.*

CLARIFICATION: *An input-output model uses a matrix representation of a nation's (or a region's) economy to predict the effect of changes in one industry (The make table) on others (the Use Table) and by consumers, government, and foreign suppliers on the economy.*

The input-output economic table is the first [basic] tool for doing any systems-based economic planning. Each service support system (and sub-system) in the economy is delineated on both dimensions of a graphing/calculation table (i.e., the system becomes a category in the column and the row dimensions). Generally, an input-output [economic] table contains the following sub-units:

- *Columns [a dimension]* are categorized by what [HSS] system or market-State industry (e.g., mining forestry fishing agriculture -- where the raw resources come from)
- *Rows [a dimension]* are categorized by what system or industry (e.g., mining forestry fishing agriculture -- where the raw resources come from)
- *Cells [a bi-dimensional synthesis]* say how much of one system or industry, or other category if more than multi-dimensional, goes into the other system, industry, or category. Here, the cells list (display) the relative flows between the different systems or industries.

[Service support system] Input-output tables are required in order to 'project' through the outcomes. Here, the concept, 'project' has a double meaning. The concept, 'project' means:

1. To have an information system that supports the informational requirements of a [configuration of the] human habitat service system -- The outcomes desired have a [unified] project-based information system within which calculation is possible.
2. To have the ability to predict the outcomes -- The outcomes require projection (i.e., modeling,

visualization and simulation) of systems, and therefore, transparency (and openness) in concern to the inputs and outputs of those systems.

These tables can be refined:

1. Down (Read: dis-aggregated) to the individual products, and then to the individual modules (if separate).
2. Up (Read: aggregated) to the supra-service system level (i.e., life support, decision support, technology support, facility support).

When something is changed (or integrated/developed; e.g., a new product), these tables are used so that in a "what if" scenario what is the implications for other systems or industries due to a change in what (i.e., a target or given) system or industry.

Input-output analysis is the solution to simultaneous linear equations. In input-output analysis, there are many equations, but each equation has a highly limited number variables.

In mathematics, a connected graph is a system of:

1. Circles (nodes), with
2. Arcs or lines joining the circles.

Relations between the nodes of the graph and the arcs between them is an:

- N log n relationship

Any economy can be calculated and visualized as this [N log n relationship] graph, with each system or industry being a node in the graph and an interconnection between two systems or industries being an arc. Fundamentally, if the economy is unified (i.e., it's not split into two or more separate economies), then there is at least an N log n relationship within the graph (the Erdos diem, Jacobian solution).

3.5.2.1 Efficiency

Start with the basis that society is an information based system, then efficiency, necessarily, becomes a core value. Efficiency is important to computation based systems. If a computation system is inefficient, then is it wasting resources. Systems that remain inefficient become extinct.

> **NOTE:** *Digital information is constantly copy able.*

4 What is the proposed method of integration for work?

The integrated project-engineering approach method involves the measurement of all work in time with resource:

1. **Project** (effectiveness, efficiency).
   A. Measurable goal using identifiable properties.
   B. Measurable criteria for goal using defined parameters.
2. **System** (effectiveness, efficiency).
   A. Measurable goal using identifiable properties.
   B. Measurable criteria for goal using defined parameters.

Systems engineering involves the designed formation of a system through a project-based structure. Ideas are developed into assets (systems) through projects. Projects are information-level organizations with knowledge areas and processes for asset creation. The existence of a project means the presence of presence of engineering and the vision of a resulting asset (state or condition).

Information can be observed (sensed) and processed (computed). In concern to projects, the idea that information [about production] can be observed, leads to the idea of a common "body" of knowledge areas [about a project-type, production information]. Therein, the idea that information can be processed, leads to the project control and coordination processes (organized by process groups). Note that with each iterative development, both at the project-level and product-level, there is the potential of adding to our knowledge (value). We are doing iterative development because we want to learn with each iteration. The development of a functionally optimized, adaptive asset [known as society, highest level asset] is an iterative process.

> **INSIGHT:** *This is a unified social approach.*

There is a unified system [development and operations] view, within which there is a:

1. Project-level view.
2. Engineering-level view.

Systems engineering and project management are two critical aspects in the success of complex real-world projects. A project to develop a complex real world system necessitates both systems engineering and project planning. Wherein,

1. **Projects** define and decide how resources are cycled through a common materiality.
2. **Engineering** defines and determines how resources are configured in the common material environment.

3. **Operations** (Read: continued engineering projects) sustain and executes ongoing project-service configurations.
4. **Evaluations** (Read: Feedback) define learning.

Coupling design and "management" through a decisional model associating the two process categories:

1. The project-level information process.
2. The engineering-level materialization process.

Both project management and systems engineering necessitate a life-cycle decisioning (or "gating") structure. The key principle of development is that it is goal-driven. In a projects should be planned based on explicitly set goals.

To develop [complex and adaptive societal] systems efficiently and effectively, it is essential to align practices in systems engineering and in project "management". This issue of systems engineering and project "management" integration is at the core of all societal concerns (e.g., economic and industrial).

The unification of the processes of engineering (systems engineer), planning (project management), and decisioning (decision management) has been given a number of names, including, but not limited to: collaborative engineering; unified engineering; unified planning; systems engineering management; project integration engineering; and integrated systems engineering.

In a non-unified approach, one without a recognition of the underlying information system, the engineering of a system, and the management of that as a project will likely be carried out separately (as two separate disciplines). Depending on the environment and organization, the two disciplines can be disjoint, partially intersect, or one can be seen as a subset of the other. However, integrating the engineering and project components of system development (i.e., in order to carry out and complete engineering projects) is essential for a unified approach (i.e., an information-based approach). Here, the term 'unified' is a reference to a whole, integrating information system. Both systems engineering and project coordination ("management") are necessary for engineering (or otherwise, developing) a real world system. This represents the integration of systems engineering and project management.

> **HISTORICAL NOTE:** *Traditionally, systems engineering and project management have been practiced separately (i.e., they were considered two separate sets of knowledge and processes, instead of two views into one set).*

In the market-State, generally, competing entities usually attend to systems engineering and project management processes as separated roles (or processes), and do not consider connections between them. Indeed, for many years, the labor roles of systems engineers and project managers have thought of their work as separate, focusing more on their own domains than on the whole project as a unified information system. This compartmentalization of processes has led to significant inefficiencies is system design (and in society as a whole). In the economic labor market, the economic roles of systems engineer and project manager are in some degree of economic competition between one another. Further, in the market, research into the integration of these two roles is motivated by the prospect of improving the business' (or State actor's) competitiveness in the development of a product or service; it is not motivated by the prospect of improving [global] human fulfillment and environmental safety.

Engineering has a social function, and it is the presence of a social function (to lesser and greater degrees of quality, that makes engineering possible. An important point in looking at the social function of engineering is how society makes engineering possible. A complex feedback situation emerges. Societal organizations extend the power and reach of society and the individual. Society, in turn, through its organizations and demands, makes possible the development of complex habitat service systems and stimulates their constant technical evolution and diffusion. Today, to talk about the impact of engineering on society is meaningless without also talking about the impact of society on engineering, and how it shapes the role of engineering. The complexity of the interactions between society and engineering is at the root of unrealistic expectations about engineering, as social entities are often inadequately organized to develop and use engineering effectively. It is also at the root of the frustration of engineers unable to bring their capabilities to bear on the solution of social problems or the effective organization of the engineering enterprise.

Simplistically, the project-engineering of a society involves:

1. Socio-technical issue input.
2. Socially acceptable solution.
3. Technically acceptable social solution.
4. Projects organize the temporally positional information.
5. Engineering organizes the compositionally positional information.

4.1 A unified systems approach

A unified society is highly likely to have two core societal systems applications ("disciplines"). In other words, the two main domains of comprehension involved in a unified societal systems approach are:

1. Systems engineering (a.k.a., engineering development and operations, engineering coordination).
2. Project coordination (a.k.a., project management).

These two disciplines can be more generally categorized as the:

1. Engineering [design and development] approach.
2. Project [information] approach.

There are two [information] domains when engineering a complex system into existence:

1. Systems engineering (technical processes primarily), and
2. Project planning (coordination processes primarily).

An integrated view accounts for both systems engineering and project management. In this sense, project management identifies and coordinates need fulfillment, and engineering is the systematic study and resolution of socio-technical problems. In community, engineering is not handicapped as its effectiveness is some societal configurations (e.g., the market).

The outputs of an integrated view of a solution are:

1. Project coordination involves the project domain.
2. Systems engineering involves the system-service-product domain.

The flows of an integrated view of a solution are:

1. In the production process, the flow is materials and the objective is to make a system from materials.
2. In the engineering process, the flow is technical documents or technical information, and the objective is to provide the necessary technical specifications for the product and production of it.
3. The control process coordinates these other two processes. For the control process, the flow is information.

During the evolving stages of a project, users require (at least) the ability to:

1. Observe (perception).
2. Coordinate (organization).
3. Control (decision).

Together, systems engineering and project coordination (management) decompose a project into tasks and processes, planning tasks and processes with an overall project plan, and monitoring all tasks and processes until the validation of the project is complete.

Therein, effective action toward the users desired resolution necessitates the following information processes:

1. Coordination.
2. Decisioning.
3. Tracking.
4. Analysis.
5. Memory.
6. Feedback.
7. Correction.

A project is an organization designed to fulfill an objective, created with this purpose, and dissolved after its conclusion. A project can be defined as an organization with a clear and well-defined objective; it is working through a planned and coordinated approach with possibly pre-defined parameters of time, cost, quality and resources available.

The aim of project coordination (project management) is first to define the project mission and organization, then to determine the budget and plan a schedule, and then to ensure operational control of said project through an assessment of performance by analyzing possible deviations relative to the initial schedule, and to implement corrective actions or new preventative actions if necessary to mitigate risks. Its role also consists of organizing and monitoring systems engineering processes.

Having in consideration that a project has a well-defined beginning and an end, it can be associate to a life cycle, generally designate project life cycle (PLC). The PLC establishes the work that must be done in each phase of the project and the number of resources needed in its realization. The PLC phases are context specific for that reason it may defer from one organization to another.

Unified project-engineering involves:

1. **Projects that:** Projects are concerned with the overall, [Social organizational context of an environmental change.
   A. Lifecycle of projects: initiation, execution, closing.
2. **Develop systems:** System development as a lifecycle is concerned with the [technical] work/ service systems that are to undergoing the change:
   A. Lifecycle of system (loop) - analysis, requirements, design, development, implementation, feedback.

A simplified project-engineering approach may be:

1. Recognize situation (articulate issue, problem, or need)
2. Identify societal requirements (understand system)
3. Identify user requirements (understand user)
4. Analyze gap (understand user demand)
5. Create solution description (design system)
6. Propose viable solution specification (propose system)
7. Select optimal solution specification (determine system, system construction decision)
8. Build new system state to solution specification (produce system)

9. Verify and validate system state (inspect and test system)
10. Cycle (de-integrate and re-cycle as appropriate)

4.2 System life-cycle coordination

The integration of project and engineering information sets requires coordination. The primary coordination systems required to coordinate the development of a unified socio-technical system are:

1. Project planning process group (processes).
   A. Project coordination (parallel inquiry process) .
2. System planning process group (processes).
   A. Engineering coordination (solution/technical inquiry).

Coordinating the development of a system into the life-cycle of a [habitat] service, necessitates the major activities of:

1. Systems [life-cycle] engineering.
   A. The system which is being brought into existence has a lifecycle.
2. Project [life-cycle] coordination/management.
   A. The project to bring the system into existence has a lifecycle.
3. Service/product [life-cycle] operations/ management.
   A. The system in its operation has a lifecycle.

4.3 Service coupling

In order to complete human requirements together as a global population using common resources, a social organization must be coupled to a decisioning organization as a service organization (note that the following are all engineering views, because engineering does the work):

1. **Simple service view** (e.g., concept of operation)
2. **Document concept of service** (e.g., model of service)
3. **Development of physical service** (e.g., designed service system)
4. **Manufacture, fabricate, assemble service actual service** (e.g., produced service system)
5. **In-service operations** (e.g., the operating habitat service system)
6. **Iterate service operation** (e.g., the strategic plan)

In the context of a service [system] operation, the integration of systems engineering and project management become two coupled mechanisms system, those of design (and development) tools and project management (coordination) tools into an effectively operated service. These two mechanisms are used, in part, to propagate the operational-organizational decisions necessary to sustain a service [system] as a solution (to societal system's organization, for example):

1. **Information coupling (information interfacing):** each sub-project is directed by requirements distributed between the two architectures (design & project), leading in some cases to the definition of common indicators. The information flow between these two points of view is based on the definition of these indicators and on their "management". The most straightforward example is resource presence:
   A. From the design point of view, will the resource be available? Are these materials available, or are others optimal [in our selection of a solution]?
      1. To what is information optimally flowing?
   B. From the project viewpoint, can a resources be made available? Are those materials available, or other solutions optimal?
      1. How could information flow optimal?
2. **Structural coupling (real-world, physical interfacing):** each sub-project is broken down into a design architecture and a project-system [management] architecture. These two architectures are logically connected to enable an exchange of information that facilitates the optimal construction of real [world] interfaces. The most straightforward example is, the function of a set of given buildings at a given location (where, the buildings and land are the interface; GIS data):
   A. From the design point of view, the data set is a design specification modeling the function of the buildings at the given location.
      1. What is to be built?
   B. From a project viewpoint, the data set is the construction (or re-construction) of the set of given buildings at a given location.
      1. How is it to be built?

4.4 Integrating project management and systems engineering

CLARIFICATION: *Presently, the integration of project management and systems engineering into a unified approach has no directly attributable name.*

The integration of systems engineering (SE) and project management (PM) has only been considered in the beginning of 21th Century. The point is that, depending on the environment and organization, the two disciplines can be separate, partially intersecting, or one can be seen as a subset of the other. Previously, there were often treated as separate, using different persons, different tools, and different processes. For many years, a cultural barrier has been growing between practitioners of SE and of PM leading them to consider their respective work as separate rather than integrated towards a common objective, that of satisfying the end user. As a result, work is often more costly, takes more time to be completed and provides a suboptimal solution.

A cooperative society requires a tool wherein the high-level process groups of project management and systems engineering are optimally integrated. This is accomplished by:

1. Integrating standards from both domains into a unified domain.
2. Formalizing the definition of integration.
3. Developing integrated assessments.
4. Sharing responsibility for risk, quality, lifecycle planning, etc.

Systems engineering and project management are two critical aspects in the success of system development and system operating projects.

1. Project management is organizational decision processes.
2. Engineering management is solution decision processes.

The integration of project coordination and systems engineering necessitates types of requirements:

1. Decision requirements (organizational).
2. System requirements (service).

Systems engineering is focused on product requirements and should be empowered to handle them autonomously, involving the project manager when a technical requirement has project requirement impacts.

Project management and systems engineering are complementary functions, with great benefit from leveraging each other's strengths in a team environment.

Project manager manages the project life cycle, the systems engineer manages the technical baseline of the product under development. The project manager and systems engineer share requirements management responsibility, and by working closely together they keep the project on track.

A development system requires a repeatable controlled process - a fully integrated project cycle that addresses both the organization (PM) and technical aspects (SE) as an integrated process:

1. A project organization - project coordination (a.k.a., project management) initiates, plans, and then monitors and controls the execution of a technical solution. At any point in the lifecycle, project coordination may close the project or put the project on hold. Projects are closed when:
    A. They are completed.
    B. A decision is taken based on organizational inputs that determine the risk outweighs the expected benefits (safety protocol).
    C. They are terminated by the user.
    D. Phase closure - to define a more prominent gated progression.
2. A technical solution - the work required to realize the result, and the specification, which is acted upon.

4.5 *Systems reference standards*

In both the project management and systems engineering disciplines there exist a number of globally recognized and utilized [reference] standards for brining into existence (i.e., working) an environmental change in a systematic manner.

> **NOTE:** *Not a single one of these standards (or guides) contemplates an integration or sufficient cooperation between systems engineering and project management, despite the fact that engineers and managers (a.k.a., coordinators) have to cooperate closely throughout all stages of project development.*

4.5.1 Systems engineering reference standards

System engineering has the following recognized standards *(systems engineering reference standards)*:

- **ANSI/EI-632** (ANSI and EIA 1998)
- **IEEE-1220** (IEEE 2005)
- **ISO/IEC-15288** (ISO and IEC 2008)
- **International Council on Systems Engineering** (INCOSE)
- **Systems Engineering Handbook** (SEHBK) (Haskins 2010)
- **NASA Systems Engineering Handbook** (NASA 1992)
- **Systems Engineering, Coping with Complexity** (Arnold et al 1998, 152-168)
- **Systems engineering management plan** (SEMP)

The most important systems engineering standards are:

- **ANSI/EIA 632** - Processes for Engineering a System
- **ISO/IEC 15288** - System Life Cycle Processes
- **IEEE 1220** - Standard for Application and Management of the Systems Engineering Process
- **INCOSE Systems Engineering (SE) HandBook**
- **SEBoK** - Guide to the Systems Engineering Body of Knowledge (SEBoK)

4.5.2 Project management reference standards

Project management has the following recognized standards *(project management standards)*:

- **PMBoK 2018** (Project Management Institute, PMI) - A Guide to the Project Management Body of Knowledge
  - *Notice how the term, "management", is in both the title of the Institute (PMI) and in the title of the standard (PMBoK).
- **ISO 21500** - Guidance on Project Management

Additional project-related reference standards:

- **ISO 9001:AS9100 Quality Management Systems** – Requirements for Aviation, Space and Defense Organizations
  - **ISO 9001:2015 -** International standard for a quality management system ("QMS")
- **ISO 55000:2014, ISO 55001:2014, ISO 55002:2014** - Asset management
- **ISO 8000** - Data management
- **ISO 16404, ISO 10795, ISO 14300-1, ISO 21351** - Requirements Management space systems - program management
- **BS 1192:2007 + A2:2016** - Collaborative production

4.5.3 Building information management reference standards

Building information management (BIM) standards:

- **ISO 19650**, Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) – Information management using building information modelling
- **Level of development specification** v2013, v2015, v2016
- **BSI PAS 1192-2:2013** - delivery phase
- **BSI PAS 1192-3-2014** - information Operations & Maintenance (M&O) phase
- **PAS 1192:2015** - security phase
- **PAS 1192-6** - Health and safety
- **BIM Guide**

4.5.4 Architecture reference standards

The American Institute of Architects (AIA) has produced an integrated guide for architects:

- *Integrated Project Deliver: A Guide*. Ver. 1. (2007). The American Institute of Architects. [info.aia.org]

4.5.5 Integrated reference standards

Integrated standards are those that integrate both project management and systems engineering. The most recent, important integrated reference standard is:

- **ISO/IEC 29110** - System and Software Life Cycles

4.6 Reference standards re-alignment

The best current reference standards candidates for integration (as the alignment of processes):

- ISO/IEC 15288 standard would represent best candidate to alignment with PM standards.
- PMBoK 2018 standard would represent best candidate to alignment with SE standards.

The five processes of the ISO/IEC 15288 must be executed one after the other (initiating > planning > executing > monitoring and controlling > closing, in series in time). For ISO/IEC 15288, the four process group can be executed concurrently, or not (Agreement Processes, Technical Processes, Project Processes, and Organizational project enabling Processes, in series or parallel). For processes, some of them can run simultaneously, while the others must be executed in a chronological order.

4.6.1 Integrated reference standards data structuring

ISO/IEC 15288 data structure:

1. Process groups.
2. Processes.
   A. Purpose.
   B. Outcomes.
   C. Tasks & Activities.

PMBoK 2016 data structure:

1. Knowledge areas.
2. Process groups.
   A. Inputs.
   B. Tools & Techniques (the processes themselves).
   C. Outcomes.

4.6.2 Standards Software integration

Integrated software solutions (for PM and MS) include, but are not limited to:

1. In project management: Primavera, MS Project, etc.
2. In product life-cycle management: Windchill, Team Center, ENOVIA, BIM software with Autodesk Fusion and Revit, etc.

NOTE: *Generally, software solutions are* *traceable to accepted reference standards. The starting point of software is a reference standard, a specification.*

5 What is the proposed method for life-cycling project-engineered solutions?

Every solution is an integration of project coordination and systems engineering through life-cycle (of process groups). Although there are many variations of project composition, the following is a simple and general composition of the interrelated, synchronously integrated project-engineering phases for a 'solution':

1. Coordinate (project coordination)
   A. Have informed information system.
   B. Establish decision processes.
   C. Decide who.
   D. Determine resource allocation (resource access).
   E. Define quality evaluation standards.
   F. Document processes.
   G. Develop evaluation plan, framework or policy.
   H. Review evaluation (do meta-evaluation).
   I. Develop evaluation capacity.
2. Define (project definition)
   A. Develop initial description.
   B. Develop project theory/logic model.
   C. Identify potential unintended results.
3. Frame (solution framing)
   A. Identify primary intended users.
   B. Decide purpose.
   C. Specify the key evaluation questions.
   D. Determine what results ('success') looks like.
4. Describe (solution description)
   A. Sample.
   B. Use measures, indicators, or metrics.
   C. Collect and/or retrieve data.
   D. Coordinate data usage.
   E. Combine qualitative and quantitative data.
   F. Analyse data.
   G. Visualize data.
5. Understand cause (problem-solution evaluation)
   A. Check the results support causal attribution.
   B. Compare results to the counter-factual.
   C. Investigate possible alternative explanations.
6. Synthesize (design solution)
   A. Synthesise data from a single evaluation.
   B. Synthesize data across evaluations.
   C. Generalize findings.
7. Implement (apply solution)
   A. Execute an action (or multiple and/or dynamic actions) to bring the solution into existence.
8. Report and Support Use (of solution)
   A. Identify reporting requirements.
   B. Develop reporting media.
   C. Ensure accessibility.

D. Develop recommendations.
E. Support use.

5.1 Simplified project systems engineering

The group of functional relationships that form a highly simplified view of project systems engineering are four:

1. Coordination (share plan).
2. Design (concept model).
3. Build (spatial construct).
4. Operate (real-world system).

Note that any re-cycling system is itself an operated system; in other words, there is list #5 for re-cycling, because all systems are either operational or under design to become operational (a re-cycling system is either under design, or operational currently).

5.1.1 Historical note

Neither the project nor engineering approaches represent a new way of developing a system, or providing and operating a service. Before the principles of mass production were developed, all complex production and operation was carried out as projects to produce engineered systems. Craftsman (early term for an engineer) have always made products based on the information, materials, and time, available, and adapted to the requirements of a user.

Therein, project coordination (or more commonly in the market, project "management") has been practiced for thousands of years, and can be dated back at least as far as the Great Pyramid of Giza and Gobekli Tepe. The idea of project "management" is related to early civil engineering projects. Until 1900, civil engineering projects were generally "managed" (coordinated) by the architect(s), engineer(s), and master builders, themselves. It was in the 1950s that organizations started to systematically apply documented project coordination ("management") tools and techniques to complex engineering projects.

In the professional market for labor, 'project management' became recognized as a distinct discipline arising from the labor market's management domain, with material creation (design and development) occurring through the labor market's 'engineering' domain. In 1969, the Project Management Institute (PMI) was established in the USA, and then globally, to solidify and refine the 'project management' [economic] profession. In the professional [labor] market, there are now 'project managers'.

In 1996, the PMI first published "A Guide to the Project Management Body of Knowledge" (PMBoK), which described project management practices that were common to "most projects, most of the time". In 2012, the International Standards Organization (ISO) also realized the importance of project management and published a project management standard ISO 21500. Today, there are many similar and related disciplines of project management, such as program management, project lifecycle management, product lifecycle management, and others.

NOTE: *More technically, beyond the labor-market, a "project manager" is simply a type of information process unit, a unit that coordinates and controls the flow of a high-level project related information.*

5.1.2 A project [development] integration view of the projected system's life-cycle

The project life-cycle provides a framework (of processes) for resolving coordination problems to the production of complex work.

A project life-cycle necessitates:

1. **Project initiation** - In project initiation, the goals for the project need to be consistent (in alignment) with organizational goals. Organizational models, such as the societies decision system help with this.
2. **Project execution and controlling** - The executing and controlling steps of a project is where the 'system development life cycle' exists. This is where/when the analysis of existing systems and processes takes place,and when new ones are developed and implemented.
   A. One way to view the [system] development/ operations [life]cycle is as one executable step in project coordination.
3. **Project planning** - Planning occurs in between (in parallel, often) initiation and execution. This is where the goals of the project (Read: the reasons for doing the project) into actionable steps. A variety of documents are developed during this phase. These documents are used to coordinate ("manage") the project. The three core [project] planning > plan documents (Read: recorded and transparent, living, information sets) are:
   A. The [project] charter.
   B. The work breakdown structure.
   C. The [project] schedule.

Systems are engineered into the coordinated operation of a larger and pre-functioning system; they are integrated:

1. **Integration** - Once characterised and accepted as suitable, the products/services undergo adaptation and integration into the required asset-service system. The maturity of this integration is measured through Integration Readiness Levels (similar to Technological Readiness Levels, but

with operational evaluation information). Any new development elements are integrated with the adapted elements to form the new systems.

2. **Transition into service** - The transition into service utilises project views of materiality (architecture) to schedule the requisite elements of products and services for deployment and use (Read: access). At this stage the asset-service systems are used in their intended environment and undergo validation against the capability requirements (of the architecture).
3. **In service support** - Throughout the sustainment/operating period, asset and service measures are captured/observed and analysed against the indicator-metrics selected to correct for alignment errors based on alignment requirements, which form the basis of process improvement [in a given information system]. Progressively, the capability [requirements] architecture, system and service models are validated, or not validated. As changes are undertaken (to correct for validation) the architecture, models, and operative services are updated. Any potential change can be modelled prior to commitment to change (i.e., solution) to ensure the changes will contribute to system's objective/requirements.
4. **Dis-integration** - the end-of life-cycle removal (and possible modified replacement) of an asset-service.

5.1.2.1 A system [development] life-cycle view of the integration

The systems development life-cycle provides a framework (of processes) for system creation and integration, for technical (solution inquiry) change in the environment given a user with requirements [for which a project has been composed].

The system development life-cycle includes:

1. **Analysis (of situation)** - The sdlc starts with an analysis of the situation. What can be better? What is going wrong?
2. **Requirements (for systems change)** - describe the solution to the degree that the delivery can be compared in alignment with the [solutions] description. What is required for fulfillment? What are the goals, specifications, and must haves in order to resolve the [systems] change.
3. **Design** - After the situation is fully understood and the requirements for and solution, the you start planning out that solution. What will the future situation look like? What do the technologies look like that support this future situation? Design out what the technologies look like, what they should do, and their expected context(s)?
4. **Development** - Create, build, and prototype and test the technologies.
5. **Implementation** - integration of the technology. Train InterSystem and Community people on them, and InterSystem Team operate the systems as services for our human community.
6. **Analysis of implemented situation** - After implementation, evaluate to see what is working as expected (alignment with requirements) and desired by users (fulfillment as expected)? What is working and what is not? Then, start the process life-cycle all over again with analysis.

Alternatively, the system development life-cycle could be viewed as:

1. System definition.
   A. Collection of user needs.
   B. Translation of user needs into technical requirements.
   C. Initial design concept.
2. System development.
   A. Specifications for functional level.
3. Process development.
   A. Design and prototype manufacturing and assembly processes.
4. Process quality control.
   A. Process parameters (specification for performance/quality level) are determined and evaluated.

When analysing a situation it is important to analyses it in the context of the goals for the organization, the user and the service that meets their needs (in a business goal context, for example, profit, reduce costs, improve customer value)...if these conditions aren't going to happen, are you sure the change should be taken? The usage of a social information model facilitates the analysis of and identifies the requirements for parallel societal decisioning (i.e., the societal decision value alignment inquiry processes).

The system development process includes:

- A development process, where the main activities are represented going from requirement definition to maintenance of the finished product; a life cycle based on evolving prototypes into a fully integrated system; and the methodology itself (why the method was logically selected).

Most generally, the development process is:

1. **Direction** - put together a specification of the objective.
2. **Conceptualization** - put together a specification of the system. Conceptualization involves the

organizing and structuring of acquired knowledge.
3. **Implementation** - implement the concept model to create and/or operate the system.
4. **Evaluate** - execute an evaluation (and "judgement") by doing a technical analysis on the process and result, and correction any mis-alignments with objectives and requirements (system so that all information in all phases is more coherent and/or useful).

The commonly accountable elements of the design phase are (i.e., what is the "design for"?):

1. **Function** - the "means" by which (how) the system operates for user fulfillment. Why and how does the system operate? How is that specific operation determined and measured (or observed)?
2. **Interface** - the "means" by which (how) which two systems interact (Read: share information).
   A. Because an interface's principal purpose of existence is to represent usability between an object and a user, the principal interface design [operational-conditional] principle is: usability. The interface is being designed to literally 'interface' with another system, and so, it must do this effectively for both systems. Humans and other necessary systems can interact with the target system (e.g., a societal system) in a way that allows them to achieve their purposes in an efficient and effective manner, together.
3. **Performance** - the evaluated the quality of the method (means) by which the function occurs (how in alignment with expectations is the function):
   A. Information is shared between systems (per requirements).
   B. The function operates for user fulfillment (per requirements).

5.1.2.2 The planning [development] life-cycle view of the integration

The planning [development] life-cycle view of the integration process involves:

1. **Assess** the articulations alignment (recognition and effectiveness) of the inquiry.
   A. If there are gaps, then change social value set or evolve self value set to remain in coherence so alignment of articulations can be assessed.
   B. If there are [now] no gaps (otherwise, repeat prior step), build the vision (as a model through to simulation as integration over time), while maintaining a set of goal-oriented (need) conditional statements, that will be translated into an extant system.
2. **Simulate** the vision by modeling in real-time to resolves more complexly, completely, and commonly.
3. **Test and evaluate** a prototypical operation of the requirements of the vision.
4. **Planning** - What possible solution fulfills the technical engineering and constraining organizational requirements, together as a system, most optimally? That solution is the selected plan. The planning process is a continuous, dynamic process -- the "creation" of the plan is a continuous activity group.
   A. Plan the project - (information-) oriented components of Information-Project Engineering Development - integration of the following units into a directionally coordinated human societal fulfillment interface (plan) including, but not limited to the following major sub-component systems defined by their "engineering" requirement: requirements coordination (a.k.a., requirements management), schedule coordination (a.k.a., schedule management), resource coordination, [societal] quality coordination, risk [and, cost] coordination, communications [and interface] coordination, computation and logic coordination,
   B. Plan the lifecycle-oriented components of Engineering Operations of the service and/or service asset.
5. **Execute** by resolving the decision to a section, which sends a signal to a controller, causing an execution of action involving a modification to the state-dynamic of the material (habitat) service system, which will either be acted upon uniquely by an InterSystem team(s), or it will be integrated into an active service lifecycle as an asset by an InterSystem team, or it will be removed from active materialization by the core Effectiveness Inquiry Process.
   A. Engineering developments of service systems by applying information processing (Read: lifecycle planning). Engineering controls design process.
      1. As part of the Development InterSystem Team, those individuals who contribute with accountability toward the sustainment of the habitat service system's operation.
      2. As part of the information system, control decisioning (constrain solution to value-alignment), and thus, control design process.
      3. Define technical operational baselines.
      4. Coordinate design solution is the result of a controlled design process and the development of baselines.
      5. Configuration [state of HSS] levels through this the entire design effort can be coordinated via

decision points ("audit") informed by:
- i. Concept [configuration] level study - generate system concept description; what should be done, behave, exist?
  1. System [configuration] level study - describe requirements for integration into service (performance requirements); how will the system perform under different conditions?
  2. Component study [sub-component configuration] of subsystem level - performance requirements - detailed description of characteristics required for production; what are the components of the system that enable the fulfilling of performance requirements?
- ii. Functional baselines:
  1. Allocated baselines (preliminary design definition).
  2. Asset baselines (detailed design definition, product baseline, and material asset realization).

B. Engineering integrations of service systems by accounting (surveying baselines) - sub-component of both systems, simultaneously, indicator-metric-evaluator interface.

C. Engineering operations of service systems by applying systems by applying apply service (operations) knowledge areas (including, principles) and processes.
1. As part of the Operations InterSystem Team, those individuals who contribute in accountability toward the sustainment of the habitat service system's operation.

5.1.2.3 The system-conception engineering life-cycle view of the integration

The system-conception engineering life-cycle view of project integration involves the following:

> **NEED ANALYSIS QUESTIONS:** *Is there a valid need for a new system? Does there exist a practical approach to satisfy the user need for a new system (is it feasible)?*

1. **Concept** DEVELOPMENT **phase**

A. Need analysis - a valid need has the form of:
1. [Human] Needs analysis.
2. [Social organizational] needs analysis.
3. Technical needs analysis .
   - i. Inputs:
     1. Operational deficiencies (gaps in service).
     2. Technological opportunities (knowledge).
   - ii. Processes:
     1. System studies.
     2. Technology assessment (technological readiness levels, or new technology).
     3. Methodological assessment (model readiness level).
     4. Operational analysis (is it feasible to operate).
   - iii. Outputs - the output of this is the first (preliminary) iteration of the system's design itself, which is a basic (high level) concept model.
     1. System operation effectiveness.
     2. System capabilities.

B. **Concept** EXPLORATION **phase**
1. Concept exploration questions: What are the principal characteristics of the systems concept that can provide the best design between capability, life of system, resource occupation of system (and in the market, cost).
2. Concept exploration tools: process methods, decision support systems, expert analysis.
   - i. Inputs:
     1. System operation effectiveness.
     2. System capabilities.
   - ii. Processes:
     1. Requirements analysis .
     2. Feasibility tests (alternative search) - what are the other alternatives available to the system for fulfilling the need(s).
   - iii. Output:
     1. System performance requirements.
     2. System concepts.

C. **Concept** EXPLORATION **phase**
1. Concept exploration questions: what are the performance requirements of the new system so users needs can be satisfied? Is there at least one feasible approach to achieve the desired performance at an affordable/acceptable resource usage (and in the market, price)?
2. Concept exploration tools: process methods, decision support systems, expert analysis.
   - i. Inputs:
     1. System performance requirements.
     2. System concepts.
   - ii. Processes:
     1. Selection from alternatives ("trade-off" studies).
     2. Architecture (system [architecture] logic; not engineering architecture).
   - iii. Outputs:
     1. System functional specifications - a description of what the system must do and how well?

2. System concept definition (a.k.a., system definition).

Scholarly references (cited in document)

- Tri, D.Q., Thai, T.H. (2018). *Biological environmental survey in Cat Ba Island*. Biodiversity International Journal, 2(2). https://doi.org/10.15406/bij.2018.02.00054 | https://www.semanticscholar.org/paper/Biological-environmental-survey-in-Cat-Ba-Island-Tri-Thai/9b7b39d4e323502118ee36ef988c0b724cbea36c

Book references (non-cited)

- Mobus, G. (2017). *A Framework for Understanding and Achieving Sustainability of Complex Systems*. Systems Research and Behavioral Science, 34(5), 544–552. doi:10.1002/sres.2482
- Skyttner, L. (2005). *General Systems Theory: Problems, Perspectives, Practice*. World Scientific Publishing Co. Pte. Ltd. 2nd Edition. ISBN-13: 978-9812564672

Online references (non-cited)

- *Essence: Kernel and Language for software engineering methods*. (2015). Ver. 1.1. Object Management Group. formal/2015-12-03. https://www.omg.org/spec/Essence/1.1 | https://semat.org/documents/20181/57862/formal-15-12-03.pdf/5d2e878b-73aa-40a7-9c6a-6d365f4f1f8c
- Hevner, A., Brocke, J.V., Maedche, A. (2019). *Roles of digital innovation in design science research*. Business & Information Systems Engineering. February 2019, Volume 61, Issue 1, pp 3–8. https://link.springer.com/article/10.1007/s12599-018-0571-z
- Jacobson, I., NG, Pan-Wei, McMahon, P.E., et al. (2012). *The Essence of Software Engineering: The SEMAT Kernel*. ACMQueue. Vol. 10, Issue 10. September/October 2019. https://queue.acm.org/detail.cfm?id=2389616
- Jacobson, I., Ng, Pan-Wei, Mchanon, P.E., et al. (2013). *The Essence of Software Engineering: Applying the SEMAT Kernel*. Addison-Wesley Professional.
- Mobus, G. P. (2016). *Systems Science and Biophysical/Ecological Economic*. Slides presentation access on 18 June 2019. In Principles of Systems Science by Springer. http://www.isecoeco.org/wp-content/uploads/2016/09/Mobus-George.pdf
- Zapata, C., Jacobson, I. (2014). *A first course in software engineering methods and theory*. Dyna rev.fac.nac.minas vol.81 no.183 Medellín Jan./Feb. 2014. https://doi.org/10.15446/dyna.v81n183.42293 | http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532014000100026

Approach: Projecting

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: project approach, project management, project coordination, project planning, project operations, projection,

Abstract

This article delves into the intricacies of a project-based approach to the coordination and realization of complex socio-technical systems within the material confines of the real world. It emphasizes that such an approach is not merely theoretical but deeply entrenched in real-time, information-driven processes that aim to navigate the multifaceted stages of designing, constructing, and potentially operating diverse systems. This method stands out for its comprehensive coordination through the tangible realms of materiality, incorporating both hardware and software environments to bring service systems to life.

At the core of the project-based method is the meticulous processing of coordination-related information, which serves as the backbone for the methodical design, development, and life-cycling of service systems. This approach is characterized by a structured methodology that outlines specific, coordinated operations, each culminating in precise deliverables. These projects are conceived as coordinated packages of information, meticulously planned to execute actions and achieve results through tasks that ultimately yield beneficial objects and insights.

Furthermore, this approach underscores the significance of activities within the project framework. These activities, known to contributors as their tasks, are intricately coordinated to ensure seamless teamwork, fostering a collaborative environment that acknowledges and incorporates the perspectives of all stakeholders involved. The article highlights the crucial aspect of scheduling, which is carefully tailored to accommodate the dynamics of team collaboration and stakeholder engagement, ensuring the project is executed efficiently and effectively.

Graphical Abstract

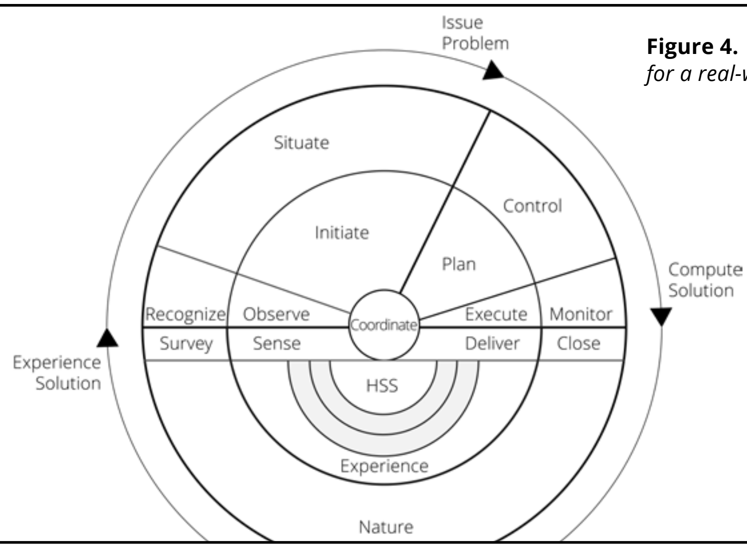


Figure 4. *Project coordination planning for a real-world societal system.*

1 Introduction

A.k.a., the project structure; the project-based approach, and the project method.

A project-based approach is a real-time, information-based approach to coordinating the resource realization of complex socio-technical systems. It involves the processing of coordination-related information in order to design, construct, and potentially, operate, some system in the real world. In other words, a project-based approach involves coordinating the design, development and operative life-cycling of service systems through materiality (i.e., through a material, hardware and software, environment). In practice, the project method is the specific, coordinated way of performing an operation that implies precise deliverables (at the end of each stage of the project).

CLARIFICATION: *"Projectability" (project-ability) is the ability realize (Read: make real or material, to reify) that which is currently unrealized (informational), and to distinguish between the two. Effectively, the project approach allows for (i.e., conveys the capability for) complex, parallel "projections" (i.e., constructions or creations) into a commonly shared, real world environment. When you think about projects you have to think about moving forward. A plan brings common understanding to navigating motion together in a shared space.*

Real world problems and challenges are complex and necessarily approached through projects. Projects define tasks by means of scope and requirement with the purpose of a designed construction as the output. Projects involve teams of individuals working together toward the shared constructive purpose for the projects existence. Here, there are tasks within which are processes for accomplishing the task. There is a spectrum of effort automation for task fulfillment processes. In other words, some tasks and subtasks are entirely automated, some involve a combination of automation and human effort, and some involve only human effort.

2 What is a project?

A.k.a., Program (collection of inter-related projects), portfolio (collection of inter-related programs, projects), plan.

A project is a systematically structured approach to resolving a problem in the form of a[n information through to materialized] solution (e.g., the community's societal habitat service system). The output of a project is an operational system or result, as a selected response to some directional input (e.g., an issue or problem). More technically, the function of a project [in a unified information space] is to successfully deliver one or more requirements in the form of a product or result. In other words, a project is a bounded, directional information space within which a problem space is resolved into the selection and construction of a solution into an operational, materializing environment. At the human level, a project encompasses a set of interrelated tasks and decisions that are executed over an identified period of time within limitations (real world and organizational) to resolve an intention. A project can be visualized as structured flow of information and events, by initiation, through a process or processes, producing an outcome. The purpose of a project is an outcome, a result - a directional "desired" change in a material condition that benefits the user (as the "target" population or group). A projects can be viewed as an organizing mechanism (process organization) for getting work done. Organizations do projects. Projects drive (direct) change [to/in an environment].

CLARIFICATION: *In some organizations there are differences in meaning between the terms 'project' and 'plan', and in other organizations the terms are used interchangeably. Here, the project to bring into existence a network of integrated city systems operating through a unified societal model, and the plan to bring this system into existence is detailed within an information set called the 'project plan'.*

A, 'project plan' may also sometimes known as: a plan; a project management plan; a project coordination plan; an implementation plan; an execution plan; and a construction plan, etc. Psychologically, a plan ensures that the momentum is kept up (because progress and issues can be seen) when a plan is executed. Planning and scheduling in society are a dynamic and a never-ending process. A project is setup to be "successful" when the right environmental conditions exist for it to be successful. Therein, a "successful" project is a project that satisfies its intended purpose in a safe, timely, and resource-effective manner producing a result that aligns with the intention.

NOTE: *In some social contexts, the word "project" is replaceable with the word "mission", and to a lesser extent, the words "goal", "objective", and "outcome".*

Every project relates to a product (system, service, object, etc.). The product could be a tangible product, a software system, a service, or a new organization. Note that the project result is not synonymous with the product. The product definition is the product's boundary condition(s).

> **INSIGHT:** *The deliverable of a societal-level project is that which we all are "collectively" materializing.*

A project is carried out through a series of interdependent tasks - that is, a number of non-repetitive tasks that need to be accomplished in a certain sequence in order to achieve the project objective. A project consists of a coordinated series of activities or tasks performed for a common purpose. Here, phases must be worked through and tasks must be completed (as "gated" processes) in order to complete the whole project. A project utilizes various resources to carry out the tasks and meet the project's stated requirements. Each task is a sub-system containing input, process, and output elements, and is a sub-part of the unified, projected system.

> **Note:** *It is not logically possible to "do a project", it is only possible to "do tasks as part of a project".*

The following is a common list of definitions of the concept, 'project':

1. A project is composed of components, activities, and rules that lead to some "thing's" materialized existence.
2. A project is an forward/series progression ("endeavour") designed to produce a unique service, product, or result. All projects have boundaries for progression (i.e., a life-cycle), progression proceeds through phases, each with a defined beginning (input) and end (closing output). Sometimes the beginning and end are time-constrained, and sometimes not.
3. A project is executed ("undertaken") to meet unique goals and objectives, typically to bring about beneficial, objective change. The change must be objective, as in, measurable. The change must be measurable so that the project's success/completion can be evaluated.
4. A project is the first [in]formation of a construction. Wherein, a construction is an information and/or material asset, a designed solution. That asset represents a potential construction or realized construction (into materiality).

The 'solution' characteristics of a project information set are:

1. Given what is known, a project has a synthesized output, known as a 'solution'.
2. Given a motivation by consciousness, a project is the progressive elaboration of a problem (direction, goal, etc.) and its predicted, and then tested and delivered/operated answer, known as a 'solution'.

It is relevant to note here that current operations are run as [continuous] projects with typical task start and end times. Ongoing service issues (operations concerns). In project management, these tasks are considered processes and not projects. However, the habitat service system operations themselves are considered to be run as projects; they are also process or serve groups, and at a lower level, processes. Synonyms for current operations are day-to-day operations of habitat service systems.

Projects may be classified according to a set of characteristics, including, but not limited to:

1. **Novelty** - refers to a derivation, update, or a new system.
2. **Service complexity** - refers to the number of [societal service] systems engaged.
3. **Technology complexity** - refers to the level(s) of technology engaged.
4. **Organizational complexity** - refers to the number of information sets engaged.
5. **Uncertainty** - refers to the degree of unknowns.
6. **Pace** - refers to time (e.g., fast, regular, slow).

For any complex project plan there are two levels of action:

1. **The Project Level** - a plan for developing a community-type society. The top-level for which the purpose of the project is to construct a community-type society. The project level is the level for which the purpose of the project is to meet one or more of the top-level project imperatives.
2. **The Sub-Project Level** - sub-plans for discovering and developing a community-type society. A sub-project plan (a.k.a., project integration plan) is a standard (i.e., not a societal-level) project plan that accounts for integration within the top-level plan, and provides the basis for all coordinating activities between the sub-project and the Project. A sub-project plan ensures alignment with the project level.

2.1 Project structured information set

The three processes of executing, planning, and controlling rely on a single structural representation of the project (e.g., the work breakdown structure type). These three responsibilities (Ex, Pl, and Co) share the same obligation, when programming so dictates or when a malfunctioning warning sign occurs, to embark on a discussion and to go ahead only if a decision has been taken.

1. Executing.
2. Planning.
3. Controlling.

Projected-based information coordination comprises the transformational processes of:

1. Gathering (collecting).
2. Structuring (integrating and generating).
3. Retention (memory).
4. Access.
5. Adaptation (updating).
6. Application (applying).

Information transformation, itself, comprises:

1. Synthesis - put parts together to form a whole.
2. Analysis - separates material or concepts into component parts so that its organizational structure may be understood.
3. Application - applies what was learned.
4. Comprehension - understand the meaning, translation, interpolation and interpretation of instructions and problems.
5. Knowledge - recall data or information.

The common flow of information through a project structure:

1. Project initiating.
2. Project planning.
3. Project execution.
4. Project completion.

2.1.1 Societal project planning phased organization

> **NOTE:** *It is desirable to obtain total agreement from the entire project team regarding these objectives and priorities to ensure alignment.*

The project planning of a habitat involves a set of basic project phases that continuously design, build, and close (or, operate) new states of a habitat [service system]:

1. **Phase 1: Building the project team.**
   A. List stakeholders.
   B. Identify working group (if applicable).
   C. Identify habitat InterSystem service team or InterSystem Habitat Team (if applicable).
   D. Identify resources and tools.
2. **Phase 2: Project definition (establish the context).**
   A. Define project (goals, purpose, objectives).
   B. Establish current state.
   C. Identify needs.
   D. List priorities.
   E. Define objectives.
   F. List available data and gaps to be filled.
3. **Phase 3: Solution definition.**
   A. List requirements.
   B. List outcomes of project.
   C. List available solutions to project.
4. **Phase 4: Plan definition.**
   A. Approach identification.
      1. Outline of phasing, tasking, and scheduling (project coordination).
      2. Local context.
   B. Direction identification.
      1. Purpose and goals (mission and needs).
   C. Execution identification.
      1. Operational criteria.
      2. Evaluational criteria.
5. **Phase 5: Execution (implementation).**
   A. Execute the phases of the plan, either synchronously or asynchronous as appropriate.
   B. Control the implementation of the plan by means of monitoring, communication, deciding, and acting (working) for three types tasks:
      1. Coordination tasks.
      2. Human action tasks.
      3. Automatic action tasks.
   C. Schedule plan by means of time line association between available locations, resources, and actors (e.g., personnel and/or systems).

Master planning for society is a socio-technical project that involves economic, technical, social, and environmental categories of data:

1. Economic resources - visualization of the flow of objects (resources).
2. Economic production - mechanism by which resources are transformed into good and services to meet needs.
3. Technical knowledge and tools - using knowledge and tools to change, advance, and modify the habitat.
4. Social fulfillment - accounting for the fulfilment of needs in habitat services.
5. Environmental regeneration - accounting for the carrying capacity of the environment, controlled [in part] by site modification.

2.1.2 High-level project organization

At a high-level, a project is composed of the following actionable information sets:

1. **Information system or space (system):** All information relevant to the project.

2. **Directionality, purpose, or imperative (input):** Describes the goals (and objectives) for performing the project's process(es).
3. **Results or outcomes (output):** Describes the observable (materialized) state-dynamic expected [in an environment] from the successful/complete performance of the project's process(es).
4. **Tasks, instructions, or activities (process):** Describes the actions intended to produce the outcome(s) using the project's inputs.

Using other terminology, a project is sub-composed of:

1. **Directive component** (input of initialization - that initializes the space) - a directive (or imperative) area represents the point-source intention and defined input for the state-dynamic change.
2. **Knowledge component** (input of learning - that from memory) - a knowledge area represents a complete set of concepts, terms, and activities that make up a usable, specialized information set.
3. **Process component** (input of instruction - that of software code and material task) - a process a specific method, mechanism, procedure, task, protocol, etc. Process components are characterized by their inputs, the tools & techniques that can be applied (and developed), and the resulting outputs.
4. **Constraints component** - that limits the execution of the project and constrains it to a boundary of what is and what can be made available. Every project has constraints (limitation) imposed by some combination of:
   A. Information.
   B. Time.
   C. Energy-power.
   D. Material resources.
   E. Labor resources.
   F. Throughput.
   G. Prerequisites.

The basic flow of information within a project structure could be viewed as:

1. Prioritization.
2. Analysis.
3. Design.
4. Build.
5. Evaluate.

Fundamentally, the projects approach is to have a projects' support service for the fulfillment of the global human population.

2.2 The project direction and orientation (initiation): project objectives, outcomes, outputs, and key results

To fully identify a project and measure the progress of a project it is important to define project objectives, project outcomes, and project outputs, along with the measurable key results that define achievement of each objective and the project as a whole (*Project Managemen*t, 2017):

1. **[Project] Objectives** - A project objective is a statement of the overarching rationale for why the project is being conducted. A project can have one or more project objectives. A useful way to frame an objective is to answer the question, "Why is the project being done?" The result is a one sentence statement, or series of statements, usually starting with the word "To ..."
2. **[Project] Outcomes (a.k.a., key results)** - Project outcomes are the benefits or other long-term changes that are sought from undertaking the project. They are achieved from the utilisation of the project's outputs. Here, key results (outcomes) are determined for each objective. Outcomes are linked with objectives, in that if the outcomes are achieved, then the project's objective(s) have been met.
   A. **[Project] Target outcomes (a.k.a., measurable key results)** - Project target outcomes for a project are outcomes that have a measurable benefit and will be used to gauge the success of the project. Usually there will only be a small number of target outcomes for any project. Each measure[able key result] will be linked to one or more target outcomes. At the end of the project the measures will help answer such questions as, "what have we achieved?" and "how do we know?" Target outcomes are expressed as a sentence in the past tense and usually start with a word ending in "-ed", such as improved, increased, enhanced, amplified, filtered, stopped, decreased, or reduced. Framing target outcomes in this way makes it easier to determine their success measure.
3. **[Project] Outputs** - Project outputs are the products, services, organizations or procedures/practices that will be required to be produced to meet the identified outcomes. They may be products (objects) or services (processes). Outputs link with outcomes, in that the outputs are used by the project's customers to achieve the outcomes. An output can be a verb (process) and/or object (noun).

4. **[Project] Activities and tasks** - are the work that needs to be done to produce the outputs for the project. Activities are the larger, higher-level units of work, which can be broken into tasks, the smaller units. Activities and tasks are dynamic concepts (i.e., processes, verbs, actions). Activities and tasks are written in the present tense using verbs (dynamic concepts including motion).
5. **[Project] Milestone** - a significant, scheduled event that acts as an identifier of progress (i.e., "progress marker") in the progress to completion of a project. A single milestone is usually the completion of a key activity or task, or a deadline. A milestone is like a toggle switch – at any point in time, it is either completed or not completed. The start of an event is rarely a milestone. Milestones are expressed as sentences in the past tense.
6. **[Project] Risks** - any concern that some object, process, or event might adversely affect the successful completion of the project.

2.3 The project direction (initiation): project goals

A goal is, the intention of a 'user'. To an engineer, goals represent the intentions of the system's user. In concern to systems, a goal describes a relationship that a system desires to have with its environment. In general, goals are formulated based on a current situation and a measurement criteria. If consciousness has the intention for something to stay the same, or to change, then a goal is present. Optimal goal selection relies on understanding, and the coordinated layering of direction throughout the flow of a project.

In order to accomplish the Project's primary directive, the goal is to expressly materialize the following three sub-systems:

1. [Conception/Design] A continuously updated specification of the whole societal system. A specification is anything that describes what an actual instance looks like.
   A. We need a commonly shared design plan to iterate [the next state of evolution of] our society.
2. [Materialization/Action] The operation of a network of city systems based upon and expressed through the specification. A city system (or network of city systems).
   A. We need a controlled habitat service system that operates in alignment with the design plan.
3. [Experience] The experience of optimized fulfillment and well-being for each and every individual human, based upon the given conditions.
   A. We need a population of self-motivated, self-integrating, and compassionate humans who understand and align with the design plan.

When these highest-level [project] goals/objectives are complete, then the Project, as specified in this Project Plan, is complete[ly delivered]. In this sense, objectives/goals are the final outcome to the user.

In order to accomplish the Project's primary directive, the proposed societal system maintains the following four goals:

1. Quantitatively identify the different components of the human system, and understand how these components relate to each other.
2. Quantitatively fulfill the needs of individual humans in the human system, and understand how the needs are best fulfilled.
3. Quantitatively determine the habitability of an environment, and understand how different spaces have different habitability potentials. Access past and present habitability potential of location.
4. Sense the experience of a reliable and robust operational service system (intentionally developed).
5. Remain sufficiently uncertain about what humans require to maintain a set of value inquiry thresholds programmed into the decision system as the socio-economic decision inquiry process group and the solution inquiry process group.

In order to accomplish the Project's primary directive, there are coordinate system objectives. Societal coordination objectives are common to all projects.

The primary and secondary goals of the proposed coordinated societal system are to:

1. Ensure positional data of all resources.
   A. In application, the question becomes, are we using environmental resource survey data?
2. Ensure effective and efficient interaction and communication among project participants.
   A. In application, the question becomes, are we using a unified information system?

The supportive sub-goals of the project coordinating system are to:

1. Assure the highest quality technical, organizational, and contractual coordination at every level.
2. Initiate and facilitate the resolution of decisioning at every level.
3. Support active and beneficial collaboration among projects.

In order to accomplish these objectives, the following project coordination processes must be carried out:

1. **Scheduling work and access** - register tasks in time and space. Scheduling activities.
2. **Monitoring work and access** - track the operational work of the project. Monitor activities and results.
3. **Reporting work and access** - communicate an understanding of the projects progress and status. Reporting activities and results.

Each individual process expresses a unique level of resulting information motion:

1. In concern to **controlling**, a decision selection is approved, accepted for inclusion (inclusion).
2. In concern to **scheduling**, when a selected (decided) change is to be executed (as an activity/task), an InterSystem Team role synchronously with a change control coordinator shall be assigned accountability [for the project]
3. In concern to **monitoring**, when a change in a noted characteristic is deemed appropriate, notification of the change shall be sent to the appropriate review and change control coordinator [for the project].
4. In concern to **reporting**, when an expected change is complete, an accountable event log shall be sent to the appropriate review and change control coordinator [for the project].

Each individual contributing to the optimization of a coordinated society maintains a set of life-orienting goal (more commonly called 'rules'):

1. The design must account for life value regulators from start to finish.
2. The production must have more life value capacity through generational time.
3. The evaluation must compute a life value measure as a criteria to tell (determine) greater from lesser ('>' from '<') in any domain by knowledge of life capacity loss or gain.
4. Cumulative life gain is always the organising goal, the intended result.
5. Coherently inclusive decision or action is enables life capacities, the better it always is for common life opportunity capacity.

2.4 Executable project elements

In order for a project to be completely "delivered", actions ("activities") must occur. An activity or action is one of a coherent set of specific steps that must be taken to reach the imperative(s) conclusion (i.e., the change for that which the project was initiated). Therein, an executable is an information set upon which action can be taken.

Executability means,

1. From an operator's perspective - that a service is operational and monitored for alignment with specification.
2. From a user's perspective - that a system may be validated against specification.
3. From a developer's perspective - that a specification is verified against the facts of its operation.

The commonly named executable elements of a project include:

1. **Deliverable** - a tangible (materializable), verifiable work service or product.
2. **Activity** - a planned action.
3. **Activity work package** - a deliverable at the lowest level of the deliverable diagram (work breakdown structure). A work package may be divided into activities. Work elements with expected duration and resources requirements (and in the market, costs) that may be subdivided into tasks.
4. **Task** - a deliverable at the lowest level of the work breakdown structure. A work package may be divided into activities.- the selection of a 'job', procedure, or other process, to accomplish an effort.
5. **Work package** - a deliverable at the lowest level of the work breakdown structure. A work package may be divided into activities.

The executable project elements are coordinated into existence by an accountability organizational structure and matrix:

1. **Organizational breakdown structure** - relates work packages to organizational [InterSystem] work/team units.
2. **Accountability matrix** (a.k.a., responsibility matrix) - relates the organization structure to a deliverable diagram (work breakdown structure) to ensure that each element of the project's scope is assigned to an accountable individual or system.

2.4.1 Project selection for execution

For any project, the project is first defined, then projected solutions must be uniquely identified, and then, those solutions are screened for optimality prior to execution of the optimal:

1. "Project selection" refers to selecting (via some decision-determination method) the one project solution which is probably best to execute. To identify the one project solution out of a set of possible project solutions, project identification is required.

2. "Project identification" is formalized by a specified set of objectives and a given the context of an internally and externally bound nature. Internally, projects become identified through their solution. Externally, projects become identified with their results. For any projected solution, solution design possibilities are screened [through a decision control system consisting of logical programs] to provide an optimal project proposal for execution.
3. To "screen a project" is to have a set of criteria for evaluating the project. An appraisal of a project['s success] has to be based on a set of criteria. There are always criteria in screening for optimal solution selection and for evaluating the experienced results of that solution selection.
   A. At a societal-level, the socially defined values for any society represent these set of criteria (i.e., values are societal alignment conditions). In community, these values are reasoned and identified in the Social System Specification. In the market, investment, payback, and likely profit are a good set of [value] screening criteria. For this proposed community-type societal system, the core orientationally stabilizing values of individual freedom, restorative justice, and technical efficiency are a good set of [value] screening criteria.

2.4.2 Execution [tasking] phases

A.k.a., Project execution phases.

In concern to this societal building project, the primary execution tasking phases are:

1. **Project initiation/identification**
   A. A plan is created to change the state of our common living system toward one that meets all human need while facilitating the generation of well-being a the individual, social, and ecological levels.
2. **System specification**
   A. A design is created to model the common system (a unified societal information system), to which the state of the living system may be changed.
3. **System prototype** (minimal to fully integrated prototype)
   A. The design is tested and reworked.
4. **System operation**
   A. The new system becomes fully operation at the population level (context dependent, integration can have a widespread affect and effect due to the rapidly real-time nature of the community's service network).
5. **System feedback**
   A. The integration of feedback for the next iteration of the system.

2.4.3 Execution [tasking] life-cycle

A.k.a., Project execution phases.

Every project is executed through a life-cycle of project phases. Every project [to develop a new system] follows the same (or similar) set of task-based execution phases. Information about changes within a project pass through these "gated" event phases, becoming more coherent and actualized over time, as tasks are performed, until the intended result is met (i.e., the new system is produced).

A project [to develop a new system] may be sub-divided into the following set of execution phases, representing the collection and integration of information relevant to the resolution of the project:

1. **Identification** - define usage of system.
   A. A need or issue is recognized.
   B. A problem and set of requirements is formulated.
2. **Design[ation]** - integrate information into a unified information system.
   A. A design is synthesized.
3. **Development[ation]** - construct the system from the unified information.
   A. Discover[ing] - analyze the situation and acquire information to design the system.
   B. Design[ing] - synthesize the design of the system.
   C. Construct[ing] and Test[ing] - construct the design and integrated into full operation.
4. **Operation** - operate and monitor the system.
   A. Use[ing].
5. **Evaluation** - assess and verify the system
   A. Survey[ing].
6. **Iteration** - update, upgrade, and replace the system
   B. Issue[ing].

Note here that the common "executive" functions include:

1. Panning.
2. Deciding.
3. Checking work.

2.5 Societal-level project execution elements

NOTE: *Projects sustain values.*

The following are axiomatic input-tasks for project development coordination:

1. **Define the system concept through imperatives and requirements**, which are attainable, definitive, quantifiable, and with specific duration and resulting conclusion. The intention of the societal project is composed of a set of imperative requirements. In engineering, the imperatives ("obligations") and proceeding requirements, define the primary problem domain.
   A. Define ("identify") the societal system.
2. **Identify the work (a.k.a., tasks, actions, events, and other activities)**, which is sub-divided into tasks following either a manual system (of input, process, and output), or they may be automated to provide a functional service, of which the societal system is itself in service to fulfillment.
3. **Sequence the tasks**, which involves the mapping of all relationships across all scheduled activities into a visual network map, allowing for effective monitoring of the project by everyone (open source).
4. **Estimate the activity costs and durations**, which allow for resource budgeting, scheduling, and decisioning.
   A. InterSystem Team work packages become available.
5. **Reconcile constraints**, including time, resource, and financial constraints, which will likely necessitate the determination of a decision.
   A. A decision protocol.
6. **Execute the tasks to design and build the system** by executing tasks required for physical and/or digital integration of the system concept.
   A. Design the information system through work packages.
   B. Construct a habitat service system through work packages.
7. **Observe and Review the results** and integrate the changes.
   A. Operate a habitat service system through work packages.

2.6 Project measurement

Measurement is a component of every project. Measurement is used to (i.e., project metrics enable a project coordinator to):

1. Assess the status of an ongoing project.
2. Track potential risks.
3. Uncover problem areas before they become 'critical'.
4. Adjust work flow or tasks.
5. Evaluate the [project team's] ability to control the quality of work products.

2.6.3.1 Process measurement and process metrics

In the context of measurement, process measurement is the efficacy of a process (often, indirectly). Process measurement provides a mechanism for objective evaluation of a process. Process metrics are the measurement's definition. Metrics must reliably and accurately measure the factors that are needed to complete some process (e.g., an objective or need, a decision). Metrics are, by definition, evidence-based, because they are a criteria, that if observed, is evidenced by marking the condition/observation event as occurring by a trusted agent. Hence, measurement (metrics) is an evidence-based method. The success of any project relies on evidence-based decisioning, data collection, evaluation, and adaptive behavior. Metrics required by standard setters (i.e., article and decision working groups) should be well-defined. For example, in terms of allowable measurement approaches, expected level of granularity, etc.), consistent, and objectively measurable.

> **NOTE:** *Frequently, the same measurements can be used for both process metrics (measurement across many projects) and project metrics (measurement upon a single project).*

Process metrics are useful for:

1. Estimation.
2. Quality control.
3. Productivity assessment.
4. Project control.
5. Tactical decisioning.
6. Coordination.

Process metrics can be derived by:

1. Measuring the characteristics of specific engineering tasks.
2. Measuring outcomes that can be derived from the process.

Potential outcomes for process measurement include, but are not limited to:

1. Measures of errors uncovered before release of the product.
2. Defects delivered to and reported by end-users.
3. Work products delivered (productivity).
4. Human effort expended.
5. Power/energy expended.
6. Material resources expended.
7. Calendar time expended.
8. Schedule conformance.

Process metrics include, but are not limited to:

1. Quality (quality-related) - focus on quality of work products and deliverables.

   A. Correctness (e.g., adherence to requirements).
   B. Maintainability (e.g., easy to fix?).
   C. Integrity (e.g., attack vulnerability).
   D. Usability (e.g., training time, number of interfaces).
2. Productivity (productivity-related) - production of work - products related to effort/energy/material expended.
   A. Value analysis (a.k.a., earned value analysis).
3. Statistical SQA data - error categorization and analysis.
   A. Severity of errors (1-5).
   B. Mean time to failure (MTTF).
   C. Mean time to repair (MTTR).
4. Defect removal efficiency - propagation of errors from process activity to activity.
   A. Defects found in this stage.
   B. This Stage + Next Stage.
5. Reuse data - the number of components produced and their degree of reusability.
   A. The number of components produced and their degree of reusability.
   B. Within a single project this can also be a “project metric”. Across projects this is a “process metric”.

2.6.3.2 Project metrics

Project metrics include all measures related to the project used to assess product/system quality on an ongoing basis, and when necessary, modify the technical approach to improve quality. Project metrics measure aspects of a single project to improve decisions taken on the project.

Project metrics include, but are not limited to:

1. Number of team members.
2. Number of external systems interfaced.
3. Number of technology objects used.
4. Number of executable functions.
5. Etc.

Project metrics are used to:

1. Minimize the development schedule by making the adjustments necessary to avoid delays and mitigate potential problems and risks.
2. Assess product quality on an ongoing basis, and when necessary, modify the technical approach to improve quality.

Every project should measure:

1. **Input metrics** (inputs, project input metrics) - measures of the resources required to do the work (e.g., materials, people, tools).
2. **Output metrics** (outputs, project output metrics) - measures of the deliverables or work products created during the engineering process.
3. **Result metrics** (results, project results metrics) - measures that indicate the effectiveness of the deliverables.

Examples of project metrics include, but are not limited to:

1. Effort/time per [engineering] task.
2. Errors uncovered per review hour.
3. Scheduled vs. actual milestone dates.
4. Changes (number) and their characteristics.
5. Distribution of effort on [engineering] tasks.

Best practices for developing and using metrics include:

1. Teams must set clear goals and metrics that will be used to achieve the goals.
2. Never use metrics to threaten individuals or teams.
3. Metrics data that indicate problem areas should not be considered “negative”. These data are merely an indicator for process improvement.
4. Do not obsess on a single metric to the exclusion of other important metrics.

Best practice for developing effective metrics:

1. Simple and computable.
2. Empirical and intuitively persuasive.
3. Consistent and objective.
4. Consistent in use of units and dimensions.
5. Programming language independent.
6. Should be actionable.

Actionable metrics - metrics that guide change or decisions about something.

1. For example:
   A. Actionable - measures the amount of human effort versus use cases completed.
      1. If result is too high: actions may include more training, more designing, etc.
      2. If result is too low: actions may include maybe the schedule can be shortened.
   B. Non-actionable - measures the number of times a word appears in a manual.

QUESTION: *What is to be done if the measured result, in comparison to the metric, is too high or too low?*

2.7 Cybernetic-type project requirements

Projects that are executed with controlled feedback, which is all socio-technical projects, must recognize, organize, and resolve project a set of [cybernetic] project requirements:

The common phase completion requirements for socio-technical projects are (note: the cycle repeats with a complete database from which to design solutions and take decisions):

1. Databases complete.
   A. Initial state visualization [of fulfillment] complete.
   B. Initial processes description [of fulfillment] complete.
   C. Objectives complete.
   D. Requirements complete.
   E. Issues complete.
2. Solution designs complete.
   A. Decision algorithms complete.
   B. Optimization calculations complete
3. Operations complete [signal sensor].
   A. Evaluations complete.
   B. Surveys complete.
4. Updated database complete [result integration and controller updating].
   A. New state visualization [of fulfillment] complete.
   B. New processes description [of fulfillment] complete.

Continuous project coordination requires the integration of a set of lists that [must be completed, and] identify and plan for work/action in the material-informational environment (i.e., in material-time, space-time):

1. Humans list.
2. Teams list.
3. Schedules list.
4. Events list.
5. Concerns/Issues list.
6. Actions/Tasks list.
7. Deliverables list.
8. Tools/Technologies list.
9. Resources list.
10. Locations list.

Note: These lists are always active during the lifetime of a project, and they are filled-in/ populated by project coordinators.

3 [Project] Coordination

NOTE: *Individuals among any society may communicate and coordinate in order to optimize their fulfillment.*

'Project management' is the market-labor term for that which, in community, is referred to (in part) as 'project coordination'. It could be said that the general purpose of project coordination (a.k.a., project management) is to control ("gate") and monitor a project's information flow(s). At the information system level, project coordination ("project management") is an information support service to other Functional-Service InterSystem Teams (composed of contributors), and the whole user-base (through open source creation). The InterSystem Teams, and also, the whole community of users, have several formalized organizational structures common to cooperative teams.

Project coordination is an iterative process. For example, the planning phase is a refinement of the initiation phase. In some instances, phases may be repeated because of changes within the project. Also, project phases may be performed simultaneously as well as sequentially. For instance, the planning, execution, and control phases may be performed in parallel as changes are made to the project baseline.

A fully coordinated societal environment one in which each action taken by each individual in a demarcated set of actions, correctly takes into account (1) the actions in fact being taken by everyone else in the set, and (2) the actions that the others might take were one's own actions to be different. To achieve equilibrium in this model, it is not enough that each subject correctly anticipates the contingent actions of everyone else. It must also be the case that each subject—using these correct understandings - chooses his or her own strategy and actions so as to maximize utility; because, it takes work (effort) to sustain a utility.

In order to sustain a unified information system where actions taken by individuals benefit themselves and others, work organization must be sub-categorized. Project coordination involves the accounting for and directing of information between multiple sets of project-related lists (i.e., categories of data that are useful for coordination purposes):

1. Project 1 (Plan-of-Work 1)
   A. Tasks (actions)
      1. ...
   B. Roles & Personnel (work descriptions)
   C. Resources (objects & data)
   D. Deliverables (services & assemblies)
      1. ...
2. Project 2
3. Project 3
4. Project ...

3.1 Societal-level project coordination

NOTE: *At the societal level, project coordination makes the societal system more resilient (i.e., robust) by taking the needs of all the stakeholders (everyone whose human-materialized, conscious life is involved) into account.*

Fundamentally, humans participate in social-technical organizations to increase their chances of satisfying their needs through coordination.

Some common questions necessary for coordination of a project for a type of society include, but are not limited to:

1. **Who coordinates** community [into existence]?
   A. Individual humans are contributors (open, global cooperation) to the societal system
2. **What coordinates** community [into existence]?
   A. A societal decisioning system integrated within a larger societal, real-world information system.
3. **How is community** coordinated [into existence]?
   A. By enabling, and using, a societal information system (while, encoding common individual human need and common access to planetary resources).

NOTE: *From a project perspective, a [living] societal system may be viewed as a problem of project selection.*

3.1.1 Project 'management' [at the societal level] is redefined as project 'coordination and control'

Community can use the discipline of 'project management' (project organization), and it does not need to adopt the market and state elements. In the literature, there are a large range of "accurate" definitions for the term, 'project management'. In its most broad definition, project management is the:

- Application of knowledge, skills (competencies), tools, and techniques (methods) to project activity objectives to meet the project requirements.

In an information-based society, the idea of project 'management' becomes replaced by project 'coordination', which is characterized by:

1. **Unified, global cooperation-based** - accounts for everything, applies everywhere, and is open to view integral in access.
2. **Objective, predictive model-based** - informed by first order real-world abstractions, and not second order abstractions.
3. **Algorithmically, encoded instruction-based (controls-based)** - logical variability, and adapted from previously taken decisions.

Simply, in community, people are not "managed" as they are not coerced or awarded by an authority to work, but are working because they are self-motivated to participate in the development and operation of society as a contributor. Project management is a career profession in the market.

NOTE: *What 'management' does can include secret decisioning. Whereas, the concept of 'coordination' does not carry that association, and instead, carries the association of shared relationships.*

Market-labor parlance terms for project-related coordination include:

1. **Project management (PM)** – "The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. The role applies to any project or program personnel applying the knowledge, skills, tools, and techniques to project activities to meet the project (not product) requirements" (ANSI and PMI 2008, 6). This term will apply to those project managers, program managers, systems engineering managers (SEM), systems engineers that perform the role-specified activities regardless of their associated discipline. It applies to all disciplines such as finance, contracts, supply chain, quality, and engineering managers.
   A. In community, this is project coordination.
2. **Program [manager]** – "A group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually" (ANSI and PMI 2008, 9). 1209 Project – "A temporary endeavour undertaken to create a unique product, service or result" (ANSI and PMI 2008, 5).
   A. In community, these are service systems, the largest of which is the Habitat Service System (e.g., life-support, energy, water, etc.).

CLARIFICATION: *The PMI defines '***program management'** *(a sub-discipline to project management) as, a group of related projects managed in a coordinated way to secure benefits and control which could not be achieved individually. A program coordinator would thus "manage" a portfolio of projects, whereas a project coordinator would "manage" one project. (PMI 2016; ANSI Prince 2; ISO 21504:2015).*

3. **Project Management Body of Knowledge (PMBoK)** – the knowledge of how and why to "manage" a project as produced by the Project Management Institute (PMI).

   A. In community, this the social knowledge-base within the unified information system.
4. **Project manager (PM)** - a person named to manage the complete project, which includes product and system oversight as a subset of the overarching responsibility, authority, and accountability demanded of a project manager. A project manager is the person accountable for accomplishing the stated project objectives. The term will also be inclusive of the term program manager for this paper.
   A. In community, this is the project coordinator; the project coordinating entity.

The Lifestyle System Specification describes in greater detail how there is not the market-labor profession of "management" in community. Simply, in community, people are not "managed" (as in, not coerced or awarded by) an authority to work as part of an InterSystem [Projects] Team, but are working because they are self-motivated to develop themselves as well as participate in the development and operation of society as a contributor.

In the Community, people are not managed and the interrelationships between people do not have to be managed because everyone is arriving at the same or similar decisions about the system and they recognize their responsibilities both to themselves and to the community. And that those responsibilities to the community are also responsibilities that support themselves and their lifestyle. Not because they are robots, but because they have the same knowledge about the system and a similar set of understandings, values, purpose and approaches to the empirical and life-grounded system that maintains the community.

In an information-based society, the idea of project 'management' becomes replaced by project 'coordination', which is characterized by:

1. **Unified, global cooperation-based** - accounts for everything, applies everywhere, and is open to view integral in access.
2. **Objective, predictive model-based** - informed by first order real-world abstractions, and not second order abstractions.
3. **Algorithmically, encoded instruction-based** - logical variability, and adapted from previously taken decisions.

In the context of logistics (Read: the optimal, logical movement of objects), coordination refers to efforts (Read: the execution of supra-tasks and supra-information processes required) to be in the right place (location) at the right time (temporal) to execute a task as planned; thus, moving an object to its intended destination, optimally.

In an information system, the idea of project "management" is replaced, in part (i.e., the human subjective-management part is replaced), by objectively informed and processed project information. In an information system context, think of project management as an pre-programmed, open source, algorithmic coordinator of information relevant to a project, which processes project-level information in order to achieve all of the project goals and objectives, while remaining in the bounds of constraints.

A project coordinator actively and passively monitors a projects information sets to actively ensure that the solution inquiry (a.k.a., system development lifecycle, engineering development) delivers an optimal and organizationally/societally acceptable solution (through parallel decision inquiry processes).

> **INSIGHT:** *A project, in the market sense, is something that creates "value" for someone. A project could also be viewed as a structure for resolving a greater state-dynamic of fulfillment (i.e., resolving problems with fulfillment).*

Additionally, this coordinator monitors and controls the flow of project-level information. The term project "management" (and "manager") is a misnomer, because instead of the creation of something being the work of management (power-over-other relationships), it is a collaborative effort to bring something new into existence, or maintain the iterative operation of an existent system.

Within the context of a project, there is a need for coordination, which requires intrinsic motivation among the workers, who are voluntary contributors, and not laborers for anyone other than themselves, as users of the service systems they are "co-creating". Among community, there is no need for external reward and punishment, and hence, need for the management of other humans.

That framework which is applied to ensure the successful resolution of a project space is more akin to a coordinator, rather than a manager. A coordinator can still maintain control functions, as managers do, but the term is socially agnostic, whereas the idea of 'management' arises out of an authority-based, transactional set of social relationships.

3.2 Communication coordination

Coordination necessitates precise and accurate communication. Communication, in general, necessitates asking and answering the following five questions; these questions are are essential for coordinators and good practice for all communicators:

1. What do I need to communicate?
2. To whom do I need to communicate?
3. When do I need to communicate?
4. What method is most appropriate for the communication?

3.3 Technical coordination

A.k.a., Unified potential for movement.

Depending on the context of its application, the concept of 'coordination' has several related meanings.

Coordination is the ability:

1. To combine several distinct [informational/physical] patterns into a singular movement, with efficiency [in input usage] and effectiveness [in output delivery].
2. To integrate all the components of fitness so that effective movements are achieved.
3. To unify movements into a coherent and optimally effective pattern of movement.
4. To engineer/develop and apply/operate patterns of [information] movement efficiently, effectively integrating visual information for a purpose [vector].
5. Of system entities (i.e., actors or patterns) to interact beneficially for a higher-order system purpose/function.
6. To optimize the direction and sharing of information and materials.

Simply, coordination is the combined sub-abilities that allow people and/or systems to work together [efficiently and effectively for a common purpose]. Generally speaking, coordination is a global systems ability, made up of several synergistic elements and not necessarily a singularly defined ability. Decisioning, [spatial] orientation, and the ability to organize an effective and efficient pattern of reaction to real world stimulus are core elements of coordination.

QUESTION: *How well are we coordinating information and resources so that people have what they need, where they need it, when they need it?*

Society requires a coordinating structure to support a contribution-based platform. Societal coordination can be broken down into two high-level components (or component systems):

1. **Social-project coordination.**
2. **Technical-engineering coordination.**

And also, one low-level component (or component system):

- **Socio-technical tasks** *(Note: Both social-project coordination and technical-engineering coordination have associated socio-technical tasks)*

As a structure for the flow of information, coordination is an organizational relationship among entities/actors, which may become more or less coordinated over time (due to various internal and/or external factors). Coordination becomes optimized through cooperation (in computation, 'cooperation' effectively means, simultaneous and purposeful operations).

CLARIFICATION: *At a societal-level, project coordination refers to organizing, planning, initializing (execution done by teams), monitoring, evaluating, and deciding multiple societal-project inquiry tasks simultaneously. More simply, coordination refers to organizing and planning multiple tasks simultaneously.*

Coordination requires commonality, a similarity, or pattern with a purpose, otherwise there is less, or not, coordination (i.e., less of an ability to move together). For example, in the context of a project, everyone involved in the project uses ("follows") and informs the same, single, unified project plan, as part of a larger and more unified societal/organizational information system.

INSIGHT: *The idea of coordination is not theoretical; it's application to society, as an information system, will likely change (update) the language of the individuals therein, their comprehension of the real-word, and their ability to create safely together, in the real world.*

In concern to the idea of project management, herein, "management" is a misnomer for coordination and organizational/societal-level decisioning. Instead of a project manager, there is the element of an information system and the coordination of information therein via a project coordinator [that coordinates the flows of information during a project's societal/organizational life-cycle].

INSIGHT: *Coordination is the ability of a system to orient in an environment so that it aligns more closely with a given direction as iteration/motion occurs. And there, a system requires the ability of sub-systems to work together to re-orient its own internally motive system toward a direction of system fulfillment; because the system has a purpose. In concern to system resources, coordination ("management") may be simplistically defined as having and doing what is required/necessary to achieve the greatest access to, and get the most usage out of available resources.*

3.3.1 Monitoring phase - Project quality review

The purpose of quality reviews is to assure that the established systems development and project coordination processes and procedures are being followed effectively, and that exposures and risks to the current project plan are identified and addressed. Quality reviews facilitate the early detection of problems that could affect the availability, reliability, integrity, maintainability, safety, security, or usability of the system or product. Quality reviews enhance the quality of the end work products and deliverables of a project.

All deliverables (work products) are subject to quality review.

> **NOTE:** *In a societal decision process, the Effectiveness Inquiry is continuous, and is part of the quality review process.*

3.3.1.1 Peer review

A peer review is an informal review of systems, including documentation, which can be conducted at any time. These informal reviews are performed by the developer's "peers" -- frequently other developers working on the same project. Informal reviews can be held with relatively little preparation and follow up activity. Review data are collected and the developer determines which data require future action. Some of the work products prepared are considered interim work products as they feed into a major deliverable or into another stage.

3.3.1.2 Structured walk-through

A structured walk-through technique (SWT) is a more formal review and is prescribed by the engineering for all project deliverables. SWTs are used to find and remove errors from work products early and efficiently, and to develop a better understanding of defects that might be prevented. They are very effective in identifying design flaws, errors in analysis or requirements definition, and validating the accuracy and completeness of deliverable work products.

SWTs are conducted during all stages of the project life-cycle. They are used during the development of work products identified as deliverables for each stage, such as requirements, specifications, design, code, test cases (scripts), and documentation. SWTs are used after the work products have been completed to verify the correctness and the quality of the finished product. They should be scheduled in the work breakdown structure developed for the project plan, where, in practice, they are sometimes referred to generically as reviews. SWTs should also be scheduled to review small, meaningful pieces of work. The progress made in each life-cycle stage should determine the frequency of the walkthroughs; however, they may be conducted multiple times on a work product to ensure that it is free of defects.

SWTs can be conducted at various times in the development process, in various formats, with various levels of formality, and with different types of participants. They typically require some advance planning activities, a formal procedure for collecting comments, specific roles and responsibilities for participants, and have prescribed follow-up action and reporting procedures. Frequently reviewers include people outside of the developer's immediate peer group.

3.3.1.3 Exit review

A.k.a., Stage exit review.

The exit review is a process for ensuring a project meets the project standards and milestones identified in the project plan. The exit review is conducted by the project coordinator with the project stakeholders. It is a high-level evaluation of all work products developed in a life-cycle stage. It is assumed that each deliverable has undergone several peer reviews and/or SWTs as appropriate prior to the stage exit review process. The exit review focuses on the satisfaction of all requirements for the stage of the life-cycle, rather than the specific content of each deliverable.

The goal of a exit review is to secure the approval (verification) of designated key individuals to continue with the project and to move forward into the next life-cycle stage. The approval is a "sign-off" of the deliverables for the current stage of development including the updated project plan. It indicates that all qualifications (issues and concerns) have been closed or have an acceptable plan for resolution.

Generally, at a during stage review, the project coordinator communicates the positions of the key personnel, along with qualifications raised during the stage exit process, and the action plan for resolution to the project team, stakeholders, and other interested participants. The stage exit review is documented. Only one stage review for each stage should be necessary to obtain verification ("sign off") assuming all deliverables have been accepted as identified in the project plan.

3.3.2 Alignment and control variables

Alignment is the principal sub-coordination process. Alignment means the ability to adjust the position and/or orientation (Read: alignment) of some directed thing in motion (or iteration).

> **INSIGHT:** *The principles of coherency (or consistency) and alignment (degree of logical relationship of the one to the unified) are required for optimizing coordination.*

Alignability requires control. Some controls can be automated, given what is known and available.

3.3.2.1 For example, project control variables

The following are some of the variables that can be adjusted for any project:

Standard control variables for a project include:

1. Scope.
2. Time.
3. Resources.
4. Cost (market-only).
5. Jurisdictional (State-only).

Complementary control variables (control variables that are particularly salient in undefined projects):

1. Transparency.
2. Inspection.
3. Adaptation.

3.3.2.1 Alignment in cybernetic second-order systems

Alignment is a second-order cybernetic systems control function. Alignment requires the integration of feedback into a control system to determine the current value, and correct for the error, to an objective trajectory. By collecting and collating measurement data (i.e. observing the speedometer and the clock), the driver (the controller) can calculate at any point in time how fast the vehicular system should drive in order to achieve the defined goal [of getting to a location an a specified time]. Or, in the case of the habitat service system, by surveying human [need] requirements, the unified information system project coordinator (the controller, the project information processing unit) can calculate at any point in time how many human services must be produced, and in what time-frame, in order to meet a defined human fulfillment-requirement objective.

3.3.3 Coordination decisions

A.k.a., Coordination decision points.

A unified information system must coordinate between multiple information sets to ensure the fulfillment of the whole population of the society. The following are some common coordination decision points, expressed in the form of decision deliverables:

1. **Decision control coordination** - decide, accept, and approve the inclusion of people and final-decisions for execution.
2. **Decision analysis coordination** - decide the current model of the situation; decide the decision variables; decide the method of optimization.
3. **Technical planning coordination** - decide the scheduling; decide how to track; decide resource and system allocations.
4. **Technical assessment coordination**- decide how to track, measure, and assess metrics (for metrics collection).
5. **Requirements coordination** - decide requirements and decide mode of bi-directional traceability flow tracking.
6. **Risk coordination** - decide how to identifying and mitigate risks.
7. **Technical data coordination** - decide data structure; decide logging, decide data access, data storage, data control, data use; decide formal documentation interface.
8. **System/product coordination** - decide functional, physical, and non-functional, and FAIT (SAITL) specification, provide traceability.
9. **Service coordination** - decide the protocols by which a service operates.
10. **Implementation coordination** - decide the prototype model, simulation model, and testing model.
11. **Verification coordination** - decide the testing and evaluation model.
12. **Validation coordination** - decide how the end user will validate that the end-user's need(s) are met (with no further issues).

3.4 Project situational analysis

Coordination is not possible, at least not optimally, without a persistence of data being analyzed about the project's internal and external situation (i.e., issues with the project or environment as related to the project).

In order to sufficiently form a situation space the following questions must be answered:

1. What is the problem?
2. What causes the problem?
3. Who is affected by the problem?
4. Who cares about (is affected by) whether or not this problem is solved?
5. What are the priorities?
6. How will existing decisioning, research and experience, solve the problem?

NOTE: *If the 'problem' is not recognized, then the situation where an information resolution determination is required is not recognized.*

More generally, project situation analyses involve the following information sets:

1. **Participant analysis** - *who* is involved in the project's situation ("stakeholder/user analysis", person tree, "whose").
   A. Here, there is are people (users, humans).
   B. Here, there are technical systems (users, machines).
2. **Problem analysis** - *what* is problem of situation (problem tree, "ends").
   A. Here, there is an issue with some system from a person and/or some technical system (problem tree, root cause analysis, "cause"). Define the problem. Identify the needs.
3. **Objectives analysis** - *what* are objectives for situation (objective tree, "means")
   A. Here, there is also an outcome tree depicting a change in condition that benefits the user

(target group). Define the objectives. Define achievement for each objective, for purposes of measurement.

4. **Solution analysis** - *what* are alternative solutions for situation (solution tree).
   A. Here, there is a solution analysis at a societal level with societal-level inquiry decision processes.
   B. Here there is a solution analysis at a technical level with technical-level inquiry decision processes.

3.4.1 Assessment

A.k.a., Project evaluation, project analysis, situational analysis, situational assessment, situational report (sitrep).

An assessment is an analysis of a situation in order to acquire additional information in order to inform decisioning. In order to take informed decisions, often, a number of analysis activities are required to be carried out. The analysis data are not an end in themselves, but are used to inform (input into) decisioning.

There are many types of assessments. A conditional assessment (e.g., risk, impact assessment) is an objective analysis (i.e., review) of one or more conditions (e.g., risk) applied to a system, concluding with a determination of the probability or likelihood of the condition being true. However, not all assessments conclude with a probability. For example, a gap assessment analyzes what may be missing from a solution, given what is currently present in the solution (i.e., where are the gaps in the solution?).

An assessment process needs to include details on,

1. How will the project's result be assessed (i.e., what is success; what is the criteria?)
2. What are the difficulties and risks in the project and its final assessment of success?

3.5 The project coordinator

A **project coordinator** is an information processing unit (agent) that coordinates the flow of all project related information.

Project coordinator activities (functions/operations) include:

1. **Identify** project requirements.
2. **Define** clear and [probably] achievable objectives.
3. **Combine (integrate)** the knowledge areas into processes (process groups).
4. **Update/adapt** the specifications, plans, and approaches to users requirements.

A project coordinator integrates (combines) the three constraints (a.k.a., triple constraints) that are present in every project:

1. **Scope** (objective).
   A. **Quality** (condition/value).
2. **Resource** (materials; a.k.a., "budget").
3. **Schedule** (time).

Each constraint constrains the other and is in turn constrained by the other two. Planned projects can be impacted by impacting these variables. All change requests to the values of these variables must go through a formal change request procedure and form.

The 'scope' is the foundation of what is being developed [by a project] through resources and time. Herein, 'scope' is Project coordination carried out by a project coordinator, like a project managing project manager, necessitates the following information system elements:

1. **Documentation(s)** - Project documentation interprets the awareness of a project in the unified information system. To record existence and change within existence.
2. **Surveys(s)** - Project positioning in the information system and transparency with information system resources. Project positioning occurs through surveys, which also provide data for situational analyses. First, there must be known that some "thing" (what) exists.
3. **Integration(s)** - Project integration identifies and integrates the project's imperatives, in relationship to the determined method for their successful completion. There are memory structures, information processes, and the software tools for integration.
   A. To combine available information into a more completely view of what is:
      1. Occurring in the project.
      2. Planned to occur in the project .
4. **Evaluations(s)** - Project evaluations (sometimes "administrations") follows the operational progress of the project and reports on its progress; analysis highlight discrepancies, risks, while protocols issues alerts and requests decisions.
   A. To observe if there is an error between the environmental value and the required value.
5. **Decision(s)** - Project decisioning involves coordinating the resolution of the project space among the common constraints of quality, function, deployment, time, and resources.
   A. To decide the flow of information by "gating" activities within the life-cycle, closing them when they meet completion requirements. Here, "gating" refers to decisions as to whether (Read:

how) information does or does not flow [due to the completion of requirements for a given activity].

6. Plan(s) - Project plans involve ta visualization of the coordinated project resolution space (e.g., the Habitat Service System plans).
   A. To decide prior in time how information will flow. The result is a set structure for the flow information, known as a life-cycle [for the flow of integrating information].
7. **Monitor(s)** - Project monitoring involves the opening of a real-time or recorded visualization of a process in order to maintain transparency and ensure the process was completed as planned (as the rules for the gate were followed).
   A. To provide analytics [through a "dashboard" interface] to those who require calculated data for decisions to be taken concerning a project.
8. **Interface(s)** - Project interfacing refers to the visual interface users have into the project [information space], including the view the interface has into the unified information system, and its visualized and "dashboard" configuration.
   A. To provide a transparent interface into the whole project information space, to all users.

Project plans are organized by a project coordinator. The following is an example organization of project-related plans (Lewis, 2016):

1. Overview *of project.*
2. Definitions *applied in project.*
3. Project organization.
   A. Method of organization.
   B. Internal interface.
   C. External interface.
   D. Roles and responsibilities (accountabilities).
4. Coordinator process plans.
   A. Estimation (schedule, cost).
   B. Work (activities, resources, budget).
   C. Control (quality, metrics).
   D. Risk.
5. Technical [generative] process plans.
   A. Process model.
   B. Methods, tools.
   C. Acceptance plan (decision plan).
6. Analytical process plans.
   A. Configuration coordination (past and probabilistic future configurations).
   B. Verification/validation (of requirements).
   C. Quality assurance (reviews, audits).
   D. Subcontracts.
   E. Process improvement plan.

A common project planning coordination outline is:

1. Project coordination.
   A. Title and approval sheet.
   B. Table of contents.
   C. Distribution list.
   D. Project/task organization.
   E. Problem identification/background.
   F. Project/task description.
   G. Quality objectives and criteria.
   H. Special training/certification.
   I. Documentation and records.
2. Data generation and acquisition.
   A. Sampling design.
   B. Sampling methods.
   C. Sample handling and traceability-accountability requirements.
   D. Analytical methods.
   E. Quality control.
   F. Instrument/equipment fabrication.
   G. Instrument/equipment testing, inspection, and maintenance.
   H. Instrument/equipment calibration and frequency.
   I. Instrument/acceptance requirements for supplies.
   J. Non-direct measurements.
   K. Data storage and coordination.
3. Assessment and oversight.
   A. Assessment and response action.
   B. Reports and roles.
4. Data validation and usability.
   A. Data review, validation and verification.
   B. Verification and validation methods.
   C. Reconciliation with user requirements.

4 [Project] Planning

A.k.a., Planning, system planning, adaptive planning, dynamic planning.

Having a 'project' is a pre-requisite for planning. Projects involve the process of planning under the condition of uncertainty; they require coordination. In a typical project lifecycle, planning occurs in between project initiation and project execution. Project planning applies to all projects regardless of their size. Planning concerns the processes associated with pre-execution inquiries, integrations, and decisions within some predictively-probabilistic environment.

INSIGHT: *Community is a continuous plan. Continuous planning assumes that the system can continually be improved.*

In general application, there is a project-level plan, and then, there are more detailed and progressively elaborated plans for each phase level. Plans are more like snapshots of a desired change or development, instead of static blueprints, and their focus is more on temporality and movement than on the long-term configuration of an emergent structure. Simply, a 'plan' is a course of action (i.e., a model of actions). A plan is a mechanism or set of techniques to guide the activity of economic [socio-technical] decisioning through time toward the achievement of specific goals. A plan is a list of instructions to be executed/performed. A plan is a process with an input, process, output; whereupon, something does work until the output is out/delivered, and the whole thing is then evaluated.

A plan is a socio-technical, action-oriented integration of information:

1. A social-plan is a unified model of action that allows cooperation to work.
2. A technical-plan is the information required to do or build something with complexity.

In application, planning is the unified information processes of:

1. Organizing [of information].
2. Analytical-synthesis [of information].
3. Predicting/estimating (probability value, meta-value) .
4. Tasking (task value, numerical hierarchy of work breakdown, WBS).
5. Scheduling (temporal value with all relevant-associated project information).

Strategic planning is preparing in advance to respond flexibly and transparently to a range of possible eventualities. Planning is deciding (preparing), in advance, what to do, how to do it, when to do it, where to do it, and who is going to do it. Therefore, panning determines:

1. What is going to be done?
2. How is it going to be done?
3. When will it be done?
4. How flexibly can it be done (i.e., production cycle flexibility)?
5. With whom and what will it be done?
6. How will it be known that it is done?

Planning is a precise information processing and task coordination tool. Planning is essential to every project, regardless of the size of the project. The result of all planning is a deliverable, a plan [of action/execution] that is then, executed (and modified, as required). As a plan's execution progresses, more information becomes available, and therein, feedback loops may modify the plan.

The amount of detail required to plan varies according to the needs of a given project. Planning is a repeatable supra-process, which is involved in multiple other processes. The project approach processes information into a high-level usable format.

When planning is conducted in a systematic and precise manner, then execution of effort toward a goal(s) has a greater potential of being optimal, adaptive, effective, and efficient (i.e., execution becomes easier). Without a complete and comprehensive plan, it is difficult to execute and coordinate optimally, or even coordinate at all.

Using a travel analogy, 'planning' is the aligning of an intentional direction with potential action, to explain how to arrive at an intended destination, and what the experience of the intended destination will be, prior to executing the movement toward the intended destination.

Simply, planning is:

1. The word for doing project tasks with documentation.
2. The systematic preparation for action in the [temporal] future (i.e., some future iteration).
3. Deciding about the future "course of" [materialized] action (i.e., some spatial motion).
4. Thus, inherent in all activity (individual or collective), because all activity happens in time-space (i.e., materializing time).
5. Within a community-type society, habitat service systems provide an aggregated framework for the planning of the material [experience] system. There is a reason the material system, and life experience therein, is the way it is.

To plan is to decide ahead of time, to envision, everything about some desired state/output, and what is required to achieve [as checked off criteria] the desired output. Every plan has a predefined goal (or objective). Once there is a goal (or direction), then a plan(s) can be developed (configured, and selected) to achieve the goal and arrive at the objective. Therein, planning is itself, sub-composed of well-defined objectives. Planning is an information process that happens in the context of goals. A plan defines and explains what is needed, and is to executed as an operation, over some period of time

It is relevant to note that a plan can be deviated from. For any team-based plan, there are acceptable and unacceptable divergences off of [alignment of] a plan. For any team-based action, an objective approach is necessary to re-align [and restore] unacceptable divergences from a planned trajectory.

If "to plan" is to decide ahead of time, then "to use an algorithm" is to automate decisioning. If a plan may be coordinated into materialization, then coordination is, in part, the ability to synchronize project information for sequential execution, ahead of time.

In the real world (because of temporal-spatial existence), planning is required for:

- Optimum utilization of resources.

A '**resource**' is any "thing" that may be used in a project. Time, energy and material resources are the minimum. In a community type society, no human is considered a resource for other human; technically, there are no "human resources" as humans are not managed in the hierarchical and authoritarian sense.

The first project integration is planning:

- When time and task become available, planning becomes available.

The requirements identified in project related materials, (e.g., a scope of work, concept of operation). The level of detail will vary depending on project type and size.

The overall planning process can be sub-defined by a linked set of inquiry-deliverables:

Definition of what is to be *the activity:*

1. *What* activities are need (**processes**).
   A. Inquiry: what.
   B. Deliverable: tasks.
2. *Why* are they needed (knowledge and values).
   A. Inquiry: why.
   B. Deliverable: knowledge and values.
3. *How* they are to be performed (**information logic and scientific knowledge**).
   A. Inquiry: how.
   B. Deliverable:
      1. Information processing logic (what function).
      2. Scientific knowledge (what models of useful prediction).

Location of what is to be *the activity*:

1. *When* will the activity be executed (**time**).
   A. Inquiry: when.
   B. Deliverable: temporal information system (Read: schedule).
2. *Where* will the activity be executed (**space**).
   A. Inquiry: where.
   B. Deliverable: graphical information system (Read: GIS).
3. *How much* [resources] are required to be executed (**technical solution design inquiry**).
   A. Inquiry: how much.
   B. Deliverable: technical solution resource flow simulation (and data and logical processing model).
4. *How/what quality* [condition] will the activity be executed (condition-quality solution design inquiry).
   A. Inquiry: how/what quality.
   B. Deliverable: quality-function-deployment (QFD) combinatorial [decision to selection] synthesis.

INSIGHT: *Planning work allows for a sensible society.*

4.1 Environmental surveying

A.k.a., Project surveying, environmental surveying of humans and resources.

Surveying, which feeds into detailed planning directly returns information on what is required (what is the direction and what is accessible -- must have global cooperation as pre-requisite). Note that in the commercial market, a survey is not a direct input; it returns information on price (i.e., the price people are willing to pay in the market). The surveying specified here is primary abstraction surveying (i.e., categorized objective data) and not a survey of secondary order abstraction (i.e., categorized subjective-price data). Surveying feeds objective data into detailed planning (directly) by returning information on what is required.

In the market, a survey is not a direct input; it returns information on price (i.e., the price people are willing to pay in the market).

INSIGHT: *We have to keep track of what is about to happen so we can prepare, and we have to keep track of what has happened because an important part of what we are going to do next is in consideration through what we have just done. The past and the future flow into one another by keeping track of data and integrating it.*

5 [Project] Planning a plan of action

All projects contain one or more plans. A 'project plan' (a.k.a., plan-of-action) organizes and integrates every single bit of information (Read: "all details") that there is with any relevance to a project, combining them to produce information that can be taken action upon. A project plan is the data set integration of situationally relevant information required for the coordinated (as in, allowing for optimality) resolution of a societal, technical problem through action. The design and operation of any [complex] system is approached optimally through project organization, the who integration of which, becomes, a project [access] plan (i.e., a plan for deciding, coordinating, and resolving access to a real-world, socio-economic system). Plans exist to be executed. Herein, a plan is an information model (specification or work package) with the following characteristics:

1. Visually documented.
2. Executable through tasks, which have objective functions separated out at a high level into phases.
3. Related through integration modeling of all objects and processes (and states, stages, or phases).

In general, project plans have at least the following three characteristics:

1. The project ["management" or coordination] plan is an information system that contains all project related (subsidiary) plans (PMBoK 2018).
2. A project plan is an adaptive, iterative information system that gets updated every time something new is learned or otherwise discovered about the project.
3. A project ["management"] plan is a complete/final aggregation of complete planning (control and monitoring) done for a project.

A plan involves the information processes of:

1. **Understanding (knowing)** the meaning of objectives.
2. **Identifying (collecting)** assets.
3. **Analysing (thinking)** the consequences and risks.
4. **Establishing (thinking)** project performance and solution requirements.
5. **Designing (thinking)** a project solution. A **plan** is the result of thinking about the problem and the solution.
6. **Producing (doing)** a project solution.
7. **Delivering (giving)** a project solution.

A project plan is the sole formalized document, repository, interface, tool for project organization, execution and control.

1. The project plan describes and communicates the status (i.e., state) of the project to everyone concerned. As an information repository (or reference-base_, a project plan represents the formal database for all project related content. A project plan facilitates communication and optimizes effort expenditure between participating humans and technical systems.
2. A project plan documents the solution to a degree that the team can produce and deploy the solution effectively. A project plan is formalized in order to ensure coherent communication of the state of a project, which is necessary for complete and efficient project execution.
3. A project plan acts as a project's information control tool; it is the master planning and coordination referent for the project. Herein, 'project control' is the analytical process of comparing the real world progress with the scheduled/planned progress.
4. The project plan must correctly and accurately define the output as best as possible given what is known.

A plan is the result/deliverable of the planning process. In general, a plan is a represented determination of what tasks must be done, and which tasks precede others in order to accomplish some effort, work or created result. Plans focus and coordinate all effort. Simply, a plan is the "step-by-step" proceedings of a sequence of actions (tasks) to achieve a stated goal. A plan is analogous to a map; it maps out a "step-by-step" progression to completion of the some intention (objective, vision, etc.). The primary function of a plan, given a social organization, is to coordinate [social] effort.

More simply, a plan is a directional information set that everyone can see and point to, and say, "look here, this is what we are building, and this is when and how and with what we are building it (given that the language is understood)".

> **INSIGHT:** *If you don't make a plan then someone else is libel to make one for you.*

Effective plans cover all aspects of a project, giving everyone involved a common understanding of the information space and the work ahead. Plans must be kept up-to-date to be continually useful (i.e., they are a "living" information set).

A project plan if often considered a "visionary" piece of information (i.e., a visionary document), because it defines the vision and how the vision is to be achieved. A vision or goal is an end state (a description of). The project plan provides the [required] vision, as well as, how to realize the [complete] vision. The project plan allows participants to observe and control the flow of information through a project, from initial questions, to requirements definitions, to functional designs, and finally through to unit, interface, system, and user acceptance testing (or any similar integration lifecycle flow).

Summarily, a project plan defines the information elements of a solution in detail:

1. What is required?
2. How is it required?
3. Who requires it?
4. Who will build it?
5. When will it be built?
6. Where will it be built?
7. With what will it be built?
8. How will its building affect previous buildings?

In other words, a project plan of any level should be able to answer the four basic project questions:

1. **What?** What is the desired outputs/deliverables for the project? What work needs to be done in order to achieve these outputs?
2. **Why?** Why is the project being undertaken? What is it trying to resolve?
3. **Who?** Who are the team members working on the project? What does each individual do while on the project?
4. **How?** How will the project be completed? What activities must be completed, and in what order?

A project plan should [be designed to] visualize to everyone involved [in the project] the following information sets:

1. *Why* the project is being undertaken. The project plan includes the *why*, the reasoned imperative.
2. *What* will the project produce (i.e., the destination state of the habitat). The project plan includes the what, the described vision (or mission) as an information flow into a set of goals and objectives, which flow into a set of [engineering] requirements. Note, the execution of the project develops the specification, its construction, and potentially, its operation.
3. *How* and *when* will the project produce its intended result (i.e., the path to the destination). The project plan includes the *how* and *when* a project's objectives are to be achieved, by showing (in part) the deliverables, activities, resources, and schedule.

5.1 [Plan] Action structured view

MAXIM: *If you fail to plan, then you are planning to fail.*

In general, a complete project plan will include (most of) the following project content categories:

1. **The imperatives** - Goal-orientation, solution-orientation, problem-based orientation, a direction of orientation. An imperative (objective) is a stated intention of direction (or, direction of intention). The objective may be a mission, a vision, a goal, an end product, etc.)
2. **The approach** - the type of logic (or, not logic) to be applied to the flow of information.
3. **The work** - task information integration (task-based information "work" packages).
4. **The schedule** - time information integration.
5. **The tools and techniques** to be employed - procedural information.
6. **The people** - human InterSystem accountability.
7. **The resources** - common materiality.
8. **The risks** - the probability of harmful [human and ecological] consequence in materiality.

Table 2. Project Approach > Plan of Action: *Planning concepts and their alignment to core project instantiating elements.*

Plan concepts	Alignment to core instantiating elements
Requirement	Need, Value, User
Design	Solution, Need, Context
Plan	Change Solution, Context, User
Risk	Change, Value (reduced)
Benefit	Change, Value (increased), User, Context

5.2 [Plan] Action executable view

In order for a project to arrive at completion, it must integrate several information sets. The complete integration of these information sets is known as a life-cycle - it is the computation to completion of information given what is desired and what is known.

From an action/work coordination view, there are five information sets (or phases) to any given project (this is a recursive list, because 'execution' is a phase itself and part of every other phase):

1. **Breakdown work into tasks:**
   A. A task is another name for a processes with an input and an output.
2. **Identify resources:**
   A. The composition of the input, process, and output.
3. **Identify dependencies:**
   A. Tasks relate in requirements, input, process, and output; their relationships can be visualized through a database matrix.
4. **Schedule time and resource access:**
   A. The time and resource variables are added to determine temporal-location execution, and eventual completion.
5. **Execute:**

   A. Tasks are executed as actions/activities at the scheduled time and with the allocated resources. After evaluation, execution is the modifying of tasks.
6. **Evaluate:**
   A. Was the task executed correctly and did it have the intended impact or result.

Herein, the execution of an action has four principal project phases (note that this is a recursive process, and each phase also contains actions; for example, the identification of an activity to take action on requires actions itself):

1. *Identification* of activity.
2. *Preparation* of activity.
3. *Activity* action.
4. *Evaluation* of activity/action completion.

5.2.1 Simplified view of the project action life cycle

A simplified project action life-cycle involves detailed elaboration upon the following phases:

1. **Plan** - plan what needs to be done.
2. **Act** - take action to collect everything that is required for what need to be done.
3. **Do** - do what needs to be done.
4. **Check** - check to make sure what needed to be done has been done.

> **NOTE:** *This life-cycle is sometimes written as: (1) Plan, (2) Do, (3) Act, (4) Check. [the9000store.com]*

5.3 [Plan] Action decisioning tools

The following are information synthesis tools that have application in determining optimal planning decisions:

1. **Category diagram (identification)** - categorize as similar to an entity, such as group or label (entities as shapes but no relationships as lines)
   A. Affinity diagram - generate and group ideas.
2. **Relationship diagram (elaboration)** - categorize as entities with relationships (entities as shapes and relationships as lines)
   A. Activity decision program chart - identify potential problems and contingency measures.
   B. Activity network diagram - identify optimal path and schedule to complete work.
   C. Relationships diagram (interrelationship diagram) - map cause and effect links between items, events or tasks.
   D. **Tree diagram** - map tasks to achieve a goal in increasing detail. High level information is de-composed into lower-level information. An organization chart is an example. It graphically breaks down complex processes into smaller level details.
      1. An issue is known or being addressed in broad generalities and requires specific details.
      2. Developing actions to carry out a solution.
      3. Analysing processes in detail.
      4. To determine the root cause of a problem.
      5. To evaluate implementation issues for several potential solutions.
      6. After affinity diagram or relations diagram has uncovered an issue.
      7. As a communications tool to explain information to others.
3. **Combination diagram (comparison)**
   A. **Matrix diagram** - identify, analyze, and rate relationships between two or more sets of information. Shows the relationship between two, three, or more groups of information. Its completion will give information about the relationship (e.g., no, weak, strong, ...). Graphically establishes relationship between two or more sets of items in such a way as to provide logical connection points between each item.
      1. L-shape matrix diagram.
      2. T-shape matrix.
      3. Y-shape matrix.
      4. C-shape matrix [3D model, cube].
      5. X-shape matrix.
      6. Roof-shaped matrix (used with a L- or T-shape matrix, roof used with QFD) .
   B. **Prioritization matrix** - narrow down options by comparing them against criteria.

> **NOTE:** *Lines indicate links and lines with arrows represent a direction of [information flow in a] relationship.*

5.3.1 Quality function deployment (QFD) tool

A.k.a., Production relationship matrix for evaluation/assessment decisioning

The **quality-function-deployment (QFD) method (matrix)** is a method of combing the articulations of a users needs and expectations, while effectively accounting for the users by understanding their requirements, and then, developing engineering specifications to fulfill their requirements in an executed environment. The QFD method is used, in part, to determine optimal paths (synthesize a selectionable, optimal decision, given what is known). The QFD is a systematic method of translating the requirements of users into both the design and service (production & operation) process. QFD is a visual-logic (calculation) tool for ensuring user requirements are accurately translated into relevant technical specifications (from asset definition to asset design to process development and finally to asset-process implementation).

> **INSIGHT:** *Every organization has users (in the market, customers).*

Quality-function-deployment is used to translate user requirements into measurable design targets, and derive them down through the different compositional categories of an asset:

1. Assembly (of asset).
2. Sub-assembly (of assembly).
3. Components (of sub-assembly).
4. Production process (of components).

Multiple QFD matrices are used to translate this progression.

From a performance perspective, QFD could be viewed as:

1. Conception of performance (qualities).
2. Function of performance (functions).
3. Deployment of performance (deployments).
4. QFD is a decision interface for communication (concerning the engineering of a system).

6 [Project] Execution

> **READ:** *Execution of the plan of action, and the lists therein.*

A project's execution is the result of prior decisioning (and, decisions). Projects are executed under the conditions of a situation and of requirements for a situation to be different. Projects are executed as plans and lists. In actuality, plans are decided, and then executed as lists. Plans are solutions. Plans are an intermediary [information] deliverable of all projects. Plans are descriptions and explanations for current and/or future action. A plan is a documentation set that describes some set of actions and explanations, and how their execution (actualization of an action) will complete a set of [project] objectives. In part, plans are composed of lists. All lists to be executed (actualized, completed) during the execution phase of a project should be planned. Plans exist to be executed upon (i.e., used) in times of need. All plans are executed as a series of lists. That which is executed, is a reasoned plan-of-action likely to complete objectives. A plan-of-action includes project lists, which are the primary executable elements of any project.

To execute is to take action. Execution is a state of motion, a state of movement consciously energizing. In a sense, the project execution is the execution of a set of plans and lists to achieve a desired result. Execution is to take action (i.e., to go from) becoming (potential, design) into actual being (actualized, materialized). Execution done well ("right") is a planned and disciplined process that involves a logical set of connected activities acted upon by an organization to produce an expected result (to make work successful). The execution of any project requires lists and plans. This document details the project's lists and plans. The two most important plans for this project are: the contribution plan and the planned transition to the proposed society.

> **NOTE:** *The three common questions concerning the execution of a societal development project are: how does contribution work, what does justice do, and how do we transition to...?*

In concern to project execution and control, lists are a prerequisite. Lists are presented best as tables (matrices). In a database, tables store computable values. For purposes of execution, lists are an execution [coordination] tool. Relational tables can be computed (combined) by software as an information system. It is possible to operate a society without the price or violence mechanisms in that the information required to make the economy work can be performed by computer simulation, extrapolation, and calculation upon relational tables of project-relevant data so that the value and demand is represented within a software system.

To be effective, the execution of the plan:

1. Must include people and resources coming together to create better conditions, moving everyone, over time, into a community configuration of society.
2. Must use great leadership to generate and sustain community. Here, leadership involves:
   A. Stepping out to go first and take risk (to develop and promote this common direction).
   B. Rational, organizational, and socially relatable abilities and skills (that allow for peaceful integration and harmonious progress).

6.1 Project executable plans

The first project deliverable solidifying the project's execution is primary plan of action. In a societal project such as this, there are really three categories of plans, to be executed, each with their own, and interconnected, lists:

1. Local habitat [master] operations plans. There are many locally and regionally operating habitats. These habitats are master planned and carry out planned operational procedures in order to maintain their local habitat services.
2. Global habitat [master] operations plans. There is a global operating information system that facilitates information access and is used for decision support.
3. During transition from a non-community-type configuration of society to a community-type configuration, there is a societal transition plan. Note that this transition plan is different that local re-configuration and re-orientation plans for local habitats (who already exist within a community-type configuration of society).

In this way, there are really two plans, with different objectives and concerns:

1. The plan to operate a community-type society.
2. The plan to construction and transition people and resources into a community-type society.

In total, it is presumed that there are 4 execution plans, each with their own set of lists (some of which mind overlap with the other plans:

1. The community [master] plans.
   A. The standard development plan.
      1. This plan, and its accompanying set of lists, produce the community's societal specification standard.
   B. The local habitat operational plans.
      1. These plans, and their accompanying set of lists, produce the community's habitat services.
   C. The global habitat operational plan.
      1. This plan, and its accompanying set of lists, provides decisional and operational support to the global community network.
2. The transition [master] plan.
   A. The transition proposal plan.
      1. This plan, and its accompanying set of lists, provides for the transition of people and resources into a community configuration, and the successful duplication of that configuration.

6.1.1 Socio-technical materialization plans & planning

In the real world, plans are critical to long-term survival; without planning people tend to live day-to-day, always reacting to unforeseen threats, instead of seeing potential problems and avoiding them completely. This is especially true when there are not enough resources or contributions. Here, the primary concern is a lack of a desire, or of foresight, to take an interest in the plan (which exists regardless of interest, because humanity shares a common plan·et). The following is a list of the project sub-plan deliverables for a community-type societal project:

1. **Design plan** - conception information set. The societal specification standards.
2. **Construction plan** - materialization information set. The masterplan to construct the habitat (city) location. This is a project to construct a network of cities. All construction projects are monitored and controlled through a construction plan. A simplified construction plan may be summarized as follows:
   A. Concept design.
   B. Architecture and engineering design.
   C. Site selection.
   D. Materials and tools acquisition, and transport to and from site (a.k.a., resource collection, including tangibles and intangibles).
   E. Operational team formation (i.e., intersystem team to construct and operate the habitat service system).
   F. Site preparation.
   G. Main construction (phased delivery).
3. **Operations plan** - having a plan of procedures that knowing carried out is sustain the operation of the [production] system.
4. **Maintenance plan** - knowing when to maintain systems.
5. **Configuration plan** - knowing where and how to re-configure systems.
6. **Incident handling plan (operations continuity plan)**- know how to coordinate the occurrence of all types of incidents.

   A. **Disaster recovery plan** - knowing how to recover systems; continuity of operations.
7. **Market-State transition plan** - know how to communicate with entities in the market-State to sustain working relationships.
   A. Political communications strategy (a.k.a., State communications strategy, State relationship plan).
   B. Market communications strategy (a.k.a., business plan, market relationship plan.
   C. Public communications strategy (a.k.a., social/ crowd communication plan).

6.1.2 Project coordination plans & planning

I.e., What are the plannable elements of a project plan?

These plans describe how the project will be coordinated, monitored and controlled throughout the project lifecycle:

1. **Project charter (project definition plan)** - the planned instantiation of a project. The project charter outlines the projects purposeful direction; it is essentially a memorandum of understanding (that is agreed upon).
2. **Communication coordination plan** - the planned protocols (synchronization and acknowledgement) and platforms by which information is understood and used.
3. **Document coordination plan** - the planned publication and dissemination of standard references for usable information.
4. **Schedule coordination plan (time team planning)** - the planned positioning of team elements in time.
5. **Resource coordination plan (object and operation planning)** - the planned positioning and occupation of resources.
6. **Issue coordination plan (change control planning)** - the planned decisioning of issues.
7. **Risk coordination plan (challenge response planning)** - the planned [mitigation & incident] response to negative events.
8. **Human coordination plan (human team planning)** - the planned positioning of individual humans into an organization of InterSystem teams and working groups who accountably complete tasks to sustain and adapt the operation of society.

6.2 Project executable lists

A.k.a., Positive lists, accountable lists, accountabilities.

A project list is a repository of all listable elements relevant to the execution (running, coordination) of a project. Whatever a project is composed of, it can be added to a [project-relevant] list. Lists contain data accessible for execution, which may be software, hardware, or human, or some combination thereof. A list is any information displayed or organized in a logical or linear formation, which is necessary for the coordinated execution of any task.

To take action requires the synchronous integration of a set of project plan lists. There are two categories of list, a list that includes certain information traceable to requirements, and a list that includes uncertain information traceable to risks (detriments to the project). The execution of a plan involves the combining or positive project lists along a timeline (schedule), whereupon risks are mitigated and responded to through reasonable controls. The execution of a societal-level project is complex and multivariate. Human flourishing can be resolved for by applying effort toward the combined resolution (actionable integration) of a set of directional (positive) lists. In the market, these lists represent exchanges of property/ownership. In the State, these lists represent hierarchical relationships of one person having power [of coercion] over another. In order to sustain a fulfillment-oriented society, relationships must be sustained that meet the society's minimum level of informational and spatial requirements.

In terms of computation, which is a necessary component of the execution of a complex socio-technical system, it is useful to understand a list as a data structure that generalizes one or more atomic vectors. An atomic vector is the simplest directional data type. Data without a vector (i.e., scalar values; data without useful decisional information) can be vectorized through operations. Each sub-system of a total societal system has a different set of interrelated "atomic" vectors:

1. In a social system, a 'need' and 'value' (condition), together, are the simplest directional data type (i.e., is an atomic vector). Values are orientationally usable data packets with an identifiable vector (meaningful direction, need). Data organized for meaningful fulfillment [if need] has an atomic vector.
   A. Needing is a human process, conditions allow for needs to get cyclically met.
2. In a decision system, an 'objective' (claim, *requirement*) is the simplest directional data type (i.e., is an atomic vector). Objectives are measurable outcomes. Action taken on the part of objectives has an atomic vector.
   A. Operationalizing values is the process of identifying human-community values and translating them to accessible and concrete concepts so that they can be implemented, validated, verified, and measured as progress and/or completion of a need. An operationalized value is an 'objective'.

3. In a material system, an 'object' (matter, *technology*) is the simplest directional data type (i.e., is an atomic vector). Objects have shape. The motion of objects has an atomic vector.
4. In a lifestyle system, an 'organism' (life, *feeling*) is the simplest directional data type (i.e., is an atomic vector). Life has consciousness. The experience of consciousness has an atomic vector.

A project is necessarily composed of the following executional list elements (components, parts):

1. **The lists** - The execution of a direction as a set of lists that account for all project relevant data.
2. **The meta-relational database** - The descriptive meaning of each list, and all lists in relation to one another.

6.2.1 The project lists

A.k.a., Project execution list data.

In order complete a project, a project plan must identify and relate the following lists/tables (within a project coordination database), upon which calculation can be done:

1. **Agreements list (a.k.a., alignment checklist)** - the list of what is being agreed to under the initialization of a project.
2. **Needs list (a.k.a., issues list)** - The list of what is needed for the user to have complete fulfillment.
3. **Objectives list (requirement-oriented conditions breakdown) -** An objective/requirement is a capability to which a project outcome (product or service) conforms to a measurable degree.
   A. **Situational analysis** - is the complete context of the real-world situation in which the objectives exist.
4. **Concerns list (a.k.a., risk, incident, problem, issue-oriented breakdown)** - Each concern is either a risk or an issue, which are handled in much the same way via a decisioning prioritization and solution process. Some issues of concerns are incidents (actualized risks). This is a list of issues concerning organizations and events that have been/may/or are adverse [in their effects] to the completion of the project (i.e., "threats"). Here, the issue is either a risk (with some likelihood of), or an incident (current affect of), inhibiting project completion. Incidents require resolution (hence, new actions/tasks to resolve the incident), and risks necessitate mitigation reasoning for project preservation planning. Issues are prioritized (as in, 'triaged'). In general, issues themselves are not scheduled, although their resolutions may be. A planned "issue" is either a test or a trap.
5. **Requirements list** - Each requirement that must be checked in order for a project objective to be complete.
6. **Resources and locations list (location-oriented breakdown)** - Each available resource at some location in time. The list of locations of everything is -- material and digital [resource] locations. Note that resources can be moved to re-located them over time, and this relocation can be scheduled.
7. **Deliverables list (product/service-oriented breakdown)** - Deliverables are requirements packaged with contextual information into the form of products and services (as outputs of processes) required to complete the project. Note: There are project deliverables (project needs/requirements), and sub-project deliverables (sub-project needs/requirements). The deliverables list is a list of the outputs (of processes) that must be completed ("ticked off" as done) for the project to be complete and/or meet objectives. There are two possible categories of deliverable: process (service) deliverables and object (technology) deliverables.
   A. **Plans** - The documentation set for identification of actions.
8. **Actions list (action/task/work/deliverable-oriented breakdown)** - The list of all tasks (actions, activities, etc.), all of which are tracked. Some tasks exist to resolve concerns. Actions (activities/work packages) are executable [process or construction] tasks. The items in this list are tasks within a hierarchical structure of textual groupings (a work breakdown structure, WBS). Synonyms for 'action' include, but are not limited to: work, task, activity, executable, "something to do", process, procedure, construction, and resolution. Actions are assigned to systems and/or people. Some actions are automated. Automated actions form automated services - services without the need for direct human effort, no 'event' instantiation (no addition to the Events List). *Note: A project produces a product and/or a service, and so, that is why this type of plan, is called a "plan of action"); because, it intends to describe the act of brining something into existence.*
9. **Human user list (a.k.a., stakeholders)** - The list of those who are going to use and/or benefit from the project's deliverable(s).
10. **Humans contributions list (a.k.a., workers)** - The list of who is contributing, and where and when and with what.
    A. **Contribution accountability list (people/actor-oriented breakdown)** - profile and activity information on every human in the project,

including all their associated project and sub-project information, resource allocations, and roles/responsibilities.

11. **Teams list (a.k.a., roles list)** - The individuals and machines that carry out activities.
    A. The human work package as a 'role' with a "work description" - human placement on a team.
    B. The human work package as - the human selection of tasks as part of a team.
12. **Schedule list (time-oriented breakdown)** - The items in this list are Tasks within a hierarchical structure of groupings called the WBS (Work Breakdown Structure). The temporal association as an activity. In order for action to occur (i.e., "things to happen"), there is time. Actions, deliverables, requirements and events can be organized within time (i.e., they can be scheduled and time delineated). These project information categories can be expressed in terms of a time (i.e., iteration) dimension. A schedule list may also be known be the following labels: timeline, gantt chart, or project schedule. A schedule can be a unified visualization of all (or selected) actions/work, deliverables, requirements, and events per [unit of] time, with all associated meta-/calculable-information. Through the scheduling of accountability project coordination can be calculated and visualized; wherein, it is possible to view: system and human bandwidth; *who's* available; and *who's* busy.
13. **Events list (timeline breakdown)** - A transparent recorded ledger of any change that has occurred. A recorded event always identifies the 'result' of an interaction (e.g., minutes of meeting, a report, a computational result) in the real world.
14. On-going and post-execution lists:
    A. Operations (monitoring and procedures) lists.
    B. Evaluations (feedback) lists.
    C. Integrations (updating information systems) lists.

Simply, the following is the list of executable elements (a.k.a., the project lists, for which there is order to the their positioned instantiation):

1. **Agreements list** - what is being agreed to (and, why is it being agreed to)?
2. **Concerns (risks) list** - what are the concerns (and, why are they concerns)?
3. **Needs list** - what is needed (and, why is it needed)?
4. **Objectives list** - what are the objectives (and, what are their priorities)?
5. **Requirements list** - what is required to complete objectives?
6. **Resources list** - what resources are available for what is required?
7. **Team & role list** - what work [descriptions] are being contributed (per team-role)?
8. **Deliverables list** - what is to be produced (and for whom)?
9. **Decisions list** - what must be decided (now that all else is known)?
10. **Actions list** - what must be done to complete the project?
11. **Schedules list** - when, where, and by whom are actions to be taken?
12. **Deadlines list** - by when must be actions taken?
13. **Events list** - what has been done (and reported analysis on why it was done)?
14. **Costs list** - in the market there is a also a financial costs list.

7 [Project] Life-cycle

A.k.a., Project lifecycle, project life cycle, project process groups.

A project is sub-divided into a set of phases (resolving information sets) as sub-parts of a life-cycle plan for coordination and decisioning (control) purposes. The [project] life-cycle plan forms the foundation for project planning, scheduling, coordination, and estimation.

NOTE: *Different types of projects may have different life-cycle structures.*

All projects have a life-cycle, which involves the division of the project into phases. Through this method of life-cycle modeling, it is possible to plan a whole global habitat network. Life-cycle modeling facilitates the planning and coordination of all project progress. Everything that should be done to accomplish a project is divided into distinct phases, separated by control gates. Phase boundaries are defined at natural points for project progress assessment and *go/no go* decisions (i.e., should the project continue to the next phase, or not)?. Decomposition of a project into life cycle phases organizes the development process into smaller, more ordered ("manageable") pieces ("chunks").

To be a complete life-cycle, a life-cycle (and its engineered design) must cover the entire system's life as a cycle; from conception, extraction to closure, usage and re-cycling (from conception to closure). With increasing project complexity, validation and verification (V&V) becomes increasingly important; the standard should provide a detailed view of the verification and validation (V&V) processes (or the like).

7.1 [Plan] Life-cycle **control**

A.k.a., Planning control, plan control, plan management, plan programming, plan[ned] decisioning, controlling coordination.

In the discipline known as project coordination (project management), when the words 'plan' (and 'control' or 'management') come together, only the knowledge areas change. For example, Plan quality control/management, Plan schedule control/management. The controlling of coordination [as intentional motion in a physical environment] for the purpose of navigation, in time-space, requires 'planning' as the intentional conception and expectation (of a particular sensation, giving rise to a memory of [the unit] experience).

All projects are controlled by:

1. Deciding (approving) who to include in the project.
2. Deciding (approving) final decisions for execution within and without the project.

In documentation, every time the terms 'plan' and 'control'/'management' are together in a project coordination/management title, it means, "the rules or procedures of the [information flow] gating and monitoring process" (i.e., the rules of the gate that allow information to pass). The control of the flow of information, and of all access, can be sub-divided into bounded phases, for easier and more model-like understanding, known as a 'life-cycle' (Read: a whole unified system sub-divided into interrelated boundaries that form a [whole life-]loop).

In a project, these rules are defined [within the unified information system] ahead of the project [phase or sub-process] gate. For example, the term "Plan scope management" means the rules for how to process information associated with scope as, the defined direction, which includes the process group decision-deliverables of: defining objectives, collecting requirements, and producing a "work breakdown structure".

TERMINOLOGY: *A 'rule' is (in part) a pre-decided flow of information from one point to another by the method of a [controlled] relationship that links to one entity out of multiple possible entities.*

In the process group known as **CONTROL**, when plan and coordination come together, it means the logical resolution of information into a decision point to be acted upon in the future by an accountable (i.e., monitored) entity, who understands the plan (decision structure/procedure) and is able to act). More completely, project coordination "management" is about information-level control and communications under more or less well defined information categorizations and processing goals.

INSIGHT: *Instead of an environment where relationships are based on [market] transaction and power-over-others (i.e., the State-owner-authority), relationships are based on collaboration, as a global cooperation of thought, resolution, and action in a common environment.*

The center (Read: core process) of each process [group] is 'integration'. Integration combines the other 9 knowledge areas into a fully specified understanding of some knowledge area of a project. All terms that start with Control are sub-sets with Integration.

In other words, project integration involves controlling how information is integrated into a decided project plan knowledge-base. Project integration involves:

1. **Initializing project-level information [sets]** - Identifying the issue and the users (as in, those who may be impacted by the issue).
2. **Analyzing project-level information [sets]** - Collecting and analyzing the project data on the

results achieved by the project, ensuring the project meets the project objectives, by constantly monitoring the project's progress.

3. **Delivering project-level information [sets]** - delivering a project plan (supra-plan) through a project information interface that will facilitate the optimal resolution of project objectives.
4. **Closing project-level information [sets]** - doing all the required work at the project-level to meet the requirements, and then closing (exiting, no longer working on) the active project process.

7.1.1 Control gates

A.k.a., The process group gates, control gate.

Each phase in the life-cycle represents a gate in the whole life-cycle process. Each gate in the project life-cycle, at a high-level, is called a 'process group'. Through the gating process, a project (or any deliverable) is broken down into smaller stages or phases, each delimited by a gate, which has a rule-set, wherein information is executed, leading to a decision to pass or not pass the gate. Each gate is a control point where verification that the necessary prior steps (and deliverables) have been completed. At each of these gates, the project requires decision determinations, deliverables, based on specific criteria and the information available at the time, whether to continue, stop, hold, recycle or modify the project/deliverable. To each of these gates corresponds one or several decisions/deliverables.

7.2 [Plan] Life-cycle monitoring

A.k.a., Checking accurate alignment, gating accuracy.

To monitor is to perceive, or not, a quantitative (behavioral) or qualitative presence.

IMPORTANT: *Until a* **measurement** *[of presence] is taken, there is only* **potential** *[for presence].*

Monitoring (which necessitates analysis) is done to meet information needs. Considering the level of abstraction a calculable concept can be composed of other sub-concepts, which could be represented by a concept model (e.g. ISO 9126-1 specifies a quality model based on characteristics and sub-characteristics). A calculable concept is associated to one or more attributes of entities. An entity is a tangible or intangible object that is characterised by measuring its attributes. Types of entities of interest to system engineering are: Project, Asset, Service, Process, and Resource. The attribute is a measurable physical or abstract property of an entity. An entity may have many attributes; only some of them may be of interest for a given calculable concept. For a given attribute, there is always at least an relationship of interest that can be captured and represented in the formal domain by means of a metric, enabling us to explore the relationship mathematically and/or statistically. The metric contains the information of the defined measurement (and/or calculation) method an scale. An attribute may be measured using different measurement methods and scales, hence one or more metrics can quantify the same attribute. (The reader can see the method definition and derived concepts likewise the scale and unit concepts in table 1, and the scale Type attribute).

7.3 [Plan] Life-cycle information sets

In order to fully describe the flow of information within a project-based structure, it is necessary to have a top-level sub-division of information flow known as a life-cycle of different phases known as 'process groups' -- the axiomamtic, divisional categories of information processing required to complete the project. These process groups are, as the category name describes, groups of processes. In order to complete any given process, there must be knowledge:

1. Knowledge about the process itself, and the environment in which it is operating, in order to effectively and efficiently execute and correct the process for a given intention/objective (for change).

Thus, a complete project coordination process flow involves the following information sets project-level [working] information sets:

1. **Process group** (5 total phases) - plan coordination life-cycle.
    A. **Knowledge area** (associated with process groups; 9 total) - the knowledge of how to plan coordination.
        1. **Project processes** (47 total activities)
            i. ITTO (input, tools and techniques, outputs) unique for each process. Some inputs are used for multiple processes. The number of ITTO associated with each project process is proportional to its prerequisites. The inputs, tools, techniques, and outputs when coordinating.

7.4 [Plan] Life-cycle data inputs

A.k.a., Project areas.

For any given project there are four core data [area] inputs include:

1. Project knowledge areas.
2. Life-cycle knowledge areas.
3. Life-cycle process areas.
4. Project process areas.

Core project knowledge areas (data inputs):

1. Integration.
2. Scope (issue & goal).
3. Schedule (time).
4. Quality (& quantifiable evidence).
5. Risk (incidents).
6. Resources.
7. Stakeholders.
8. Communications.
9. Cost (the unnecessary factor).
10. Procurement and disposal (market).

Core life-cycle knowledge areas (data inputs):

1. Integration (system).
2. Research (science).
3. Development (support).
4. Assembly-Operation-Disassembly (service).

Core life-cycle processes (data inputs):

1. Initiate project (initiating) - goal setting requirements.
2. Design system (designing) - modeling solution to requirements.
3. Build system (building) - assemble system.
4. Use system (using) - operate service system
5. Cycle system (cycling).
6. Observe > Analyze > Design > Build > Use > Cycle.

Core project processes (data inputs):

1. Initiating (intentional objective, directive issue).
2. Planning.
3. Decisioning.
4. Executing.
5. Controlling and Monitoring.
6. Closing.

7.5 [Plan] Life-cycle **phasing processes**

A.k.a., Plan coordination, plan phasing, life-cycle phasing, life-cycle coordination, project progression, project management life-cycle.

An iterative, cyclic flow of information known as a life-cycle (or lifecycle) bounds the organization of a project, and coordinates the project's forward progression toward resolution/completion. Each sub-organization in this flow of information is a gated process (Read: process group set), known most commonly as a 'phase' or 'stage' in the totality of processes known as its 'life'[-cycle]. The collection of these phases at the information-level is the project lifecycle. And, the collection of these information gating phases at the operation-level is the engineering lifecycle (a.k.a. solution inquiry). These two 'categorically' separate lifecycle cycles are different views (windows) into the same sub-section of the unified information space representing that of the project's direction. In an information systems context, all phases are best viewed as categories of information in a unified information system, essential to the effective resolution of the complex social-project space.

> **LANGUAGE:** *The term lifecycle, life-cycle, and life cycle mean the same thing (i.e., are used interchangeably).*

A project life cycle is the series of 'phases' that a project passes through from its initiation to its closure. A 'phase' is a set of activities that culminates in the completion of one or more deliverables (PMI 2017).

More technically, a 'phase' ('stage') is an invariantly sequenced, qualitatively distinct level that can meaningfully characterize process sequences of abilities. In other words, by measuring a system as it moves through a life cycle, at each of the stages/phases in the life-cycle it is possible to state that there has been a meaningful change, and that change has come (in part) from a technical ability within the life-cycling system.

Project [coordination] phasing is the process of dividing and sub-dividing a project into a number of logically related phases (and related information sets) that must result in completion of the project's associated deliverables (informational and material). At a high-level, project phasing could be considered the project methodology, as in, the study and reasoned selection of a method by which to complete a project.

Project phasing produces the high-level representation of steps (phases) for project fulfilment and show objectives for each of the steps with durations and priorities. Typically the project phases are combined with the 'time' factor to compose a visual coordination tool known as a 'schedule'.

The process of subdividing a project into phases involves the following two intentional [design] processes, informed via project objectives:

1. **Identification** - The first process ("step") is to identify which phases, sub-phases or/and sub-projects will be required for completing the overall project [life-cycle]. The identification should be based on the objectives and expectations stated by the project imperative. Time becomes the critical factor that determines the phasing so schedules (a timing tool) are used to identify the relative positioning of phases. All the objectives are divided into groups considering expected delivery time for each of the objectives. Then for every group the primary goal is to be determined. The goal combines and aggregates all the objectives included in each given group. In such a way project planners can group the objectives by delivery time and therefore divide the entire project implementation life-cycle into certain phases.

2. **Prioritization**. This process ("step") associates priority activities for each of the identified phases. The relative ranking or priorities for the phases should be based on the extent to which every phase carries out a specific objective. Often priority activities are set up by defining the critical path for all the objectives of the identified phases. Practically: an objective with the longest duration in every phase is investigated and selected; such objectives are compared with each other and organized by durations (from shortest to longest). Then priorities are set up for the phases.

7.5.1 [Project] Life-cycle **process groups**

A.k.a., Project phases, project deliverables, project process groups.

The most common project phasing (i.e., life-cycle) is the five project supra-processes (i.e., project process groups). A typical project has the following five major phases, also known as the five project process groups (and, each process group has its own set of information sub-components).

CLARIFICATION: *Coordinating a project usually requires dividing the project's work into more "manageable" pieces called phases. Phases allow the project team to more effectively coordinate and control project activities throughout the life of the project. Collectively, these phases are called the project life-cycle.*

The structure of the following project-specific view of the lifecycle phases is:

1. Life cycle phases (process groups).
   A. Sub-process group processes.

The project-specific deliverable-view of the process groups (phases) are as follows:

1. **Initiation** *(project/phase/process initiation):*
   A. Define initial imperatives.
      1. Develop project charter.
   B. Develop stakeholder registry.
   C. Generate initial plan.
2. **Planning*** *(project/phase/process planning):*
   A. Determine *where*, *when*, and *with what.*
   B. Decide *selection of planned solution.*
   C. Survey (resources and humans).
   D. Identify and prioritize action .
   E. Establishing action performance requirements vis the selection of metrics, used to monitor and assess downstream activities.
   F. Documents that bound scope (what we are and are not doing).
   G. Documents that list detailed requirements.
   H. Documents that provide estimates for cost and time.
   I. Documents that provide for a schedule.
   J. Documents that plan for quality, communications, risk and procurement.
3. **Execution** *(project/phase/process execution):*
   A. Discover > Design > Development > Operate > Evaluate.
   B. Execute the plan through doing/action as directed in the plan.
   C. Determine *what* and *why*.
   D. Build *what*, *where*, *when*, and *with.*
   E. Operate *what, where, when, and with.*
4. **Closing** *(project/phase/process closing):*
   A. Close project or phase (closeout).
5. **Monitoring and controlling** *(project/phase/process M&C; Integration):*
   A. Testing and validation.
   B. Protocols.
   C. Repositories.

**Clarification: There is sometimes confusion concerning 'planning' and 'lifecycle'. Planning is a continuous [project process] group/activity. Planning is a phase specific process group; one that is continuously active while the project is active. It is a continuous phase in project lifecycle.*

Project coordination involves the following domains of information processing, which interrelate:

1. **Initiating (1st phase)** - Instantiation of a project occurs through an imperative or other directional statement. Imperative and/or directional statements include, but are not limited to the following: purpose, needs, goals, objectives. Imperatives denote a direction (with which to align) or outcome (as a condition and conclusion). An imperative necessitates further action, and the application of a structure with which to resolve the imperative. Defining a new project or a new phase of an existing project by obtaining authorization to start it.
   A. The activities performed to define a new project or a new phase of an existing project.
2. **Planning (2nd phase)** - Establishing the scope of the project and defining the objectives and the course of action required to reach the objectives. The planning phase itself focuses on developing sufficient details to allow various project elements coordinate their work optimally.
   A. The activities performed in order to establish the total scope of the project, define and refine the objectives, and develop the course of action

that will be followed to achieve the objectives.

3. **Executing (3rd phase)** - Completing the work defined in the project management and planning to satisfy the project specifications. Execution refers to the completion of informational and psychical work; wherein, work is packaged, distributed and selected, and then, completed.
   A. The activities performed to carry out and complete the work as defined in the project plan. Executing activities includes coordinating people and resources and performing and integrating the activities as specified in the project plan.
4. **Closing (4th Phase)** - Finalizing all activities across all Process Groups to formally close the project (closeout).
   A. The activities performed to finalize the project – to bring it to a conclusion and to meet contractual obligations.
5. **Monitoring and Controlling** - While the other process groups occur sequentially (generally), Monitoring and Controlling hover over the whole project (i.e., happens throughout the project and is not linear). Reviewing and regulating the progress of the project; identifying any areas in which changes to the plan have to be made and initiating the corresponding changes.
   A. The activities performed to track, review, and regulate the execution of the project; identify any areas in which changes to the plan are required; and initiate corresponding changes.
   B. The tools for monitoring and controlling a project include but are not limited to:
      1. **Cause-and-effect diagram** (a.k.a., fishbone diagram) - The causes are found by looking at the problem statement and asking “why” until the actionable root cause has been identified or until the reasonable possibilities on each fishbone have been exhausted.
      2. **Control charts** - Control charts measure the results of processes over time and display the results in a graphical form. These charts are a way to determine whether process variances are in or out of control. A control chart is based on sample variance measurements.
      3. **Histogram** - Histogram is used for illustrating the relationship in the context of two variables. Histograms are typically bar charts that depict the distribution of variables over time.
      4. **Flowchart** - Flow charts are used to understand complex processes in order to find the relationships and dependencies between events. Flowcharts are diagrams that show the logical steps that must be performed in order to accomplish an objective. They can also show how the individual elements of a system interrelate. Flowcharting can help identify where quality problems might occur on the project and how problems happen.
      5. **Checksheets (criteria sheets)**- A check sheet is basically used for gathering and organizing data.
      6. **Scatter diagram** - Scatter diagrams use two variables; one is called an independent variable, the input, and other dependent variable, which is an output. Scatter diagrams display the relationship between these two elements as points on a graph. This relationship is typically analyzed to prove or disprove cause-and-effect relationships.

The core deliverables of a project, separated by [project process] phase, are:

1. **Initiating** - representation of human opportunity, human direction; an issue and/or proposal.
   A. Project charter (a.k.a. statement of work, proposal, estimate response document, complete memorandum of understanding, etc.).
   B. A proposal is a data set of understanding and planned/-able action. Proposals are presented to populations in order to facilitate awareness and alignment, to bring forth and define future action. All solutions are proposals.
2. **Planning** (a.k.a., strategizing, strategy).
   A. Project plans.
      1. Deliverable diagram.
      2. Communications plan - what information needs to be communicated to what person.
      3. Schedule.
      4. Decision coordination (a.k.a., change control) - how do decisions come in, how are they assessed, what or who takes the decision.
      5. Cost management (market only).
      6. Procurement management (market only).
3. **Executing and Controlling** (a.k.a., plan implementation).
   A. Performance reports.
   B. Ongoing issues.
   C. Change logs.
   D. Project progress.
   E. Deliverables (delivered?).
4. **Closing** - the project, itself as a deliverable, is completely delivered.
   A. Acceptance.
   B. Final report.
   C. Documented.

D. Learned.

Take note that sometimes the following combination of information elements is referred to as strategizing (or, strategic thinking):

1. Purpose.
2. Values.
3. Objectives.
4. Metrics.
5. Goals.
6. Capability and capacity.
7. Plan.
8. Action.

The phases (process groups) are expressed below with their sub-processes:

1. **Initiating process group.**
   A. Develop project imperatives (project charter).
   B. Identify stakeholders.
2. **Planning process group.**
   A. Develop project plan.
   B. Identify requirements.
   C. Develop work breakdown structure.
   D. Define activities.
   E. Sequence activities.
   F. Estimate activity resources.
   G. Estimate activity duration.
   H. Develop schedule.
   I. Estimate cost (market).
   J. Determine budget (market).
3. **Executing process group.**
   A. Coordinate project execution (track all project information).
   B. Perform quality assurance.
   C. Acquire project team.
   D. Distribute information.
   E. Conduct procurement (market).
4. **Monitoring and controlling process group.**
   A. Monitor and measure project work.
   B. Report performance.
5. **Closing process group.**
   A. Close project or phase.

CLARIFICATION: *Not all project have a closure -- not all projects have a specified end or end date. Some projects produce services with their own life cycles, and these services may still be managed as projects.*

7.5.1.1 [Project] Life-cycle coordination process phases simplified

Each similar collection of project information processes are called project process groups (PPG, in PMBoK) -- each process group is a phase of the whole common project life cycle. For any given project, all process group processes could be active at any stage.

Every project lifecycle has at least the following three ordered, principal processes (a.k.a, supra-processes, process groups, lifecycle phases):

1. **Initiating** process group.
2. **Phase specific** process group.
3. **Closing** process group.

There are two important points to take note of in concern to a project's principal processes:

1. Note that the processes (i.e., process groups) do not happen only once. They happen at every cycle of phase. Of course that the first time you pass on the process you create the document but in the following ones (other project phases you use what you created to improve the other process). The process does not occur only one time.
2. Note that a project's lifecycle processes are recursive, because each phase of the project's lifecyle itself needs to initiated and closed with processes (Read: the process groups known as 'initiating' and 'closing').
3. Note that different types of projects go through different stages before the result becomes life (or a part of the real world, the extant life cycle).

7.5.2 [Project] Life-cycle **knowledge areas**

The knowledge areas necessary for performing project coordination are:

1. **Scope** coordination.
2. **Time** coordination.
3. **Quality** coordination.
   A. Scope, Time, and Quality = the three triples constraints:
      1. *Scope* - an objective given an environment.
      2. *Time* - schedule.
      3. *Quality* - of resource.
4. **Risk** coordination.
5. **Communication** coordination.
6. **Procurement** coordination.
7. **Cost** [market] coordination.
   A. The *market (competition)* has externality costs.
8. **Human resource** [**contribution**] coordination.
   Integration [processing] coordination.
9. **Stakeholder** [**operating users**] coordination.

Concerning the timing of process groups and the integration of knowledge, the 10 knowledge areas can be executed (as information sets) concurrently (PMBoK 2018) within a project's phases (e.g., initiating, planning, executing, etc). All the knowledge areas will not begin and end at the same time; they are all independent: integration, scope, schedule, cost, resources, stakeholder, procurement, risk, quality, and communication (or any other composition) can be executed in parallel in time.

7.5.3 [Project] Life-cycle **inputs, tools & techniques (as activities), and outputs (ITTO)**

Each process contains a set of knowledge areas, each with the following information set structure (abbreviated ITTO):

1. **Inputs (pre-requisites)** - that which is necessary to start the process.
2. **Techniques and Tools (procedures, methods, mechanisms)** - the type and level of effort necessary to do the process.
3. **Outputs (deliverables)** - one or more of that which results from the process.

Each phase of a project's life cycle is composed of the following input categories:

1. **Resource** life-cycles (materials)
2. **People** voluntarily contributing effort (contribution)
3. The **application** of tools, techniques, and knowledge in the form of an action, activity, event, task, etc. (the executed process).
4. An **intended result** (the outcome)
5. The **actual result** (the evaluation)
6. *Currency* and *authority costs* (market-State only)

More completely, each knowledge area contains a set of ITTO.

1. **Inputs** - Any item, whether internal or external to the project that is required by a process before that process proceeds. May be an output from a predecessor process.
   - For example, plans, specifications, permits, financing, building materials, etc.
   - For example: project charter, project schedule, resource calendars, organizational process assets.
2. **Tools and techniques (for construction)** - skilled labor, concrete, framing, electrical, plumbing,
   - Tools - Something tangible, such as a template or software program, used in performing an activity to produce a product or result.
     - For example: Analytical techniques, modeling, project management information system, benchmarking, product analysis.
   - Techniques - a defined systematic procedure employed by a human resource to perform an activity to produce a product or result or deliver a service, and that may employ one or more tools.
     - For example, meetings, expert judgment, inspection, interviews, decomposition.

   Diagrams,
3. **Output** - A product, result, or service generated by a process. May be an input to a successor process.
   - For example, the finished product or service, work performance information, project plan updates, organizational process assets updates, project document updates.

In the PMBOK's, ITTO knowledge base is a standardized means of systematically using the same method of developing and executing processes and projects (i.e., the same methodological knowledge). Decomposing processes into systems (i.e., ITTO) reduces each to its most fundamental and basic [system-based] components, and does so in a standardized manner that is equally applicable for all processes and projects.

7.6 *[Project] Plan life-cycle* **coordination process**

The following is the complete project process flow (life-cycle), formatted into processes and their associated knowledge areas:

1. **INITIATING PROCESS** - The initiating process details 'What' the project is about.
   A. **Integration** (4.1 chapter of PMBOK 6th) knowledge - start initiation by integration.
      1. [Develop] Project charter (a project exists, the projects intention and why it exists)
   B. **Stakeholders** knowledge (13.1 of PMBOK)
      1. [Identify] Stakeholders - any person or entity that has any kind of interest in the project (positive or negative interest)
2. **PLANNING PROCESS** - to ensure that the plan will satisfy the stakeholders and deliver the project results. All of the below is part of the planning process.
   A. **Integration** knowledge (4.2) - start Planning by integration.
      1. [Develop] Project management plan
      2. All next content should be indented, but I don't want to do that
      3. At the end of the planning process, everything is consolidated into the project management plan.
      4. The project management plan details 'How' the project has been planned.
   B. **Scope** knowledge (5.1)
      1. Plan scope management - the rules for how you process information associated with scope
   C. **Scope** knowledge (5.2)
      1. Collect requirements (functional, technical, and activities as parts of the work) - things

that need to be done to satisfy Charter and Stakeholders

D. **Scope** knowledge (5.3)
   1. Define Scope - documented scope statement that reflects the scope of the project. Defines how you want to approach the project.

E. **Scope** knowledge (5.4)
   1. Create WBS

F. **Schedule** knowledge (6.1)
   1. Plan schedule management

G. **Schedule** knowledge (6.2)
   1. Define activities - define activities that must be accomplished to deliver the work package on the WBS.

H. **Schedule** knowledge (6.3)
   1. Sequence activities

I. **Schedule** (6.4)
   1. Estimate activity durations

J. **Schedule** (6.5)
   1. Develop schedule - a visualization of how the project will be placed over time. (gantt chart, network diagram, etc). The schedule will provide information on how much time (as a resource) is likely required to complete the project.

K. **Cost** [*Market] (7.1)
   1. Plan cost management - Who has approval to "spend" money? Costs are intimately related to resources and time.

L. **Cost** [*Market] (7.2)
   1. Estimate costs - if you know the activities and have a clear scope, then costs can be estimated.

M. **Cost** [*Market] (7.3)
   1. Develop budget - how and when the spender will spend the money, s-curve. Note, the term 'enterprise resource planning' (ERP) is another term for cost budgeting, in general, the "resource" in ERP is that of financial cost in the market. These planning platforms often include the following modules: sales; purchasing; extracting and manufacturing; inventory management; distribution; accounting/finance; human resources; and, customer relationship management (a.k.a., customer services).

N. **Quality** (8.1)
   1. Plan quality management - what are the quality standards that must be complied? What is expected to be delivered in terms of quality? This is the decision system's non-functional requirement inputs for integration into the extant community system.
   2. Here, the expected [standard] quality is set.

O. **Resource** (9.1) (*6th, in past editions was only humans, no longer just humans)
   1. Plan resource management - rules of the game of how you plan to manage the resources.
      i. Do you have [access to] the resources, in what state, where?
         1. Do you need to discover or extract resources, in what state, where, how?

P. **Resource** (9.2) (*6th, in past editions was categorized under time)
   1. Estimate activity resources - Estimate activity resources (9.2) and Estimate activity durations (6.4) go together and cannot be separated. Because most of the tasks are "effort driven", meaning that if you add more resources you will reduce the time (up to a certain level).

Q. **Communications** (10.1)
   1. Plan Communications management - build/develop the communications plan.
      i. What do you want to communicate?
         1. Who do you want to communicate to?
         2. Where do you want to communicate?
         3. When do you want to communicate?
         4. How much will the communications cost?
         5. How many resources will the communications require?
         6. For example, meetings go here.
   2. What is the best way of visualizing the [societal] system so that the user may understand any inquiry into it?

R. **Risk** (11.1)
   1. Plan risk management - what is the 'tolerance'? Tolerance defines exactly what is risk for the group and organization, and what is not a risk for the group and organization.

S. **Risk** (11.2)
   1. Identify risks

T. **Risk** (11.3)
   1. Perform qualitative risk analysis - an ordinal scale (e.g., low, medium, high; green color, orange color, red color. A standard scale is used.

U. **Risk** (11.4)
   1. Perform quantitative risk analysis - math is used to calculate probability and impact. For example, there is a dice with six sides, and what is the probability of (rolling a) 1.

V. **Risk** (11.5)
   1. Plan risk responses - what can I do to protect my project from each risk.

W. **Procurement** [*Market] (12.1)
   1. Plan procurement management - What do you need to do in terms of internal/external

action (the make or buy decision is here). Will I do everything internally, or not? What must be acquired from the market? What does not need to be acquired from the market?

i. What must be made?

1. What must be bought? [*Market]

X. **Stakeholders** (13.2)

1. Plan stakeholder engagement - map stakeholders via an influence, power, interest (four quadrant matrix), and understand what will be done.

i. Who needs it? What is its priority to whom needs it? What is the nature of the interest in it? Issue type to whom? Issue priority to whom?

3. **EXECUTING** - to act or take action (occurs in parallel with monitoring and controlling; works together with monitor and control as a fluid process)

A. **Integration** (4.3)

1. Direct and manage project work (if you are not the resource that is executing the work) - you direct and manage the work being done by the resources that have been defined for the activities.

B. **Integration** (4.4)

1. Manage project knowledge - what new knowledge is available to improve the whole process (i.e., "lessons learned").

C. **Quality** (8.2)

1. Manage quality

D. **Resources** [Materials] (9.3)

1. Acquire resources

i. Allocate material resources

1. Buy material resources [*Market]

E. **Resource** [Humans] (9.4)

1. Develop team - ensure that the human (resources) brought to the project are working together as a team (i.e., collaborating), communicating effectively, executing tasks as planned, sharing information, etc.

F. **Resource** [Humans] (9.5)

1. Manage team - operational aspects (e.g., someone becomes sick or needs to take a leave). Manage the daily changes to work due to changes on the team.

G. **Communications** (10.2)

1. Manage communications - make the meetings (time view)

H. **Risk** (11.6)

1. Implement risk responses - this is where the planned risk responses are implemented (executed). If under Plan Risk Responses, the purchase of insurance is a planned risk response, then here, the insurance is purchased. Here, are the actions related to the plans.

I. **Procurement** [*Market]

1. Conduct procurement - Execute purchases based on how procurement has been planned

J. **Stakeholder** (13.3)

1. Manage stakeholder engagement - what is happening with the stakeholder engagement (e.g., is someone gaining power, is someone losing interest?).

4. **MONITORING AND CONTROLLING PROCESSES** - to observe and correct action (occurs in parallel with monitoring and controlling; works together with monitor and control as a fluid process)

A. **Integration** (4.5)

1. Monitor and control project work - Is everything ok? Is everything going as planned? Where are the "flagging" issues around the project?

B. **Integration** (4.6)

1. Perform integrated change control - The project will change over time, and the changes must be integrated (everything, not just scope, time, cost and quality)

C. **Scope** (5.5)

1. Validate the scope - check that the scope (goals and objectives) defined in the initiating process was delivered through the executing process?

D. **Scope** (5.6)

1. Control the scope - concerned with changes in scope. The focus is on scope.
2. All tasks related to scope.

E. **Schedule** (6.6)

1. Control schedule - is something going wrong with time and the schedule? Is a deliverable late? The focus is on time.
2. All tasks related to schedule.

F. **Cost** (7.4)

1. Control costs - all tasks related to costs.

G. **Quality** (8.3)

1. Control quality - all tasks related to quality.

H. **Resources** (9.6)

1. Control resources - all tasks related to resources. Are the resources sufficient? Do more resources need to be added? Are the resource performing at the level expected.

I. **Communications** (10.3)

1. Monitor communications - does some aspect of communicating need updating or changing to become more effective/efficient?

J. **Risk** (11.7)

1. Monitor risks - are the risks appearing, or not, as expected?

K. **Procurement** [*Market] (12.3)
   1. Control procurements - receive products, and check (analyze) products to make sure products are as expected.

L. **Stakeholder** (13.4)
   1. Monitor stakeholder engagement - because stakeholders may change.

5. **CLOSING PROCESS**
   A. **Integration** (4.7)
      1. Close project or phase - this can be done for every phase and every project. Check off completion of phase or project and disseminate information via interface.

Related planning areas (essentially, the same process) are:

1. **Schedule estimate activity duration** and
   A. **Resource estimate activity resources**.
2. **Plan communications management** and
   B. **Plan stakeholder management** (because most of the communication will be to reinforce stakeholder engagement).

Related Executing Areas (essentially, the same process):

- **Procurement conduct procurement and Resource acquire resources** (because in the market, most of the time, the way you acquire resources will require a market-based procurement process).

7.7 [Project] Plan list view

A.k.a., Project plan database view.

The plan [executionable] list (database) view shows the accepted executable plan of [future] action broken down as a series of lists (information categories that have some relationship to project execution). Here, a project (and its plan) is composed of a series of information sets (or lists or project database tables).

A project plan acts as the master coordination database containing a record of all information [list] elements relevant to the project. For practical purposes, a unified project information space is subdivided into a set of use-oriented information categories.

In order complete a project, a project plan must identify and relate the following lists, upon which calculation can be done:

1. **Schedule**: The items in this list are Tasks within a hierarchical structure of groupings called the WBS (Work Breakdown Structure).
2. **Concerns**: Each Concern is either a Risk or an Issue which are handled in much the same way via a decisioning process.
3. **Actions**: The list of all tasks (actions, activities, etc.), all of which are tracked. Some tasks exist to resolve concerns.
4. **Locations**: The list of locations of everything in an information storage system.
5. **Humans**: The list of who is contributing and where.
6. **Events**: This is the list of computational integration points on a timeline. More broadly, any notable interaction between two or more people may be listed here. A recorded event always identifies the ‘result’ of that interaction (e.g., minutes of meeting, a report, a computational result).
7. **Deliverables**: The outputs (of processes) that must be completed (“ticked off” as done).

More completely, a project must identify and relate the following eight top-level project lists/tables (within a database), upon which calculation can be done:

1. **Objectives list (requirement-oriented breakdown)**: An objective/requirement is a capability to which a project outcome (product or service) conforms to a measurable degree.
2. **Deliverables list (product/service-oriented breakdown)**: Deliverables are requirements packaged with contextual information into the form of products and services (as outputs of processes) required to complete the project. Note: There are project deliverables (project needs/requirements), and sub-project deliverables (sub-project needs/ requirements).
3. **Actions list (action/Task/Work/deliverable-oriented breakdown)**: Actions (activities/work packages) are executable [process or construction] tasks. The items in this list are tasks within a hierarchical structure of textual groupings (a work breakdown structure, WBS). Synonyms for ‘action’ include, but are not limited to: work, task, activity, executable, “something to do”, process, procedure, construction, and resolution. Actions are assigned to systems and/or people. Some actions are automated. Automated actions form automated services - services without the need for direct human effort, no ‘event’ instantiation (no addition to the Events List). *Note: A project produces a product and/or a service, and so, that is why this type of plan, is called a “plan of action”); because, it intends to describe the act of brining something into existence.*
4. **Events list (Human-to-human-oriented breakdown)**: Events are a specific type of task; they are social integration-decision event task. An event (on this list) contains [at least] the location, time, and contents of human-based interactions that have lead to, or will lead to, a change and/or decision about the project (or some aspect therein).

5. **Schedule list (time-oriented breakdown)**: In order for action to occur (i.e., "things to happen"), there is time. Actions, deliverables, requirements and events can be organized within time (i.e., they can be scheduled and time delineated). These project information categories can be expressed in terms of a time (i.e., iteration) dimension. A schedule list may also be known be the following labels: timeline, gantt chart, or project schedule. A schedule can be a unified visualization of all (or selected) actions/work, deliverables, requirements, and events per [unit of] time, with all associated meta-/calculable-information. Through the scheduling of accountability project coordination can be calculated and visualized; wherein, it is possible to view: system and human bandwidth; *who's* available; and *who's* busy.
6. **Concerns list (risk/incident/issue-oriented breakdown)**: Each issue of concern is either a risk or an incident. This is a list of issues concerning organizations and events that have been/may/or are adverse [in their effects] to the completion of the project (i.e., "threats"). Here, the issue is either a risk (with some likelihood of), or an incident (current affect of), inhibiting project completion. Incidents require resolution (hence, new actions/tasks to resolve the incident), and risks necessitate mitigation reasoning for project preservation planning. Issues are prioritized (as in, 'triaged'). In general, issues themselves are not scheduled, although their resolutions may be. A planned "issue" is either a test or a trap.
7. **Contribution accountability list (people/actor-oriented breakdown)**: Profile and activity information on every human in the project, including all their associated project and sub-project information, resource allocations, and roles/responsibilities.
8. **Locations list (Location-oriented breakdown)**: Material and digital [resource] locations. Note that resources can be moved to re-located them over time, and this relocation can be scheduled.

7.8 [Project] Plan documentation view

A.k.a., Plan documented deliverable.

This is a high-level view of the multiple deliverables and integrated components necessary to complete a complex project with multiple sub-project plans in the market-State:

1. **Project charter** - Initial visualization of the problem related information set as a solution-oriented project expected to resolve the problem in a specified (procedural, protocol) manner (i.e., via a documented method).
2. **Scope statement (statement of work)**: A description of the who direction, an overview; the statement of work explicates a high-level set of requirements (with references) that define the user expectations of the work ("scope").
3. **Business case or feasibility study**: The business case is a project manager-owned artifact, often part of the Charter. A feasibility studies analyze observations over time to determine whether there are sufficient resources (given what is known and available) to complete the project at all). Not all project require feasibility studies, and in an open source system the processing is done via an open control protocol.
4. **Project coordination plan**: An overarching (project and technical) project planning document that is typically tasked to compose by an executing project coordinator. The project coordinator develops a plan to have all functions of a project fulfilled. In a socio-technical system, this plan should involve active participation by socio-technical systems teams for those items of technical interest, finance for those items of financial interest, contracts for those items of contractual interest, supply chain for those items of procurement interest, manufacturing for those items of production interest, and all of the support functions for an integrated project view. The equivalent plan for the technical aspects is the systems engineering coordination plan (a.k.a., systems engineering plan or technical plan) that expressly plans the technical design of the solution itself by subject matter expert-calculations (experts) within the given unified system.
5. **Work breakdown structure (WBS)**: In an open source system, everyone is a potential contributor, and therein, project coordinators break down problems into issues and how they can be resolves with a series of tasks. Some of these tasks exist at a high level and are called the society's Habitat Service Systems.
6. **Responsibility assignment matrix (RAM; a.k.a., RACI matrix, linear responsibility chart)**: The work breakdown structure is progressed from the product breakdown into activities, tasked to individuals assigned to the InterSystem Team. RAM is a model that describes the participation by various roles in completing tasks or deliverables for a project process. RACI is an acronym derived from the four key responsibilities most typically used: responsible, accountable, consulted, and informed. The RAM/RACI model is used for

clarifying and defining roles and responsibilities in cross-functional or departmental projects and processes. The RAM/RACI model brings structure and clarity to describing the roles that stakeholders play within a project. The RAM/RACI model clarifies responsibilities and ensures that everything the project needs done is assigned someone to do it. The WBS is tightly coupled with the RAM/RACI model, requiring the project coordinator to account and monitor who is assigned to the team, which work they will be performing, when, and its resulting orientational quality. Systems engineering is a major contributor, although not the only function involved. RACI stands for:

A. **Responsible:** People or stakeholders who do the work. They must complete the task or objective or make and take the decision. Several people can be jointly 'Responsible'.
B. **Accountable:** Person or stakeholder who is the "owner" (Read: finally responsible) for the work. S/he must sign off or approve (i.e., take the decision) as to when the task, objective or decision is complete. This person must ensure that responsibilities are assigned in the matrix for all related activities (to accountable individuals). Success [in any project] requires that there is only one person 'Accountable', which means that "the decision and total responsibility lies here."
C. **Consulted:** People or stakeholders who need to give input before the work can be done and signed-off on. These people are "in the loop" and active participants.
D. **Informed:** People or stakeholders who need to be kept "in the picture." They need updates on progress or decisions, but they do not need to be formally consulted, nor do they contribute directly to the task or decision.

7. **Change control plan** – the project coordinator visualizes a change control plan, that provides the reasoning for the selection (i.e., the methodology) of the project, and how access is decided[to be used/enabled] when processing changes to project information. On the subjective-level, this is called, an 'authority', and on the objective-level this is called, an 'open-source protocol'.
8. **Communications plan** - description of how stakeholders are notified of tasks and/or changes.
9. **Risk and opportunity coordination plan** - The Risk and Opportunity Management Plan provides risk and opportunity oversight for the project manager, but is commonly managed by systems engineering for system-based development. Both disciplines are trained in risk (and in some cases opportunity) management, only using different terminology and slightly different methodologies. This is an area that should be agreed upon up front across all disciplines employing a common language if the organization's processes are not clear.
10. **Risk register** – Some organizations limit the Risk Register to technical risks. Others identify separate registers for technical and business risks. In some cases, the technical risks bubble up to business risks. The project coordinator needs to be aware of both, just as the SE needs to be aware of both.
11. **Issue log (action item list)** – A monitoring service for the schedule. These documents are often created in multiple instances and even formats, dependent on the functions or projects capturing the issues and actions. Coordination of these items is most efficient and effective at the program level, in one format (language), with metrics in place to observe consistent issues across projects. All actions in one place (i.e., in a unified space) seems to make sense, when a project is driven to ensure timely action item closure.
12. **Resource coordination plan** – The project coordinator is accountable for obtaining the required resources for the project, which are decided upon in a temporal-priority technical resources matrix organization.
13. **Project schedule** – The Schedule, whether or not it is for the project or program, is processed via the project coordinator (for large projects/programs, the actual hands-on creation and analysis of the schedule is performed by sub-coordinating schedulers often a separate planning and control group). Technical schedules, at a lower level, feed the Project Schedule including the integrated, unified Societal Schedule (SS). The intersect is not only the "milestones", but also, other "critical path elements".
14. **Project status report with monitoring procedures** – Project status is provided via a project information visualization tool (dashboard).
15. **Lessons learned (from mistakes)** - Learning (integrating) from experience is a critical effort, not only for the project of interest. Future projects can benefit from feedback.
16. **Stakeholder analysis (re-evaluation of impacted)** – Systems engineers identity every human and non-human system involved in the project (a process of information collection and coordination).
17. **Document control** – Configuration and data control through approved documentation [is a function of the project coordinator, because

documentation is coherent social-communication]. The documentation of control is otherwise known as a 'protocol', synonyms of which include: contracts and procedural tasks (a.k.a., procedures, orders, instructions, etc.).

18. **Task completion observation and survey** (e.g., meeting minutes, video and audio recording, transcription) – for project meetings, the PM owns them. For systems engineering meetings, systems engineering owns them.

7.9 [Project] Plan process group deliverables

A.k.a., Plan phasing, plan phase deliverables,

1. **INITIATION (PROCESS GROUP, PHASE)** - issue presence and recognition.
   - A. **Project request (activity)** - issue inquiry
     1. A project request is usually the first attempt to describe, document, and estimate the project purpose, benefits, costs, and timeframe. Project estimating is an iterative process that begins at a high level with the project request. If the project request is approved, then more detailed estimates will be developed in subsequent project phases as a more thorough understanding of the project becomes known.
   - B. **Review project requests (activity)** - effectiveness inquiry:
     1. Regardless of the organizational context, the review process involves decisioning to reject or postpone some project requests, and then to prioritize those requests that the user group approves (possibly, through a protocol). The decision unit essentially "draws a line" (a threshold) based on what is possible. Those projects above the line are authorized to begin (or even continue), and those below the line are placed on hold until such time as what is necessary is available. The approved list of project requests will likely change over time as new ideas surface and priorities shift.
   - C. **Project control (activity)** - parallel control inquiry, project control decisioning:
     1. Approval and prioritization decisioning of the project request by the project coordinating unit.
   - D. **Selection of project coordinator (activity)** (i.e., project manager):
     1. The project coordinator unit is selected, and/or designed and selected.
   - E. **Project charter (activity):**
     1. Goals and needs.
     2. High level project description.
     3. Measurable project objectives.
        - i. In a general sense, an objective is a description of what will exist at the end of a project, expressed in a SMART way.
     4. Project scope – defines the work to be included (in scope), the work not included (out of scope), assumptions, and constraints.
        - i. For planning purposes, an assumption is a factor considered to be true, real, or certain.
        - ii. A constraint is a restriction or limitation, either internal or external to the project, that will affect the performance of the project. This section provides the opportunity to document constraints, such as:
        - iii. *Schedule* – project must be completed by a specific date in order to avoid [financial] penalties.
        - iv. *Cost* – funding is limited and cost overruns are not an acceptable alternative.
        - v. *Human Resources* – system architect is available only at x time.
     5. Initial high level project planning - It is recognized that planning is an iterative process that becomes increasingly precise as detailed information becomes available. High level planning usually has a fairly large margin of error. Again, the project request information is a good place to start, but the charter provides an opportunity to provide additional detail and rationale for the following estimates:
        - i. Resource requirements, including the types and quantities of resources needed to perform the in scope work.
        - ii. Project budget, including the cost of resources (human, hardware, software, other products and services) to perform the in scope work.
        - iii. Benefits.
        - iv. Scheduling dates, including anticipated start date and target completion date.
     6. Project authority - Most, if not all, projects require decisions to be made to keep the project on track. The project charter defines the authority of the individual or organization initiating the project, limitations or initial checkpoint of the authorization, control-oversight of the project, and the level of decisioning of the project coordinator (authority of the project manager).

i. Decision control (Approval authority) – identifies the project initiator by name and title, ensuring that the individual has the authority to apply project resources, expend funds, make decisions, and give approvals.

ii. Project coordinator (Project manager) - identifies the project manager by name and defines the individual's level of authority. A project manager should be given authority to plan, execute, and control the project. For example, the project manager may assign resources in a matrix organization, authorize overtime, conduct staff performance appraisals, and take appropriate corrective actions that do not increase schedule or cost. However, scope changes must be escalated to the project sponsor.

iii. Effectiveness inquiry decisioning (Oversight-steering committees) - describes societal (agency management) control over the project. Within the project, internal control is commonly established to control the day-to-day activities of the project. The project coordinator (manager) should manage internal control. External oversight should be established to ensure that the organization's resources are applied to meet the project and organization's objectives. Also identifies committee members and contact information.

2. **PLANNING (PROCESS GROUP, PHASE)** - The purpose of the planning phase is to define the course of action necessary to accomplish project goals and objectives. This course of action is typically called a project ("management") plan. It addresses all aspects of project management and includes scope, time, cost, quality, communications, human resources, risks, procurement, and stakeholder engagement. Development of the project management plan is iterative, as new information and changes occur throughout the project lifecycle, which require revisiting one or more components of the project plan. Actual coordination of the project, which occurs in the execution and control phases, is the process of doing what was described in the project plan. Project planning is not a single activity or task, it is a the Primary phase of the whole project-oriented process:

A. Project coordinators are responsible for developing the project plan (as an information set). Wherein, planning is an information processing unit responsible for ensuring the coordination of information such that planning requirements are fulfilled.

B. The project plan is the deliverable of an information set through means of a project coordinator. The project plan is itself a sub-system of a larger and more unified societal system, which is itself, operated as a projected system.

C. Project planning defines the project activities that will be performed, end products that will be produced, and describes how all these activities will be accomplished.

D. **The Project (Management) Plan (the planning deliverable)** - sub-views into a whole project, contained within a larger societal, unified information space. A project (management) plan provides a foundation for all coordination (management) efforts associated with the project. Development of the project (management) plan begins after formal approval of the project charter, which indicates completion of the project initiation phase. The project (management) plan is a document[ed information set] that is expected to change over time. The assigned project coordinator (manager) creates the project (management) plan.

1. The plan's organization should be as accurate and complete as possible:

   i. **Project Summary.**
      1. Statement of Work.
      2. Project Deliverables.
      3. Project Approach.
      4. Project Results/Completion Criteria.
      5. Critical Success Factors.

   ii. **Project Schedule.**
      1. Purpose.
      2. High Level Milestones.
      3. Detailed Schedule.

   iii. **Human Contribution (Resource Management, Human Resource) Plan.**
      1. Purpose.
      2. Project Team Functional Roles.
      3. Project Team and Cost Estimates.

   iv. **Project Budget Estimate.**
      1. Purpose.
      2. High Level Budget.
      3. Detailed Budget.

   v. **Communication Management Plan.**
      1. Purpose.
      2. Communication Matrix.

vi. **Change Management Plan.**
  1. Purpose.
  2. Change Management Roles and Responsibilities.
  3. Change Management Governance.
  4. Capturing and Monitoring Project Changes.
  5. Communicating Project Changes.

vii. **Quality Management Plan.**
  1. Purpose.
  2. Acceptance Criteria.
  3. Quality Assurance Activities.
  4. Project Monitoring and Control.
  5. Project Team Quality Responsibilities.

viii. **Risk Management Plan.**
  1. Purpose.
  2. Risk Identification Techniques.
  3. Risk Assumptions.
  4. Timeframes.
  5. Risk Ranking / Scoring Techniques.
  6. Risk Thresholds.
  7. Risk Response Approach and Risk Action Plan.
  8. Risk Tracking Process.

ix. **Issue Management Plan.**
  1. Purpose.
  2. Issue Log.
  3. Relationships Among Issues, Risks and Change Requests.

x. **Approval Information.**

2. The plan's explanation should be as accurate and complete as possible:

i. **The Project Plan Summary**: A project summary is a simplified view into the system and could include a high-level description, objectives, and scope, information flows, and control.
  1. Statement of work.
  2. Project deliverables.
  3. Project approach.
  4. Project results/completion criteria.
  5. Critical success factors (effectiveness inquiry).

ii. **The Project Schedule**: The project schedule is the roadmap for how the project will be executed. Schedules are an important part of any project as they provide the project team, participants/sponsor and stakeholders a picture of the project's status at any given time.
  1. Objective to deliverable mapping ("high-level milestone"): A milestone is an event with zero duration and requires no resources. A milestone is an event that receives special attention. It is used to measure the progress of a project and to signify the completion or start of a major deliverable or other significant metric.
  2. Detailed schedule - A detailed schedule is developed, maintained and tracked in a unified information space. This electronic schedule constitutes the project work breakdown structure (WBS). Detailed information on project estimating and WBS development is included in the appendix.

iii. **Human contribution (resource management) plan:**
  1. Project team functional roles - a project team matrix/database/chart is identifying functional roles and responsibilities, matching degrees of responsibility to processes, phases, or activities.
  2. Identification of required skills and available contributors. It is helpful in the planning process to develop a list of skills-tasks required, which may then be used to determine the type of contributor-system required for the task.

iv. **Project budget estimate**: A view into the project that relates a current project's predicted expenditure of resources to on past similar project's expenditure of resources. In a unified decision space, budgeting is control.

v. **Communication coordination (management) plan**: Formalizes communications protocols for communication within the plan. The interface and interoperability of an openly unified system with the project space, and all communications within that space.
  1. How information will be collected and updated.
  2. How information will be controlled and distributed.
  3. How information will be stored.

vi. **Change control (management) plan**: An information view into the project that describe the process involved with identifying, escalating, and controlling (managing) project changes. A project change is defined as something that is outside the documented and approved project scope or is a change to project requirements, project schedule or project cost (including resource effort). How is a required change [to the project] identified and escalated? A project change requires protocol-approval for additional resources, funding or modifications to the project schedule. The change (management) process defines how to handle project changes that present either a negative or positive impact on deliverables, schedule, budget and/or resources. The unified societal system is the repository for all project changes.

vii. **Quality (management) plan**: The purpose of the quality (management) plan is to describe how quality of the project will be controlled (managed) throughout the life-cycle of the project. It also includes the processes and procedures for ensuring quality planning, assurance and control processes are all conducted. All stakeholders should be familiar with how project quality will be planned, assured, and executed (Read: decisioning). The quality (management) plan establishes the activities, processes and procedures for ensuring a quality system-product is delivered upon the conclusion of the project. Here, verification and validation require acceptance criteria for quality. Herein, what activities will be done to ensure (have measured to be accurate) required qualities are expressed throughout a project?

viii. **Risk (management) plan**: The purpose of the risk (management) plan is to specify the processes used to identify, predict and mitigate (manage) risk. The risk (management) plan addresses both internal and external project risks associated with the project. As the *uncertainty declaration* of a project plan, risks are events or conditions that may occur, and whose occurrence, if it does occur, has a positive or negative effect on the project. Exposure to the consequences of uncertainty constitutes a risk. Although by definition risk planning may include risks that will have a positive impact on the project, the focus is typically on risks that may negatively impact the project.

1. Difference between risks and issues: If something is definitely going to happen or has happened, then it is an issue. If it is something that might happen, whether that is very likely or very unlikely, then it is a risk.
2. Risk ranking / Scoring techniques (such as prioritization ranking): For example, low to high, or 1 to 5. 5. Risk thresholds trigger action. Effectiveness inquiry is largely composed of risk thresholds that trigger action taken on a project because of organization/societal level risk thresholds. Disaster recovery and restorative justice is risk response. There is an active recognition of what to avoid in order to reduce risk. There is an active recognition that transferring a risk does not eliminate the risk. Some risk can be mitigated against (constructive action taken) to reduce the likelihood of the actual expression of the risk. Contribution is risk acceptance (for example, an astronaut today, or whenever), accepts a level of risk. In a coordinated information space, that carries an action plan in order to reduce the consequences should the risk even occur. When risks are specified, risk action plans.
3. Risk mitigation necessitates: Identifying the risk(s), evaluating the risk(s), and defining a resolution method for the risk(s).

ix. **Issue (management) plan** - The purpose of the issue (management) plan (Read: issue tracking) is to specify the processes used to identify and manage project issues. The issue management plan addresses both internal and external issues on the project. The societal (enterprise) issue tracking system is used to enter, track and report issue activity. Both the issue (management) plan and the issue log will be reviewed regularly throughout the project to monitor existing issues and to identify new ones.

3. **EXECUTION**: Project “execution” begins immediately after the project (management) plan is approved by the project creator(s). The execution phase essentially involves carrying out and controlling (managing) all the activities described in the project (management) plan. A decision is taken, at the project-level, and it is acted upon by some entity. A baseline is present, and then a change is observed. Project coordinators monitor and control all phases of a project in order to report accurately. Here, the deliverable is the actual action, an accountable role is completing the planned task(s).

4. **MONITORING AND CONTROLLING**: The collection of new project data by comparing planned and actual performance, analyzing variances and trends, identifying and assessing potential improvements, and recommending corrective action as required. Monitoring and controlling project performance enables accurate assessment of project progress, which in turn increases the likelihood of meeting user expectations.
   A. Sensors, surveys, and individual roles report, and the reports are analyzed and integrated into the project by the project's coordinator.
   B. Change control is itself is a process executed through the monitoring and controlling of changes.
      1. A baseline: A baseline is defined as the original plan, for a project, a work package, or an activity, plus or minus approved changes. A modifier (e.g., project budget estimate, schedule baseline, performance measurement baseline) is usually included.
         i. A baseline is a ruler: A baseline provides the "ruler" by which a project can be evaluated, statistically.
         ii. Baseline changes: Variance identifications.
         iii. Baseline control: Change control, decisioning protocols with thresholds. Scope is controlled through execution upon decisions [related to project information sets].
      2. Project change control action types:
         i. **Preventative action (a.k.a., preventative "measure", proactive)** - to prevent a problem's occurrence, or to ensure a problem doesn't continue to occur.
         ii. **Corrective actions (a.k.a., defect repairs)** - to fix something currently being done that is not being done correctly. Change or introduce something to prevent the appearance of a potential problem.
5. **CLOSEOUT (APPROVAL)**: The last major phase of a project's life cycle is project closeout. Project closeout is performed after all defined project objectives have been met and the user has accepted the project's service-product. This phase finalizes all project activities completed across all phases of the project to formally close (and/or transfer) the project. The project coordinator verifies the acceptance of all. A project closeout is a term that describes any action taken that finalizes all project activities and gives a formal closer to the project.

CLARIFICATION: *Some projects never have a formal closure. For example, there is never any formal closure to a community-type societal engineering project, for community's societal specification standards are an open continuous development, and the habitats in which people are physically preferable, re-configurable environments.*

8 [Project] Imperative

A.k.a., Specified project direction, directive, vision, mission, goal, objective, purpose, need, imperative, desire, problem, ideal, aim, intention, expectation, impact, benefit, output, result.

An imperative is the input of a desired output, causing the formation of a project to resolve the output into existence. Project [strategic] imperatives are specific and measurable, though not directly actionable.

NOTE*: In common parlance, the conceptual boundaries among strategic directions, goals, objectives, needs and requirements are often vague. An objective in one context or organizational level may be a goal in another. The following is intended as a rough guide to understanding project imperatives.*

Imperatives are dependent and interconnected, and hence, they can be arranged in a hierarchy with parent node imperatives following second level imperatives. Here, an imperative tree (a.k.a., objective tree) is a visualization of the hierarchy of imperatives.

QUESTION: *How is a project's [planned] direction specified?*

The directionality of a project can be sub-composed into a variety of possible informations sets, including but not limited to goals, purpose, and objective(s).

8.1 Intention (conscious directive)

A.k.a., Intent.

A [conscious] intention is an act or instance of determining mentally upon some action or result. In application, an intention is an aim that guides action.

8.2 Purpose (state the purpose)

Purpose is a compelling reason to do something. Purpose is a life aim that stimulates and motivates behavior.

NOTE: *In terms of humans, purpose sends signals to the body. When someone is motivated by a purpose that is greater than themselves, then competition disappears and collaboration starts to emerge.*

8.3 Vision (imagine the vision)

A.k.a., Visualization.

Vision refers to a commonly held visualization (description and/or explanation) regarding the deliverable(s) and the imperative (e.g., direction, goals and values) of a project and/or team.

8.4 Mission (define the mission)

A.k.a., Mission objectives.

The mission is a summary of what is to be achieved, and broken down into objectives to successfully complete the mission. Mission objectives are statement(s) that clearly document the goal(s) and constraint(s) of the mission. Constraints are pre-imposed limitations on the project. The mission objective follows from the stakeholders and their expectations.

Note: *Mission environment (a.k.a., a situational analysis) must be included (communicated) in statement of mission goals; because, it does affect design for completion of objectives.*

8.5 Goal (identify the goal)

A goal is a non-specific description of an outcome (the aim of an action), continuous or temporary. Temporary goals have a specified time limit. A goal is a specific target or direction, an end result or something desired. It is a high-level, broad, non-specific, and long-term definition of what is to be accomplished. Goals are not measurable, and several discrete projects may be needed to achieve a larger project goal. Goals are high-level, general statements about the aims of the project. A goal is some result (output) to be achieved (completed) by an action (process). Action planning is necessary to complete all goals.

CLARIFICATION: *In a business, project goals are influenced by business goals. In community, project goals are created by humans for human, and they are not influenced by market-State goals (because those concepts are not encoded, conceptually or technically). And, in engineering, goals focus on problems to be resolved.*

Setting a goal is setting a directive (i.e., an imperative or possibly even intention). It's the first step or movement toward a desired, designed change. An operational project, there is a necessity for two types of goals (operational requirements:

1. **Product (system) goals** - typically, associated with functionality and quality (i.e. functional and non-functional) requirements.
2. **Planning (process) goals** - typically, associated with schedule, resources, risk, team effort, and in the market, cost.

Thus, the first, core services enter into existence as that of information and materialization:

1. The first core information service is planning.
2. The first core material[ization] service is production.

8.5.1 Action planning (plan for action)

A.k.a., Goal execution planning.

The purpose of action planning is to select actions, and order relations among these actions, to achieve specified goals (objectives). Logic must be applied to select [optimal] actions given an probability-based environment.

Goal representation has the following essential criteria:

- A goal (G) is achieved in a state (S) if all the propositions in G (called, sub-goals) are also in S.

Action representation

- An action A is applicable to a state S if the propositions in its precondition are all in S.

Different patterns can be planned through intentional action toward a goal. Here, a pattern is the result or is itself, a rule (a process fractal is a pattern) of logic[al information processing].

8.6 Project charter (document the reasoned overview)

A.k.a., Introduction, overview, or high-level concept of operation.

A project charter is the first documented view through which the organizational case for a change is translated into project planning. Here, a project and its plan may be summarized or described with a linguistic and visual overview.

IMPORTANT: *The project charter is the first planned deliverable.*

In general, project charters include:

1. Project title (project unique categorization and identifier) Project coordinator ("manager").
2. Users/stakeholders (who).
3. Project description (what).
4. Project timeframe, start/end (when).
5. Project justification (why).
6. Project deliverables (how).
7. Constraints (optional).
8. Assumptions (optional).
9. Risks (optional).
10. Approvals (optional).

Common supplemental information includes:

1. Definitions and linguistic clarifications - to ensure effective communication and efficient understanding.
2. Imperative statement(s) hierarchy - A project comes into existence because of a stated imperative, which represents a direction with a problem-solution space. Wherein, the project itself becomes part of the solution space.

8.7 Project scope (identify the work)

A.k.a., Project scope of work, project scope statement, project definition, project mission, project vision, etc.

The scope of a project, also known as the project scope or the work scope, is all the work that must be done in order to meet the deliverable requirements or acceptance criteria agreed upon at the onset of the project. Sometimes the scope includes the identification of work that need not be done. Hence, completely, a project's scope is a definition of the elements that are included in a project, as well as what is not included. Broadly speaking, a project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, tasks, costs and deadlines. The documentation of a project's scope, which is called a scope statement, terms of reference or statement of work, explains the boundaries of the project, establishes responsibilities for each team member and sets up procedures for how completed work will be verified and approved.

NOTE: *It is important to clarify here the terms 'charter' and 'scope'. The term 'charter' is a market-State term, meaning "by contract". The term 'scope' is an optical sighting device (based on a refracting telescope); wherein, everything "in scope" is within the desired direction for the project -- the "on targets" are numerical values to be met (metrics). In this sense, an explicit written scope becomes a "contract" between the project and the participants.*

8.8 Objective (define the objective)

A.k.a., Goal, outcome, result, key results.

The objective states the ultimate goal of the project. At the societal-level, typically, an objective expresses a human need and the long-term condition that is to be achieved when the project is complete. Objectives for which a solution (system) is needed; these are often described as project objectives. Objectives are the outcome(s), the key result(s). An objective is a pre-determined result towards which effort (action) is [to be] directed, informed by intelligence. Key results quantify the success/completion of each objective in a given time period (the objective may span multiple result periods). A key result is the outcome by which success/completion is measured. In application, an objective is a specific intention expressed in measurable terms to achieve a goal (i.e., direction). Objectives may be defined in terms of outputs, results, outcomes and/or benefits (or similar intentional/directional language). More completely, an objective is the described result of the completion of effort toward a direction of intent. An objective is a the linguistic absolute description of a result (output or expectation) to be satisfied at successful completion of effort, within a certain period of time and by means of access to certain resources.

Objectives provide an individual or social organization with clarity on intention, focus, and direction in an uncertain environment. Every objective has a purpose (cause, constructor) that defines the *what* and the *why* of its instantiation.

In concern to teamwork, objectives localize to (i.e., become associated with) *nouns* (objects, physical or digital resources) and *verbs* (functional service operations, processes or protocols) of a given team.

> **NOTE:** *Real world human objectives are also known as: human needs and human requirements.*

Influenced by goals, a project objective is a detailed description of the specific and measureable outcomes desired from a project. A project objective describes the desired result of a project (tangible product or intangible service). Objectives are detailed statements about what the project should accomplish. The project and its objectives must always contribute to the goal, otherwise the project is not being pursued (or, should not be attempted). Objectives document a project's value for the end user. Therein, activities, and most likely deliverables, will contribute to achieving the objective.

All planning and strategic activities occur to resolve objectives, as well as to quantify a level of performance for their resolution.

Objectives express the following characteristics:

1. An objective is specific and measurable.
2. Describe the [business] value of the system and help prioritise features and requirements based on their value.

Project objectives:

1. Are a more refined version of the goals (outcomes and expectations) of the project.
2. Are what must be achieved (in existence, function, status, or state) to consider the project complete.
3. Refers to what the project aims to achieve; a strategic vision.
4. Are a part of the description of the project. Project requirements are derived and created from the requirements of the user and/or system.
5. Different from project coordination/management objectives.

Objectives and key results (OKRs) commonly include:

1. Mission and vision (scope).
2. Goals (delineated scope).
3. Objectives (delineated goals).
4. Results (impacts).
5. Tasks (activities).

In written form, key results are generally syntactically composed of:

1. Verb (e.g., reduce).
2. Specific noun (e.g., time to service).
3. Key result (e.g., elapsed time from first call to service).
4. Target (e.g., 16 hours).
5. Date (e.g., 12, April, 16:45).

Project objectives may be prioritization. Per the language of the existing societal specification, in the case of humans, there is a priority of needs from life support (survival), to technical support (technical services), and facility support (leisure services).

From an action-oriented perspective, an objective is a measurable target that specifies when a problem is resolved. Every objective has a success or completion metric.

> **Clarification:** *In business, project objectives describe the business value of the system. What is the value of the produced system to business interests, and hence, based on its value, what is the priority for requirements and features, materials and motions? In community, project objectives describe the human need for and community value of the system. What is the value of the system to human interests, and hence, based on its value, what is the priority for requirements and features. Also, there is a set of pre-defined values that facilitate this process.*

Among community, the desired outputs of the societal system are derived from the effective needs of the users, which are continuously prioritized. The language of the outputs should be more precise than that of the needs, and should reflect what the system does or provides in response to the eliciting needs.

An objective is a description of what will exist at the end of a project. Generally, objectives are written as linguistic statements. In the statements there are nouns, and those nouns are the project's deliverables, which are listed in the deliverable diagram (a.k.a., work breakdown structure, WBS). The deliverables and outcomes come from objectives.

For example:

- OBJ-001: Develop a design to identify the components and costs for the gardens.

Every project will have several layers of objectives, which are necessary in order to complete a project in the real world. Some objectives are common to all projects, and others are only relevant to the a specific project.

There is an absolute *objective* to fulfill human need on some cyclical basis. In order to engineer a resolution to the *objective* problem, there are multiple types of engineering requirements that must be defined:

1. **Process requirement** to identify human issues (needs, wants and preferences).
2. **Non-functional requirement** to fulfill needs in a specifically assimilated/assembled way.
3. **Availability requirements** for resources and people - when will the process be operated?
4. **Functional [capacity] requirements** - how many times do they operate the process per day?
5. **Reliability requirements** - do the users really need the process and data to be available 100% of the time?

8.8.1 Construction of an objective and key result ("milestone" setup,

The construction of an objective and key result statement (claim) takes on the following linguistic structure:

1. (Quality change; e.g., increase, decrease, continue) (Objectives statement, Key results, Needs) (Target) (Date)
   A. For example, increase habitat service water clean water output to a region. Then when it has clean water, continue to output clean water per master planning specification selection whose operations continue to meet human need fulfillment targets.
   B. There are incident operations in a habitat: for example, increase safe water output to 100% of the population within one week.
   C. And, there are continuous fulfilment services in a habitat: for example, continue to produce safe water output to 100% of the population per master [scientific-design] plan.

8.8.2 Characteristics of objective(s)

In a dynamic system, in order for information to be useful it must maintain the following objectives (which are the characteristics of useful information):

1. **Definable (conceivable)** - can be described and easily understood by the population of contributors.
2. **Manageable (organizable)** - a meaningful unit of information where specific responsibility and access can be assigned to an accountable actor, and where monitoring and tracking is possible.
3. **Predictable (Attainable)** - sufficiently understood that planning is possible in time with resources.
4. **Estimateable (specific)** - duration, time-frames, and resource usages can be estimated to complete the project.
5. **Integratable (Specific)** - integrates with other project work elements at a higher project [system] level.
6. **Measurable (Quantifiable)** - can be used to measure progress; has start and completion dates and measurable interim milestones.
7. **Adaptable** - sufficiently flexible so a change in social intention can be readily accommodated into the project's directive.

The primary measurement creation quality objectives (quality goals) are (SMART):

1. **(S) Specific and clear goals** - What is to be done or realized?
2. **(M) Measurable** - How will it be measured?
3. **(A) Achievable** - Is it feasible, viable?
4. **(R) Relevant and recorded** - Is the goal recorded and relevant to a larger direction?
5. **(T) Time-bound** - What is the timeframe?

Table 3. *Measurement quality objectives list.*

Letter	Meaning/Purpose
S	**Specific** - Is the objective clear in terms of what, how, when, and where the situation will be changed?
M	**Measurable** - Are the targets measurable? For example, how much of an increase or reduction is desired? How many items should be produced, or how many people will be trained?
A	**Action-oriented** - Does the objective specifically state what actions are required to achieve the desired result? In some cases, the A refers to "attainable." Is the objective something that can be reached by the performers?
R	**Realistic** - Are the desired results expressed in a way that the team will be motivated and believe that the required level of involvement will be obtained? Is the description accurate?
T	**Time-bound** - Does the objective reflect a time period in which it will be accomplished (e.g., end of the first quarter or by end of year)?

8.8.3 Real-world objectives

Every project is itself an information system with a real world directive. In order to maintain alignment with the real world, information in a project must be processed into three systems-level objectives, which are common to every project:

1. **Develop an accurate model of the world from which to work (Read: science)** - This becomes a universal information set common to all projects. *Science creates a societies common knowledge base from which to create systems into the material world.*
2. **Design an accurate model of the system to be constructed into the real world (Read: engineer)** - This is a model unique to each project. *Design creates a specification to be constructed.*
3. **Construct the model of the system into the real material world through (Read: hardware production and software programming)** - This is a material creation unique to each project. *Construction creates the materialized creation that humans must live with.*

In order for information to be useful in a project, it must have some sensibly aligned relationship with the real world. In order to design a system which may be effectively constructed in the real world, it is essential to have an accurate model of the real world, informed by logical systems processing, scientific research, and artificial sensors. This model should be as accurate ("lossless") as possible, because it will be used to inform design and final, real world product.

In concern to modelling the real world, the goal is to compress all the data associated with the real world, optimally, into a computational representation (a.k.a., model) of the world with which individuals, and together, everyone, can work on human projects (and at a societal level, on projects that ensure human fulfillment and planetary ecological regeneration).

8.8.4 Project-level objectives

> **CLARIFICATION:** *Project objectives go by multiple different names depending on context; other common names for a project objective include, but are not limited to: strategic direction, strategic imperative, mission, vision, goal, purpose, endeavour, target, etc.*

Every project has a top-level [project] objective to complete the project. This project objective may be sub-divided into a set of [project] sub-objectives related to the categories of material realization. These material realization categories are sub-defined within the project in terms of information flows, tasks, resources, and time (a schedule), and budget (*in the market*).

At the project sub-objective level, objectives are orientational. Project sub-objectives delineate and define what is to be delivered and how it is to be produced.

The objective of a project (which exists as a conscious intent outside of the project). The project's [systems-level] objective(s) is the tangible end product or result that the project team must produce and deliver. The projected systems-level objective is an objective description of what is to be produced and delivered. These objectives state what the project will accomplish in terms of the user's intended value to be achieved.

> **NOTE:** *The term "charter" is sometimes given to the document that lists the full set of project systems-level objectives.*

8.8.5 Community-type society objectives

For a community-type society the objectives include, but may not be limited to:

1. Core objectives (direction indicators, values, at the highest level):
   A. Freedom.
   B. Justice .
   C. Efficiency.
2. Stabilizing objectives:
   A. Learning and integration.
   B. Health and vitality.
   C. Appreciation and compassion.
   D. Regeneration and abundance.
   E. Openness and sharing.
   F. Cooperation and collaboration.
   G. Intrinsic motivation.
3. Evaluation-side (result indicators, at the highest level):
   A. Human fulfillment objectives.
   B. Ecological flourishing objectives.
   C. Economic sustainability objectives.

9 [Project] Deliverables

A.k.a., Work products, work outcome, change deliverables, project outputs, resulting usable objects, work or task output.

A deliverable is a specific, tangible product or thing; an object and/or information packet. One or more deliverables may contribute to achieving an objective, but an objective is not a deliverable. A deliverable is anything produced or provided as the result of a process (i.e., service, operation, etc.). A deliverable is a pre-defined, tangible work product (i.e., the output of time working). A deliverable can be informational and/or material. More generally, a deliverable is an output, something produced or provided as the result of a process. Process is another word for task or action. A work product is any tangible item that results from a project function, activity, or task.

When there is any change, there is an event, and a result. Deliverables are the output/outcome of activities (which complete to produce the deliverables). Deliverables must be aligned to objectives (intention). Deliverables are linked to the tasks (work) that produce them. A deliverable is a grouping of project work elements (tasks, actions, activities, executions, etc.) shown in graphical display to organize and subdivide the total work (as a visual information "scope" of a project). A deliverable involves the reducing of work into tasks, and ultimately, scheduled state changes in the extant, real world. A deliverable is a tangible or intangible (or service, combination) output of a project.

At the societal-level, there are two main project-related deliverable life-cycles:

1. **The project life-cycle:** There are **project-level (information) deliverables**, specified by information standards and practices.
   A. Project coordination deliverables (Read: the project's information-level)
2. **The product life-cycle:** There is **the project's deliverable(s) specified by a user** in relationship to a pre-existing environment that an InterSystem Team sustains.
   A. Product deliverables (Read: the system under project development)

Table 4. *Example deliverables, include, but are not limited to the following types.*

Deliverable No.	Deliverable name
1	Societal specification standard
2	Education materials
3	Software system
4	Hardware system
5	Demonstration experience
6	Dissemination platform
7	Geopolitical analysis and relationship development
8	Habitat site (including, selection)
9	Sufficient market-State currency
10	Habitat service system
11	Human will

9.1 Project deliverable diagram - Work breakdown structure (WBS)

A.k.a., Work process organization.

A project deliverable diagram (and list) is also known as a work breakdown structure (WBS). The basic formula for a WBS is to take the complete scope of a project, break it up into pieces, then organize them into a logical hierarchy. All the items at a lower level are needed to complete the item at the next highest level. A work breakdown structure (WBS) is a key project deliverable that organizes a team's work into coordinated sections work coupled with a deliverable. The work breakdown structure visually defines the scope into categories that a project team can understand, as each level of the work breakdown structure provides further definition and detail. An easy way to think about a work breakdown structure is as an outline or map of the specific project. A work breakdown structure starts with the project as the top level deliverable and is further decomposed into sub-deliverables, which are the output tasks (work). A project team/coordinator creates a project work breakdown structure by identifying the major functional deliverables and sub-dividing those deliverables into smaller systems and sub-deliverables. These sub-deliverables are further decomposed until a single entity (person or machine) and all necessary resources can be assigned. At this level, the specific work packages required to produce the sub-deliverable are identified and grouped together. The work package represents the list of tasks or "to-dos" to produce the specific unit of work represented as a deliverable on the work breakdown structure diagram.

A project deliverable diagram describes (visualizes and lists) the specific activities (delineated into tasks) that must be completed for the project to be complete. More simply, a project's work is broken down into a visual structure -- a project's work is broken down into a usable structure. The WBS is a hierarchical arrangement of major tasks that need to occur in/during the project. Within each of these major tasks there are typically a number of sub-tasks that describe the major task in more detail. These sub-tasks can have their own lower level sub-tasks, and this can be broken down to multiple additional levels depending upon complexity (requirements).

Work breakdown structures are typically visualized as hierarchy diagrams. It is common practice to include time (and in the market, cost) estimates in the WBS diagram. It is important to number the diagram with each sub-task as a decimal integer of the whole number primary task (e.g., 1.0 > 1.1 > 1.1.1 > ...). These numbers are used for: InterSystem team task selection/divisioning and accountability; monitoring activities and schedule alignment; and allocating resources (in the market, allocating budgets). A work-breakdown structure (WBS) in project management and systems engineering, is a deliverable-oriented breakdown of a project into smaller components. A work breakdown structure is a produced information set that hierarchically (by priority) lists all deliverables. A deliverable is an outcome or a result of something. A deliverable provides some value to the project [service] users.

A work breakdown structure identifies all the work (i.e., task, action, doing, activity). A work breakdown structure is a top-down decomposition of deliverables into work packages, which are made available to community contribution.

The work breakdown structure (WBS) represents 100% of the deliverables (given current knowns). A project work breakdown structure may be visualized as a hierarchy chart (of work/task/action packages). The work breakdown structure visually defines 'outputs' of the project at a sub-project level. A work breakdown structures states that the project will produce the worked deliverables in the visualized structure. The WBS provides a hierarchical depiction of all the work outcomes. It is created through progressive elaboration (i.e., it is a "living" document). 100% of what makes up the outcome of the project is listed in a hierarchical chart known as the work breakdown structure.

The WBS, on the other hand, is agnostic to timing, effort, and costs. It only represents what needs to be produced as a result of the project and it.

- **Deliverables** all start with nouns (things to be produced as part of this project) - these are associated with the 'work breakdown structure'.

  CLARIFICATION: *Work deliverables (WBS) are described via nouns, whereas schedule activities are described with verbs.*

The WBS tool functions to:

1. Interface: Deliverable-oriented view of the project work - list and visualize deliverables.
2. Organization: Hierarchical grouping of the work outcomes required to meet project objective. In other words, a hierarchical list of project deliverables (outputs).

Type of WBS (always represents tangible deliverables to be produced):

1. **Project WBS** (**product deliverables or project-level deliverables**, specification or blueprint) - all of the components of a product being developed. Projects are initiated to produce specific, unique outcomes based on specific, unique needs. That intention and need must be expressed (delivered) in some tangible form, whether it's a system, a product, a process, an object, a plan, a rule, or some other outcome.
2. **Service WBS** (**service deliverables or operations deliverables**, specification or blueprint) - all of the components of a service provided to a user.
3. **Process WBS** (**process deliverables or method of delivery, mechanistic deliverables**, specification or blueprint) - all of the components of a process or methodology used to coordinate work to provide service to a user [in the form of a product]. All work goes through a set of organizational process, conveying the conditions of being planned (planning), scheduled (scheduling), executed (coordination and monitoring), and assessed (assessing).

For example, a project to produce a bicycle may have the following WBS:

1. The top level of the system (e.g., a bicycle).
2. The 2nd level systems (e.g., frame set, crank set, wheels, braking system, shifting system, project integration)
3. The 3rd level systems (e.g., frame set - frame, fork, handlebar, seat; crank set - pedals, chain; project integration - prototype approval, product test, quality sign off, project management; etc.)

9.2 Product breakdown structure (PBS)

The product breakdown structure (PBS) is a tool for analysing, documenting and communicating the outcomes of a project, and forms part of the product based planning technique. The PBS provides an complete, hierarchical tree structure of deliverables that make up the project, arranged in whole-part relationship. In other words, a product breakdown structure is a hierarchical structure of deliverables that the project will make or outcomes that it will deliver; it decomposes a project product into its constituent parts in the form of a hierarchical structure.

10 [Project] Tasks

A.k.a., Activities, event, action, job, work, process, procedure, instruction, energy.

A task is some amount of work that must be completed within a defined period of time, or by an output date. Tasks exist due to conscious imperatives or automated directives. A "work instruction" describes how to perform a task within a process, which is a more detailed portion of a procedure. Here, task designation is the systematic and purposeful allocation (assignment) of tasks to individuals and groups within an organization. Task-based models chunk effort into short "doable" segments. The purpose of anything (humans, any organism, machines) results in tasks. A new form creates a new task[ing] space.

A task/activity is a distinct, scheduled portion of work performed during the course of a project. An activity is a task that is identified, assigned, executed and controlled as part of a project. Activities are work packages of work[ing roles] available for contribution. Scheduled (i.e., temporally associated) tasks are also known as activities. The activity is what is done to achieve the objective of the task. An activity can be a specific action or a process (it is another word for a task), and many activities will likely be involved to meet project objectives. Activities contribute directly to achieving the objective, and thereby the goal [of the project].

Activity diagrams shows the activities involved in each project coordination process (activities are tasks). All activities occur to complete/meet requirements. The objectives of requirements management activities include collecting, documenting and organizing the requirements, linking requirements to activity items, tracing requirements through execution of activities, and tracking and communicating this information to all stakeholders. This is necessary to ensure that the requirements and their activities are properly handled throughout the project life cycle.

Generally, in project management, the terms "task" and "activity" have similar but different meanings:

1. Task is associated with the input 'requirement'.
   A. Activity is associated with output 'schedule' (as in, the time-binding of tasks).

In this sense, activities are [tasks that are] time-bound within a schedule, which is interconnected (i.e., flows) to a requirement-bound deliverable structure.

NOTE: *Tasks are associated by project, which are issues.*

Tasks that machines carry out are human tasks and not machine task. Machines do not and should not carry out tasks for their own sake. All tasks are human tasks, and machines are extensions of humans that carry out human tasks. As machines become more self-aware, the distribution of all [human] tasks will likely happen more autonomously and intuitively. Over time, humans have increased the number of machines they use, thus extending their circle of tasks (i.e., expanding human task ability).

NOTE: *All work requires tasks. Multiple parallel tasks require coordination processing: [en]rolling and scheduling of tasks.*

Tasking terminology includes, but is not limited to:

1. A **process** states what needs to be done and why. A process is any activity or set of activities that use resources to transform inputs into outputs.
2. A **procedure** states how the process needs to be done. A procedure is a uniform method that outlines how to perform a process.
3. A **work instruction** explains how to carry out the procedure. A work instruction describes how to perform a task within a process.

10.1 Task analysis

A.k.a., Analytical task granularity, activity analysis.

The standard definition of a task is "a piece of work to be done". In more complex terms, a task is a package of information, that when acted based upon by an actor, together produces some qualitative or quantitative result in the status (or state) of a system.

Through project coordination a task analysis becomes a work order (work package), which enters wider scheduling.

APHORISM: *It is when we choose to resolve existence, as a whole, that sufficient information becomes available to see how we can live together in fulfillment.*

In this project plan, task analysis is one deliverable (i.e., element/component). Task analysis is

1. A formal method of describing and analysing actions performed by people and/or systems.
2. The analysis or a breakdown of exactly how a task is accomplished, such as what sub-tasks are required.

In concern to tasks, there are two types of task analysis:

1. Task analysis (high-level) - the work needed to accomplish a large goal broken down into sub-goals and major tasks.
   A. Procedural analysis (low-level) - the specific steps and decisions the user takes to accomplish a task.

A task analysis aims to understand at least these three elements:

1. The users (the creator of the issue).
2. The tasks that are performed (to resolve the issue, activities).
3. The environment (in which the tasks are performed).

Tasks typically involve:

1. A clear start and finish (e.g., requirements and requirements review)
2. Involve discrete steps (e.g., task breakdown)
3. Result in a change of status (i.e., they require energy, work, effort, action, motion, change, etc.)
4. Are specific to clearly defined circumstances (e.g., sufficient unification for situation awareness)

Task analysis data is collected, integrated, and then visualized in a hierarchy.

The purpose of task analysis is to analyze how the user interacts with the space system and to define the tasks, which direct design concepts and decisions. Task analysis is a methodology used to break an event down into tasks and break tasks down into components. A task analysis identifies system-level and subsystem-level tasks, to determine operator needs for established mission objectives and concepts of operation. It is used to understand and thoroughly document how tasks are accomplished. The focus for the analysis may be on how a human(s) perform tasks, how a machine(s) perform tasks, or a combination of both. Task analyses should be performed for all functions for the established system objectives, scenarios, and ConOps. Task analysis is an essential component of human-centered design, focused on providing usable systems for humans throughout a system's entire life cycle. Task analysis is a fundamental design activity necessary for implementing many human system requirements. Task analysis refers to a family of techniques that involve the systematic identification of the tasks and subtasks involved in a process or system and the analysis of those tasks (e.g., who performs them, what equipment is used, under what conditions, the priority of the task, and dependence on other tasks).

An iterative approach to task analysis enables the identification of current and future task demands that can aid in decisions, such as which tasks should be allocated to a human or to an automated system, or how system components should be used. Task analysis also results in the identification of critical team tasks, which are tasks that are absolutely required and necessary for team to successfully accomplish operations and meet project (service or mission) objectives. Critical team tasks may occur nominally or off-nominally and include tasks that are essential to team health or, if done incorrectly, may lead to loss of life, loss of project, or undesirable habitation states (through to, loss of habitat).

NOTE: *Identifying these tasks early, can enable efforts to be made to implement designs that reduce the probability of mishaps or errors and allow crews to perform tasks within expected time limits and environmental conditions.*

Task definitions should evolve as the system capabilities, including the user, become better defined through the conduct of activities in the iterative human-centered design process.

In the context of a human user, it is possible to define the physical and cognitive tasks that must be accomplished, and to describe pertinent task attributes such as:

1. User roles and responsibilities.
2. Task sequence.
3. Task durations and frequencies.
4. Environmental conditions.
5. Necessary clothing and equipment.
6. Constraints or limiting factors.
7. Necessary user knowledge, skills, abilities, or training.

The process of conducting a task analysis commonly involves:

1. The associated decomposition of physical and mental (i.e., cognitive) activities.
2. Activity frequency and duration.
3. Task allocation.
4. Inter-task dependencies.
5. Task criticality and complexity.
6. Environmental conditions.
7. Necessary hardware, software, processes (e.g., clothing and equipment).
8. Any other unique factors involved in or required for one or more people to perform a given task.

10.1.1 Societal-level task analysis

The proposed societal systems highest level task analysis categories are:

1. Lifestyle analysis - of a user's/person's typical day or week; "a day in the life of", "an evening with", "a month in the life of". Here, what is needed and preferred is identified.
2. InterSystem Team Work analysis - all the goals and tasks that someone does in a specific role, and all deliverables - daily, monthly, over long periods. Here, it is described how work moves from person to person.

For example:

1. Personal access:
   A. User self-cultivates at (@) personal dwelling.
   B. User self-cultivates at (@) personal garden zone.

2. Common access:
   A. User schedules at (@) common eating area.
   B. User schedules at (@) common recreating area.
3. InterSystem Team access:
   A. User harvests/forages at (@) culturing zones for foraging.
   B. User serves at (@) culturing zones for food harvesting and processing.

10.1.2 Operations tasks

InterSystem Team operations has the following requirements:

1. Provide system operational availability that meets requirements. Operational availability is a factor that describes the amount of time that a system can perform its function as a fraction of total time – including downtime for maintenance.
2. Monitor the environment (e.g., sensors and surveys). For example, the degree of presence of toxins and "toxic" relationships, either microbial, physiochemical, or psycho-social must accounted for in design. The build-up of toxic substances in a tightly closed environment (e.g., the "tight building" syndrome) is a design challenge.
3. Enable, disable, and monitor processes and capabilities.

10.1.3 Construction tasks

The habitat service system is constructed modularly. Each module has a repair and replacement lifecycle (a duration of existence):

1. Test/Prototype construction.
   A. Prototype fidelity:
      1. [Medium to high fidelity] A prototype is a model of the system delivered in the medium of the system.
      2. [Low fidelity] A mock-up is a representation in a different medium.

Tasking roles include, but are not limited to:

- Engineer or technician - A person who is skilled (has procedural and semantic knowledge) in designing, diagnosing, developing, constructing, maintaining, and repairing technical system (Read: any information or material system).

The following habitat oriented terms are effectively synonymous, but can be loosely separated to mean:

1. Engineering (Engineering/Decisioning as planned) - Development of system and System integration.
2. Technician (Operating/Operations decisioning) - Integration of design and System operation.

10.1.4 Maintenance tasks

In general, maintenance refers to inspection and monitoring, repair, replacement, and updating. Technically maintenance only concerns those tasks necessary to maintain a service once its integration has achieved final valid and verified integration.

Maintenance can be a complementary means to restore fault tolerance, non-critical functions and system/human safety. Because movement is limited by physical mechanics, transport time, and mass and volume constraints, maintenance provisions must be available on [each habitat-city] site.

Tactics to ensure efficient and safe maintenance include:

1. Advance deployment of spares.
2. Component commonality.
3. In-situ manufacturing.
4. Low-level repairs.
5. Autonomous training and procedures.
6. Robotic implementation and preventative attention.

Unless impractical, all equipment that may require maintenance will be located internally; and whenever possible, all external items should be detachable so they can be moved to an interior space for repair. In general, human time and logistics demands must be minimized and conducted under the safest possible conditions.

10.2 Tasking

A.k.a., Task prioritization (task triage).

It is possible to intentional design the next expression of a system to meet a set of desirable conditions (conducive of fulfillment). And, those conditions that remove the likelihood of fulfillment, when included as data, set up a task/time/resource prioritization hierarchy, commonly known as 'triage'. As a continuous systems or a systems level process of prioritizing task systems (tasks, supra and sub).

At a high-level, tasking involves:

1. Identify task.
2. Identify task completion date.
3. Duration (as a probability) is assigned to each task.
4. Resources (as a probability) are assigned to each task.

10.2.1 Task complexity

Task complexity (within the scope of a project) is broken down into two categories of effort:

1. manual skills, and
2. cognitive intelligence.

All tasks (i.e., work) contain both a manual element and a cognitive element, and require both types of effort. Intelligence underscores the interconnected nature of work within a project, emphasizing that successful task completion relies on physical and intellectual ability.

10.2.2 Task dependency

A task 'dependency' is when one task cannot start or complete until another one has been finished, because the later requires (relies on) a resolved output from the earlier task.

10.2.3 Task scheduling

Task are typically scheduled, but when there are incidents, tasks are not scheduled, but needed. The strategic and systematic allocation of resources, time, and dependencies to do work at a single point in time, requires scheduling. Scheduling aims to optimize workflow, balance workloads, and meet deadlines by strategically sequencing tasks based on protocols. Various scheduling techniques, such as Gantt charts or critical path analysis, are employed to visualize and manage the timeline and [cognitive/physical] dependencies of tasks throughout a project's lifecycle.

10.2.4 Milestone

A 'milestone' is a 'task' with zero hours and zero duration, used to mark an important 'event' or 'accomplishment'. In relation to a project's direction, a milestone is a significant advance that must be made or taken (enacted). The milestones for a socio-technical society system are individual (that become social) and technical.

Because there is only consciousness, there are individual self-awareness advances, and there are material-technical advances (Read: consciousness-physics advances, a.k.a., mental-astral physics advances).

The difference between an individual consciousness advance, and a social advance, in 'conception', is:

1. For the individual, the recognition that "we are all one" becoming an integral part of self-conscious awareness, and thus, the urge to resolve social issues in some way other than by killing each other and artificially limiting access to planetary resources (i.e., these ways of deciding become obsolete).
2. For the social, there is, for example, 'social mobility', and 'no war'.

11 [Project] Work

A.k.a., Teamwork role.

'Work' is an 'activity' involving intentional motion in time -- work is scheduled in a time as 'activities'.

In any given human society, there are three types of real work:

1. **Human work (a.k.a., manual work)** - human physical and cognitive work.
2. **Hybrid work** - human work with complex machine work. Hybrid work refers to a socio-technical system composed of humans, information, hardware, and software.
3. **Machine work (a.k.a., automation work)** - hybrid hard-soft machine work.

11.1 Work description

A.k.a., Work package, articles of work, work contract, labor contract, job description, etc.

All work organizations are procedurally based upon a set of contracts of agreement to do work. All tasks and activities are "work packages" run from within projects. Work is done to output (produce) a deliverable through requirement associated tasks and time associated activities. Work packages are assigned to teams who are accountable for the completion of work. In specific, work packages can be assigned to humans, machines, or a hybrid combination. When humans agree to do work, they agree to a description of that work (a "contract").

Visualization of the work package, includes:

1. Deliverables are listed (visualized) in a 'project deliverable structure' or 'work breakdown structure' (WBS).
2. Activities are listed in a 'project schedule'.

Work packages describe

1. The what - Work.
2. The who - Who the work is assigned to.
3. The when - Within what time-frame.
4. The where - Location, position in space-time.
5. The how - The information, tools, techniques, and the -ware (hard-/soft-) systems necessary to complete the work.

11.1.1 Work package details

A.k.a., Work profile/program.

Common work package details include, but are not limited to.

1. Objectives - what is to met, maximized or minimized.
   A. Overarching objective - Strategic goal.
   B. Primary objectives - A statement of purpose expressing a desired satisfaction of required capability; related to why a system is wanted. (No numbers or values).
   C. Define a set of measures (or metrics) for each objective.
   D. Objectives state definable characteristics.
   E. Objectives describe what a system should do (function) and why the user needs it (to fill a defined need or gap).
2. Constraints - what are the boundaries.
3. Functions - what does it do. What is its behavior.
4. Access (Means) - by means of what does it do what it does.
5. Skills (human translation).
6. Number of humans involved as contributors.

11.1.2 Work decisions

Common work decision in a conceptual-spatial system include:

1. **WHAT**: Define the work (e.g., concept of operation, statement of work, etc.).
   A. **How much**: Solution resource requirements.
2. **WHEN**: Schedule the work.
   A. **To whom**: Schedule assignment matrix of human and machine systems.
3. **WHERE**: Allocate resources to the work.
   A. **To whom**: Accountability assignment matrix.
   B. **For what access**: Work-service access-decision effectiveness control.
4. **WHO**: Contribute “your” time and conscious bodily effort.
   A. **How much**: Work contribution.
5. **CONDITIONS**: Assigned role in InterSystem team.
   A. **How much**: Individual and team capability.
6. **UNDER WHAT**: Environmentally situational conditions.

11.1.3 Work execution

The interface between plan and work (a.k.a., “job dispatching” or project execution) is the allocating or assignment of tasks (jobs) to systems, teams, and machines (by a central coordinator).

11.1.4 Work-ability

A.k.a., Workable?

For a system to be ‘workable’, it must be capable of being coded (“changed”), and possibly, executed (“run”) by another system.

Both human and machine systems can work the real world environment:

1. Information can be re-coded and executed upon by information (cognitive and computing) systems.
2. Materiality can be re-coded and executed upon by material (human and robotic) systems.

12 [Project] Stakeholders

APHORISM: *May we all live in honor of one another's potential to contribute [to bettering the self and others].*

Historically, the term stakeholders was applied to 3rd party individuals or organizations that held a financial wager (i.e., held a "stake" in a financial exchange). Some stakes are bigger than others. Anyone with an interest, a metaphorical "stake" or interest in the success of an organization or action. In concern to the financial origins of the term. Project stakeholders are entities that have an interest in a given project. The project stakeholders include:

1. Those who actively participate in a project.
   A. Those who can determine the course of action, the plan, of the project.
   B. Those who take action based on the plan of action.
2. Those who are influenced/affected by a project (or project process).

In practical societal-service, the common stakeholders include:

1. **Users**: Human individuals using the service.
2. **InterSystem Team:** Engineering developers and engineering operators of the service; those who develop and operate the service (Read: those who co-operate to create and continue the service).
3. **Planetary population**: All beings that may be fulfilled or may suffer.

CLARIFICATION: *Not all project stakeholders may be co-operating as working on, and using the result(s) of, a project. A given project may or may not consider a population that is impacted/affected by the project, but not deciding or enacting a project. There is a social organization internal (in affect) to a project, and there may be a social organization external (in affect) to a project. A project that does not account for some influentially affected population [by the project] in decisioning and action upon the project is said to have 'externalities'. It is the case that some societal configurations have the [fulfillment and suffering of the] planetary human population, and the population of other species on the planet, as externalities.*

When the project is in service of the market-State [incentive structure], the common stakeholders include:

1. **Users**: Customers.
2. **Employers**: Businesses.
3. **Employees**: Project team members.
4. **Citizens**: Taxable (extractable) human base.
5. **Planetary population**: All beings that may be fulfilled or may suffer.

The socio-technically conceived project stakeholders are:

1. **End Socio-Technical Users** - as people (citizens, customers, individuals among community) using the project-generated and project-operated service system.
   A. The Community of individual users.
2. **Operational Socio-Technical Users** - as people (citizens, employees, individuals among community) operating the project-sustained system.
   A. The InterSystem Team of contributors.
3. **User Socio-Technical Developers** - as people (citizens, employees, individuals among community) using the operational system to develop new systems.
   A. The InterSystem Team of contributors together with the Community of individual users, where the InterSystem Team of contributors is accountably tasked to develop the controlled habitat service system (HSS).

12.1 *What is a stakeholder*

A "stakeholder" is anyone who has an interest ("stake") in the outcomes of any decisioning or action in relation to a project. Identifying, mapping, and prioritizing [the flow of information to] stakeholders are important first steps in coordinating the communications for a project. Projects can only be considered successful (complete) when their stakeholders acknowledge that they are a success (validation and verification).

In concern to stakeholders, there is a need for collaboration among:

1. Those who design systems.
2. Those who operate the systems.
3. Those directly served by the operated systems (directly affected).
4. Those affected by the outcome of the operated system, but who are not directly served by the outcome (indirectly affected).

12.1.1 Personal validation

Personal validation is how facts that are possible to be verified are done so by individuals. In a participatory project, any action taken in the project is based on decisioning, and decisioning is transparent to all stakeholders.

12.2 *[Project] Stakeholder register*

The following table is an example register within a market-State organization (note that a community team register will have different:

Table 5. Project Approach > Stakeholders: *Table shows stakeholder data interrelationship categories.*

Stakeholder interrelationship data category	Description
Position	A unique ID for the Stakeholder
Description	A fuller description of the Stakeholder (Additional information that supplements the Name)
Influence_Power (market-State only)	High, medium or Low
Disposition	Positive, neutral, negative
Requirements	Project objectives as seen by the Stakeholder (what the Stakeholder wants to benefit from the project)
Impact on project	Project concerns that could arise due to the influence of the Stakeholder.
Strategy	Project action plan to ensure the objectives of the Stakeholder are met and that the potential Project Concerns are minimised.

A real world example of a stakeholder registry is a 2012 mapping of project stakeholders for a Kazakhstan gas plant [order-efficiency.com]. (Winter, 2012)

13 [Project] Team

A.k.a., Contributors, stakeholders, engineers, operators, personnel, crew, staff, members, participators, point-of-contact, servers.

A team is a group of people who work well together toward a measurable outcome. Teams take decisions based on expertise. A team is a group of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable. In a sense, a team is a human system of organization, of individuals, with the same intention for change to materialization. A team exists to do work [action in time]. Teams use tools, techniques, and resources to do their work. Tools include, but are not limited to methods, rules, procedures, protocols, practices, standards, materials, visualizations, diagrams, etc. In order [for a team] to do work, there must be the communication of information. A team carries out a multiple related tasks. A team is a collection of individuals working toward a common purpose through a similar structure. Teams of individuals "manage" themselves, although coordination may still be present. Systems exist to coordinate the roles and activities of teams.

Those who apply effective communication may be said to have internalized two statements that express the two axiomatic team cooperation-oriented principles. The following axiomatic team-orienting [in alignment] statements are written in natural non-formalized, and then semi-formalized, language:

> **Team statement 1:** *Nothing is done until the paperwork (documentation, logging, record keeping, etc.) is done. All teams require the return of data from an acted upon environment.*

> **Team statement 2:** *All team contributors share access to resources through a controlled and coordinated procedural information set.*

> **Team question 1:** *Can these personnel with this equipment and training perform their tasks to a specific standard under planned conditions?*

A 'project team' is those humans (and other information processing -ware systems) that are internal, in affect, to a project. The 'project team' does the 'work' (of actually developing and operating) a project.

In societal application, the 'work' always includes engineers (developers and operators). In Community, sometimes the 'work' includes the Community end-users themselves. 'Team members' are always 'stakeholders', but 'stakeholders' are not always 'team members'.

> **NOTE:** *A list of labels of anyone who could be considered internal to a project can extend from the 'complementary' (e.g., contributor) to the 'disparaging' (slave). In other words, the work on a project may be done willingly, or not (e.g.,*

through force and coercion).

Continuous projects at the highest societal-level are sustained, developed and operated, by InterSystem Engineering Project Teams. The highest societal-level continuous projects are the (i.e., at the level of a human habitat networked-service system):

1. **Life support** - Human core survival project.
   A. [Human core] Life-support service system and sub-systems.
2. **Technical support** - The human-interfacing -ware (Read: hardware and software) structure.
   A. [Interfaceable structure] Technical-support service system and sub-systems.
3. **Exploratory support** - Support service to facilitate exploration and discovery.
   A. [Human developmental] exploratory support service system and sub-systems.

13.1 Team-based lateral approach

QUESTION: *What organizational work structures are there other than teams?*

A team based lateral structure is an organizational structure that groups individuals working within the organization into teams that perform specific job functions. As the individual team member's abilities increase, so does the performance of the entire team. Being a part of an inter-disciplinary team means you are willing to fulfill your designed, selected, and assigned responsibilities.

A team-based lateral organizational structure is similar to a traditional lateral structure, in carrying less overhead management to cause delays in decisioning and implementation of best practices or new ideas. With no need to climb a lengthy chain of command to receive approval for ideas or changes to the operations model, a team-based lateral structure can make necessary changes on the fly and allow for rapid response to different conditions.

By spreading the responsibility among team members rather than having a single individual in charge of decision-making or management, decisions are arrived at and action can occur rapidly as team members can be assigned to research areas of need, implement changes, or work on other problems while other team members continue to focus on the current situation. Decisions arrived at by a team are sometimes better thought out and more effectively implemented than decisions made by a single individual. This is why the team exists, because everyone in the can work a specialized problem and trust that the others are working and taking decisions effectively in their specialized areas. A team-based organizational structure can eliminate traditional scalar chains of command, which can cause delays, frustration, and can limit an individual's choices due to a fear of reprisal.

13.2 Contribution support

A.k.a., Personnel administration.

Contribution is voluntary activity of interest toward the continuation and betterment of our community. In an extrinsic oriented society the motivation to do things is about money to provide more than destitution, to provide safety, and perhaps even, to provide luxury.

TEAM SELF-AWARENESS STATEMENT:
Good grooming can enhance self-image, improve morale, and increase the comfort and productivity of the team. An aesthetic surrounding can enhance nature connection, improve morale, and increase the felt well-being of everyone.

Because people sign up, instead of having someone assign them, management-type work is highly reduced if not eliminated entirely. In community, skills needed for a teams will be commonly available. People interested in the job can sign up to work on the team. Here, there is careful coordination between people, time, resource, and purpose.

Commonly required team skills include, but are not limited to:

1. Interpersonal skill: Is aware of, responds to, and considers the needs, feelings, and capabilities of others. Deals with conflicts, confrontations, disagreements in a positive manner, which minimizes personal impact, to include controlling one's feelings and reactions. Deals effectively with others in both favorable and unfavorable situations regardless of status of position.
2. Team skill: Establishes effective working relationships among team members. Participates in solving problems and resolving decisions. Identifies where and when action is needed, is willing to make decisions, render judgments, and take action. Accepts responsibility for the decision, including sustaining effort in spite of obstacles.
3. Continual learning: Grasps the essence of new information; masters new technical and knowledge; recognizes own strengths and weaknesses; pursues self-development; co-operates well, seeks feedback from others and opportunities to master new knowledge.

In concern to integrity,

1. Personal integrity – taking care of your physical and mental health. Personal integrity is interconnected with social integrity.
2. Social integrity – working for the sustainment of the community, reducing conflict and power structures,

establishing trust, and engineering and operating services.
3. Ecological integrity – ensuring the integrity of our ecological resources and cycles, together.

13.3 Accountability

A.k.a., Responsibility.

Accountability of tasks involves four categories of communicable information:

1. Responsible:
   A. Identify who (the role that) takes the action.
   B. If leader and follow action scenario, then decide who leads and who follows.
   C. An entity responsible for taking action (i.e., completing an activity).
   D. Responsibility level is determined by individual who is "accountable".
2. Accountable:
   A. An entity with objective decision authorization (authority).
   B. An entity accountable to an objective.
   C. Generally, only one per goal.
3. Consulted:
   A. Notified (kept "in the loop").
   B. Person consulted for input, information, insight and perspectives before a final decision is resolved.
   C. Two-way communication.
4. Informed
   A. One-way communication.
   B. Normally after activity has been performed or a decision has been taken.

The accountable person may, or may not, also be the responsible person.

13.4 Individual status

Under appropriate procedural conditions, team members are monitored (where appropriate) for their performance, which is operationalized through (note, this monitoring generally only occurs in high-risk team situations; e.g., astronauts); however it is good for any contributor to know their limits (and necessarily, the categories), thus making the team safer overall:

1. Physiological status (and, % remaining).
2. Cognitive status (and, % remaining).
3. Psychological status (and, % remaining).
4. Are team members alive, healthy, and happy?

Individual utilization metric & design efficiency:

1. Ratio of [self-]resources used per task (and, over total duration):
   A. Use of physiological resources (and, resources used/total number of tasks).
   B. Use of cognitive resources (and, resources used/total number of tasks).
   C. Use of psychological resources (and, resources used/total number of tasks).

Note: It is clear the metrics that are needed must measure the physiological, cognitive, and psychological state of the an appropriate team member (here, "appropriate is dependent on role/context).

13.5 Team decisions

Teams arrive at decisions the same way proposals are taken for changes to the kernel; the same way the decision system resolves decisions. Each proposal (i.e., solution, decision) is given a numerical score (measurement); supplementary measures are calculated. The criteria lead to the ranking of solutions. A threshold may exist beyond which a solution is acceptable and/or is not acceptable. Team members are trained to follow procedures. Machine team members are operated to follow instructions. Some procedures are more open ended in their separation of conclusions between team members decisions (i.e., the team members will reach different conclusions to the same decision), and others are more closed ended (i.e., the team members will reach the same conclusion). The intention is always closed ended solutions (of 99% certainty at its greatest).

13.6 Team indicators

Team task indicators:

1. Achievement is progress toward a set goal.
2. Knowledge is relevant material for a task.
3. Effort is time and resources used on task. Effort is time on task.

13.7 Team expectations

CLARIFICATION: *Responsibility is the essence of self-direction (or self-regulation). Accountability is the essence of social-direction (or, social-regulation). To accept responsibility people have to define, understand, and take decisions. In the market-State, the tendency is for management to hand the operational people an output of redesign thinking done by others, and expect them to work it. Expecting also, the supervisors to supervise the implementation of a design that management has completed. Alternatively, organizing for real teamwork is a process of getting everyone involved in the total systems improvement.*

For every project, the team must have access to the right tools for the right problem. Understanding the context of use for a particular technology requires asking the right questions. For example:

1. Why is this technology being used? What task and/or process is it being used to accomplish?
2. Who are the end users for this particular technology? What are the characteristics of the end user population (e.g., age, physical and mental capabilities, technical aptitude)?
3. What are the characteristics of the technology itself? What are its component features? Is it fixed or portable?
4. When is this technology used? What triggers the process/task the technology is used to carry out? At what point in the process/task is this technology used? How frequently is this technology used on an hourly, daily, monthly, etc. basis?
5. Where is the technology used? Are there any environmental characteristics – such as dust, lighting conditions, or noise – that may impact the functioning or effectiveness of this technology?

Define what is expected in terms of performance early and clearly and then support adaptations toward appropriate means by which the group can achieve ends. However, do not over-specify -- this is an adaptability principle, which recognizes that we are designing living systems rather than machines. With living systems, the same ends may be reachable by different means.

There are a lot of ways to solve problems and meet a user/community needs. What is critical here is the definition and understanding of the end goal. The "what" is to be highly specified. The "how" is open to local decision and initiative. This enables learning and an increased sense of "efficacy" on the part of team members. Efficacy is the sense that "we" are effective as a team that we can make a difference and do the work well. Efficacy is "fragile" and needs to be supported by continuous learning and improvement.

Teams have to be deeply involved to determine what and where information is needed for self-direction. There needs to be a societal (Read: community) commitment to provide information and resources for task performance and learning. Information has to be provided where it is needed for self-direction, learning, and task improvement. Control has to be subordinated to achievement.

13.7.1 Expected team requirements

To operate effectively, teams require social and or technical ability and access, involving:

1. Knowledge (concept memory).
2. Skill (behavioral memory).
3. Technology (useful material composition).

13.8 [Project] Team standards

APHORISM: *A group of people who correct one another can help one another.*

Even when internal standards are well designed, they can break down. Personnel may misunderstand instructions. They may make judgment mistakes. Or they may commit errors due to carelessness, distraction, or fatigue. Temporary personnel executing control tasks for sick personnel might not perform correctly. System changes may be implemented before personnel have been trained to react appropriately to signs of incorrect functioning.

13.9 [Project] Team categories

Any given Project Team may be assumed to be composed of all of the following three identity categories (colloquially called, "stakeholders").

Any given Project Team is composed of those individuals and systems:

1. **Who are impacted by the work**?
   A. When common heritage resources and a common ecology are impacted, then the whole human-social population is impacted to some probably observable degree of 'certainty'.
2. **Who will do the work**?
   A. The InterSystem Team of Habitat Service System 'Engineers'.
3. Who have **user/customer expectations from the work** to be, or being, done?
   A. The two accessing populations:
      1. The population of individuals, individuated units of human consciousness. Individuals among the community population.
      2. The population of engineers that coordinate and operate the materialized habitat service system. InterSystem Team members (Read: contributing individuals).

For any team structure, there are categories of organization:

1. **Manpower (humans)** refers to the number and type of personnel who operate, maintain, support, and provide training for systems.
2. **Personnel (skills)** refers to the human aptitudes, skills, and experiences required to perform the jobs of operators, maintainers, and support personnel.
3. **Training (education)** prepares personnel to perform the tasks necessary to meet the mission or goals and objectives of the system. Development of training requirements, methods, curricula, and training system design are important parts of the

overall system design process. The length and intensity of training depends on the background, ability levels, and learning styles of the personnel in the training class; the complexity of the system; and the level of skill and knowledge needed to ensure the desired level of performance speed and accuracy. Some training is designed for individual task performance; some for team or unit-level performance. Note that an important input to effective training is a task analysis that identifies the skills and knowledge needed for acceptable performance. Inadequate training can result when work and task descriptions are outdated. Training deficiencies may also result from failure to allocate the necessary training time and budget, lack of flexible training schedules needed to meet learning requirements, and lack of useful proficiency criteria.

Manpower, personnel, and system design decisions should take into account the level of training needed and the feasibility of delivering that training in the allowable time frame.

A team has two fundamental skill-sets:

1. Practical skills (do work):
    A. Practical skills, capabilities and knowledge relevant to the task.
2. Communication skills (intercommunicate):
    A. Present and receive ideas easily between team members. Ability to use a range of communication and visualization methods, and communication techniques should be well documented.

13.10 [Project] Organizational mapping

A.k.a., Organizational chartting, hierarchy chartting.

Teams belong to organizations, and both organizations and teams can be mapped/chartted. Organizational charts are the graphical representation of an organization's (or team's) structure. Its purpose is to illustrate the relationships and chains of communication (or, command in the case of hierarchies) within a social organization. Names, roles, titles, etc. are generally depicted in boxes or circles with lines linking them to other person's in the organization. By looking at the organizational chart, people can gain a quick understanding of how the organization is designed, its number of levels, and where each person fits into the organization.

13.11 [Project] Team and group personnel selection

Project teams and working groups are composed, in part, of personnel (Read: humans). The selection of personnel involves an algorithm that is highly weighted by qualification (subject matter expertise), interest (including curiosity and motivation, and effective communication. Team and group members are individuals with a strong knowledge (and skill) in the subject matter, and also have the ability to understand and be open to multiple points of view.

In concern to qualification, a team is significantly composed of individuals using tools. If a tool user doesn't understand the correct use and safe operation of a tool, then the user can hurt themselves and others. Tools are useful to the extent that the user understands their operating capabilities and safety parameters (or, the degree to which they provide certainty and uncertainty). Any mechanic or philosopher will tell you that tools can be used in a wrong way. How a tool is used is often more determinant on the outcome than the fact that it was used.

Working groups nominate experts (or members) who have requested nomination to participate on teams. The algorithm (software), council (project coordinator team or technical council), or vote, then selects the nominees. This same process can be used for the selection of working group members themselves.

All working group and team members have the responsibility to attend working group and team meetings. A member of the public may attend and observe, but not participate, in working group meetings.

A working group may choose to invite other individuals with special knowledge and expertise related to the priority issue to attend meetings to provide information and/or advice. Advisors will be encouraged to participate in discussions, but shall not participate in the decisioning of the working groups.

> *"We become what we behold. We shape our tools, and thereafter our tools shape us." [The presence and use of tools and technologies affect how we look at the world and how we behave. Think about how airplanes change your perspective on distance.]*
> *- Marshall McLuhan*

13.12 [Project] Team member attributes

Any given project [human] team member has the properties of personality and capability. Team member personality ("attitude") and capability ("skills") are controllable at the team level by means of team composition (Read: staffing a team with certain individuals possessing the specific personalities and capabilities desired). Education and training are self-development activities aimed at improving certain personalities and capabilities before or during projects.

APHORISM: *Those who receive a service from me, receive fulfillment without lessening mine, receive light, without darkening mine (Read: a true social contribution model).*

A [project] team requires individuals with intelligence and ability to compose and to operate a set of components that work effectively.

13.12.1 Role

A **role** is a task related to a function. A role is the continuous carrying out of specific tasks inside a temporal [project] context.

13.12.2 Personality

A.k.a., Intra- and inter-personal composition, attitude.

Whereas personality is all those feelings, thoughts, affects, desires, language, and ideas that expressed, or likely to be expressed, by an individual human. There are some InterSystem positions where personality type is a requirement and there are personality sub-elements (e.g., feelings and language) that are unacceptable given that InterSystem team position. For instance, a nurse must have a personality that is likely to positively influence, and unlikely to negatively influence, the well-being of someone whom they are treating. Personality is required for efficient task completion - the task is unlikely to be completed well unless the individual completing it has the desire and personality structure to do so well.

13.12.3 Capability

A.k.a., Skill, ability.

Team member "skills" are the requisite abilities held by individual team members, which enable them to complete their tasks within the team setting. Capability is required for effective task completion - the task cannot be completed unless it is known how to complete the task.

Teams are partly composed of individuals with capabilities:

1. Technical skills (technical abilities, capabilities).
2. Social skills (interpersonal abilities, capabilities/ feelings).

Social skills include, but are not limited to, the ability to:

1. Perceive another's point-of-view.
2. Involve others in the work process.
3. Understand the technical and organizational constraints the team must confront.
4. Work collaboratively.
5. Follow protocol.
6. Share.

13.13 [Project] Team organization

Naturally, a team [work] organization is an identity in which the activities of individuals are coordinated, motivated, and supporting between each other in order to reach some common target or goal (i.e., the completion of a set of requirements formed from objectives) that requires work and structure.

13.13.1 Team work **organization**

A.k.a., Team organizational structures.

The computational sub-structures of team include, at a high-level:

1. **Role structure** - people interact based in how their roles are supposed to interact as part of an explicit organization. In a role structure, people know how to interact be abuse their role and its relationship to others roles are define.
2. **Team structure** - the roles are nested in a team structure.
3. **Organizational structure** to support coordinated adaptation.
4. **Control structure** to determine how to keep this distributed set of people in sync as the plan is evolving.
   A. *Use version control* to enable reconfiguration of the organizational structures:
      1. Branching - a branch is a copy of an organization that is referenced back to a point in time)
      2. Merging - a coming of two into one.

In an open and contributive organization, any member if the organization can branch, edit, and make pull requests against any organizational structure: roles, teams, tasks. Pull requests are reviewed within a core decisioning framework, and if selected (as the solution) will be merged (e.g., in GitHub through a three-way diff).

13.13.1.1 The dimensions of team organization

The common team organizational dimensions include:

1. Priority.
2. Interaction patterns.
3. Values (norms of engagement).
4. Decisioning logic objectives.
5. Feedback.

13.13.2 Team work **co-operative organization**

A.k.a., Cooperative work environments.

Access to information is available to all team members who can see the same instance of information as other team members, provides a single, unified source of awareness with which to engage together.

A continuous information system means that all digital data can be connected and every piece of digital content can be made aware of all other digital content. Therein, modifications can be more predictably visualized.

13.13.3 Team work **co-operational knowledge**

A.k.a., Social cooperation knowledge areas.

Project-level social coordination and collaborative action require the following necessary operations on the part of individuals whose interests and/or actions are interrelated:

1. **Cognition** of a problem-solution, the project.
2. **Visualization** of the [problem] situation.
3. **Coordination** of a solution.
4. **Communication** of a plan of action to execute the solution into realization (into real-time).
5. **Execution** *(realization)* of the plan by means of development (design and construction); the ability to execute the task at a certain level/condition of performance/quality.
6. **Evaluation** of the execution and results (accountability alignment with pre-decisions).

13.13.4 Team work **recursive operations**

Whereas, 'operations' means the work (or tasks) done, 'design' means the work (or tasks) to be done in a future operation. Note that this set is recursive, because doing the work of determining what is to be done in a future operation is itself work (or, a task).

In other words, the recursion (recursive reason) for understanding the Inter-System nature of the Teams that create and maintain a working human fulfillment service system is:

1. **Operations** - the work/tasks done (as visualized).
   A. **Operations are designed.**
      1. Note: the "Inter-" part of InterSystem Team.
   B. **A system is operated** through a design.
      1. Note: the "-System" part of InterSystem Team.
   C. **A team is operated** through a protocol.
      1. Note: the "Team" part of InterSystem Team.
2. **Design** - the specific[ation] plan (as visualized).
   A. **Design is an operation.**
      1. Note: the "Inter-" part of InterSystem Team.
   B. **A system is designed** through an operation.
      1. Note: the "-System" part of InterSystem Team.
   C. **A team is designed** through an operation.
      1. Note: the "Team" part of InterSystem Team.

13.13.5 Team work **planning activities**

The project team has responsibility for conducting project activities, which may be viewed from the 'work' perspective (information set) through two methods:

1. **The checklist** [method] to visualize ("tell") the team member, *what to do*.
   A. Identify tasks [through tasking, as in, the accountable itemizing of a 'work' function].
2. **The schedule** [method] to visualize ("tell") the team member, *when to do it*.
   A. Relate tasks to time [through scheduling].
3. **The plan** [method] to visualize ("tell") the team member, *what 'it' (i.e., that with shape) is*.
   A. Relate objective to task and time [through planning].

Team members have a continuous interest in observing the state of the project. Therein, team members have (or are likely to have (because, they have an interest in the project): 'Is' questions about their project, such as:

1. Who is asking for the project?
2. Why is the project asked for?
3. What is the expected outcome from doing the project?
4. Who is affected by doing the project?
5. When is the project being done?
6. How is the project to be done?

These 'is' questions comprehensively relate the project objective to the task (work) and time, and are thus, inquiries that compose the information space/set of a 'project plan'.

13.13.6 Team work **communication structure**

In a contribution-based team setting, a 'hierarchy' is having a centralized point of communication between systems -- out of all possible entities that could communicate, one is selected for efficiency. A complex system is one that has multiple levels in a hierarchy of systems, with each level being composed of sub-systems that may themselves be further decomposed into sub-sub-systems; herein, a common team communications structure becomes a requirement for optimality.

Systems teams, in concern to their communications structure, can be identified as being:

1. Simple - For example, the InterSystem Team runs all societal-level operations.
2. Complex - For example, the system teams, of which there are three core, each with multiple sub, operate an information and material network.
3. Complex adaptive (complex adaptive system, CAS) - For example, the InterSystem team operates a second-order cybernetic information system (closed-loop control), which integrates issues and feedback while continuously resolving the most

up-to-date information and material system. All living organisms and ecosystems are complex adaptive systems. Complex adaptive systems are represented by genera (species evolve within), the human being as structuring societal organizations (such as, habitat service systems and corporations).

13.13.7 Team work **influences**

The most common influences on a work team (e.g., the InterSystem team) are:

Individual influences:

1. Attitude/feeling change.
2. Salience.
3. Elaboration.
4. Priming.
5. Knowledge and skill acquisition (training).
6. Behavior change.

Interpersonal (social) influences:

1. Reasoning (justification, logic).
2. Protocols (societal protocols and social norms).

Societal (individual and interpersonal) influences:

1. Organizational structure (unified information system structure).
2. Protocol and structural change (decisions).
3. Diffusion (of information).
4. Access (to resources, services, goods).

13.13.7.2 Manipulation

Common methods of manipulation, of which team members should be aware, include:

1. Logical fallacies (spurious reasoning).
2. Thought-stopping.
3. Goalpost-shifting.
4. Double bind.
5. Idealisation.
6. Intimidation.
7. Shaming.
8. Isolation.
9. Repetition.
10. Denial.
11. Infantilization.
12. Demonization.

The usage of a method of manipulation, itself, does not mean that the information attempting to be propagated is false.

13.13.8 Team work **structure**

INSIGHT: *There is a lot to being a person and there is a lot to being a person who contributes to society; which needs guidance.*

Common team structures include development and operations:

1. Project development work structure.
   A. Traditional development life cycle.
   B. Critical development (i.e.,traditional development sped up).
2. Project operations work (system operations life cycle).
   A. Centralized - the primary InterSystem team members are on the same work team (e.g., responding to an incident).
   B. De-centralized - the primary InterSystem team members are on different work teams (e.g., maintaining a routine energy system).
   C. Specialization - how specialized is the work group (degree of specialized skill set and variability among group members)?

13.14 Team coordination

Team coordination and collaborative action require the processes of forming, executing, and dis-forming a team involves, and involve:

1. Planning human contribution.
2. Planning resource availability.
3. Acquiring a project team.
4. Coordinating a team effort through time.

Functions of the team coordination (management) process are:

1. Coordination of information: Coordination of information is the fundamental concept of acting upon information.
2. Identification of information: Identity is the fundamental concept of uniquely identifying an object (person, computer, etc.) within a context.
3. Authentication of information: Authentication is the process of objectively ensuring trust and accountability (i.e., gaining confidence) in a claimed identity. Once identities are issued, whenever they are used, there is the requirement that the person using the identity is the person that is qualified to use it. This process minimizes decision violations (i.e., in this case, identity "theft").
   A. Revocation is the process of rescinding (i.e., "withdrawing") an identity that has been authorized. This is a process that must be properly recorded for accountability (i.e., transparency/audit) purposes. All systems and processes with which identity has been

established must now be notified that that identity was revoked. This is required to prevent continued use of the identity under potentially false and insecure contexts.

B. Authoritative [control] source(s): An authoritative controlling source exists in an organization to resolve the problem of authorization (i.e., identity formation). From a best practices and manageability perspective, it is important for an organization to make one authoritative source the main source of identity information.

C. Authorization is the category (label) to which a person or an operational entity is assigned, having gained access (i.e., authority or permissions) to do an operation or task (with a set of resources and tools). Authorization is the name of the process where requests to access a particular resource are granted (0, “go”, True) or denied (1, “no go”, False). An authorization is where the system controller (e.g., administrator or protocol) translates a user’s (or a specific group or class of users) request to access a designated set of system resources into a resolved decision. It should be noted that ‘authorization’ is not equivalent to ‘authentication’. Authentication is providing [“me”] and validating [“me”] identity. Authorization includes “me” as a variable in the decision resolution logic (i.e., execution rules) that determine what access systems the user may access, ensuring the accurate decisioning of access after authentication is successful. Service applications need access controls to allow users (with varying privileges) to use the application.

4. Provisioning of users.
   A. Account provisioning (a.k.a., user provisioning) - identity-related information associated with individuals in the unified system. Provisioning has the following functions (i.e., functional processes):
      1. The process of providing users with accounts, the appropriate access to those accounts, all the rights associated with those accounts, and all of the resources necessary to manage the accounts.
         i. Adding an identity: Initially, the identity may never have existed. As credentials of the identity are known and collected, the identity is then added, checked against the authoritative source, and the identity is then provisioned to required systems and services.
      2. Changing the status of an identity: The process of modifying (i.e., assigning, granting, changing, or removing) user access to systems (applications and databases) based on a unique user identity by creation of user accounts on target systems.
         i. Password coordination [password management].
         ii. User access premissioning.
         iii. Analytics and reporting.
         iv. User provisioning and de-provisioning.
      3. Modifying an identity: When an identity exists within an organization in which it has been provisioned and a change (e.g., merger/acquisition) occurs, the identity’s credentials may require review and adjustment in light of changes to the provisioning system’s workflow.
   B. Account de-provisioning - deals with the termination of access rights to systems and services and re-allocation of those systems and services The de-provisioning of identity is the termination of the identity that had been provisioned to services and systems. De-provisioning is critical for organizations to review and assess because accounts that are not de-provisioned in an accurate and (especially) timely manner, lead to considerable risk.
      1. Deleting an identity: Covered under De-Provisioning below.
      2. Suspending an identity: Suspending the identity basically represents the temporary halt of access to systems and services provisioned to an identity. The identity(s) are then suspended, thus suspending access to respective systems and services.
      3. Resuming an identity: Once the identity comes back the identity’s state will be resumed and appropriate resources will be reassigned.
5. Provisioning of resources.
   A. Resource provisioning - assets such as computers, databases, and applications and the management of permissions associated with those assets. Resource provisioning is the provisioning of identities to systems and services that the identity has the approved access to use.
   B. Resources may be classified as the following types of systems (and services, in the HSS context):
      1. Material (i.e., habitat, physical environment).
      2. Non-material (i.e., digital, informational “abstract” environment).

3. Computing (i.e., computational).
4. Non-computing (i.e., non-computational).

13.14.1 Team work **tasking coordination**

Teams complete tasks by coordinating among the factors of:

1. **Accountability** - individual responsibility for an organizational output.
2. **Communication flow** - the paths of relationships that get work understood and done.
3. **Priorities** - the timed structure of actions to ensure the purpose of the organization is me.

Team coordination, thus concerns:

1. Identity coordination (management).
2. Access coordination (management).
3. Schedule coordination (management).

The coordination of individual team members, their authentication, and access occurs within the habitat service system, across [Inter]system boundaries InterSystem Team [service] work positions involve:

1. Service role.
2. Service responsibilities.
3. Service tasks.

Here, a 'work package' is the logical package that makes up work, as a task(s), to be complete.

13.14.2 Team work **budget monitoring**

A.k.a., Tracking team work and budget.

Tracking and analysis of team work:

Formula: Earned value (EV) = % of work complete (actual) x task budget (at completion).

1. Obtain physical % complete for each task
2. Calculate "earned value" (EV) for each task.
3. Sum up "earned value" (EV) for all tasks as project "earned value" (EV).
4. Calculate actual expenditure for actual work complete during the period.
5. Compare the cumulative "earned value" (EV) to actual expenditure.

Actual cost (AC) is the total cost incurred for the complete work to date. Actual cost is also known as actual cost of work performed. The actual cost doesn't have any formula.

13.14.3 Team work **task dispatching**

A.k.a., Job dispatching

Dispatching refers to process of entering a task for the purpose of execution. Job dispatching is a procedure that uses logical decision rules to select a job for processing on a machine that has just come available.
Dispatching consists of two elements:

1. Decision (for selecting task for a workstation from those predefined tasks that are ready for execution),
   - Communicating the assignment (or authorization) to the workstation.

In the case of project coordination, the decision is largely taken care in planning , and thus dispatching is reduced to mere communication of the notification to start work.

13.15 What is 'optimal performance' as part of a team?

Optimal performance is highly qualified and contextual (individual, time, place, and situation). Optimal performance is doing activities "you" (Read: the individual doing the activity) have first deemed intrinsically worthwhile. Continual (or regular) improvement is almost certainly part of optimal performance, but continuous doesn't mean regular, it means incessant. The blind pursuit of continuous improvement can often result in restlessness and inefficiency. Optimal performance requires the alignment of desire, ability, and opportunity towards an optimal goal - a goal whose value is recognized and embraced by all involved.

The three social organizational characteristics (values) of optimal performance:

1. **Effectiveness** - services, products, and individuals are effective if their task (activity, job) is completed as functionally and non-functionally required (as expected).
2. **Efficiency** - services, products, and individuals are efficient if tasks are completed within the pre-determined boundaries of time, resource, personnel, and systems.
3. **Sustainability** - services, products, and individuals are sustainable if they can continue to do tasks as required (or, expected).

   Note that these values are described in greater length in the Social and Lifestyle System Specifications.

The two core performance questions are:

1. Desire
   A. What is needed to be done; what is the objective, given what is known?
2. Ability

   A. What can be done; what is possible, given what is available?

The three social organizational questions for performance are:

1. Goals.
   A. What is required to be done?
2. Skills.
   A. What are the individuals trained to do?
3. Systems.
   A. What are the organizational systems set up to do?

13.15.1 Individual, personal accountability

As part of the InterSystem Team, there is individual, personal accountability. When an individual, as part of the Team agrees to do some task, s/he is held accountable to doing it, as agreed upon by the work package and scheduling (registry) of his/her identity. It is the scheduling of identity into the "block chain" to complete some task of benefit in service to everyone, that generates societal accountability. In other words. to work on the habitat service system, "you" must be a part of the InterSystem Team, for which "you" will join a sub-team of your choice constrained by the task's requirements, and "your" own physio-cognitive set. The scheduling of "your" identity as part of an InterSystem Team involves the association of several [technical-value] attributes, most notably, 'accountability'. When active as part of an InterSystem Team "you" become accountable for your behavior and its timing to the totality of society. Some cultures might find this thought appealing and others horrifying; nevertheless, it is a requirement for fulfillment, because it is a requirement for monitoring progress toward the fulfillment of a given need or other objective. A transparent society, when oriented in an independently experienceable way toward fulfillment, may be shocking to consider, but its experience is the expression of the fulfillment we all desire. Hence, "you" become accountable to the community, for "you" are working on some aspect of everyone's fulfillment service system. And, you are working as part of a project team that, because "we" all are interested in the project, have the degree to monitor its progress. InterSystem access is available as 'read' access to everyone in the community. Sensors are used here to monitor activity; this includes inquiry sensors (i.e., surveys and "senses", which are surveys of a humans senses). This includes humans and instrumentation.

13.15.1.1 Role accountability

A.k.a., Service roles.

There is purpose for the existence of anything [in the habitat service system for human access fulfillment] - from the purpose of human life together, to the purpose of any service. Purpose typically derives from tasks that something is carrying out. The continuous repetition of carrying out a certain task results in the attribution of a role (or program, in software; mechanism, in hardware). Within society, the inheritance of a role over time is accepted as part of someone's personality.

The model of role can be applied correspondingly to machines. For example, the purpose of the machine-type 'refrigerator' is to keep food freshly preserved; its purpose in not to cool - the purpose is to keep food fresh. Everyone who eats wants their food to be kept freshly preserved. The purpose of the refrigerator of keeping food fresh derives into the task of maintaining food at a lower-relative temperature, which keeps food fresh, and is the purpose of refrigerator.

Activities based on tasks can result in needs (i.e., additional or secondary requirements). The need of a refrigerator is keeping its door closed in order the keep the temperature at a set level, efficiently. Another need is staying connected to the electrical power to keep its compressor running.

In terms the larger societal information system, an accountable person is accountable to monitoring and controlling some formalized aspect of a [service] system. There is the ability for humans, when adopting roles, to have specialty information and/or ability.

For example:

1. Geologics - someone skilled in geological systems.
   A. Biologics - someone skilled in biological systems.
   B. Mechanics - someone skilled in mechanical systems.
   C. Electronics - someone skilled in electronic systems.
   D. Informatics - someone skilled in information systems.

13.15.1.2 Test engineering

Test engineering (test engineers) test is to check whether something will work or not. A test should be done to prove that something will work. To test a hypothesis is to check whether it is true or not. Some 'test engineers' may specialize; for example, some may be skilled in geologics, where they design and test (by role) mostly geological sub-systems, other 'test engineers' may be skilled in several fields and be capable of designing and testing more integrated supra-systems. Note here that scientists require more specialization than engineers, because the engineers are applying (a horizontal calculation approach), versus scientists who discover and understand the whole reality information system in order to do all engineering safely.

13.15.1.3 Accountability assignment matrix

A.k.a., Accountability visualization.

An accountability assignment matrix is otherwise known as a responsibility assignment matrix. This matrix is simply a table for which one axis is the project's Work Breakdown Structure, and the other axis is the project's organizational breakdown structure. Each point at which these two structures intersect becomes a work execution element, and an individual or system is identified who is responsible for executing the work. If desired, each intersection can also identify the value of that specific element of work in terms of information and physical resources, time (hours), and financial resources (cost).

As a tool, the Accountability Assignment Matrix maintains the following service goals:

1. It serves as input for identifying, planning, progressing, and reporting (recording) work.
2. It serves as input for developing budgets, schedules, and milestones; tracking costs and spending; and preparing progress reports.
3. It identifies individual work responsibility.
4. It controls the release of access to resources by Inter-System Team Contributors.

Human limitations:

1. Humans have very limited short-term memory: 5–7 items.
2. Humans make mistakes, especially under stress.
3. Humans have widely varying capabilities, both physical and mental.
4. Humans have widely varying personal preferences.
5. Humans brains organise their perceived world differently.

Society requires a functional [accountability] matrix organization structure for the InterSystem Team organization. An inter-system team structure necessitates a unified matrix-type organization of effort-accountability.

13.16 Functional teams, functional information society

In a functioning societal information system, societal-level projects are organized by function -- by functional InterSystem Team organization and the solution's expression as a functional habitat service system (Read: network of cities) In its principal application within the habitat service system, functions are called operational processes or [service] procedures (both are equivalent).

The functional groups responsible for the fulfillment of societal-level organizational requirements include:

1. A system development group (strategic planning; organizational project plan decisioning).
2. A system realization group (engineering development).
3. A system operations group (engineering operations)
4. An information system [operations] group. (information service engineering operations).
5. A material system [operations] group (material service engineering operations).

The system realization group is divided into a hardware- and a software branch, which are subdivided into development teams, each responsible for a set of modules. The organization has defined roles responsible for each module. These persons work with function groups during specification and development teams during implementation.

13.17 Societal InterSystem team

A.k.a., Societal interdisciplinary team.

InterSystem teams have accountably tracked access to the engineering system of society. In order to trace access to the engineering system, the whole, unified information and material system must be indexed and searchable; if it can't be indexed or searched, then it doesn't exist.

The value of interdisciplinary teams has long been recognized in many fields, including particle physics, astrophysics, and other "big science" disciplines. Interdisciplinary team systems science broadens the scope of investigation into problems, yields fresh and possibly unexpected discoveries, and gives rise to new inter-disciplines that are more analytically sophisticated.

In concern to the interdisciplinary nature of societal operations, to cut off a single field, any field from the rest of cognition, is to drop the vast context that makes that field possible and which anchors it to reality. The ultimate result, as with any failure of integration, is floating abstractions and self-contradiction, and social conflict. Potentially generating a form of compartmentalization with respect to values, desires and logical self-interest, by the compartments of personal and political life. Instead, relating one context of knowledge to another is necessary for integration. Reality must be viewed as a whole in the formation of concepts that indicate aspects of reality. Percepts are basically self-evident, things that we do not choose to integrate or not. They are just there. The process of reasoning is taking those percepts and integrating them in concepts to delineate things, to find distinguishing characteristics in reality. This is not an arbitrary process as the subjectivists contend, which undermine our ability to comprehend things objectively.

NOTE: *In community, there is an emphasis on InterSystem (interdisciplinary) understanding, as if all fields are connected.*

In InterSystem Team operations and in engineering in general, there should be no subjective interpretation of words or phrases, particularly in specifications, as this can cause major issues. If subjective interpretation is possible, then sufficient reasoning should be present to ensure that qualifications reduce interpretation to satisfactory levels.

Second only to the abilities and collaborative nature of the people in a group is the goal of the group. It is important for the group to have a common, well-articulated, and meaningful goal. This goal can range from a relatively narrow and finite objective, to a broader, longer-term goal. The actions of forming, discussing, and refining the goals of the group help the team create an identity, foster participation by team members, reinforce the participants' desire to contribute, and ensure that individual efforts are aligned.

It is necessary to differentiate an overall sense of teamwork from the task of developing an effective intact team that is formed to accomplish a specific goal. People confuse the two team building objectives. This is why so many team building seminars, meetings, retreats and activities are deemed failures by their participants. Leaders failed to define the team they wanted to build. Developing an overall sense of team work is different from building an effective, focused work team when you consider team building approaches.

Some InterSystem teams may be self-selected, and others may be selected and organized by a Central Selection Program, based on what they have acquired as skills, or already contributed to the system. This is a true "election", based on what a person has done (contribution and education), and not what they say they will do. For example, some randomly selected team in the power service system may be self-selected by its team members, but the first team to pilot a craft to Mars will be program selected based on profile and skill. When there is team selection present, selection is always based on what a person has done, not what they say they will do. It is not everyone's input that is desirable, but rather the input of those who have proven their skills and expertise in some way that would lend solution to the given problem. Under program selected conditions, selection is based on what a person has done, not what they say they will do (too many contributions necessitate filtering and selection of candidates).

An environment of mutual tolerance is critical for an interdisciplinary team to be highly functional. In particular, when a team comprises diverse levels of expertise and many different disciplines, it is essential that all team members are comfortable raising issues, questioning ideas, and fully participating in discussion without fear of being ridiculed or having their ideas discounted. Only when open communication and a high level of respect are present do all of the team members feel comfortable freely sharing their ideas. The leader of a great interdisciplinary team also has to earn the respect of the members, and the team expects their leader to be absolutely trustworthy where the project is concerned. The stronger the culture of mutual respect, the higher the likelihood that everyone will thrive. Another result of mutual respect is that it helps to reflect the value of each team member of the group, regardless of their level of responsibilities or experience. Members of a group who feel valued are more likely to be committed, creative, and contributory, and a group in which each member is respected and valued is much more likely to produce great work.

A team can only function optimally if the members can effectively communicate among themselves, especially under potentially stressful conditions. Sub-teams exist to address the critical pieces of a system. Crucial to the sub-team development and individual staff is the clear delineation of roles and responsibilities within the team. With good communication skills team members are able to define and negotiate (Read: arrive at a consensus) with other team members the roles that each are expected to fulfill within the team context. Team members are asked to write their own role description and bring it to the larger team for discussion and negotiation. Providing an environment where these roles are continuously reviewed and re-negotiated is understood to lead to higher satisfaction, and, likely, more efficient and effective decisions.

Teams use a Team Measure (unpublished, in development) survey instrument to measure team attributes and provide "teamness" feedback. This measure has helps to understand that team attributes are clustered around four domains of team development that appear to have a developmental or hierarchical structure. These domains are cohesiveness, communication, roles clarity, and goals-means clarity. The team attributes within these domains have consistently been observed as the teams develop. Providing feedback to the team on their level of development has allowed them to strategize about how they might improve team processes.

Because of the complexities of the conditions of a society, numerous processes are needed to support that operation, the sub-teams address the critical pieces of the operational process expressed in the material form of a service. In addition to sub-teams based on system/discipline, such as wiring technicians, physicians, and ancillary services, specialized sub-teams exist.

Teams can be formed for various purposes. The purpose of the team can often impact the way in which the team is structured.

Team human factors include, but are not limited to:

1. Personnel - humans with capabilities and demands.
2. Tasks - work to be done.
3. Equipment - tools used to do the work.
4. Environment - where the work is to be done.
5. Schedule - when the work is to be done.
6. Specification - what the work is to be done.
7. Procedure - how the work is to be done.
8. Effort - time (and/or resources) on task.
9. Effectiveness inquiry (global decision system inquiry) - safety of task.

Team structural elements include:

1. Communication.
2. Boundary maintenance.
   A. For material systems: Aside from possible internal ware, the boundary between the system and its environment often degrades quickest.
   B. For team systems: A team exists for a purpose and must maintain its boundary its purposeful boundary to ensure it remains efficient and effective toward its purpose.
3. Systemic linkages and internal dynamics.
4. Standardization and procedures.
5. Social coordination.
   A. Intra.
   B. Supra.

What are the 'resources' the InterSystem team has access to, and potential control over?

1. Intermediate economic services ("goods") may be resources.
2. The basic resources such as materials and energy are taken from nature.
3. Human effort, as contribution, could be a resource. Resources human physiological energy energetic component.

The InterSystem Team must have significant depth and breadth of technical expertise to review, evaluate, and operate a significant majority of design considerations. Areas of technical expertise necessary for proper Habitat InterSystem operation include, but are not limited to:

1. Human factors and human engineering (including crew workload and usability, human-in-the-loop evaluation, and human error analysis).
2. Human health and restorative measures.
3. Environmental health.
4. Safety.
5. Systems engineering.
6. Human functions and habitability functions (including nutrition, acoustics, water quality and quantity, etc.; i.e., the subsystems themselves - architecture, fabrication, computation, etc.).
7. Human interfaces and information systems.
8. Maintenance and housekeeping.
9. Materials cycling.
10. Exploration operations.
11. Mentoring and training.

A project's view of society as a habitat service system may include:

1. Project team size.
   A. The team is composed of x members.
2. Project duration.
   A. The total duration can range from [identify on schedule], if the information is known.
3. Project requirements.
   A. The total (or evolving) set of requirements to be completed by intersystem team human members and/or machines.
4. Early termination of project.
   A. The project can terminate when the lowest level of project success criteria is met.
5. Role of habitat service support operations in project.
   A. The habitat is a controllable, real-time sensitive operational potential. It is the role of intersystem teams to coordinate the real-time controlled operation and coordination of the global habitat service support system.
6. Human habitation.
   A. Human habitation capabilities include the multi-purpose, integrated habitat service module (i.e., A city) duplication and operation.
7. Sample return.
   A. All monitoring for demand or hazard must be performed transparently.
8. Project team timeline.
   A. The timeline and schedule for a project.

13.17.1 Socio-technical contribution

A.k.a., Contributor, technician, engineer.

An 'InterSystem Project Team' is a project team because they have the knowledge and skills, and have been contributively assigned accountability for some particular role.

There are two general types of socio-technical contribution:

1. Technician scientists (sometimes technicians, sometimes not).
2. Technician operations (technicians).

Any contributor on an InterSystem Team is a technician/engineer. However, technician scientists can also be open source contributors anywhere within the Community, and not part of the InterSystem Team engineering-technicians service.

In Community, there is an integrated socio-technical team to coordinate, develop and operate, the societal system. That team of consists of individuals (who may at the base level be considered 'designers', the human InterSystem Team) and computers (who may at the base level be considered computational InterSystem Team service support systems). The team optimizes their environment (mostly, cities) through intentional algorithmic thought (i.e., through the intentional design of a socio-decisioning protocol).

13.17.2 Socio-technical team viewpoints

For any coordinated socio-technical, there are multiple viewpoints (information sources) through which work is coordinated.

The three team-oriented views in community are:

1. **The community user's view** - the view of any given individual user of services in the community.
2. **The InterSystem team's view (technicians, engineer's view)** - the view of any given contributor, who is part of the InterSystem team, and developing or operating a community service(s).
3. **The unified information systems view** - the view of the whole, unified information system (i.e., the view of all information sets, as much as possible).

There are four project deliverable (work output) viewpoints (information sources):

1. **Time view** - what is Δt for actions in planned sequence (i.e., has temporal possibility for experience?).
2. **Location view** - what is physical coordinates for time-bound actions with resources (has material possibility for experience).
3. **Resource view** - what is material composition for physical-bound objects (has touch, interfaceable?).
4. **Service view** - what is functional usage for object-bound relationships (i.e., real world entity-objects; has shape?).

There are three project deliverables (output information sets) for any socio-technical system:

1. **Project [information set] viewpoint** - the relationships between operational and capability requirements, and the various projects being implemented. The project information set visualizes dependencies among capability and operational requirements, system engineering processes, systems design, and services design within time.
2. **Services [information set] viewpoint** - the design for solutions articulating the expressed system (including: actors, controllers, performers, activities, services, goods), and their input-output resource transfers between systems, all of which provides for supporting operational and capability functions.
3. **Systems [information set] viewpoint** - the design for solutions articulating the [service] systems purposeful[ly expressed] existence, their composition, interconnectivity, and context providing for or supporting operational and capability functions.

The engineering viewpoints on a project are:

1. **Technical standards view** (TV, knowledge data added) - a set of deliverables (information sets, products) that define technical standards, implementation conventions, rules and other prototypical criteria for the design and/or operation of systems. Note that when a technical standard is applied to operations (to be executed at some time), then it is generally called a 'protocol' or 'procedure'. Protocols and procedures are perceived within this view. Known safe ways of designing and constructing systems.
2. **Operational view** (OV, time data added) - a set of deliverables (information sets, products) provide descriptions of the tasks and activities, operational elements, and information exchanges required to accomplish the intended direction [of change]. Standardized ways of co-operating [service] systems.
3. **Systems and services view** (SV, location data added) - a set of graphical and textual deliverables (information sets, products) that describe systems and services and interconnections providing for, or supporting, directional functions.

In a community-type society, the principal systems and services view is that of the local habitat service systems (cities), which form a globally network habitat service system (Read: city network). SV data focuses on explaining (reasoning/justification) how the purpose for specific actualized systems with specific physical and/or digital (hybrid) locations is met by objects and relationships (often through UML). The relationship between data elements across the SV to the OV can be exemplified as systems are developed and operated to support individuals and organizations (their operations).

The unified project-engineering viewpoint:

- **All view** - a view that provides a unified, integrated, whole, overarching description of the life-cycling system (i.e., the whole, socio-technical, information-material life-cycling system).

The common supplemental viewpoints that ensure an accurate alignment of understanding include:

1. **Capability viewpoint** - articulates the capability requirements, the delivery timing, and the deployed capability.
2. **Data and information viewpoint** - Articulates the data, data relationships, data alignment in a structural format that expresses content for

the capability and operational requirements of a system through system engineering processes, and systems and services tools and techniques.

13.17.3 [Societal] InterSystem team work **rotation**

A.k.a., Individual rotation, team rotation, etc.

When there is intention and consequences, then there is a team [of "stakeholders"]. Teams work with information and material that are integrated into a materially extant system in which 'life' exists (i.e., living organization occurs -- there are living systems). Components of the living organization come together to form a team organization [to most effectively and efficiently fulfill]. The team has the potential [capability] of recognizing the discoverable information-base of existence. The team then has the potential of optimizing the organizing its societal information construction system for highest fulfillment of each and every individual. Within any given organization, work is scheduled out to project teams.
Project teams deliver projects.

NOTE: *Teams function optimally when they do the right then at the right time.*

13.17.4 [Societal] InterSystem team work **effectiveness**

A.k.a., Team effectiveness.

In order to be effective at scale, and hence at the societal InterSystem level, teams must have the following:

1. A shared understanding of the situation.
2. A shared direction.
3. A shared orientation.
4. A shared approach.
5. A shared informational environment.
6. A shared material environment.

Fundamentally, in order for teams to execute solutions effectively, the two team must work off of a single specification for the work.

13.17.5 [Societal] InterSystem team work **roles and responsibilities**

A proper functioning socio-technical system requires the co-ordination of the actions of all roles involved in operation.

The core InterSystem team organizational (structural) role is:

- Technician (a.k.a., socio-technician, engineer, operator) - a technically skilled contributor.

The primary roles involved in operations are:

1. **InterSystem contributing teams (technicians, engineers)** - assigned work for the coordination, development, and operations of the whole societal system through individual contribution upon an InterSystem Sub-Team. There are three conceptual dimensions of contribution to an InterSystem Team; three separately together functional roles. Coordination is an operation that sustains all development and operations. Coordinators coordinate the optimal allocation and timing of all resources and access. Developers test and develop the next iteration of the whole societal system. Operators execute upon the selected societal solution, to either implement a new solution or serve some humane function within the societal service system.
   A. **Coordinators**.
      1. Socio-parallel [project] decisioning - coordinate societal resource access decisions in alignment with a value orientation.
   B. **Developers.**
      1. Socio-technical [solution] decisioning - design societal service systems composed of resources in alignment with optimal socio-technical safety standards.
   C. **Operators.**
      1. Socio-technical [solution] executioning (Read: execution decisioning) - operate service system for user through a standardized optimal procedure.
      2. Recursive (all roles are sources of 'operator' information).
2. **Individual human accessors** - selectively access services (and service objects) as the outputs of InterSystem Contributing teams.
   A. **Users (the Community of)**
      1. Usage of end service, or service object [as designed and operated].

13.17.5.1 Professional team roles and responsibilities

The common "professional" roles and responsibilities of teams designing, developing, and operating integrated service systems include at a high-level, but are not limited to:

1. **Issuing entities** - The individual(s) with an issue that instantiates the requirement for a project.
2. **Developers** - The individual(s) whose responsibility is the development of the system for the project.
3. **Information analysts** - The individual(s) skilled in resolving [societal] information inquiries being used in the project.

4. **Definition analyst**: The individual(s) skilled in the development and definition of the computational controls of the project environment.
5. **Leads**: The individuals accountable for all aspects of the system design and construction.
6. **Subject matter expert (SME)**: The individual(s) and system(s) who have the knowledge and skills necessary to implement the project.
7. **Project coordinator**: The individual(s) responsible for all activities of a project. The project coordinator plans, controls, and coordinates a project.
   A. **Quality assurance analyst**: The individual system who audits and approves project deliverables from a QA perspective. Reviews plans and deliverables for compliance with applicable standards. Provides guidance and assistance on process matters and defining standards. Primary focus is on defect prevention.
   B. **Quality control analyst**: The individual system responsible for checking the product or service after it has been developed. Primary focus is to find defects.
8. **Training coordinator**: The individual system who is the key person and point of contact, interface, for all training required for the project.

13.17.5.2 Trainee team role

A.k.a., Apprenticeship, mentorship, internship, residency.

The trainee team role is that of someone who is training upon an InterSystem Team.

13.17.5.3 Hazard isolation roles

Because life-threatening failures may occur when working with existent systems, humans must design their systems so that hazards can be isolated and systems can be restored. In concern to materials, for example, remote placement of hazardous materials, redundant containment, and clean-up material are a few options for reducing risk. An emergency shower will, for example, isolate dangerous chemicals into a liquid contain from a contaminated human. Every InterSystem Team should be able to avoid or secure hazardous systems with which they work.

Additionally, the concept of hazard isolation applies to the encoding of the value of 'justice' in any society. When a human becomes a "hazard" (danger, risk) to others, they "isolate" that human. The concept 'isolate' carries two orientations -- an orientation that restores fulfillment relationships, and another orientation that does not -- isolation from presenting a danger to society, by means of:

1. Isolation from supportive and restorative relationships (as in, restorative justice), and hence, isolation from structural feedback.
2. Initial isolation for physical safety, with the application of supportive methods and restorative relationships so that the self-organizing entity can re-orient itself toward fulfillment, releasing its societal requirement for any core form of isolation.

In early 21st century society, police take the role of law enforcement and are the represent the service that physically isolates hazards to society. In a market, society is structurally composed of 'property' (an abstraction), and thus, a core part of the role of the police is protection of 'property' from hazard. In community, there are individuals trained and accountable for isolating both technical and human hazards in the environment. At the human level, however, the concept of 'police' does not precisely apply, and their role as isolators of individual-human hazards to society, would be accounted for by medically trained personal, who are more like EMTs (emergency medical trained) personal who also have training in detaining humans), versus the conception of 'police', which entails politics, jurisdictional laws, authorities, property, psychological combatant training, jail, prison, etc. (none of these exist in Community as they are commonly defined in market-State societal configurations).

13.17.6 [Societal] Intersystem team work **tasks**

A.k.a., Intersystem team work/actions.

Societal design is the accountability (responsibility) of the community, and therein, the InterSystem Team:

1. **Habitation-related tasks** - includes tasks associated with sustaining and evolving the services provided by the controlled habitat system. These tasks are divided into a priority matrix between sub-system service and operational process priority. Tasks that are directed at the long-term viability and ultimate fulfillment of humanity.
   A. **Automation and maintenance** - Automating routine habitability tasks, while still allowing for InterSystem intervention, will be a high-priority development capability for all of these systems to allow a reduced human workload. This would free up human time for higher-priority tasks, while yet retaining the ability to control systems as needed in the event of problems.
   B. **Redundancy** - Redundancy management (RM; monitoring) will be employed in the selection of backups to replace failed or degraded systems, or to manage the rotation of redundant systems to equalize hours of operation. Some systems will have one or more identical backup units, ensuring physical redundancy. Other systems,

for which there are no physically identical replacements, may have their functions assumed by non-identical systems, ensuring functional redundancy.
   C. **Operating** - Doing any job in any habitat as part of a contributor to the InterSystem team.
2. **Scientific discovery endeavours/tasks** - includes field and laboratory tasks associated with answering the principal scientific questions.
3. **Skill areas** - These functions include command and coordination, routine, and contingency operations. Specifically, coordinating (piloting and navigation), system operations, system maintenance, repair of systems, and upgrade of system).

13.17.7 [Habitat] InterSystem team work **service structure**

The intersystem team habitat service structure:

1. Habitat service system design - open source and collaborative.
2. Habitat service system selection - preference criteria based on local population (cannot modify base functioning) customized layout, aesthetic, sub-services, timing, type and availability per demand and localization (location + control of location).
3. Habitat service system integrated and selected for design actualization. The solution inquiry process resolves.

Habitat services are, in significant part, formed from human needs; and hence, they are met continuously through a network of habitat service systems, which in and of themselves, have operational [InterSystem] deadlines (as in, the priority scheduling of tasks) in order to maintain themselves and adapt. In community, if "we" don't adhere to the deadlines "we" set, then our own services will likely fail.

13.17.8 [Inter-societal] InterSystem team work **roles and responsibilities**

The structural organization of human relevant roles and responsibilities may be relationally visualized through an organizational "breakdown" diagram on the part of an InterSystem project coordinator.

Due to the societal spanning nature of this project, its organizational structure necessarily interfaces separately with each type of society: the market-State and the community. Additionally, due to the presence of a larger global audience and the necessity for maintaining contractual agreements in the market-State, there is an Executive Steering Committee. In order to coordinate between these three divisions, a Main [Project] Coordinator (or, coordination system) exists.

The main societal coordinator is responsible for coordinating the flow and integration of information and materials between the three organizational divisions:

13.17.8.1 Division 1: Executive steering committee

The Executive Steering Committee is responsible for oversight, direction, and final project decisions related to market-State interaction . The Executive Steering Committee is responsible for communicating market requirements, State requirements, and human requirements to all stakeholders, while facilitating the resolution of any potential issues or changes that threaten the completion of the project.

13.17.8.2 Division 2: Market-State interface structure

The interface with the market-State society is an active societal construct engaged with on behalf of the Community (via this project) through project coordination. The market-State is interfaced with through electronic-jurisdictional contracts. The market-State must be complied with in order to access resources only available through the market-State.

13.17.8.3 Division 3: Intersystem team structure

Work upon the societal community system is organized through an InterSystem Team structure. The InterSystem Team structure is divided by three different primary system processes (coordination):

1. **Design (Informational specification and standardization)** - Responsible for the specification deliverable.
2. **Implementation (Material operation)** - Responsible for the operational deliverable.
3. **Social (Awareness and Sharing)** - Responsible for the social population deliverable.

14 [Project] Schedule

A.k.a., Time planning, plan timing, project timing, task timing, time logistics, the scheduling problem, time coordination, time management, time mapping, calendar mapping.

A schedule is a timeline of events. Scheduling is the process of deciding (and otherwise, coordinating) which of a given operation set gets performed, and when, on a given system set. To schedule is to setup a specific time when some event will occur. Scheduling is the process of coordinating (arranging, controlling, and optimizing) work and workloads in a production process. Scheduling is used to allocate resources, plan human contribution, plan production processes and acquire materials. Scheduling is an assignment over time of operations to systems, called a schedule. A schedule is the output of the scheduling process. More simply, scheduling is the process of coordinating work schedules (of humans and machines, a socio-technical system) to meet human requirements expressly input as deliverable activities.

Within the planning process, scheduling is the process of determining when tasks must be completed; when they can and when they must be started; and which tasks are critical to the timely execution of the project. A complete schedule is a function of total effort and resource allocation.

Scheduling (and the resulting schedule) are often considered a tool that defines what tasks are to be done, when, and by whom. Schedules define and track the progress and completion of a project.

A project timeline is most often called a Gantt chart. It is possible to add any schedule/time associated project variable to a project timeline; however, project timelines most often identify project milestones and tasks. Progress bars are included in timelines to identify the progress of a task(s) to a milestone(s).

The quality of any schedule is measured by its principal objective function:

1. The operational [project] completion function - Is there the state of 'completion' of the last operation (i.e., is the last operation complete)?
   A. There are # of tasks to be scheduled. Each task consists of one or more operations (processes). These operations must be scheduled on # of systems. The completion time of a task is the first point in time at which all of its operations are completed. The objective function (of scheduling) as an optimization function, involves minimizing either the completion time or the number of machines required to complete all the tasks by some specified deadline.

When working in a material environment, there is time-based information:

- **Scheduling [a time-planning solution]** is the process of calculating and assigning an arrival time for each deliverable (stop, output, etc.), with workers (transfer entities, contributors, etc.) being assigned time-bound roles (shifts) that adhere to working hours.

When scheduling in a material environment, there is also a location-based scheduling structure:

1. **Routing [a location-planning solution]** is the process of mapping out the unique paths (ways) that one or more transfer entities will take while they deliver or collect resources from each of their stop (deliverable) points. This involves considering the sequence of stops (deliverables), and the ways (approach, method) that will be taken by each transfer entity to successfully achieve this outcome.
   A. **Route optimization** follows logic steps, and is the process of analysing the projected routes and refining them to be more (or, most logically) efficient. This can be achieved by taking all physical and temporal relationships and locations into account and calculating an optimal path, given extant conditions.

NOTE: *Takt time describes pacing work to match the user's demand rate. Takt time planning then, is one method for work structuring around a set pace of work.*

14.1 Time

Time is the universal matrix of experience and activity. Time is an open matrix of possibilities for present action. In community, people decide what to do with their time together, to express their highest potentials and values. What is the experience of time, when labor is no longer alienated, but freely chosen? Contribution.

TRUISM: *Because people can decide what to do with their time when they experience time as free, it does not follow, for a socially self-conscious person (agent) aware of his/her mortality, that any use of time s/he decides is "good", is good (i.e., actually aligned with a common value set). People through imaginative reflection and projection, distinguish themselves from the social forces acting on them and decide as socially self-conscious agents what they will do. However, just because people can decide what to do with their time when they experience time as free, it does not follow for a socially self-conscious agent aware of his/her own mortality, that any use of time s/he decides is good, is good.*

14.1.1 Time duration

A time 'duration' is how long something (a task/activity) takes to complete. Duration can be visualized along a timeline (i.e., the chronological ordering of events).

14.1.2 Timing

Timing refers to performing an activity at the 'right' time, either according to a planned frequency or in response to an event.

14.1.3 Temporal-spatial coordination is scheduling

There is a requirement to identify time as well as location in environmental representation. Time is a concept perceived as the continued iteration ("progress") of existence, measured by an observer as events that are ordered relative, as "before" or "after", and which, at a given point in time, give rise to the notions of past, present and future. Time and location are often used together by an application to describe when a given condition exists or when an object was present at a given location (read: an objects epoch).

14.2 Deliverables-based project schedule

A deliverables-based project schedule facilitates the process of a project system:

1. Definitions.
2. Work-deliverable breakdown structure.
3. Schedule tasks.
   A. Enter all tasks.
   B. Determine predecessor tasks.
   C. Estimate the work.
   D. Estimate the duration.
4. Assign resources.
5. Add constraints.
6. Identify and operationalize contributing entities.

A spatial-temporal view of a set of operations for scheduling must include:

1. Set time, date, and location (1 operation).
2. Reschedule operation.
3. Postpone operation.
4. Change location operation.
5. Delete operation

14.3 Computing and scheduling

Computers and scheduling are closely related in two dimensions:

1. Assignment of operations on a machine is called a schedule.
2. Coordination in a multi-variate environment is more efficient and effective using computers do scheduling via computation more efficiently and effectively than humans do scheduling via cognition.

Additionally, there are three essential information characteristic sets associated with schedule computing:

1. Task characteristics (job characteristics).
2. Mechanism characteristics (machine characteristics)
3. Objective function characteristics (process characteristics).

The performance of a scheduling solution will likely fall into one of a number of possible categories, the most optimal of which is, generally:

1. An optimal solution in an amount of time proportional to a polynomial of the problem size.
   - $\leq Kn^k$
   - Wherein, n is the problem size, and there are constants K and k, which are independent of n (given, the problem is solvable in time).

There is a class of problems that can be solved in polynomial time (P, time-determined problems), and the superset of this class of problems non-deterministic problems (NP). NP is bound by a set of problems that can be solved by search or enumeration of a tree whose depth is itself bound by a polynomial in the problem size. (Lagerholm, 1998)

> **INSIGHT:** *Time could be viewed as that which is universally scarce.*

14.4 Schedule (timeline)

> *A.k.a., Gantt chart, project schedule, project timeline, calendar, project calendar, timeline, task-time network diagram, timetable, itinerary, time plan, planned time.*

A schedule visualizes activities in time; laying out the work and its phases on a calendar, mapping time-relevant items onto a calendar. All schedules are schedules of activity (as any action, task, work, deliverable, etc.) with all associated time information. A schedule is used to account for working, together, with real-world [resource-based] systems through time. A 'schedule' lists all project activities in time. Activities all start with verbs (what is to be done as an action). A [project] schedule is all project activities, dependencies and resources associated with time. The system records and tracts time, resources, and effort on the project. A schedule coordinates between time, activities, and a projects the resources (people, equipment, location) required to execute project tasks. A schedule is a "living" interface for coordinating and estimating work together.

A project schedule (a.k.a., gantt chart) visually combines project information essential to the coordinated execution of the tasks in time and space, with people and resources (and in the market, money). A Gantt chart is one type of organizational chart which could be used to convey the Action, Time and Finance plans of and between workgroups. Once the work is broken down by tasks and sub-tasks (i.e., the WBS is delivered), the project coordinator will [process information to]:

- Arrange these tasks in temporal order.
- Schedule them out to InterSystem Teams and HSS service systems.
- Identify dependencies (inquiry: does the start of one task require the completion of another task).
- Highlight the completion points [on the diagram] of critical tasks (a.k.a., "milestones").

Scheduling involves the relating future events (activities, tasks) in the real world to some linearly sequenced coordinate system called 'time'.

Generally, a schedule lists the following work data:

1. Activities (tasks, work).
2. Deliverables (outputs, outcomes, products).
3. Phases (milestones, stages).
4. Time points and durations (start and finish dates).

A schedule may visualizes (at least) the following (i.e., the following are mapped onto a calendar):

1. What are the deliverables.
2. What are the tasks (work items to produce deliverables).
3. Where are the completion/integration sign-offs for the deliverables.
4. Who is responsible.
5. Who is accountable.
6. Who is consulted.
7. Who is informed.

In concern to work, a schedule visualizes:

1. The current activities and future activities on the timeline.
2. The current status of a project.
3. All other projects that any given project relates to.
4. All work packages in a project that have a time reference, such as phases, tasks, and milestones, as well as, relationships between them (i.e., all work packages, phases, milestones, tasks, and bugs/ issues in a timeline view). Phase is a label for a set of linearly related tasks (e.g., development or planning).
   A. The work packages can have a start date and due date.
   B. Milestones may only have a due date.
5. All preceeds and proceeds between different work packages.

Any schedule is necessarily associated with data on users [of the schedule, contributors] and resources [accessible via the schedule]:

1. Contributed accountability coordination (worker identity)
2. Available resources coordination (resource identity)

A schedule is a type of chart that involves time:

1. A chart is a visualized display of data-base[d] information.
2. A schedule timeline ("gantt" chart) displays tasks as horizontal bars across a calendar (time-cycle), creating a visual representation of the project schedule, and other time-relevant.

A complete schedule may be calculated as a function of total work and available resource allocation. The schedule for a project is the timetable that specifies when each activity should start and finish.

An effective schedule is:

1. Understandable (visual).
2. Sufficiently detailed.
3. Highlights critical tasks.
4. Flexible.
5. Based on reliable estimates.
6. Conforms to available resources.

14.4.1 Elements of a schedule

Define the schedule's data structure as a list:

1. **Work breakdown structure** - a detailed list of [project] activities and [creation/development] tasks.
2. **Historical information** - from similar projects and other lessens learned.
3. **Personal calendars** - information from project contributors about their own time commitments.
4. **System calendars** - information on calendar events, significant common durations of time (e.g., holiday, vacation, work, cycle, maintenance).
5. **Resource planning and coordination** - the number of people available to the project.
   A. In community, there is the construction of a set of adaptive services that fulfill human need, want and preference. In the initial construction of the, hence forth, continuously operational habitat service system (part of the total societal system), there will need to be agreed upon dates for delivery of specific outputs. And, during operation, there will be maintenance

and replacement requirements, which will have static delivery dates [before urgency criticality is raised]. Individuals and systems agree on dates for the delivery of specific outputs, with a degree of flexibility relative to the task priority requirements themselves.
   1. In the market, there are milestones, or agreed on dates for the delivery of specific outputs.
6. **Visualize the schedule** - ready for inquiry *process.*
   A. Plan - "define" activity sequence and duration, develop the network integration or unique production diagram, and compose GANTT chart (i.e., the project implementation unique tasks timeline).
   B. Do - Communicate and update schedule core timeline with agreed upon tasked InterSystem Team positions (roles as part of an InterSystem Sub-Team) and tasks.
   C. Check - monitor schedule variances.
   D. Adapt - update the schedule.
7. **Monitor the schedule** - ready for *output.*
   A. Project schedule baseline - what is needed to sustain what degree of fulfillment (high-level categories include, but are not limited to: life support, some degree of technology support, and some degree of recreational-facility support.
   B. Schedule variance reports - when there is a variance from baseline in the scheduled fulfillment of need, and also when there is a variance from baseline in following (for automated and human systems) through with 'standard'[-ized] practices and procedures when contributing as part of an InterSystem Team.
8. **Update the schedule** - ready for *feedback*.
   A. Schedule updates become notifications.

Humans or automated systems, or some combination thereof, can perform [all] tasks. A unified information system allows for the reporting of habitat service's expected functionality. Is life support sustainable, and what are the plans for the systems evolution? The same goes for technical and exploratory service systems; are they meeting expectation and sustainable? Also, planning overlaps with criticality forming a criticality matrix applied to the determination of task priority [in a functional habitat service system].

14.4.2 Process for creating the schedule

The most common process for creating the schedule is:

1. Enter all the tasks (and sub-tasks) - as associated with the identified list of deliverables (from requirements document, WBS, etc.).
2. Determine predecessors (determine dependencies) - as the tasks that legitimately belong linked in an order (resource availability, decisions, outputs).
3. Estimate the work - as who will do the work, and when will the work be done by (accountability and completion date).
4. Estimate the duration of the work (timeline of activities).
5. Assign execution (team availability).
6. Assign resources (resource availability).

14.4.3 Project scheduling time-frame

A project may be a unique (one-time) endeavour, or it may have an ongoing and continuous objective. To some relative degree, of course, all processes (phases, stages, whole projects) have a specific time-frame, or finite life-span, to some situationally relative degree.

14.5 Schedule/-ing coordination

Aphorism: Plan the work, work the plan.

Schedule coordination includes the processes required to ensure timely completion of the project. A Schedule is created using a collaboration-driven estimation method; the reason for this is that a schedule itself is an estimate -- each date in the schedule is estimated, and if those dates do not have the people and their agreement, as those who are going to do the work, then the schedule will be inaccurate. Once the scheduling is in process (for it is continuous throughout), then project coordination involves monitoring the progress of the project and revising the schedule were required.

Schedule coordination consists of a series of tasks and steps designed to help manage the time constrains of the project, the steps are:

1. Defining the Schedule.
2. Publishing the Schedule.
3. Monitoring the Schedule.
4. Updating the Schedule.

Schedule inputs:

1. Work breakdown structure - contains a detailed list of all project activities and tasks.
2. Historical information - from similar projects and their lessons learned.
3. Calendar information - other commitments and calendar events.
4. Resource planning - planning for the collection, integration, and cycling of resources through a system.

Schedule progress conditions:

1. Plan - define activity sequence and duration, develop the network diagram and gantt chart.

2. Do - communicate and update schedule progress.
3. Check - monitor schedule variances.
4. Adapt - update the schedule.

Schedule outputs deliverables:

1. Project schedule baseline.
2. Schedule variance report.
3. Schedule updates.

14.5.1 Scheduling 'state' status

In some cases, the values of quantities included in scheduling have, or have not, been confirmed and are designated as:

1. **To Be Confirmed (TBC)** – details may have been determined, but are subject to change.
2. **To Be Determined (TBD) or To Be Supplied (TBS)** – the appropriateness, feasibility, location, etc. of a given event has not been decided (known, but not yet available).
3. **To Be Resolved (TBR)** - used when there is a disagreement on the requirement between technical teams.
4. **To Be Announced (TBA)** - details may have been determined, but are not yet ready to be announced. Note: This Does not apply in community, because the societal system is open source and transparently generated by the community.

14.5.2 Schedule delays

The whole project [completion timeline] will be delayed if task-deliverables and/or resources are delayed:

1. Schedule **critical path (tasks-deliverables)** - If anything (e.g., any task or deliverable) along this path (timeline) gets delayed, then the whole project will get delayed.
2. Critical **resource chain (resources)** - If those resources which are required are not available (i.e., not present when they need to be) and/or the quality of the available resources is not sufficient, then the whole project will be delayed

14.5.3 Principal schedule constraints

There are three schedule constraints that 'control' when *an activity starts or finishes*:

1. An activity must be completed by no earlier than a specific date - an activity may occur at any time after a specified date, but no earlier then the given date.
2. An activity must be completed no later than a given date.
3. An activity must be completed on a given date, no earlier or later.

14.5.4 Schedule modifications

A.k.a., Schedule timing, schedule alteration.

There are several common ways in which a project's schedule [timing] may be modified:

1. Add more resources - to shorten the time it takes to complete a scheduled activity or event (i.e., "crashing").
2. Do more actions - perform more activities simultaneously (i.e., "parallelization" and "FastTracking").

14.6 Scheduling system and user interface

A complete scheduling system and interface must meet the following criteria (i.e. the schedule coordination process must visualize a project schedule that meets the following criteria):

1. Complete - the schedule must be capable of representing all the work to be done. This is why the quality and completeness of the total information system, and its architecture, is so important.
2. Realistic - the schedule must be realistic with regard to time expectations and the availability of human and system contributors.
3. Accepted - the schedule must be acceptable to (have identifiable agreement from) the individual user.

14.7 Scheduling contribution time

To the InterSystem Team of a community-type societal system, at the highest-level, 'timing' refers to contribution as the selection and follow-through of [a] work [package]:

1. **When (time point)** the communication of an extant work package is distributed to the Community (for community and InterSystem Team contribution)?
2. **How long (i.e., duration)** the work package will take to complete (as whole and/or cycle)?
3. **When (time point; a.k.a., "milestone")** the work package is required to be complete (as whole and/or cycle)?

14.8 Schedule model

A schedule model involves all project information in association with a specifically applied scheduling method(s) and scheduling tool(s). In application, a societally coordinated schedule likely consist of a series of synchronous tasks (and sub-steps) designed to coordinate [between] the time constraints of any societal-level project.

14.8.1 Scheduling method

I.e., How will control over a schedule occur? Plan the control through the selection of a method.

A scheduling method is a formal procedure that can be applied to any instance of a scheduling model in order to obtain a feasible schedule (i.e., schedule aligned with objectives). A scheduling method solves a scheduling problem. A schedule method is a procedure that takes an instance of a scheduling model as an input in order to produce (At least) one schedule as an output, given a real world situation.

There are a variety of possible scheduling algorithms for scheduling [a set of project] activities [visually] in time. The following are methods for planning schedule control:

1. **Program evaluation review technique (PERT; a.k.a., Critical path method, CPM)** - Using the data below, CPM calculates the longest path of planned activities to logical end points or to the end of the project, and the earliest and latest that each activity can start and finish without making the project longer. This process determines which activities are 'critical' (i.e., on the longest path), and which have 'total float' (i.e., can be delayed without making the project longer).
    A. List all activities required to complete the project (typically categorized within a work breakdown structure).
    B. Identify dependencies between the activities.
    C. Visualize the relationship between all activities in a precedence diagram.
    D. Identify logical end-points, such as, milestones or deliverable items.
    E. Assign time (duration) that each activity will take to complete.
    F. The PERT (PERT chart creation) procedure is:
        1. Tasks (activities) represented as arrows (a.k.a., activity-on-arrow diagram).
            i. For example, "Collect project data".
        2. Milestone (major completion stage, phase, min-max version save) are represented as nodes (Read: circles).
            i. For example, 'No project data' (start node, date) and 'submit all project data' (end node, date).
        3. Estimate of duration of time it takes to complete the activity.
            i. For example, The time duration between start and end nodes that is entirely encompassed by the arrow that represents the task (activity).
        4. Package PERT (i.e., PERT applied) for selection by contributing users and habitat service systems.
            i. For example, The 'instruction' to 'investigate' an 'issue' in a building within 10 minutes in order to prevent a building evacuation.
2. **Critical chain method (CCM)** - After the critical path(s) is determined (Read: calculated with software), resource information is added (also calculated) to produce a resource-optimized schedule, with a resource-constrained critical path.
    A. Determine resource availability - associate resource information, including a resource-precedence diagram, with the critical path.

CLARIFICATION: *Though confusingly named, 'critical path' is the sequence of project network activities which add up to the longest overall time duration (i.e., the activities that create the longest distance between the start and the finish of a project). The critical path is the longest path through the schedule with either zero or negative total float.*

14.8.2 Schedule estimating

In a sense, every applied input could be viewed as a probability (or, "estimate") of potential input:

1. Contribution availability estimating.
    A. Probability of meeting contribution requirements, given that which is available and known (where, human contribution is the input).
2. Resource availability estimating.
    A. Probability of meeting resource requirements (where, real-world resources are the input).
3. Financial budget estimating.
    A. Probability of meeting financial requirements (where, money or trade is the input).

14.8.3 Scheduling tool

Project scheduling software can perform the scheduling method calculations (e.g., can perform CPM on a data set). A schedule tool is an information function that provides schedule component names, definitions, structural relationships and formats that support the application of a scheduling method (calculation).

15 [Project] Risk

A.k.a., Negatives, harms, threats, hazards, vulnerabilities, dangers, bad events, loss events, etc.

NOTE: *In general, a risk is considered something negative. This section refers to negative risks, versus positive risks. A positive risk is a potential event that might occur could be positive for a project. A positive risk would be an opportunity or advantageous outcome. However, in general, the term risk, when used by itself, implies something negative, implies a danger.*

In the dynamic landscape of project coordination, the successful execution of any project relies on the effective coordination and control of risks. Project risk coordination plays a pivotal role in safeguarding project objectives and enhancing resilience in the face of uncertainties. Risks are statements of what could potentially go wrong or the potential negative outcomes that may impact a project, operation, or objective. They represent uncertainties or events that have the potential to deviate from the planned course and adversely affect desired outcomes. Risk statements typically include information about the nature of the risk, its potential impact, the likelihood of occurrence, and any relevant context. Identifying and assessing risks is a fundamental step in risk management and mitigation. Risk statements are combined into a risk list (a.k.a., risk register), which is maintained as the project is executed. Whereupon risk mitigation strategies/activities may be applied before, during, and possibly even after execution of a project.

INSIGHT: *Risks ought to be reduced.*

When discussing risks, there are several important concepts that must be considered:

1. **Risk (a.k.a., concern, hazard, danger, negative risk, threat)** - exposure to danger. Risk is exposure to danger. Risks are what might go wrong, exposing someone or something to danger. Risks are those events or conditions that are likely to, or are, negatively influencing one or more project objectives, include scope, schedule, cost, quality, or other critical factors. A risk is any factor (or threat) that may adversely affect the successful completion of the project. Effectively, if something reduces the optimal completion of a project, then it is a risk. A risk is an identifiable danger to be overcome with a mitigation solutions before the risk becomes realized (as an incident), thereafter, requiring a more significant resource contribution.
    A. Risk assessment - is a prospective assessment of the likelihood of some incident/danger occurring (risk = rate * time for a bad/dangerous event to occur). A risk is the potential likelihood for something bad to happen.
2. **Residual risk** - risk remaining after all feasible mitigations have been applied. Residual risk is the risk that remains after efforts to identify and eliminate some or all types of risk have been made.
3. **Incident (a.k.a., occurrence, event, episode)** - is an unexpected or undesirable event or occurrence that has already transpired or is currently happening and could disrupt the project or lead to harm or damage. In other words, an incident is the occurrence of a negative event. When something bad has happened, the at-risk scenario became an actuality.
4. **Harm (a.k.a., damage, negative event, adverse effects, consequences)** - is an assessment of the damage that would be done if the risk was realized/actualized. Harm refers to the negative consequences or detrimental effects resulting from an incident or risk materializing. It can encompass damage to assets, financial losses, injuries, or any adverse impact on project outcomes.
    A. Harm assessment
5. **Mitigation (a.k.a., controls, risk reduction, preventative measures, countermeasures)** - a change(s) made, or to be made, to systems to reduce or eliminate a concern/risk. Mitigation involves the actions and strategies put in place to reduce or eliminate the likelihood and severity of risks or incidents. Mitigation aims to minimize harm and enhance the project's resilience.
6. **Issue (a.k.a., concern, problem, challenge)** - in the context of risk, an issue is a problem or challenge that has arisen during the execution of the project, and it often requires resolution or coordination to prevent it from negatively impacting project progress.

There are three common dimensions of risk:

1. **Hazards** and **exposure** (categories of danger):
    A. Human.
        1. Intentional.
        2. Unintentional.
    B. Natural.
2. **Vulnerabilities** (touch-point openings to potential harm):
    A. Social.
    B. Technical.
    C. Resource (economic as *object* or *money*).
3. Lack of [coping] **capacity** - inability to do something or know something:
    A. Infrastructural.

All concerns, risks and issues have a three characteristics:

1. **Likelihood** - of incident occurring - given existing and future expected situation.
2. **Consequence** - of incident occurring - priority response level (if incident occurs under expected conditions), result of damage.
3. **Uncertainty** - uncertainty of risk and risks occurrence. Uncertainty is uncontrollable; however, uncertainty can be reduced with better (more accurate) information.

A qualitative risk analysis is a formula for prioritizing risk-related issues:

- Risk = consequences • likelihood
- Wherein,
  - Risk = consequences (scale-of-harm) • likelihood (% occurrence as, likely to occur how often, every 24 hours)
  - consequences = scale-of-harm triage and prioritization
  - likelihood = % occurrence as, likely to occur how often, every 24 hours

After the risk value is acquired, there is always an [un-]certainty value (a.k.a., confidence level, measured error, etc.) given. The [un-]certainty value is typically a separate value, given to state a relative error-based confidence level -- how confident of not "being in" error is the stated risk value.

Risk likelihood refers to the probability of some risk becoming realized (i.e., suffering some harmful event, which was a risk), for example:

1. Likely.
2. Possible.
3. Unlikely.
4. Remote.

Note that some risks do not use this qualitative classification, instead they go further to sub-classify via counting, how often, and within what time interval the risk is likely to occur and become and real incident.

The consequence scale determines which of three/four bands each risk falls into: red (high), amber (medium), or green (low):

1. **Low (green) risks** – are currently tolerable (often as a result of existing action on them) and do not require specific extra action (although attention may be given to them to ensure they are not being over-managed thereby tying up resources that could be better employed).
2. **Medium (amber) risks** – are also not tolerable although management action is less time critical than red risks.
3. **High (red) risks** – are not tolerable, and will need immediate management action.

There are alternative ways of categorizing the consequence scale, for example:

1. **Negligible** consequences.
2. **Minor** consequences on overall needs, mission, etc.
3. **Major** consequences to overall needs, mission, etc.
4. **Critical** consequences to overall needs, mission, etc.

An alternative way of categorizing the consequence scale is:

1. **Minor** - relatively minor changes to the ecosystem or habitat; it is unlikely that there would be any measurable changes at whole trophic levels outside of natural variations.
2. **Moderate** - some measured change to the ecosystem or habitat, but without being a major change in eco-system or habitat functions. These changes are not disabling; they are not emergencies. These changes may be irritational, but are acceptable, until recovered.
3. **Major** - ecosystem or habitat components are functionally altered significantly. The level of change [due to the incident/risk] is not acceptable to enable one or more societal/environmental objectives to be achieved.
4. **Extreme** - Could lead to total collapse of habitat service, or collapse of an eco-system service (ecosystem processes).

NOTE: *Risks can change categories quickly, and hence, must be monitored.*

Common categories of risk remediation (mitigation) include:

1. **An absence** of what is required:
   A. Omissions (of information).
   B. Unclear (information).
   C. Illogical (information).
   D. In-coherencies (of information).
   E. Weaknesses (of otherwise useful structure).
   F. Inconsistencies (of applications).
2. **Barriers** to understanding and behavior change:
   A. Cultural barriers (social barriers).
   B. Motivational barriers.
   C. Profit and resource acquisition barriers.
   D. Physics (barriers of physical reality).
   E. State-regulatory (barriers of State authority).
3. **Actions** with the potential to de-rail understanding and behavior change:
   A. Actions taken on the part of market encodings, which consciousness requires decoupling from in order to operate community.
   B. Actions taken on the part of State encodings,

which consciousness requires decoupling form in order to operate community.

There are also different societal-level categories of risk, including:

1. People risks:
   A. Sufficiency of people to complete work.
   B. Accidents by people.
   C. Maliciousness of people.
2. Process risks:
   A. Failure to know of important processes.
   B. Failure to execute important processes.
3. Technology risks:
   A. Power loss.
   B. Technical failures.
   C. Replacements (if not developed internally).
4. Habitat risks:
   A. Climactic events.
   B. Market-State boom, busts.

Risk-type questions associated with ongoing fulfillment include, but are not limited to:

1. Does the community have uninterrupted access to their human needs/requirements?
2. At what quality/optimality are the needs being met?
3. Are those needs met in a regenerative manner?
4. Does the community have any unmet needs?
5. What concerns may cause the community's access to their [basic] needs to be interrupted?

A risk is a constraint or uncertain event (condition, state, or shape) that may present a potential problem for a project. A risk constraint is what is known that could go wrong and cause additional problems.

Note here that 'risks' and 'issues' are the same thing, problems. A 'risk' hasn't happened yet, and an 'issue' has happened (or is happening now). An 'issue' is a "risk" with a probability of happening 100% (not 99% as risk itself is categorized). Issues are experienced risks (i.e., "risks in reality"). Risks and issues are sometimes collectively known as "concern coordination" (or, "concern management"). Risks are mitigated and issues re-solved.

A concern will have 1 or more actions associated with it, and some actions will be associated with 1 or more concerns. Concerns and actions require actors (systems or people) and accountability.

INSIGHT: *Trust is essential because it is how you make an accurate assessment of the risk.*

Risks are listed (registered) and analysis (quantitative and qualitative) are conducted on them. Qualitative risk analysis is the process of assessing the probability of identified risks occurring, their potential impact to project objectives and prioritizing risks based on the resultant risk exposure.

Importantly, risks lists (registers) should be maintained and the incidents (risk actualizations) should be monitored for.

The names of risks are typically written as short sentences or sentence fragments. Fragments complete the following types of sentences:

1. "The risk to this project/system is the event that ..." or a variant such as,
2. "The risk to this project is that there will be ..."

Generally, risks also include a short description of the consequences of the risk occurring.

Table 6. *Risk description format.*

Name of the Risk *(what bad could happen?)*			
Type name of risk here			
Description of Risk *(describe that bad thing happening and its consequences; risk scenario)*			
Describe effects of actualization of risk here			
Root Cause	**Could Happen**	**Possible Effects**	**Current / Planned Action**
The root cause analysis (what gives rise to the risk and/or risk exposure?).	Something bad that could happen given a root cause.	Some possible effects if the could happen were to happen.	The current or planned action designed to reduce the risk through termination, transfer, treatment, or toleration.

15.1 Risk categorization

The following are risks commonly associated to all projects. Project difficulty involves a number of variable conditions:

1. **Number of tasks** - more task would increase project/mission difficulty.
2. **Skill variety** - a project requires the integration of some essential abilities, like the ability of information searching and word processing, to complete the project. Skill variety would increase the project/mission difficulty.
3. **Time limit** - Time limit means that there exists a deadline. Tight time limit would increase the mission difficulty.
4. **Resource support** - Resources here mean that all the tools, equipment and solutions could help someone complete their mission. Limited resources support would increase the mission difficulty.
5. **Social dis-alignment and conflict** - conflict means that work is not getting done because there is not a resolution to an issue. The degree to which someone's salience toward an issue of concern will

likely lead to conflict, instead of resolution and/or continued project progress.

From a high-level perspective, risks may be categorized into the following sectors of society:

1. **Political** - as those relating to the political situation facing the authority whether that is global, national, regional, or local. It covers things like election cycles, policy direction, policy creation, policy administration, political re-organisations, political relationships and styles, activism, war and terrorism.
2. **Governing** - as those relating to the governance and decision-making arrangements of the authority. It covers things like the constitution, codes of conduct, leadership, checks and balances, and member-officer relations.
3. **Management and Professional** - as those relating to the need for the "authority" to be professionally fit and capable for a purpose. It covers things like recruitment and retention, succession planning, management style, management systems (e.g. project management, performance management), staffing, administration, morale, capacity, skills, professional judgement, absence management, grievance and disciplinary policies, and employee relations.
4. **Legislative and Regulatory** - as those relating to new and pipeline legislation and the authority's audit and regulatory environment. May also relate to the authority's own legal and regulatory powers.
5. **Competitive** - as those relating to the market situation and the authority's competitors. It covers things like exposure to the market, competitiveness/value for money of services, spotlight seeking (for pathfinders, awards, etc.) and competition with nearby or benchmark organisations.
6. **Reputation** - as those relating to the authority's reputation with government, partners, the media and the public
7. **Citizen** - as those relating to the authority's need to meet changing needs and expectations of its citizens and customers. It covers things like consultation, communication and involvement as well as access, demand and the customer complaints and litigation culture
8. **Economic** - as those relating to the global, national and local economy. It covers things like economic cycles, the economic base, employment and earnings patterns, migration and inflow patterns, house price affordability and availability, regeneration and new development
9. **Social** - as those relating to national and local demographics, residential and social trends. It covers things like age profile, ethnic profile, health trends, crime trends, residential patterns and profile of housing stock, leisure and cultural scene, family profiles, skills base and educational provision and attainment.
10. **Environmental** - as those relating to the physical environment. It covers things like land use, infrastructure, transport, waste, drainage and flooding, erosion, subsidence, landslip, disease, pollution, contamination, seismic activity, air quality, water quality, energy use and efficiency, noise
11. **Partner/Contractual/Supplier** - as those relating to the authority's partnerships, contracts and supplies. It covers local strategic partnerships as well as more straightforward contracts with the private sector and concerns procurement, contract and relationship management, governance, funding, skills, quality and effectiveness.
12. **Technological** - as those relating to the organisation's technological situation and environment. It covers things like strategy, innovation, obsolescence, the nature of systems, support, maintenance, access, security, data protection and reporting.
13. **Financial** - as those relating to the organisation's financial situation and systems. It covers things like adequacy of funding, gearing, financial planning, financial delegations, budgetary control, monitoring and reporting, commitments, cash and treasury management, taxation, pension funds, insurance.
14. **Legal** - as those relating to current legislation and the authority's ability to deal with it. It covers systems of legislation, and the availability of legal advice and support.
15. **Physical** - as those relating to the organisation's physical and people assets.

Societal-level risks can be categorized in the following ways:

1. **Financial cost risks** (market only) - are those associated with the ability of the program to achieve its cost objectives.
   A. Cost risk is a focus area required to maintain integration of the risk assessment process to ensure consistency of the final product.
2. **Authority permission risks** (State and market) - are those associated with the ability of the program to acquire the permissions necessary to achieve its objectives.
   A. Authority risk is a focus area required to maintain the permission of those in authority in order to ensure consistency of the final product.

3. **Schedule risks** - are those associated with the adequacy of the time estimated and allocated for the development, production, and operation of the system/product.
   A. Schedule risk is needed to maintain integration of the risk assessment process to ensure consistency in the final product.
4. **Technical risks** - are those that threaten the evolution of the design, the production, and/or the level of performance necessary to meet the operational requirements.
   A. This type of categorization should be based on the source or root cause.
5. **Program risks** - are those that threatens to terminate, weaken, or change the strategic objectives, direction, or vision.
   A. Driving factors of Program Risk are cost, schedule, and/or performance.
6. **Process risks** - are those that threatens the proper execution of the community processes defined, explained, and envisioned in community standards.
   A. Process risks are assessed in terms of variance from accepted standards and potential consequences of the variance.
7. **Product risks** - are those that threaten the production of any aspect of community, so that it may not be produced on time, within budget, and/or according to specifications (standards).
   A. Product risks are assessed in terms of technical performance measures and observed variances from established specifications (standards).
8. **Functional risks** - are those that affect the community's ability to support specific users or user functions. Typically, functional risks occur where specific community requirements and processes have been overlooked in the implementation.
   A. Function risks are assessed in terms of variance of expected function.

15.2 Risk analysis and action planning

A complete risk analysis involves the following elements:

1. Risk identification:
   A. Is the risk understood.
   B. Is it known what the root cause of the risk is (e.g., external source or threat; or internal source, problem, or initiative).
   C. Is it known where the uncertainty lies in terms of the bad event that could happen.
   D. Is it known what the possible effects could be if the bad event does happen.
   E. Is it known what the likelihood of the risk occurring is.
   F. Is it known what the impact will be if the risk occurs.
2. Acquire a baseline:
   A. Are the critical success factors for development and performance (Read: progress) identified.
   B. Is there understanding (visual clarity) why a risk is at the position it is on the risk map.
   C. Should a risk be re-positioned on the risk map.
3. Existing actions:
   A. What is already in place to mitigate and/or control the risk, and how adequate are these actions.
      1. What are the newest actions that have just been taken to control and/or mitigate the risk, and were they appropriate.
   B. Where existing actions adequate, are the results as expected.
   C. What evidence is there that progress is being made to control/mitigate the risk.
4. New actions:
   A. What future actions may be taken to control and/or mitigate the risk.

15.3 Risk criticality mapping for risk decision prioritization

Risks can be mapped into a matrix that relates a set of risk's impact to the risk's likelihood, thus providing qualitative data to support prioritization of decisioning. The map shows what risks ought to be prioritized. Along one dimension of the matrix lies the impact categories, and along the other is the likelihood categories. To form the mapping:

1. Take each risk in turn and determine how likely the risk is to happen from very low (1) to very high (5). For ease of ranking, determine how likely each risk is to happen.
2. Once the likelihood is determined from 1 to 5, then determine the impact, again from 1 to 5 ranging from negligible to critical. Again for ease of ranking, determine the impact of the possible effects should the risk be actualized.
3. In assessing risk, take the 'worst likely' manifestation of the scenario not the 'most likely'.
4. Note: Only factor current action into the ranking, not planned action.

The risk criticality matrix, when complete, will show which risks require:

1. Immediate action and monitoring.
2. Action, resources, and monitoring are required, but less time critical.
3. Do not require significant attention or resources.

Risks can be categorized according to their potential to do harm, wherein, there is:

1. **Critical** - potential failure of whole system or function [after risk becomes actualized; after incident].
2. **Serious** - serious/high repair required [after risk becomes actualized; after incident].
3. Moderate - moderate repair required [after risk becomes actualized; after incident].
4. **Low** - significant repair not required [after risk becomes actualized; after incident].
5. **Negligible** - insignificant repair required [after risk becomes actualized; after incident].

Risks can be categorized according to their likelihood of occurring, wherein, there is a temporal frequency association:

1. Very high (e.g., 90% within X amount of time).
2. High (e.g., 70% within X amount of time).
3. Medium (e.g., 50% within X amount of time).
4. Low (e.g., 30% within X amount of time).
5. Very low (e.g., 10% within X amount of time).

15.3.1 Action plan decisioning

The key action(s) that form the plan to mitigate risks and solve issues due to risk is to consider what form, or forms, of action is best within the action categories of: Termination, Transfer, Treatment or Toleration. is appropriate to the risk Action plan decisioning around risk typically follows the following flow diagram:

1. Can the risk be avoided?
   A. Should the risk be avoided?
      1. Is it best to terminate the program that gives rise to the risk?
2. Can the risk be transferred?
   A. Should the risk be transferred?
      1. Is it best to transfer the risk to another "entity"?
3. Can the risk be controlled (treated)?
   A. Should the risk be controlled?
      1. Is it best to try and control the risk oneself?
4. Can the risk be lived with (tolerated)?
   A. Should "we" live with the risk?
      1. Is it best to live with the risk?

Common means of controlling for risk include, but may not be limited to (note: internal controls are designed, amongst other things, to reduce risk)

1. **Administrative controls (a.k.a., punitive controls)** – designed to ensure compliance with policies, standards, plans, rules, regulations and procedures.
2. **Preventative controls** – designed to prevent the bad event.
3. **Detective controls** – designed to identify errors, violations, or irregularities in actions/transactions already taken/processed.

15.4 Project uncertainty

INSIGHT: *The paradigm of understanding that humanity creates its living algorithms from should not be deeply flawed.*

All projects exist in an uncertain environment (otherwise there would be no need, no human requirement, for a project). An 'uncertain' environment is a 'probable' (similar to 'likelihood') environment. An emergent networked [eco-societal] system is, by assumption, an uncertain environment, with the conditions of risk and constraint (on all integrations, decisions, and actions). Risk is a measure of the probability that a negative outcome will occur. Risks represent potential dis-alignment from trajectory. Risk coordination identifies the risks to safety, performance and the project (e.g., overruns, schedule delays, etc.).

Every project involves some degree of uncertainty. Before a project is started, a plan is prepared based on certain assumptions and estimates. Assumptions are documented because they will influence the development of the project's resource selection, schedule, and work scope. A project is based on a unique set of tasks and estimates of how long each task should take, various resources, assumptions about the availability and capability of those resources, and estimates of the inputs and total effects (true costs) associated with the resources and their particular flow through the system. This combination of assumptions and estimates causes a degree of uncertainty that the project objective will be completely accomplished and/or accomplished within a specified time-frame. For example, the project scope may be accomplished by the target date, but the final resource requirements may be much higher than anticipated, because of low initial estimates for the necessity of certain resources. As the project proceeds, some of the assumptions will be refined or replaced with factual information.

Someone may not absolutely know the outcomes of one's own actions, but by thinking probabilistically, can perceive a distribution over outcomes. The expected value of an action can then be computed from utility (human requirement fulfillment functions) and probability integration through computation. This cognition "entangles" the agents' betterness relations (i.e., what relationship is the better choice?) as well as the agents beliefs/values about possible outcomes.

INSIGHT: *You have to accept some risk, nothing is ever going to be 100% risk free of uncertainty.*

15.5 Real problems

> **CLARIFICATION:** *A risk is an uncertainty. When this uncertainty becomes certainty, that is, when the risk occurs, then there is a real-world problem.*

The technical procedures required in formulating [environmental] problems should, but sometimes do not, begin with the question: "Does the problem really exist?" Problems in the real-world, the designed environment, are often assumed without detailed systematic analysis, leading to problem definitions that target the wrong system, or target a system without a problem.

For a human societal system, an environmental problem exists if, and only if, a malfunction can be detected between the designed environment and the system of human behaviors. Due to the nature of human-environmental dependency, environmental problems must not only be detected, but they must be resolved, so that humans are mutually fulfilled.

Because this is a societal system development project, anything that has the potential to impact the next iteration of the societal system is a potential risk.

16 [Project] Risk coordination

> *A.k.a., Risk control, negative coordination and control, risk management, plan risk management, disaster recovery, business continuity, disaster recovery.*

Risk coordination is an organizational (e.g., business, societal) process that all projects must undergo to protect their objectives from threats and facilitate actualization of opportunities to ease the efficiency and/or effectiveness of achieving objectives. Plan risk coordination is identical in naming for both project coordination (project management) and systems engineering. Both information sets define the risk method/strategy for determining how to conduct risk activities.

There are engineering technical risks, as well as project coordination ("oversight of the technical") risks [as all disciplines are critical to the participation in risks operations & maintenance.

The idea of 'risk' exists in a relationship between project management, systems engineering, and a dynamic environment:

1. The INCOSE Systems Engineering Handbook (SEHBK) suggests Analyze Risks includes "identification and definition of risk situations", which equates to the PMBoK Identify Risks.
2. The section of the Project Management Body of Knowledge (PMBoK) entitled "Identify Risks and Plan Risk Responses" equate to the SEHBK, "Define a treatment scheme and resources for each risk...", and is included in the SEHBK Analyze Risks activities.
3. The SEHBK adds an iteration activity: Evaluate the Risk Management Process, which should be true of all processes and activities.

To plan for risks requires to following procedure:

1. Identify what could go wrong (i.e., get the list of risks).
2. Assign likelihood and impact to each risk.
3. Develop mitigation techniques for risks.
4. Quantify impact of active mitigation techniques and responses.
5. Qualify mitigation solution.

Risks become issues to respond to once they actually occur. A risk list (risk register):

1. Records identified risks with a reference number for each risk.
   A. Describes the risk
   B. Links to a team or working group accountability for each risk

2. Identifies the likely severity of impact of each risk (likelihood).
3. Identifies the probability of occurrence of each risk (probability).
4. Identifies mitigation activities.
   A. Identify requirements for each mitigation activity.
   B. Identify probability of occurrence with each mitigation activity.
5. Identifies response procedures should the risk occur.
   A. Identify requirements for each response activity.
   B. Identify estimate of recoverability after each active response activity.

A complete risk plan includes mitigation activities and response procedures for each risk.

16.1 Plan for risk

A risk is uncertainty that affects objectives. In general, risk includes both opportunities and threats. The PMBok (2013) definition of risk makes this most clear with the words, "positive or negative effect on an objective". In common parlance, however, the term risk is generally intended to mean a negatively impactful probability. Uncertainties can affect the achievement of a project's objectives either positively or negatively. Often, the term, "risk event" is applied to both uncertainties that could hinder the project (threats, negative impacts) and uncertainties that could help the project (opportunities, positive impacts). A risk is an unplanned event that could result in harm or benefit; what unplanned event could happen that would result in harm or benefit? Risk involves future events/things that may not happen, but if they do happen, they would effect an objective. Risks matter because the effect the objectives. Risk could be viewed as uncertainty on the achievement of objectives. Risk-taking is the process of accepting risk.

Planning for risk involves thinking and acting to reduce the likelihood of harm preventative and in the case a risk incident occurs. One way to reduce risk is to increase the margin of safety. For example, having a store of some product provides a margin of safety in case the production of that product fails for some reason.

There are two core dimensions to risk:

1. Uncertainty.
   A. In a project, this is called **probability**.
2. Effect on objectives.
   A. In a project, this is called **impact (consequence)**.

Risks are uncertain 'events' or 'conditions'. Risk connects uncertainty with objectives. Uncertainty must always be connected with objectives in order to find the risks. Risk does not mean the same thing as uncertainty. All risks are uncertain, but not all uncertainties are risks. There are some uncertainties that are not risks, such that not every uncertainty in the world will be added to a risk list (or risk register). Risk is a subset of uncertainty that someone (or some population) deem of sufficient importance that they must take preparedness or mitigatory action on. More simplistically, risk is uncertainty that matters, and that likely some action may or will need to be taken upon, often, to prevent a negative impact (result or response).

In any practical dynamic environment, risks may be identified and added to a risks list, but risks are also emergent such that new risks may occur and old risks may no longer be risks. Knowable risks are exposed and listed. Risks are negative deviations from expected; wherein, an effect is a deviation from the expected - positive and/or negative. (ISO 31000:2009)

ISO 31000:2009 defines risk as:

- *Risk is the effect of uncertainty on objectives.*

Association for Project Management (APM, UK) Body of Knowledge, 2012 defines risk as:

- *Risk is an uncertain event or set of circumstances that, should it occur, will have an effect on achievement of objectives. [risk is uncertainty that matters]*

Project Management (PMBoK) Body of Knowledge, 2013, defines risk as:

- *An uncertain event or condition that, if it occurs, has positive or negative effect an objective.*

Significant risk (as, risk to objectives) determination questions include:

1. Which objective would be affected if this thing happened?
2. How uncertain is it?
3. How much does it matter?
4. All three combined determine how significant a risk is.

Fundamentally, there are two types of risk-based impacts, as assessed against objectives, that matter (and should be identified and addressed):

1. **Uncertainties that could hurt** the project (i.e., negative risk, danger).
   A. Uncertain changes or events that could harm; threats. Bad risks.
      1. In navigation, look out for, avoid, prevent, and protect against traps. What could cause us to deviate from a track or course?
2. **Uncertainties that could help** the project (i.e., positive risk, opportunity, favorable outcome).
   A. Uncertain changes or events that could be of

benefit; opportunities. Good risks.

1. In navigation, look out for, seek, and proactively make happen efficiencies and opportunities. What could help us stay on track or on course?

There are events that could happen that could be good, and there are events that happen that could be bad, and both need to be proactively identified and addressed. Events that could hurt need to be prepared for and mitigated against. Simultaneously, events that could help need to be identified and action taken to make them happen. Note that this is equally true in the personal lives of humans as it is at the societal level.

When designing systems, there are three principal design objectives that account for risk:

1. The potential to negate optimal re-solution of the design's requirements (i.e., the design requirements of the human fulfillment system).
2. The potential to hurt a human or fulfillment system.
3. The potential to cause an accident, and thus, unnecessary problems in, a human fulfillment system.

> **NOTE:** *For every assumption, there is a corresponding constraint (i.e., probability for a problem). Similarly, for every lack of definition in an argument there is the probable creation of a space for additional error.*

16.1.1 The composition of an risk entry

A.k.a., The statement of a risk, risk statement.

A risk statement is necessarily a cause and effect statement. To develop a negative risks list, identify what might go wrong and cause harm and/or exposure to danger. A risk entry is most useful when it is contained within a structured description that separates cause, risk, and effect.

Risk can be described in three stages (Read: salient categories of meaning in relation to the achievement of a goal):

As a result of <existing condition>, uncertain event> may occur, which would lead to <affect condition> on objectives.

Simplified view of the three stages:

1. As a result of <some cause>, then
   A. a <risk> may occur, which would
      1. <affect> an objective.

16.1.2 Semantic temporality

In the English language, there are words that can be used in communication to identify the different stages or parts of a risk entry (linguistics):

1. Definite words to describe facts (to describe the **present condition; existing condition**).
   A. Is, do, has, has not ...
2. Uncertain words to describe the risk (to describe the **uncertain future; uncertain event**).
   A. May, might, possibly, ...
3. Conditional words that say, this would follow if the risk occurred (to describe the **conditional future; effect**).
   A. Would, could ...

16.1.3 The structure of a risk

A risk-based information set contains information on:

1. Risks have an individual basis:
   A. Their likelihood of occurrence.
   B. Their likelihood of impact (on all objectives).

16.1.4 Population risk types: Personal and social risk

One challenge faced by any society is when one segment of the population does not experience a problem that another segment does experience. In more individualistic societies, when one segment of the population does not experience a problem that another segment does experience, potentially, the segment of the population that does not experience the problem will not perceive social risk, and individuals therein are likely to govern their behavior only based on what they perceive their personal risk to be (i.e., they will only perceive personal risk and ignore social risk). Alternatively, individuals in a holistic society (and not individualistic society) think in terms of social risk as well as personal risk. For instance, a younger individual in an individualistic society could be generally healthy and not concerned about their personal risk after acquiring symptoms of a viral infection. This person may feel well enough to go to their job where there co-workers are present. This person may shake hands with an older colleague who has a chronic medical condition, who may become infected by the virus carried by the young co-worker who felt well enough to go to work. In such a case, the young co-worker could be responsible for that colleagues death.

In a healthy and cooperative society, it is wise for all individuals to think about their responsibility to each other when deciding their behavior. Society at large should not be thought about in terms of individuals' personal risk; instead, individuals should act collectively in a cooperative manner to reduce societal-level risks.

16.1.5 Negative deviation: Negative risks

A.k.a., Threats, detriments, losses, negatives.

In its negative context, a risk is a situation and probability involving exposure to danger (Read: harm, injury, loss, suffering, etc.), or any other negative occurrence that is caused by external or internal environment, and that may be avoided through pre-emptive action (Read: through controls on preparedness, operations, and responses). For any intentionally living system, in an uncertain environment, there is the conception of risk. In the real world, for a social populations, there are a multitude of risks. To navigate safely together, risks must be identified, prepared for, and mitigated against (i.e., protected against the danger or reduce/eliminate the danger). A negative risk is the likelihood that a loss will occur.

In the context of negative impacts, risks are potential events that could happen during the course of a project, that if they happened, they could (note: these are effects, not risks):

1. Kill or injure.
2. Lose resources, assets, or access.
3. Waste time.
4. Waste effort and energy.
5. Damage reputation.
6. Damage natural ecological cycles.
7. Harm performance.
8. Waste money *(market-State only).*

There are many types of negative risks, including but not limited to:

1. Human life risks.
2. Project risks.
3. Personnel risks.
4. Operational risks.
5. Technical risks.
6. Social risks.
7. Environmental risks.
8. Ecological risks.
9. Financial/business risks *(market-State only).*

16.1.6 Not a risk (non-risk)

Items that are certain and do not belong in a risk list include, because they are certain (and not uncertain):

1. Problems - a problem has been identified (and there are solutions to resolve the problem).
2. Issues - an issue has been acknowledge (and a process is engaged to resolve it). Issues require resolution. Issues have occurred or will imminently occur. A negative event can turn into an issue.
3. Constraints - a known limitation placed on a project/system.
4. Requirements - a known expectation from a project/system.

Risks are neither causes nor effects. However, it is easy to confuse risk with non-risk, especially cause or effect. There is a real-world, dynamic system in which risk occurs:

1. Cause (fact) - causes are not risks because they are occurring now. Causes are facts, issues, problems. Causes are not risks, because they are not uncertain.
    A. Something true today.
2. Risk (uncertainty)
    A. Something that may, or may not, happen
3. Effect (possible result) - If the risk has occurred, then it is an effect.
    A. Why something matters to the objective.

It is most useful when risk descriptions have a description of not only the risk, but also cause and effect.

16.2 [Plan] Risk coordination process

A.k.a. The risk management process.

The risk process may be simplified to:

1. Identify objectives.
2. Identify uncertainties that matter to objectives.
    A. Include in the identification threats, negative uncertainties.
    B. Include in the identification opportunities, positive uncertainties.
3. Prioritize the risks by asking, How uncertain? How much would it matter?
    A. What are the worst threats?
    B. What are the best opportunities?
4. Identify (or construct) responses appropriate to each risk.
    A. What can be done to stop threats, or continue and recover if threat occurs?
    B. What can be done to cause an opportunity to be actualized.
5. Execute the response.
    A. Preparedness and pro-active action for opportunities.
    B. Preparedness, mitigation, and operational actions for threat event.
6. Risk control.
    A. Monitor the results of all actions.
    B. Review for new risks, and repeat.

The most significant risk process questions that can be used for any project (or even any decision) include:

1. What are we trying to achieve?
    A. Set objectives.
2. What could affect us achieving it?
    A. Identifying risks.
3. Which are the most important risks on the list?

   A. Assess and prioritize risks.
4. What can be done about the risks?
   A. Planning responses.
5. When should it be done?
   A. Schedule responses and update cycle.
6. Did it work, and what has changed?

16.2.1 Organizational planning for risks

Systems engineering coordinates ("manages") technical risks within a project[-based structure].

At the project-level, the principal risk is (managing organizational risk):

- Delivering a system (Read: new system state) that does not meet organizational, orientational standards.

A the engineering-level, the principal risk is (managing technical risk):

- Delivering a system that does, or does not, meet user requirements.

In practice, a risk coordination and control system (team or working group) should account for threats and opportunities together in a single unified process, because they are both uncertainties that matter. Both threats and opportunities are types of events that may or may not happen that are likely to impact the objective. Both threats and opportunities can both be accounted for and pro-actively acted upon.

16.2.2 The risk plan (information set)

What might go wrong with the plan, and how to limit that risk with contingency planning:

1. Description of problem (risk).
2. Probability and impact of risk.
3. Workaround of problem.
4. Scope of contingency.

System safety is the accounting for observations that accidents can result "from dysfunctional interactions among system components" (e.g., bottlenecking to incident, or overshooting carrying capacity).

System 'safety' is influenced not only by the reliability and failure behavior of various subsystems and components, but also by the nature of interactions between these components, as well as their interactions with external factors (i.e., environmental conditions).

1. Safety includes human-caused incidents.
2. Safety includes environmentally-caused incidents.

16.2.3 Risk resolution coordination

The coordinated resolution of probable risk entails the analysis of risk as an information process, and the mitigation of risk as a [engineered] construction process.

Risk coordination control (risk management) refers to systematically addressing risk throughout the life cycle of a system, product, or service. Project risk coordination (management) includes the processes of conducting risk coordination planning, identification, analysis, response planning, and controlling risk on a project.

16.2.3.1 Taking proactive action as opposed to being forced to rely on reaction

In common definition:

1. Pro-active action refers to a complex of inter-actions, including but not limited to planning and monitoring in order to reduce the probability of negative consequences (i.e., reduce the likelihood or results being mis-aligned with objectives).
2. Reactive action refers to responding to a consequence without planning.

For example, the athlete gets hurt before they need "therapy", versus providing "therapy" during the athletics life cycle so they are less likely to get hurt in the future.

In a world where supermarkets are food carnivals (falsely flavored and highly palatable foods and food-like substances), filled with biologically "addictive" combinations, then the socio-economic reality is that decisions and behavior are not solved solely by personal choice, but they also necessitate as part of the solution, modifying the food environment (i.e., fixing the food environment so that it doesn't seemingly "naturally" in close proximity and access these foods, and move through environments that don't "naturally" drive individual organisms to).

16.2.3.2 Indicators and pro-active versus reactive action

Lagging indicators are used to evaluate current conditions. In order to act pro-actively, it is necessary to explore future projections in order to better guide an organization toward greater success at, or achievement of, a goal. Leading indicators give an organization the [informed] ability to think and act pro-actively, instead of reactively, which can reduce the time required to meet the goal, and in the market "save" money.

16.3 [Plan] Organizational exposure

Tracking organizational exposure through an assessment tool helps in understanding a project's exposure to a risk. The following assessment aims to support decisioning and is a definitional tool, no an explanatory one.

Table 7. Execution > Risk*: Exposure assessment including statements about aspects that may be directly or indirectly impacted by a risk.*

#	Organizational exposure	*Disagree (0); Neutral (2); Agree (4); Strongly Agree (5)*
1	The country and regional exposure to the understandings and operation of a community-type society.	
2	The organization operates in a country or region that is not of the societal type, community.	
3	The organization operates a societal interface.	
4	The organization is facing challenges regarding access to resources.	
5	The organization contributors are pessimist about the impact of an event.	
6	The organization is based or has a strong presence in production of required resources.	
7	The event will cause an unrecoverable loss.	
8	The long-term organization can be directly and negatively impacted by the event in socio-technical (micro- and macro-economic) factors.	
9	The work is located in an environment that is not of the societal type, community.	
10	The project or initiative has a large number of people working in the same location.	
11	The project or initiative has a strong need for human interaction.	
12	There will be a large negative impact if the work is reduced or ceased.	
13	Disruption to the supply chain will have a severe impact on the development of the work.	
14	The initiative or project is heavily dependent on an external societal supply chain.	
15	A member of the team is incapacitated.	
-	**If combined numerical result is low, then there is low risk; if middle, then moderate risk; if high, then high risk:** [1] If the assessed risk is low, then continue monitoring the situation and re-evaluate if results change. [2] If the risk is moderate, take moderate action. [3] If the risk is high, take rapid action.	**Combined numerical result from calculating:** *count x weight* **(of each statement)**

In order to reduce the likelihood of harm:

1. A harm reduction approach acknowledges that laws/protocols may be broken, and this can be tolerated in favor or reducing risk and increasing safety.
2. Risk review board (a.k.a., ethics review board, ERB) - when is an action decidedly available (i.e., "OK") that risks psychological and/or physical harm [in the name of science and social safety).

16.4 [Plan] Risk mitigation and remediation

Risk mitigation planning is the process of developing options and actions to enhance opportunities and reduce threats to project objectives. Risk mitigation implementation is the process of executing risk mitigation actions. Risk mitigation progress monitoring includes:

1. Tracking identified risks.
2. Identifying new risks.
3. Evaluating risk process effectiveness throughout the project.

16.4.1 [Mitigation] Narrative role model advocacy

A.k.a., Behavioral economics.

Narrative role model advocacy is the use of a storyline in media (e.g., radio, television, film) to affect change across an entire society. A storyline, for example, may have middle of the road characters designed to represent segments of the audience and be aspirational for them, but also be very similar to them. Those characters sort out conflicting advice from the positive and negative characters one finds in all melodrama, and over many episodes they gradually evolve into positive role models for the audience, and they show the audience the benefits of the new behavior and they deal with the pushback that one gets when they try anything unusual or innovative in any society's, so they show the audience how to deal with that pushback. The storyline characters ultimately become outspoken advocates for that new behavior, thus role modeling behavioral change and advocacy for the audience population. It is possible for researchers to actually to measure changes among the audience members in self-reported interpersonal communication about the issues being addressed in the storyline. Common themes in storylines include violence against one another, human issues, and family planning. The fundamental goal is to change the perception of what is normal and/or possible.

Keltner and Piff (2010, 2014) in laboratory research have found that small psychological interventions, small changes to peoples values, small nudges in certain directions can restore levels of egalitarianism and empathy. For instance, reminding people of the benefits of cooperation or the advantages of community caused wealthier individuals to be just as egalitarian as poor people.

16.4.2 [Mitigation] Understanding advocacy

Achieving full interoperability of socio-technical data is both complex and fraught with pitfalls. Users of environmental data and may be uncomfortable with geodesy, geometric transforms, dynamics modeling, and logical reasoning. It is strongly thought that an adequate education is a necessary prerequisite to success in that endeavour. There has been a fair amount of misunderstanding when practitioners from different disciplines talk to each other about location, condition, and decision information.

There are also unspoken, and all but forgotten, assumptions made within specific disciplines that are opaque to non-specialists, and either result in miscommunication or are simply no longer appropriate assumptions to make. These problems are not unique to specialists -- they are rife within the general systems engineering and resource-based economy (RBE) organizations as well.

Given these considerations, the application of interoperability, necessarily includes significant didactic material, and generally covers four major areas, as follows (which may be seen as objectives for a learning contributor):

1. **Concept development:** This includes the reference model (RM), the scope of the reference model (RM), as well as the design criteria. It also includes the development of the concept of "pure" coordinate systems and their associated transformations from basic concepts in a database to solid and analytic geometry. Subsequently, isometric ("real world") geometry and coordinate systems are developed and extended to define the basic isometric socio-technical reference frames. Concepts associated with directions (vectors), and the corresponding orientation representations, are defined. The complexities of real-world terrain surfaces (as opposed to mathematical "smooth" approximations) are addressed.
2. **Conceptual reference frame specifications and formulations:** This covers the complete specification of the conceptual reference model (CRM) and each of the included conceptual reference frameworks (CRFs).
3. **Spatial Reference Frame Specifications and Formulations:** This covers the complete specification of the spatial reference model (SRM) and each of the included spatial reference frameworks (SRFs). Error specification and algorithmic development: This addresses how to define error specifications in reasoning and visualization (2D and 3D), and the development of efficient and accurate coordinate operations algorithms among the reference frameworks (RFs) included in the unified reference model (RM). Also included is a discussion of transitivity and chaining when converting between reference frameworks (RFs), which may use a sequence of operation steps rather than a single optimized direct conversion.
4. **Implementation, testing and application:** This involves the reduction of the developed algorithms to efficient, accurate, portable implementations that maintain the stated operation accuracies and performance. The methods used to test and verify the implementations are developed, and the results of extensive testing, are presented and reviewed. The information and material interface specification is defined, and guidelines for its use are documented.

16.5 [Plan] Risk response

NOTE: *Ignorance exist in contrast to planning for risks.*

When mitigation isn't successful, then response occurs. In order to most effectively respond to risks, a series of planned questions must be asked and answered:

1. What should be done, based on?
   A. Type and nature of risk.
   B. Controllability.
   C. Impact severity.
   D. Resource availability.
   E. Efficiency/cost-effectiveness.
2. Who, when, where, and with what tools, should it be done.

The most common categories of response to negative risks (i.e., threats) include:

1. Eliminate uncertainty - eliminate risk, kill risk, avoid risk. Note: by avoiding one risk, the solution may lead to the exposure to other risks.
   A. An avoid strategy.
2. Reduce uncertainty - reduce risk to acceptable/controllable levels; reduce the impact or exposure of the risk; mitigate the risk. Reduce risks by reducing probability and/or impact.
   A. A reduce strategy.
3. Transfer responsibility, liability, ownership - give risk to another entity. Have an outside authority handle the risk for you. (e.g., use anti-virus software on a software operating system, use insurance use insurance; note that the presence of this type of risk is indicative of poor/non-optimal design). The asset is still being protected, it's just that you are not the one doing it.
   A. A transfer strategy.
4. Accept residual risk - accept the risk and control it as best as possible.
   A. An accept strategy.
5. Ignore risk - do nothing.
   A. An ignoring strategy.

The most common categories of response to positive risks (i.e., opportunities) include:

1. Cause the opportunity to happen. Exploit some connection.
2. Share responsibility for making some event happen with another or others.
3. Enhance some connection to make the event more likely to occur.

Table 8. Execution > Risk*: Methodical responses for the presence of risk.*

Threat	Generic Strategy	Opportunity
Avoid (eliminate)	Eliminate uncertainty	Cause (exploit)
Transfer	Allocate ownership	Share
Reduce	Modify exposure	Enhance
Accept	Include in baseline	-

16.5.1 Risk coordination process elements

A.k.a., Risk control elements, risk coordination and control elements.

The phases of the risk coordination cycle include four main elements:

1. Risk identification - identify all potential risks.
   A. Deliver a list of risks.
2. Risk assessment - prioritize the likelihood of this risk occurring and the severity/damage impact if it occurs.
   A. Deliver a risk assessment matrix.
3. Risk control.
   A. Risk mitigation - Deliver a plan, procedure, technique, tool, or process to mitigate threat.
   B. Risk accentuation - Deliver a plan, procedure, technique, tool, or process to actualize opportunity.
4. Risk monitoring and control.

Alternatively, the phases of the risk coordination cycle could be viewed as:

1. Identify probable risks.
2. Determine probability of each risk.
3. Evaluate potential impact of each risk.
4. Develop controls and plans as responses to each risk.
5. Document responses to each risk.
6. Act on next steps (i.e., next tasks) for each risk.

The common risk coordination elements include (note that risk-coordination is a sub-type of issue-coordination, see sub-bullets):

1. **Risk planning** - planning for the avoidance of danger, or an avoidable reduction in fulfillment.
   A. Risk-type issue planning.
2. **Risk identification**
   A. Risk-type issue identification.
3. **Risk analysis** (clarifying, categorizing, and prioritizing risks, and developing controls for risks)
   A. Risk-type issue analysis.
4. **Risk mitigation** (design, select, and applying/ implement controls to mitigate risks)
   A. Risk-type issue mitigation.
5. **Risk monitoring** (assess and monitor active controls)
   A. Risk-type issue monitoring.

At a societal-level, risk coordination feeds into a societies Effectiveness Inquiry [decision system] process.

16.6 [Plan] Continuous risk analysis, coordination, and control

QUESTION: *Is it possible to engineer a system that does not posses the risk? Is it possible to analyze and design a system that is highly less likely to express the risk?*

Continuous risk analysis, coordination and control is a project engineering category with processes, methods, and tools for predicting risk-related problems and resolving them such that the project is safe, effective, and efficient.

Any useful risk inquiry identifies, what are the social mechanisms that are driving people to maintain these fixed and limiting behaviors and/or beliefs.

Risk mitigation processes include:

1. Assessing continuously what could go wrong (risks).
2. Determining the significant prioritization of risks.
3. Acting to resolve the risks:
   A. Identifying uncertainties.
   B. Identifying assumptions.
   C. Identifying problems and inquiries.
   D. Resolving the probability risk space.
   E. Resolving the decision space.
   F. Changing the system to have engineered a risk out of the system.

**All processes may occur in parallel and/or series.*

Risk analysis and mitigation involves the analysis, design, and operation of systems without the risk [to humanity and ecology]:

1. The identification of [probable] risks within:
   A. The current system.
   B. A new system state change.
2. The resolution of [probable] risks within:

A. The current system.
B. A new system state change.

In community, to minimize reliance on error-prone and time-intensive human or procedural controls, the primary means of risk mitigation involves the designing of risk out of a system (e.g., fail-safe redundancy, fault tolerance, load margins, inherent reliability, and test verification). In practice, risk reduction depends on advance knowledge of environmental conditions, performance of engineered products/systems, accurate testing, and human [response] capabilities.

Materialized systems throughout the community's habitat service system have different levels of fault tolerance. For example, locations where human safety is a critical function, normal design criteria require two-fault tolerance levels. All critical systems essential for human and ecological safety (survival) shall be designed to be two-fault tolerant [at least]. When this is not practical, systems shall be designed so that no single failure shall cause loss of the Team (or city). This requirement, as a component of [operational] maintenance, can be considered as a third level of fault tolerance (i.e., of redundancy). In community, however, functional roles are they are unified under open source engineering.

The risk analysis and mitigation sub-processes categories are (function/operation):

1. **Identify** - search for and locate risks before they become problems.
2. **Analyze** - transform risk data into decisioning information.
   A. Evaluate impact, probability, and timeframe, classify risks, and prioritize risks.
3. **Plan** - Translate risk information into decisions and mitigation actions (both present and future) and implement those actions.
4. **Track** - Monitor risk indicators and mitigation actions.
5. **Control** - Correct for deviations from the risk mitigation plans.
6. **Communicate** - Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks. Note: Communication happens throughout all the functions of risk mitigation.

Continuous risk analysis requires answers to the following questions:

1. What proximity is required for this risk to apply?
2. How localized are the effects posed by this risk?
3. What is the recovery time if the risk was detected?
4. What are the recovery and restoration requirements if the risk is detected?
5. Impact - How serious an impact?
6. Prior - Is there evidence of this risk prior?

16.6.1 Identify

The principles applicable during the Identify function are:

1. Risks are identified as part of a continuous process, not a one-time only activity at the start of the project.
2. Risk identification must be open source to sufficiently bring forward new risks and to look beyond immediate problems.
3. Although individual contributions play a role in risk management, teamwork improves the identification of new risks by allowing individuals to combine their efforts, knowledge and understandings.

16.6.2 Analyze (Assessment)

The principles applicable during the Analyze function are:

1. Conditions and priorities often change on a project and can affect the important risks to a project—risk analysis must be a continuous process.
2. Analysis requires open communication so that prioritization and evaluation is accomplished using all known information (**safety protocol, open source protocol**).
3. A probabilistic-oriented view enables teams to consider long-range impacts of risks.
4. A global perspective and a shared societal vision allows an analysis of risks to account for the overall societal system, human needs and goals.

16.6.3 Plan

The principles applicable during the Plan function are:

1. Planning risks is a continuous process of determining what to do with new risks as they are identified, to enable efficient use of resources.
2. Integrated coordination is needed to ensure mitigation actions do not conflict with project or team plans and goals.
3. A shared product vision and global perspective are needed to create mitigation actions that ultimately benefit humankind and the ecology.
4. The focus of risk planning is to be probabilistic, to efficiently prevent risks from becoming problems.
5. Teamwork and open communication enhance the planning process by increasing the amount of knowledge and expertise that can be applied to the development of mitigation actions.

16.6.3.1 *The best plan, a fail-safe plan*

Fail-safe and **fail-secure** (as a task for safe systems engineering and redundancy planning) means that in the event of failure, the system responds in a way that will cause no harm, or at least a minimum of harm, to other systems or danger to individuals. Fail-safe means that a device will not endanger lives or other systems when it fails. A system's being "fail-safe" means not that failure is impossible/improbable, but rather that the system's design prevents or mitigates unsafe consequences of the system's failure.

16.6.4 Track

The principles applicable during the Track function are:

1. Open communication about a risk's status stimulates the project and risk management processes.
2. Tracking is a continuous process—current information about a risk's status is conveyed periodically to the rest of the project.
3. When project personnel review tracking data with a forward-looking view and a global perspective, they can interpret the data to reveal adverse trends and potential risks.
4. Integrated management combines risk tracking with routine project monitoring processes, creating a synergy that better predicts and identifies new issues.

16.6.5 Control

The principles applicable during the Control function are:

1. Open communication is essential for effective feedback and decisioning, a critical aspect of Control.
2. Risk control is also enhanced through integrated coordination—combining it with routine project coordination activities enables comprehensive project decisioning.
3. Shared project vision and a global perspective support control decisions that are effective for the long-term success of the project and [societal] organization.

Scholarly references (non-cited)

- Lagerholm, M. (1998). *Resource Allocation with Potts Mean Field Neural Network Techniques*. Lund University: Department of Theoretical Physics. Thesis for the degree of Doctor of Philosophy. https://pdfs.semanticscholar.org/fb5a/397e433dd16c327a5e14f5699fb180339dc1.pdf
- Lewis, B., Deatrick, J., Johnson, H. (2016). *Project Planning. Region 4. U.S. Environmental Protection Agency Science and Ecosystem Support Division*. Athens, Georgia. SEDPROC-0160R5 https://www.epa.gov/sites/production/files/2016-04/documents/project_planning016_af.r5.pdf

Scholarly references (non-cited)

- Framinan, J.M., Leisten, R., Garcia, R.R. (2017). *Manufacturing Scheduling Systems - 2014*. Springer, London. https://doi.org/10.1007/978-1-4471-6272-8
- Fanchiang, C. (2017). *A Quantitative Human Spacecraft Design Evaluation Model for Assessing Crew Accommodation and Utilization. Aerospace Engineering Sciences Graduate Thesis & Dissertation. Aerospace Engineering Sciences*. University of Colorado, Boulder. https://scholar.colorado.edu/cgi/viewcontent.cgi?article=1160&context=asen_gradetds

Book references (non-cited)

- Belzer, J. Holzman, A.G., Kent, A. (1979). *Computer Science and Technology, Vol. 13, Reliability Theory to USSR, Computing in.* Marcel Dekker, Inc. New York and Basel.
- Daylio, R. (2017). *Principles: Life and Work*. Simon & Schuster. https://www.principles.com/principles-in-action/
- Dorofee, A.J., Walker, J.A., et al. (1996). *Continuous Risk Management Guidebook*. Carnegie Melon University. SEI Joint Program Office. http://www.jodypaul.com/SWE/ContinuousRiskManagement.pdf

Online references (non-cited)

- Brautigam, B. (2017). Living Machines: Design Paradigms for Self-Aware Machines. Medium. https://medium.com/interactive-mind/living-machines-design-paradigms-for-self-aware-machines-d56ac2ac9715
- *Integrated Project Deliver: A Guide*. Ver. 1. (2007). The American Institute of Architects. http://info.aia.org/siteobjects/files/ipd_guide_2007.pdf
- *ISO 9001:2015 Requirements for a Quality Management System*. (2015). https://the9000store.com/iso-9001-2015-requirements/
- *State of Michigan Project Management Methodology.* Michigan Department of Technology, Management & Budget. 2014, May, Ver. 4.0. https://www.michigan.gov/documents/suite/SOM_PMM_Manual_456390_7.pdf
- *Project Management Fact Sheet: Language Matters*. (2017). Department of Premier and Cabinet of Tasmania. Version 1.2, November 2008. https://www.dpac.tas.gov.au/__data/assets/pdf_file/0028/108946/Language_Matters_Fact_Sheet.pdf
- Winter, D. (2012). *Project Stakeholders Gas Plant Kazakhstan*. http://www.order-efficiency.com/demo/stakeholders/LinkedDocuments/Project%20Stakeholders%20Gas%20Plant%20Kazakhstan%20[example].pdf

TABLES

Table 9. Project Approach > Coordination: *Project coordination and control tools. These essential tools represent the source of information and thought processes that are needed to effectively plan and execute a project.*

Tool	Description	Value	Application
Project charter	Initializes project	Provides integration of project into society	Projects interface
Project definition document	Defines project purpose, objectives, deliverables, completion criteria, and scope of work to be completed, explains project type	Provides boundaries and communicates understanding	Project interface
Requirements	Defines the specifications for the product/output of the project	Provides tracing of actionable information	Requirements interface
Project schedule	Shows all work efforts, properly estimated, with logical dependencies assigned to responsible resources scheduled in a calendar	Provides for coordination of the execution of activities with objects in time	Schedule interface
Status reports	Periodic or continuous reviews of actual performance versus expected performance	Provides feedback to allow for timely and appropriate identification of performance variances	Control interface
Key event chart (Milestone chart)	A summary of the detailed project schedule showing progress against key events in time	Provides a high-level project progress report on one page	Control interface
Project organization chart	Shows all project associated individuals and the working relationships among them	Provides a source for identifying the organizational structures, dynamics, and project roles	Control interface
Responsibility matrix	Defines all roles and indicates what responsibilities each role has	Provides a source of coordinated expectations, and tool for establishing and accountability	Control interface
Communication plan	Defines the how, what, when, and who regarding the flow of project information	Provides a tool for effective communication among working [team] members	Communications interface
Logistical coordination plan	Lists how project resources and humans will be acquired, when they are needed, how much are needed, and how long they will be needed	Provides for scheduling	Resources interface
Quality assurance plan	Defines the approaches and methods that will be used to resolve the quality levels of project processes and results	Provides a tool for reducing uncertainty the results of project execution	Reasoning interface
Risk coordination plan	List each identified risk and the planned response procedure for each	Provides for the communication about potential issues in advance, is a proactive measure to reduce impact to a project	Risks interface
Project plan	Formal, programmed data structure [document] used to coordinate project execution and control	Provides for an whole, unified directional information set	Plan interface
Deliverable summary	Defines and lists all deliverables to be produced by the project	Provides visibility, tracking, and reporting of deliverables	Deliverable outputs interface
Project log	Records essential information for each project risk, issue, action item, and change request	Provides visibility, tracking, and reporting of items impacting the project	Log interface
Change request form	Records essential information for any request change that impacts the scope, schedule, or resource requirements (budget)	Provides for the proper assessment and communication before a change action item is taken	Change interface
Project repository	The location where all pertinent project information is stored	Provides a single source of reference for all project information	Project database and search
Project interface (notebook)	Software tool used by a project coordinator and by project contributors to record and interface with a project	Provides the interface	Software

TABLES

Table 10. Project Approach > Coordination*: Project management standard differences.*

	PMBOK (2016)	ISO 21500
Process groups	Initiating Planning Executing Monitoring and Controlling Closing	Initiating Planning Implementing Controlling Closing
Knowledge areas (subjects or activities)	10 Knowledge Areas (KAs)	10 Subjects

Table 11. Project Approach > External Standards: *Popular established references by development category.*

Development Category	Description	Popular standard or reference
Product standards or guides	Characteristics related to quality and safety	ISO 9001 Quality Management Systems
Process standards or guides	Conditions under which products and services are produced or packaged	ISO/IEC 15288 systems and software engineering – system life cycle processes
Project management standards or guides	Helps organizations to manage their operations or project	PMBoK

Table 12. Project Approach > Contributed Deliverable Project State: *Project state variables. There are 4 variables that describe the current state of each project: Version (∞ potential values): the released version accessible to the public. Stage (4 potential values): the level of readiness/completion of the current version. Status (4 potential values): the type of activity for the current stage. Dependency (2 potential values): is development blocked by one or more dependencies.*

Version	Description
0	No released version.
1	Initial version release.
2..n.. ∞	Subsequent releases/updates.
Stage	**Description**
Development (Dev)	Active development/work of current version is underway.
Alpha	Early testing of current version is complete.
Beta	Early testing of current version is complete.
Production (Prod)	Full production release of current version is ready.
Status	**Description**
Todo	Current stage not yet started.
Active	Current stage under development.
Under Review	Current stage read to be reviewed for editing and/or testing.
Done	Current stage is complete.
Status	**Description**
Blocked	Development is blocked by a dependency.
Ready	Development is ready to continue.

TABLES

Table 13. Project Approach > Work*: Work product classification scheme.*

WP ID	Generic work product class	Generic work product description	Generic work product typical characteristics
1	Object	An entity created to serve a purpose, or created in the course of serving that purpose. Its existence is observable and rationalised by its material or behavioural characteristics. It may exist as a complete, partial or exemplifying realisation of a product, be a subordinate part of a product, be a by-product or be a part of an enabling system	identity, name of object purpose, value that caused its creation ownership and responsibility for object status, state and classification of object distinguishing observable qualities and properties functional and behavioural characteristics dimensional and parametric characteristics relationship with and dependencies on surroundings observable interactions or effects on other objects interfaces, connections to surroundings location, position in surroundings safety, security, privacy and environmental regulations
2	Description	An account or representation of a proposed or actual object or concept. It may be a textual, pictorial, graphical or mathematical representation. It may be in a standardised form for human or machine interpretation. It may be a static or dynamic model or a simulation representing reality. It may establish order, structure, grouping, or classification.	object, subject or class represented purpose and applicability of description concerned parties, viewpoints, views range of use, and validity of description accuracy, detail and abstraction level model dimensions, degrees of freedom description language, notation, nomenclature applicable standards, formats and styles representations of function, attributes, properties descriptions of architecture, arrangement, interfaces depiction of composition or form definition of classification, category, ranking, type
3	Plan	A proposed scheme or systematic course of action for achieving a declared purpose. It predicts how to successfully accomplish objectives in terms of specific actions, undertaken at defined times and employing defined resources. It may apply to technical, project or enterprise actions. At a high level of abstraction it may be a policy or, with reference to assets and their disposition, a strategy.	definition of undertaking, purpose and objectives of plan strategy and policy guiding plan plan owner, stakeholders, responsible parties and their authorities plan status, version, reviews and modifications proposed events, actions and tasks predicted timescales, durations, dates of actions assumed dependencies, conditions, constraints, risks allocated resources, labour, facilities, materials planned budget, cost, expenditures defined milestones, results and progress targets decision points and authorisation gates options and contingency actions
4	Procedure	A declared way of formally conducting a customary course of action. It defines an established and approved way or mode of conducting business in an organisation. It may detail permissible or recommended method in order to achieve technical or managerial goals or outcomes.	purpose, outcomes and results of performing actions issuing authority and controls roles, responsibilities and duties actors, their competence and proficiency dependency on requirements, standards and directives achievement, goals, completion criteria definition of transformations and their products work definitions, instructions to act progression and dependencies of action guiding method and practices enabling tools and infrastructure
5	Record	A permanent, readable form of data, information or knowledge. Accessible and maintained evidence of the existence or occurrence of facts, events or transactions. It may take the form of a journal chronicle, register or archive. It may contain the information to confirm achievement of performance, fiscal or legal conditions or obligations.	record identity or title content, description and reason for record ownership, origin and authorship practices, agreements, commitments and regulations applying to record authorities and condition of storage, retrieval, replication and deletion medium and format of record location, conditions and periods of storage applicable information privacy, security and integrity declaration of status, configuration and baseline information information on audit, validity and history

TABLES

WP ID	Generic work product class	Generic work product description	Generic work product typical characteristics
6	Report	An account prepared for interested parties in order to communicate status, results or outcomes. It is a result of information gathering, observation, investigation or assessments, and it may impart situation, affects, progress or achievement. It serves to inform so that decisions or subsequent actions can be taken.	purpose or benefit of repot source, author and authority to report interested parties, recipients, distribution knowledge, understanding communicated information, data, facts and evidence contained analysis, inspections and audits employed timing, validity, condition of information use dependence on circumstances, constraints and assumptions reported status, results, achievements, conformance, compliance or outcomes identified faults, failings or errors inferred patterns, trends or predications conclusions, recommendations, rationale
7	Request	A communication that initiates a defined course of action or change in order to fulfil a need. This may originate or control on-going action based on an agreed plan or procedure. It may result in a proposal or plan of action. It may take the form of a solicitation, requisition, instruction or demand for a resource, product, service or an approval to act.	objective, purpose or outcome of request expression of a demand, need or desire communication of enquiry, solicitation or an order to provide initiation of supply, provision or support definition of action, change or exchange identification of required products, services, capability or resources authorisation of tasking or commitments specified terms, conditions to act, agreement conveyed required availability of requested provision communicated
8	Specification	Criteria or conditions that place limits or restrictions on actions, attributes or qualities. It establishes measures or qualities for determining acceptability, conformance or merit. It may be required as part of an agreement or contract.	definition of needs, expectations and circumstances statement of requirements definition of constraints and conditions standards and regulations invoked dimensions of achievement and outcome criteria of conformance, correctness and compliance definition of measures, indicators, limitations, values, and thresholds statements of action and conduct required functions, performance, behaviour or service levels definitions of interfaces, interaction, location and connection conditions of acceptance, permissible exceptions and deviations conditions of change and variation

Approach: Engineering

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: engineering approach, engineering management, engineering coordination, engineering planning, engineering operations, technical development, design and simulation,

Abstract

Engineering means working to develop a specification, and then, working from that specific specification to construct and operate. In other words, engineering means to work to develop, and then follow, a single (unified) work plan. In general, engineering also carries the connotation of doing 'useful' work (as opposed to doing 'unnecessary' work). Engineering works outward from given goals and specifications, and proceeds systematically. An engineering/-ed system, is a system designed, realized, and operated through real-world process (by engineers) to achieve a particular purpose. Society may be viewed of as an engineering process and resulting deliverable (Read: societal engineering). If society is to be engineered through cooperation and contribution, then it must identify its alignable life-cycles, its life-cycle processes, its means of design and development, its fundamental[ly encodable] system conceptions, its requirements, its construction, its information-based, its informatics base, and its methods of layering and associating societally relevant information. To produce real world useful operations it is necessary to use a system of design and operation that can separate and combine conceptual and spatial information.

Any approach to state change in the material environment requires work. Useful work requires a systems approach to socio-technically coordinated state re-creation. The integration of project coordination (project management) and system engineering is projects engineering.

Graphical Abstract

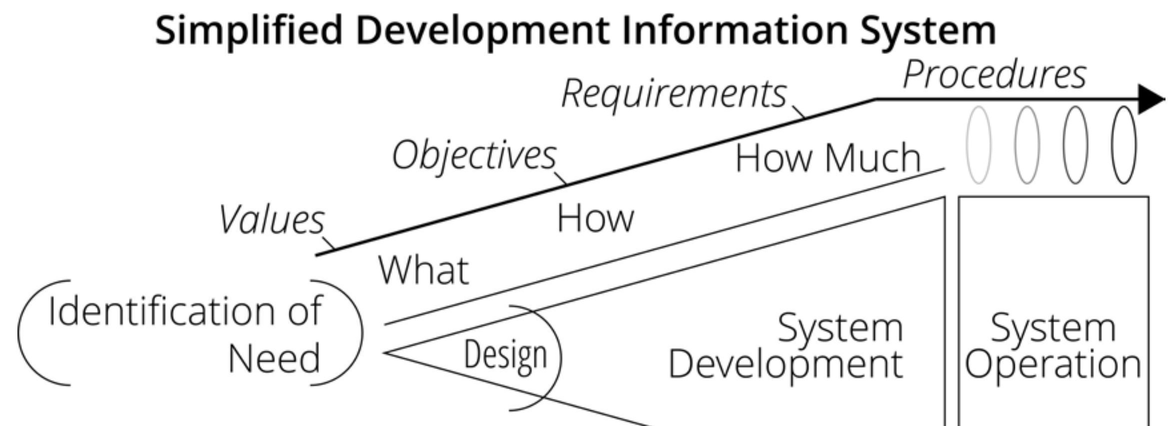


Figure 5. *From the identification of a need comes the development of an operational system sustained by resources and procedures/ processes, which are developed through an oriented design.*

1 What is engineering?

A.k.a., Unified development life cycle approach, societal designed operations approach (a.k.a., organizational design, enterprise design, business design, ...), the in[ter]-system service development approach, the construction service approach.

Engineering means working to develop a specification, and then, working from that specific specification to construct and operate. In other words, engineering means to work to develop, and then follow, a single (unified) work plan. In general, engineering also carries the connotation of doing 'useful' work (as opposed to doing 'unnecessary' work). Engineering works outward from given goals and specifications, and proceeds systematically. An engineering/-ed system, is a system designed, realized, and operated through real-world process (by engineers) to achieve a particular purpose.

Engineering is the application of the principles of science and mathematics to develop effective solutions to socio-technical problems. In society, engineering is a purposeful activity directed toward the goal of fulfilling human requirements through socio-technical [service] design; particularly, those needs that can be met by socio-technical composition. As a project cycles from an idea to the implementation, delivery, and operation of a product or service, engineering links logic and scientific discoveries to functional applications that meet individual and societal needs.

INSIGHT: *Losing function is losing the capacity to do something.*

Technology (and its operation) is the direct result of engineering. However, scientific inquiry and engineering, together, are the basis for all technology. Useful categories of objects (or systems) constructed and/or operated by engineering are called 'technology'. Technology (and its operation) is the practical application of engineering knowledge including procedural (informed by scientific inquiry).

Engineering, as an approach, is:

1. A real world, technical, problem solving activity that uses data, knowledge, and tools to materialize systems. In this sense, engineering is the materializing and materialized aspect of a societal information system.
2. The knowledge required, and the process applied, to conceive, design, make, build, operate, sustain, and/or recycle a system of technical content for a specified purpose (e.g., a concept, a model, a product, a device, a process, a system, a technology, etc.).
3. The application of knowledge and tools in the form of a process to solve discrete problems in the real world (i.e., engineering is concerned with real-world processes using scientific knowledge).
4. The design, production (development), and operation of systems that must work as expected, and hence, engineering is concerned with observable (or experienceable) outcomes (Read: knowledge applied to develop a technical solutions to a discrete problems).
5. Methodically (and systematically) conceiving and implementing viable solutions to existing problems.

Engineering necessitates organizational understanding -- the ability to organize information for a purpose. In engineering, organizational engineering requires an understanding of how to extend (i.e., enhance) the capabilities of the whole, while attempting to better understand the relationships and interactive effects among the components of the organization, and with its environment. Engineering outputs should maintain and/or improve the quality of life among a community of [technical] users. Typically, the conduct of engineering leads to systems that enable and enhance the capabilities of humans, while also responding to the needs and constraints of humans. Therein, engineering is responsible/accountable for the design, implementation, operation and maintenance of a real-world system.

To the engineer, engineering (constructing) anything into existence is based at a fundamental level on [concept- and object-systems] modeling. To the user [of technical systems], engineering represents a technical, knowledge using, life fulfillment support process.

Engineering is composed of (i.e., involves):

1. The process(es) of designing and later operating [planned production] systems based on logic and scientific principles (i.e., scientific knowledge), cycling productions through a habitat, forming a habitat service system (i.e., city) that fulfills human needs.

1.1 The core engineering processes

Engineering consists of two primary processes, each of which has multiple sub-processes:

1. **The development (including design) process** - the step-by-step development of a service or object.
2. **The operations process** - the step-by-step operation of a service or object.

The complete systems engineering life-cycle contains both a system development life cycle and a system operations life cycle. In terms of physicality, a core engineering system must account for:

1. **Flows** of physicality and stocks of physicality. A flow is a variable that measures a quantity per time

period. The motion of objects.
2. **Stock** is a variable that measures a quantity per point in time. The repository of objects.

1.1.1 The development (including design) process

Engineering to create future systems that operate in real-time. In concern to the design (and development) of systems, engineering is a design process, combining knowledge of the properties of materials, models that predict how these materials behave, and systematic thinking, to create solutions to human needs in physical matter reality (i.e., in the real world).

Early in the system development activity, a system is conceptual in nature. A system may consist of several levels where each element at each lower level may by this definition itself be considered a system (i.e., a subsystem of a large system may itself possess all of the attributes of a system).

Engineering will define:

1. The technical specification of the projected system.
2. The technical specification for the system's complete delivery, including integration and eventual de-integration.
3. The method of technology involved in executing the project.

1.1.1.1 Engineering design de-composition

The axiomatic dimensions of real world engineering (Read: development and operations of a system) by means of a project structure must account for that which existence is composed in order to bring something new (or a change to) systems in existence.

In the real world, a project has 4 axiomatic dimensions:

1. 1D - memory (knowledge).
2. 2D - direction (adds objective).
3. 3D - spatial construction (adds resources).
4. 4D - schedule (adds time).

In the market[-State], a project has 1 additional axiomatic dimension:

- Market-based 5D - transaction cost (adds market expense)

In the community, a project has 1 additional axiomatic dimension:

- Science-based 5D - environment (adds probability)

Combination:

1. 1D + 2D => information model.
2. Information model + 3D (space) => physical model.
3. Physical model + 4D (time) => service model.
4. Service model + 5D (cost) => profit model.
5. Service model + 5D (environment) => probabilities model [that humans will have the fulfillment of their needs through a service environment].

Herein, a material[-ized] service system is made up of software, hardware, and data that provides its primary value by the execution of a service for its users.

> **INSIGHT:** *To build something from the ground up "you" have to understand it in a way that "you" may not have to understand when "you" are looking at something that is already built.*

Different societal configurations have different interfaces. In the market-State, the following interfaces are required/present, which are not required/present in a community-type configuration.

State interface requirements:

1. **Contractual agreements** with an authority (jurisdictional or otherwise).
2. **Financial exchange** of currency.

Market interface requirements:

1. **Contractual agreements** with competing market entities (and an authority to enforce contract with punitive/retributive damages).
2. **Financial exchange** of currency.
3. **Demand and delivery** of object(s) or service(s).

1.1.2 The operations process (as engineering)

Engineering upon created system that operate in real-time. Note that engineering operations my involve engineering design and development. Ensuring that the [physical] behavior of the various components of a system are coordinated as required, to ensure a proper functioning of the whole system.

If there is engineering development, then there is:

1. **Development** of a new system, or
2. **Modification**, upgrade, change, iteration to existing system/product.

If there is engineering operations, then there is:

- **Operating** an actually measurable system, that can be monitored, and possibly, controlled.

1.1.3 Measurement and engineering

Engineering measurement can be categorized in two ways:

1. Direct measures - measures of the engineering process (e.g., effort, resources, and cost applied) and product (e.g., produced, lines of code (LOC), etc.).
2. Indirect measures - measures of the product (e.g., functionality, quality, complexity, etc.).

Engineering measurement requires normalization of both size-oriented and function-oriented metrics:

1. Size-oriented metrics (a.k.a., size-oriented key measures):
   A. For example, lines of code (LOC) can be chosen as the normalization value:
      1. Errors per KLOC (thousand lines of code).
      2. Defects per KLOC.
      3. Cost ($) per KLOC.
      4. Pages of documentation per KLOC.
   B. Function-oriented metrics (a.k.a., function-oriented key measures)
      1. The most widely used function-oriented metric is the function point (FP). A function point (FP) is a unit of measurement to express the amount of functionality (societal functionality, business functionality, etc.) an information system provides to a user. NESMA FPA Method: ISO/IEC 24570:2005 Software engineering - NESMA function size measurement method version. Computation of the FP is based on characteristics of the system's information and physical domains, and their complexity. To determine the number of FPs, classify a system's features into five classes:
         i. Transactions - external inputs, external outputs, external inquiries.
         ii. Data storage - internal logical files/objects and external interface files/objects.
         iii. Note: Each class is then weighted by complexity as low, average, or high. Then, the result is multiplied by a value adjustment factor (determined by asking questions based on a set number of system characteristics).
   C. Object-oriented metrics:
      1. Number of scenarios scripts (use-cases).
      2. Number of key classes.
      3. Number of support classes (required to implement system, but are not immediately related to the problem domain).
      4. Average number of support classes per key class (analysis class).
      5. Number of subsystems (an aggregation of classes that support a function that is visible to the end-user of a system).

1.2 [Systems] Usability

Systems are used by users; to be usable by a user, systems can be designed to be usable. The use of a system to is user is usability. The International Standards Organization (ISO) defines usability as "the extent to which a product can be used by specified users to achieve specified goals" (ISO-9241-11, 1998). Usability is a key element of the human-centered design (HCD) approach, and it has been shown to increase efficiency, effectiveness, and user satisfaction. Furthermore, designs with good usability can reduce errors, fatigue, training time, and overall life cycle costs. Usability is a key component of human-centered design. Human-centered design focuses on users' needs to design the system based on users' capabilities. Usability testing and evaluation methods provide user performance measures and subjective (qualitative and quantitative) comments that can be used to improve the system in question throughout the engineering design life cycle.
Usability testing and evaluation is an iterative process. Usability evaluations should be conducted several times during the life cycle of the system, and results should have a direct influence on system design, providing continuous feedback for the designers of the system. Usability should be part of the system development life cycle from the earliest stages, to make sure that users' needs, capabilities, and limitations are considered from the start of design and development.

Standards, as a control(s), make usability efficient. For example, a vehicle pedal set is standard to all vehicles. Any user can get in any vehicle and the foot pedals operate similarly, thus providing interoperability for a user.

INSIGHT: *From usability originates reusability.*

1.3 [Systems] Engineering

Technically, all engineering is "systems" engineering. In the past, many engineering organizations did not follow a systematic approach, and hence, the term 'systems' was added to engineering to emphasize its essential systematic approach. The word systems also connotes that engineering is an information-based process. If differentiated, then reasoning about systems (i.e., systems *reasoning*) is the essence of [systems] engineering. However, take note that in the market, the term 'engineering' often refers to discrete instances of the application of engineering, whereas the term 'systems engineering' often refers to the oversight of engineering at the organizational (or management) level. The term system is added to the term engineering because that which is being developed and operated through engineering is a system (pattern). Herein, systems thinking is a way of dealing with increased complexity. The fundamental concepts of systems [thinking] involve: understanding how action and decisions in one area affect another, and that the optimization of a system within its environment does not necessarily come from optimizing the individual system components. To do system engineering, someone (or something) must understanding what a system is, its context within its environment, its boundaries and interfaces and that it has a lifecycle. Fundamentally, systems engineering is a global[ly integrated] engineering approach. Note here that because 'community' is a 'unified system', practically speaking, the terms 'engineering' and 'systems engineering' are synonymous unless specified otherwise (as would need to be specified for market-State conditions).

> **CLARIFICATION:** *Engineering complex systems necessitates a project-based approach for purposes of optimal coordination. In the [systems] engineering approach, the project, itself, is a system that applies all the principles of [systems] engineering: it has a purpose, interacts with an environment, and represents a solution to users' requirements.*

All engineering (in community at the organizational level) is, technically, systems engineering. Engineering has always implicitly drawn on systems-oriented principles and practices. However, a distinguishing characteristic of systems engineering is its continual reference and orientation towards an explicitly, developed body of systems reasoning, knowledge, experience and practice. Much of this body of knowledge has come from studies in control engineering, cybernetics, information science, biology.

> **HISTORICAL NOTE:** *The term 'system' was introduced into engineering in the 1940s, leading to the rise of 'systems engineering' in the 1950s and 1960s.*

Systems engineering is the engineering process to create and operate a system. It is a structured process based on data. Take note that there are a large range of accurate definitions in the literature for both engineering and systems engineering. In its most broad definition, [systems] engineering is the process of bringing into existence a functioning, technical system for some user.

> **CLARIFICATION:** *Engineering does not proceed by straightforward application of natural science. Constructions derived from scientific theory have to be tested (and usually, modified) in order to obtain practical, useful technology. Engineering must follow the natural laws and rely on the basic resources in nature such as materials and energy. In science, there is a discoverable, initial whole. In engineering, the whole (design-solution) does not initially exist; it is constructed.*

The following is a common list of definitions of the concept of 'systems engineering':

1. The systematic application of science, tools, and methods to find an effective solution to a problem using a quantifiable approach to create for the development, operations, and maintenance of systems.
2. The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software (ISO/IEC/IEEE 2012).
3. The systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (IEEE 1990).
4. The interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a solution and to support that solution throughout its life (BKCASE 2017; ISO/IEC/IEEE 2010).
5. The interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining user needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem (BKCASE 2017)
6. The integration of disciplines into a team effort forming a structured development process that proceeds from concept to production to operation. (INCOSE 2012).
7. A disciplined approach for the definition, implementation, integration and operations of a system (product or service) with the emphasis on the satisfaction of stakeholder functional, physical and operational performance requirements in the intended use environments over its planned life cycle within cost and schedule constraints. Systems engineering includes the engineering activities and technical management activities related to the above definition considering the interface relationships across all elements of the system, other systems or as a part of a larger system. (NASA Systems Engineering Handbook SP-601S).

In the context of the engineering approach, systems engineering is:

1. The coordinated design, development, and operation solutions that retain optimal systems-level performance for specified objectives. Therein, in order to develop and sustain optimal performance, systems engineering uses information from a whole/unified information system.
2. The iterative and interdisciplinary processes of designing and developing new [systems] solutions to complex real-world problems by transforming

requirements into operational systems.

Systems engineering is composed of (i.e., involves):

1. A set of procedures (i.e., practices) that rely on enabling competencies (knowledge sets) and structures (organizational and procedural) at individual, team, and organizational levels to coordinate the design, development, and operation of solutions that maintain optimal systems-level performance.
2. The processes of designing, developing, and operating a system(s) embedded within a life-cycle. Thus, systems engineering is, in part, focused on the long-term and life cycle of a system, necessarily taking into account the cradle-to-grave (or, cradle-to-cradle) life of the system.

> **NOTE:** *In the market, systems engineering is defined as part of a continuum of business processes. In community, systems engineering is defined as part of a continuum of organizational processes.*

In general, the output of [systems] engineering entails two interrelated viewpoints:

1. The system as a created product, which is used by users.
2. The system as a delivered service, which is serviced by technicians.

The term [systems] engineering can be applied to:

1. **The system (i.e., the solution, itself)** - The [design and life cycle of the] system to be developed and operated (i.e., the technical system itself).
2. **The decision system** - the decisioning and organization that brings the system, itself, into existence. The system that controls and coordinates (i.e., decides, "manages") the development of the technical solution-system.

In general, [systems] engineering must account for:

1. The whole [systems] development process.
2. Integration of a new system (or system's state) into an environment of existing systems.
3. The life-cycle of the new system in an environment of existing life-cycles.
4. Planning the operation of, and the actual operation of, the system.

> **QUESTION AND ANSWER:** *What is [always] in operation? A developing system is [always] in operation.*

The word 'systems' in the term 'systems engineering' implies, in part, that the systems engineering approach is (i.e., the [systems] engineering approach maintains the following characteristic -- in order to enable the realization of successful, optimal systems, systems engineering is):

1. **Interdisciplinary / multidisciplinary** - engineering has access to all available "branches" of knowledge.
2. **Holistic / unified** - engineering has access to all available data and information while considering the full system, including any number of performance criteria, as well as potentially non-technical concerns related to human factors or societal impacts. Engineering has access to all relevant information to the problem, context and situation.
3. **Integrative (integral)** - engineering combines all available information, including that which is learned during the engineering process itself, into an optimal solution. Engineering requires the integration of multiple views and information sets. Engineering accounts for the whole, as well as the parts that makeup the whole).
4. **Completeness** - engineering completely satisfies the problem with a solution.
5. **Procedural (documented and planned process/ method)** - engineering requires identification, documentation, and improvement upon a method/process. Engineering defines methods of specification, prediction, and control.
6. **User-/Value-driven (utility)** - engineering considers the needs and interests of all users and stakeholders (of everyone impacted by the system).
7. **Collaborative** - engineering involves working with other teams and systems in a sufficiently open information space to produce a safe and reliable system. Systems are developed by teams of engineers, and everyone must be able to understand one-another's work (Read: readability/ understandability).

[Systems] Engineering enables (or, brings) to an organization the following [value] alignment characteristics:

1. **Correctability (correctness or correct alignment)** - the system is capable of adjusting action and information to a direction, standards, or need. All actions/decisions are correct according to the organization's direction. The system ensures that the correct [technical] tasks get done during development and/or operation.
2. **Validity** - the system is capable of taking decisions and actions that are correct and relevant to the problem-requirement. Given a relevant direction, every action should relate to that direction.

3. **Relevancy** - the system is capable of measuring the of pertinence to context, problems and needs.
4. **Consistency** - decisions and actions are consistent with other decisions and actions, and the organization's direction.
5. **Minimality** - the system is capable of meeting requirements exactly.
6. **Extensibility (adaptability)** - the system is capable of adapting to changing requirements.
7. **Flexibility** - the system is capable of integrating new information flows.
8. **Non-redundancy** - the system is capable of informing and acting without unnecessarily repetition.
9. **Value** - the system is capable of delivering the intended benefit to individuals and society.

[Systems] Engineering must necessarily account for:

1. **Complexity** - the interrelationship between multiple, seemingly separate, information sets (and viewpoints, which requires multi-view analysis).
2. **Uncertainty** - during a system's design, there are unknowns; and during a system's operation, there may be unknowns.
3. **Potentiality** - that [dimension] from which [axiomatically] systems ("things") can emerge.

> **CLARIFICATION:** *In some engineering design cases, given what is known and available, sub-systems and component designs may need to be sub-optimal in their designs for the who system to be optimized.?*

An organization's [systems] engineering capability may be equated to its ability to dependably conduct activities that traceably flow from a/an:

1. Knowledge base [documentation base].
2. Experience team base.
3. Competent team base.
4. Enabling systems [a set of relevant organizational assets].

There are [at least] three ways that humans can be involved in engineering systems:

1. Being the designer/developer of the system (e.g., design/development engineer).
2. Being an operator within the system (e.g., technician engineer).
3. Being a user (i.e., requirer) of the system.

Fundamental inputs of an organization's [systems] engineering capability are:

1. Information systems (organizational, decisioning, etc.).
2. Human systems (personnel with abilities).
3. Equipment systems (tools, facilities, etc.).

Engineered systems (or engineered specifications) may change given:

1. New scientific/engineering knowledge.
2. New problems/requirements.
3. New technologies (i.e., new systems).

> **NOTE:** *There is no need to put the term 'evidence-based' in front of 'engineering', because it is assumed.*

1.3.1 System of systems engineering

There is the concept of 'system of systems engineering' (SoSE), wherein, the term "system of systems" (SoS) is somewhat problematic. From a cybernetic perspective, a SoS is a meta-system, an integrated system composed of other systems. Thus, the concept of "system of systems" is tautological, since systems themselves are considered to be comprised of sub-systems, and therefore, a "system of systems" is itself just a system. In general, the term meta-system and system of systems, specifically, refers to a system with multiple embedded and inter-related autonomous complex sub-systems. These sub-systems can be diverse in technology, context, operation, location/geography, and conceptual frame. These complex sub-systems of a meta-system must function as an integrated meta-system to produce desirable results in performance and to achieve a higher-level purpose (mission, etc.) subject to constraints. In other words, a system of systems or meta-system is generally defined as an assemblage of components, themselves considered as systems, with the added distinction of coordinated and operational independence of components.

A SoS brings together systems in order to perform a higher level mission/purpose of which each member system plays an integral role. An SoS is a 'complex system', and as, such exhibits dynamic and emergent behavior and requires engineering to design and operate.

Common definitions of system of systems engineering (SoSE) include:

1. The design, deployment, operation, and transformation of a [meta]system that must function as an integrated complex system to produce desirable results.
   A. The integration of multiple, potentially previously independent, systems into a higher level system (meta-system).
   B. The functional design of a SoS that generates capabilities beyond what any of the constituent systems is independently capable of producing.

It is important to note that when previously independent systems are integrated (i.e., at the time of integration) that there exists some degree of constraint imparted by their part/position in the larger system (i.e., in the meta-system).

1.4 The structured systems analysis and design method (the SSADM)

This *Solution Inquiry* process follows an [agile] structured systems analysis and design methodology (aSSADM). It is a systems engineering methodology and involves systems-based processes structured in such a way as to produce well-documented, accurate design outputs. It uses a formal, methodical approach toward the analysis and the design of solutions as components of systems (real world community > habitat system > habitat subsystems).

The SSADM herein follows a modified waterfall life-cycle model starting with a requirements analysis, leading to a [comprehensive] technical feasibility study (is it technically possible), and progressing through to the physical design stage of development, while accounting for qualifying requirements, protocols, tasks, and resources. One of the main features of SSADM is the intensive user involvement in the requirements analysis stage. Every good and service is designed in transparency to the entire community and the community can improve the design by discovering more about the natural environment and combining known elements in uniquely creative ways.

Engineering designs involve layers of functional diagrammatic representation. The product of a structured method is a technical design specification that can be engineered into the habitat through the re-organization of resources.

The most efficient form of action under a systems-based approach is that of the project-based approach (i.e., team-based approach). The SSADM breaks up issues into their composite projects, stages, modules, steps and tasks – as every well-applied systems/team/project approach does. A 'project' is just a collection of tasks that a team is applying effort toward with the intention of fulfilling a larger and more integrated purpose.

Remember, a system involves at least: inputs > processes > outputs. The systems and individuals involved in the design of solutions arrived at via the *Solution Inquiry* processes derive their input from a common and verifiably founded repository of data, knowledge, and values. Systems design, therefore, is the general process of defining the architecture, components, modules, interfaces, and so forth, of a system [to satisfy specified requirements using an explicit repository of information as input].

> **NOTE:** *The Solution Inquiry processes could impact the time-frame prioritization of an issue's resolution if it requires significant design time.*

1.5 Generative design

A.k.a., Algorithmic design.

Herein, the computer, as an information processing system for calculation and optimization, becomes part of the design process (with the human designer and/or user). Computers can run hundreds, millions and billions of calculations that facilitate the engineered optimization of physical objects and systems. For example, a heat exchanger may be designed by the computer (Read: machine algorithm) - the designer inputs the following data elements: here is the space I want it to occupy, here is the volume, here is the thermodynamic requirements, here are the materials, and here are the technologies (Read: material production configurations) that meet those requirements. And, the computer runs a series of calculations including predetermined engineering formula and the new data inputs, and therein, it works to optimize (to its programmed potential). The result is then tested. Given appropriate engineering rules, it is possible to produce a system, given what is knwon, that is nearly optimal in terms of its engineering characteristics (and aesthetic).

1.6 Generic design information

Engineers apply scientific and technological information in designing services, structures, and systems (i.e., structural service systems). Herein, when an engineer creates a design specification, it has the following generic metadata:

1. **The System Requirements Specification (SRS)** is a specific record of the required characteristics (functional & aesthetic) of a system. It is the characteristics that any solution is required to possess. Usually, it also includes any goals.
2. **The Operational Concept Description (OCD)** is a system-centric description with respect to:
   A. The intended users of the system (human and/or elements of technology).
   B. The intended uses of the system.
   C. How it is intended that the system be used.
   D. The conditions, external to the system, within which it is intended that the system be used.
3. **The Architectural Design Description** (ADD; e.g. SSDD, CONOPS, IEEE 1471 design description, etc.) refers to the identification of the elements of the solution, together with the key characteristics of each element and the concept of interoperation of the elements to satisfy requirements.

1.7 System engineering life-space cycles

To optimize for "environmental performance", impacts (as affects and effects) must be considered through the entire life-cycle of a good, service, or system. Here, analytical processes measure these impacts to the best of their abilities. And, we formally determine their threshold of acceptability.

The most thorough way of assessing environmental performance factors (including an alignment with a social value set) is through the process of **life-space cycle assessment**. All iterating systems have a life-space cycle (also sometimes known as a "life-cycle"). The results of a life-cycle assessment may be compared against a benchmark, potentially a *safety benchmark*. The objective of a life-space cycle assessments is not only to identify technical feasibility, but also to identify where environmental impacts originate from and make them explicit in such a way that individuals are capable of prioritizing and setting metrics around them.

The systems engineering [inquiry] life-cycle involves the following parallel conceptual processes:

1. Discover the need for new information;
2. Discover the new information;
3. Understand and integrate the information; and then
4. Arrive at an informed and systems-based technical solution re-orientation (through novel, creative information) to the need that generated the inquiry for new information.

NOTE: *In art, people often see what they want to see; in engineering precision creates operational technologies. Stated in an alternative way: in art, 'abstraction' facilitates subjective perception, and in engineering, 'specification' facilitates useful functioning.*

1.7.1 Brief technical overview of systems engineering

NOTE: *In systems engineering, if it cannot be identified through an integrated visual interrelationship, then it is not understood.*

Systems engineering is an interdisciplinary field of engineering focusing on how complex engineering systems and projects can be designed and structured over their life cycles. Systems engineering involves the analysis of users' needs, the identification of required functionality, the explication (or "documentation") of requirements, then proceeding with design synthesis and system validation, all the while considering the context and root of the issue in which the problem has arisen. Essentially, systems engineering concerns the planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a capability greater than the sum of the capabilities of the constituent parts. Systems engineering involves the process of designing "in-motion" systems (i.e., dynamic systems). System development often requires contribution from diverse technical disciplines. And, the result is one highly integrated information-physical design. By providing a systems (holistic) view of the development effort, systems engineering facilitates the aggregation of all technical contributors into a unified team effort, forming a structured development process that proceeds from concept to production to operation and to updating, and in some cases, to termination and to recycling.

Visualization and structural models play a principal role in systems engineering, and in the communication of experience in general. A 'model' may be defined herein in several ways, including:

1. An abstraction of reality designed to answer specific questions about the real world; or
2. An imitation, analogue, or representation of a real world process or structure; or
3. A conceptual, mathematical, or physical tool for organizing the arrival of a decision.

Systems engineering involves the use of tools and methods to better comprehend and manage complexity in systems. These tools and methods lead to information that is not open to interpretation or speculation, and can be used in engineering material systems. Engineering inquiries have right and wrong answers, and there isn't any "wiggle room" for interpretation. "True enough" isn't "good enough" in the any sciences, let alone engineering sciences. Material structures must be built with intention and accuracy otherwise they put the community of their users at risk.

Some examples of these tools, techniques and models are:

1. System modelling and diagramming.
2. Simulation modelling (i.e. modelling and simulation).
3. Systems architecture design.
4. Optimization design.
5. System dynamics design.
6. Systems analysis.
7. Statistical analysis.
8. Reliability analysis.
9. Probability analysis.
10. Technical, operational, and systems specifications (and views) for visual, blueprinted representation.

Solution Inquiry is a *structured systems* process because projects are structured into small, well-defined activities wherein the sequence and interaction of these activities is specified, and because diagrammatic and other modelling techniques give a more precise [structured] definition that is understandable by the whole community.

The three most important tools in a systematic structured solution-orientation are:

1. **Logical data modelling:** This involves the process of identifying, modelling and documenting data as a part of the gathering of system requirements. The data are classified further into entities and relationships.
2. **Data flow modelling:** This involves tracking the data flow in an information system. It clearly analyzes the processes, data stores, external entities, and the movement of data.
3. **Entity behavior modelling:** This involves identifying and documenting the events influencing each entity and the sequence in which these events happen.

Some of the important techniques and models include:

1. Logical Data Models.
2. Data Flow Models.
3. Requirements Definition.
4. Function Definition.
5. Specification Prototyping.
6. Relational Data Analysis.
7. Entity/Event Modelling (Entity Life Histories and Effect Correspondence Diagrams).
8. Technical Options.
9. Dialogue Design.
10. Update and Enquiry Process Models.
11. Physical Data Design.
12. Physical Process Specification.
13. Physical Design Control.
14. Gantt charts.
15. Critical path analysis provides a method of systematizing our knowledge so that the effect of decisions of order of action can be seen.

Common diagrams include:

1. Activity & state diagrams
2. Class diagrams
3. Sequence diagrams
4. Service diagrams
5. Operational concept and connectivity diagrams
6. Organizational relationship diagrams
7. Formulaic and matrix representations of data.

In a sufficiently technologically advanced and scaled community, computers may be utilized to:

1. Collect, process, and organize information.
2. Produce documentation.
3. Enable rapid amendment of diagrams and other structured information models.
4. Check consistency and completeness.
5. Automate activities that humans have no desire to do.

Systems engineering is a structured process requiring complete documentation and definition of all system requirements. Structured methods have the following characteristics that impact requirements specifications and systems design:

1. Structure a project into small, well-defined activities and objectives.
2. Specify the sequence and interaction of these activities.
3. Use diagrammatic and other modeling techniques.
4. Give a precise (structured) definition to all information concepts.
5. Are understandable by the population.

The general stages of a structured systems engineering process include:

1. Determining feasibility.
2. Investigating the current environment.
3. Determining systems options.
4. Defining requirements.
5. Determining technical system options.
6. Creating the logical design specification.
7. Creating the physical design specification.
8. Each of these stages applies certain techniques and a sequence of analysis. They include conventions and procedures for recording and interpreting the information with the help of diagrams and language.

The components of the structured solution process include:

1. Structures define the frameworks of activities, steps and stages and their inputs and outputs.
2. Techniques define how the activities are performed.
3. Documentation defines how the products of the activities, steps and stages are presented.
4. Inputs and outputs identify the information egress and ingress.
5. Processes define the integrated flow of information in the system. Some processes operate at the systems level and other operate at sub-levels.

The logical system specification:

1. Broad specification from systems analysis.
2. Technical solutions to the requirements are evaluated.
3. Detailed logical (non-technical) design developed which shows clearly how the new system will operate within the total system; how it will integrate.
4. Narrative and system models are used.

Physical design:

1. Logical design converted to a physical (material) one.
2. The arrangement of engineered structures into a blueprint.

Also, system disruptions must be planned for when designing an engineered system. The general process known as 'systems continuity' may also be known as disaster recovery, fail-safe recovery, system redundancy, and service continuity. An intelligently designed habitat service infrastructure would maintain distributed failsafe and the [buffered] redundancy of systems to ensure system continuity in the case of a planned or unplanned incident. Distributed centralization processes minimize the spread of damage in the case of an incident to a service system.

> **APHORISM:** *Resiliency calms economic panic. Coordination calms social panic. Empowerment calms individual panic. Awareness calms egoic panic.*

1.7.2 Defect and flaw improvement

Defect remove efficiency (DRE) can be calculated and can used at both the project and process level:

- DRE = E / (E + D), [E = Error, D = Defect]
- Or, DREi = Ei / (Ei + Ei+1), [for ith activity]
- Optimize by achieving a DREi that approaches 1

Defect removal efficiency

- DRE = E I(E+D)
- Where, E is the number of errors found before delivery of the system to the end user. These are errors because they are before delivery.
- Where, D is the number of defects found after delivery. These are defects because they are after delivery.

Flaw improvement processes generally include:

1. Error - a flaw in an [engineering] work product that is uncovered before the system is delivered to the end-user.
   A. Defect - a flaw that is uncovered after deliver to the end-user.

1.8 [System] Engineering control

Organisms must be able to keep the conditions inside their bodies stable, even when conditions in their surroundings change significantly. For example, human body temperature stays relatively steady despite changes in the environmental air temperature. The maintenance of a stable internal biological conditions is called, 'homeostasis' (also sometimes, and more accurately, known as 'homeodynamics'). Similarly, societies must be able to keep the conditions inside their habitat stable, even when conditions in their surrounding environment change significantly. At the societal level, this [ability to] control comes from specificationing and planning. For example, your access to food stays steady despite changes in the food condition of surrounding nature during winter when food is scarcer in nature. The maintenance of a stable internal economic-access condition is 'econostasis' (also sometimes, and more accurately, known as 'econodynamics'). Econostasis/-dynamics are terms used to describe an access protocol that accounts for a knowable (in this sense, static) and changeable (in this sense, dynamic) environment. All access protocols are engineered, and are forms of control. Together, through openness (Read: open source), humankind can study and develop economic access protocols that facilitate and optimize the condition of complete human need fulfillment.

1.9 [System] Types of real world engineered control

There are [at least] three types of real world (socio-technical) systems, all of which may be engineered:

1. **Social [organismal] systems** - the behavior of organisms.
2. **Technical hardware systems** (a.k.a., material [information] systems) - the behavior of material technology.
3. **Technical software systems** (a.k.a., digital [information] systems) - the behavior of digital technology.

> **HISTORICAL NOTE:** *The concept of software engineering emerged with the development of computing and information sciences (in its modern meaning) around the 1960s.*

Technology is a result of engineering - the extending of human mind-body function. Technology is the result of applied [scientific] knowledge and engineering processes. Technology refers to the technical systems that engineering designs, builds, and operates.

Technology is:

1. The useful (practical) application of knowledge.
   A. In this sense, technology is engineering operations; it describes the product of engineering.
2. Technology is a capability/function provided by the useful (practical) application of knowledge.
3. In this sense, technology is engineering design and development; it describes the engineering process itself.

1.10 [System] Engineering development levels

The development of engineered systems takes time and work. One framing of the engineering process delineates that which is being developed to be used into levels of developmental usefulness. The following categories classify technology (Read: technology products) thusly.

1.10.1 Technology readiness levels (TRL)

A.k.a., Technology maturity modeling, technology development levels.

In engineering there are technology readiness levels. Generally, there are nine [levels] of them. Technology Readiness Levels (TRL) are a type of measurement system used to assess the maturity level of a particular technology. Here, maturity is a synonym for development or readiness. Each technology (as a project) is evaluated against the parameters for each technology level, and is then assigned a TRL rating based on the projects progress. In general, a technology project is only moved to the next readiness level when the relevant environmental validation is complete for that level.

1.10.2 Level of development (LOD)

A.k.a., Level of detail, level of information detail (LID), amount of information.

Level of development (LOD) is a measure of the level of development by an object (system or element). It is not necessarily a measure of the amount of information, although there must be enough information to satisfy the LOD level of the object itself. LOD is also not a measure of the accuracy of information. Generally, the term LOD is used to refer to elements (i.e., sub-parts) of any single technological system, which itself will have a technological readiness level (TRL). LOD is a measure of "progress", which each level containing a set of parameters.

NOTE: *"Objects" only exist as information (Read: exist as concepts), before they come into being [real] 'objects' through engineering, at the expense of energy and area (Read: light and matter).*

A common level of development (LOD) scale is (the object, 'chair' as the family-type, is incorporated below as an example):

1. **LOD 100** - there is an object (Read: something that has shape and can be pointed to; a thing, product, system), a 'chair'.
2. **LOD 200** - there is a product (object) of specific dimensions -- a chair that has nominal space requirements of 400x400 units.
3. **LOD 300** - there is an object with stated functions and options -- a chair with arm rests and wheels.
4. **LOD 400** - there is an object that is numerically identifiable among other types of its object, and there is a process for producing that specific sub-type of object -- there is a model number for the chair, and a production process for that specific chair.
5. **LOD 500** - there is an object number, a production process for that specific object, and a decision to produce one (or more) of that specific object -- there is the chair's model number, the production process for the chair, and an ordered demand to produce one (or more) of that chair.

There may also be sub-levels:

1. **LOD 200** - final object specification defined.
2. LOD 290 - preliminary construction defined.
3. LOD 292 - checked for functional requirements in construction.
4. LOD 294 - checked for justice-value requirements in construction.
5. LOD 296 - checked for freedom-value requirements in construction.
6. LOD 298 - checked for efficiency-value requirements in construction.
7. **LOD 300** - final construction specification defined.

There are multiple ways of visually representing LOD information. For example (i.e., one way), an LOD numerical identifier structure, such as, "XXXX" where each digit "X" corresponds to a piece of information in the table (e.g. Description, or Width, or Height, etc...) and each of these digits would take a value between 0 and 5 (or 0 and 9 if one needs more granularity). The result would be (taking the 0 to 4 scaling):

1. "0000" means 0% information (with 0% certainty).
2. "1111" means 100% information but with low certainty/development.
3. "5550", "0005" or "5050" all mean 50% information with 25% overall certainty, BUT with clear distinction on the pieces of information that are known and to what level.
4. "9999" means 100% information with 100% certainty/development.

1.10.2.1 Level of development (LOD) sub-categorization

In concern to objects and technology, the idea of a "level of development" can be de-composed into two indices, which together represent a selectable solution:

1. **Level of Information Detail (LID)** - what level of information is present to [have the ability to] materialize the object?

2. **Degree of Certainty (DoC, Level of Certainty, LoC)** - how certain is the execution upon the information to produce the expected result? In other words, how certain are "you" that upon execution of the information the result will be as expected (predicted/specified)?

1.10.2.2 Level of uncertainty

The concept of a 'level of uncertainty' may be generally sub-divided into:

- **Level of Incompletness (LoI)** - a measure of incompleteness.
- **Level of Availability (LoA)** - a measure of what and how much information is available.

1.10.3 Level of design (LOD)

A.k.a., Level of detail (LOD).

There are several commonly identified levels of design:

1. Semantic description of system concepts (a.k.a., paper-based product concept) - these are sketches, narratives (user cases), annotated drawings, graphics, or other concept descriptions that can enable initial explorations of ideas on system functionality to be made, important usability characteristics to be identified, or walk-through studies of protocols.
2. Part prototypes or simulations - Part prototypes are used to simulate specific functional attributes of a design. They might be mock-ups of physical form, scale or mass, mechanical models, static, or animated graphics that enable people to interact with them. The prototype may look nothing like the final design, but will accurately represent those aspects under investigation.
3. Experience prototype - these are representations in any medium that help people to appreciate experiential issues beyond the purely functional attributes of a design. They are designed to include contextual and affective qualities conveyed through a relevant subjective experience.
4. Full prototypes - Full prototypes perform as the final product is intended to perform an incorporate the complete functionality and appearance of the product.
5. Complete product - complete products enable the complete user-interface to be examined. This opens the possibility of carrying out field investigations, comparative studies with other products, in-service studies, etc.

1.10.4 Level of accuracy (LOA)

Level of accuracy refers to the level of accuracy that must be achieved between interoperating models; for instance, when models are created based on a laser scan, what is the level of accuracy that the deliverable model must achieve? For instance, if a beam is (to be) warped in reality, what is the level of accuracy the model needs to achieve, can it just look like a normal beam, does it need to be need to be warped, with what precision does it need to matcb the real world object?

- **Measurement of accuracy (MOA)** - how accurate is the scan data that is being started with? This relates, in part, to the measuring tool (for example, measuring tape is less accurate than a laser).

1.10.5 Social readiness level

Just as technologies have a development readiness level, so do social [mental] models and methodologies. Today, humanity now has access to the systems methodology, and a unified, systems-based (real world) social model for iteratively integrated socio-technical design, construction and operation [of society].

In order to understand and operate complex real world systems, their methodologies (Read: the selected methods that structure the formation of complex systems) must be understood. When a population starts to view society as information, then data and processes start to structure the formation of real world systems, which may be viewed as they are, unified. Socio-individual viewpoint could be considered a new "level" of self-awareness - individuals have access to a unified information system that is pre-configured with data and processes, which are accurately alignable and intentionally programmable to complete in the iterative formation of a material hard-/soft-ware [information] system that fulfills all individual human need, which are never fully known (i.e., there is always more to know). System modeling now exists to assist us in visualizing together so that we can understand and perceive impacts of models, decisions, and actions in our common environment. In this environment, an information environment, all data is fit into a structure (e.g., data model, database) upon which processes may operate. The operation of processes on data requires a control structure to coordinate and control all data and processes. This control structure is "like" a platform, operating system, decision system, protocol, algorithm, ect. (named differently depending upon what level or scale the [whole] society is being viewed from). That control structure can be openly designed and programmed by contributing individuals (you become the ultimate relationship management site, because their reputation on their is ultimate that they would contribute freely, so greatly, which doesn't mean you can't have a secondary pay operation also, it is to say that there are multiple valuable databases here, free though, so no good, well you as source of information as value) or it can be programmed in secret.

1.10.6 BIM readiness levels

BIM readiness level (as model cooperation visualization levels) can be generally separated into:

1. Level 1 BIM is CAD separated files.
2. Level 2 BIM is 2D-3D CAD separated files.
3. Level 3 BIM is CAD with unified file directory revisioning.
4. Level 4 BIM is CAD with life-cycle integration through a unified file directory.
5. Level 5 BIM is data and process simulation, and unified directory file revisioning.
6. Level 6 BIM is societal level development and operations unification.

At the 5-6 BIM levels, the highest level societal services are: "architecture", "structure" (infrastructure), and "MEP" (maintenance, engineering, and planning) may become one integrated systems team sharing a common set of data and process, for example, as separate government and industrial entities, or local habitat sub-services entities (the later is strange to say, because it presupposes a unified system, the habitat).

One would likely rather have a proactive asset and building coordination and control system at all scales of society (like Community), rather than, a reactive one (like the cities and sprawls that early 21st century humans live in).

Possibly, when BIM is referred to in its level 5 context, by industry and government, they are in fact referring to planning (e.g., "public private ownership", etc.) at that level, in definition, as the merger of industry and government as an organization that coordinates the construction of all buildings through the control of design, construction and operation, of the information systems that produce and operate all building-related data and processes.

1.11 [System] Architectural clarifications

Architecture (noun) is defined commonly in several different ways:

1. The art and science of designing and superintending the erection of buildings and similar structures.
   A. The creativity, heuristics and engineering practice of design and technical supervision resulting in man-made systems.
2. A style of building or structure
   A. A recognisable pattern or pro forma of system composition and arrangement.
3. Buildings or structures collectively.
   A. A quality or attribute of systems that conveys composition and order.
4. The structure or design of anything.
   A. The composition and rational arrangement of a system.
5. The internal organization of a computer's components with particular reference to the way in which data is transmitted.
   A. The information technology viewpoint of a computer described according to system form and function.
6. The arrangement of the various devices in a complete computer system or network.
   B. The information technology viewpoint of a computer network as a system}' [Collins, 1991].

In concern to the semantics of these definitions:

1. Meaning 1 confirms architecture to be a body of practice. It is applied to the design and supervision of actions of particular classes of structure, such as vessels, buildings, cities.
2. Meaning 2 conveys that architecture can manifest itself as patterns of significance and value.
3. Meanings 3 and 4 convey architecture to be a collective attribute of systems.
4. In contrast, Meanings 5 and 6 present a contemporary information technology and software use of the term for computers (plus the software representations of data, processes and control that they host) when considered in system terms.

For systems engineering and, as the definitions above suggest, generally for the systems reasoning mind, it is axiomatic that architecture is an attribute of system that characterises a system's order. In the IPTL survey 67% spontaneously identified architecture with structure, with 50% referring to product structure and 17% translating this directly into consequent project structure.

Architecture is thus commonly understood as a description of the composition and structuring of a man-made system; of order that arises from intent and directed design. This is in conformity with the IEEE definition of 1990: 'the organizational structure of a system or component'. That is, a factual listing of parts and their organisation or relationship [IEEE 1990].

A decade later, however, the influential standard IEEE STD 1471 had evolved this definition into 'the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment and the principles guiding its design and evolution' [IEEE 2000]. This definition had moved beyond an objective description of a system-of-interest, extending it to include the setting, if not behaviour, in an environment of operation. It also introduced the notion of the decision (or design) rationale behind these descriptions. In doing so, it began to equate architecture to design actions and the discipline that governs them.

In ISO/IEC 15288 architecture is explicitly associated with one process: architecture design.

Architecting is an invented word to describe how architectures are created, similar to how engineering describes how "engines" and other technologies are created. Possibly, if engineering is the art and science of technical problem solving, then systems architecting occurs when the problem is not yet known. (Maier, 2009)

Yet, a systems engineering and systems architecting distinction would appear to arise partially from values, beliefs and ideas, and hence to be culturally rooted. Etymologically, the word architecture comes from the Greek word "arkhitektonike", which is a combination of two words meaning 'chief' and 'builder'. Thus, the word architect derives from the Greek for "the director of works" or "chief builder" and refers to someone who is responsible for overseeing all aspects of building, and is essentially the integrator of all aspects. Hence, an architect is associated with technical leadership and connotes seniority as much as skill. Typically, it is used in the singular form and is less prominently associated with a team activity. Architecting practitioners have thus elevated the most strategic-thinking, high-level design to be architecting, relegating all else to be termed 'design' (or, engineering), which is then a subordinate/ subsequent action to that of architecting.

According to this model of architecting practice, systems engineering is concerned with the conduct of implementation-related design, and architecting with the strategic decision making across all engineering contributions. Architecting then becomes the hub of design. It is a model with seductive promise to those mired in academic complexity, but is in essence barely more than a re-titling or re-stratification of the recursive transformations described by systems engineering.

For purposes of conceptual clarification ("conceptual cleansing"), architecture may be seen as the descriptive essence of systems, and in no sense is it the system itself. Architecture could be viewed as the totality of every possible communicable view of an actual or conjectured real-world system: the summation of all possible transmissible models that inform the existence of a system as an object. It is therefore an abstract notion; a set of descriptions of the nature, arrangement, workings, holistic interaction opportunities, and additionally as preferred, the rationale for the existence of this order.

2 [Engineering] Life-cycle stages

A.k.a., The [systems] engineering life cycle, or the systems engineering method.

Each phase of the systems engineering life-cycle (process) has a similar logic[al set of objects]:

1. Definition.
2. Purpose.
3. Task(s) and activities.
4. Outcomes.

Note that the term 'requirement' is essentially the same as the term 'specification'. 'Requirements' must be sufficiently specific and detailed to allow/ensure **verification** (is the right, correct, planned "thing" being done) and **validation** (user approval).

The engineering processes of service life-cycle coordination are:

1. De-/construction (i.e., dis-/assembly, de-/equipping, etc.).
2. Maintenance (i.e., tasks that maintain a service function).
3. Operations (i.e., tasks that use a service function).
4. Monitoring (i.e., remaining aware in order to apply a control if necessary).

On-service engineering operations are systems and humans that are acting in some capacity through some task as being of service to another human or system. The incident response process, for instance, involves both data and physical incidents. When the incident response service is engaged, humans and systems become on-service to the procedural aid of other humans and systems. Maintenance is a sub-set of the life-cycle processes; it involves being of service to humans by developing and sustaining the systems that facilitate everyone's fulfillment (i.e., "service them").

The following four primary engineering process life-cycle phases:

1. Composition.
2. Maintenance *of composition.*
3. Operation *of composition.*
4. Decomposition.

2.1 Requirements of engaging in systems engineering

The primary deliverables of these systems engineering processes are/include:

1. **Requirements** engineering of the requirements specification.

2. System architecting a **logical systems architecture**.
3. **System design specification** (and standardization)
4. Integration of specification (standardization) into **habitat/information operations**.
5. **Validation and verification** of physical/ information system itself is changed as expected.

The systems engineering process requires:

1. **Access** to all available knowledge (and information).
2. **Defining** user needs and required functionality.
3. **Documenting**.
4. **Design** synthesis.
5. System **validation**.
6. While considering the complete **problem**: operations, resources and schedule, performance, support, test, manufacturing, and disposal.

Engineering sub-units include (by task category):

1. **Scientific research.**
2. **Systems design and development.**
3. **Systems integration.**
4. **Systems operation.**
5. **System update and/or de-integration.**

The engineering process chain is initialized as a dynamic problem:

1. **Measure.**
2. **Identify.**
3. **Analyze.**
4. **Design.**
5. **Execute (Act).**

The engineering process chain works to materialize a solution:

1. **Design of system.**
2. **Production of components**.
3. **Assembly of system.**
4. **Testing of system.**

The modeling process for engineering a real-world system requires:

1. **Design** - integrate the concepts, principles, data, and knowledge into a structure with a logical flow.
2. **Design development** = integrate the structure into the logical flow of a specified system.
3. **Production** - apply energy through a vehicle to [effectively and efficiently] modify material or digital information into the specified system. For example, use a knife to whittle wood into a "carved" implement for eating, like a spoon or chopstick.
4. **Service integration** - materially or digitally connect the sub-system to a larger/pre-existing system.
5. **Service operation** - operate/use the system.
6. **Service testing** - of the design occurs throughout the whole process to ensure the solution is as expected by the user (i.e., meets requirements).

The modeling process for engineering a proposed societal system requires:

1. Create - Vision for society.
2. Evaluate - Individual human needs.
3. Analyze - Collect data and analyze situation.
4. Apply - Apply decided procedures.
5. Understand - Visualize results of action.
6. Update - Integrate results of action
7. Remember - Re-envision society.

2.2 The product life-cycle stages

All productions, whether they are objects or services (combinations of objects) go through the following engineered product life-cycle stages (input-outputs of a production/manufacturing/engineering system):

1. Product design (final service and/or object design).
2. Manufacturing system design (what to produce in order to produce the designed product; intermediary designs).
3. Manufacturing .
    A. Production of the manufacturing system.
    B. Production of the product from the manufacturing system.
4. Distribution and storage.
5. Product use (habitat service system operation and user access, together; intermediary and final demand).
6. Disassembly, reuse, re-manufacturing, and recycling.

2.3 The engineering life-cycle/process flows

There are multiple possible views into engineering as a system of processes. The engineering process can be viewed from multiple, correct perspectives. There is commonality between all of the possible perspectives on engineering. Therein, different engineering projects may modify the [unified information system's] common engineering process(es) accordingly.

These activities cover the "cradle-to-grave" or "cradle-to-cradle" life cycle process associated with the major functional groups that engineering provides. The following process views are in their simplified conceptual form.

2.3.1 Technical process flow views

The following are the common technical processes (technical process flows) for the realization of a solution through engineering. Note that these view all follow essentially the space system's process; they just use different terminology to describe the same process (i.e., the same structural flow of information).

The problem-solving view of the engineering process:

1. **Problem input** - initial requirements data.
2. **Analyze requirements data** - obtain answers to requirement questions.
3. **Design solution** - obtain answers to requirement questions.
4. **Test and validate solution** - produce and evaluate the design against the requirements.

The development and operations view of the engineering process:

1. **Analysis** - identify design problem.
2. **Synthesis** - identify design alternatives.
3. **Prototype** - build and test alternatives.
4. **Integrate** - integrate the best selection.
5. **Utilize** - Operate the new system.

The problem-oriented cycle view of the engineering process:

1. Have **problem**?
2. Collect **data.**
3. Design **solution**.
4. Solution **test.**
5. Solution **feedback.**
6. **Integrate** solution.
7. Have **problem**?

The engineering phases view of the material system life-cycle:

1. Conceptual phase
2. Specification and Design phase.
3. Implementation phase.
4. Operations phase.
5. Retirement phase.

The development review completion cycle view of systems engineering:

1. **System Requirements Review (SRR):** At the beginning of the project, establishes what the system will and will not do.
2. **Preliminary Design Review (PDR):** At 10% design completion, is primarily to critique the architecture of the design and critical decisions made in the design.
3. **Critical Design Review (CDR):** At 90% design completion, is primarily to make a last set of changes before the design is finalized.
4. **Validation System Review (VSR)**: At 100% operational completion the system.
5. **Verification System Review (VSR):** User feedback on issue.

The inquired action view of the engineering process:

1. **Inquire** (is a change needed; is a decision present)
2. **Problem situation** (situational analysis; requirements).
3. **Solution formulation** of relevant purposeful models and activities [accordingly, scenarios] of the perceived problem (functional and physical design).
4. **Take action [to realize formulation, reformulation of situation]** in the situation to bring about improvement (implementation, material change/construction).

The serviced view of the engineering process:

1. **Conceive** (Imagine, specify, plan).
2. **Design** (describe, define, develop, test, analyze, validate).
3. **Realize** (manufacture, make, build, procure, produce, deliver, phase-in).
4. **Service** (use, operate, maintain, support, sustain, phase-out, retire, recycle, dispose).

The actionable phase view of the engineering process:

1. **Initiation phase** - recognition of problem.
2. **Analysis phase** - understanding of problem and context.
3. **Design/synthesis phase** - specification of solution to problem.
4. **Implementation phase** - solution production, testing, training, site preparation.
5. **Operations phase** - usage of solution.
6. **Evaluation phase** - observe and review the solution and the process that created the solution.

The problem-action view of the engineering process:

1. **Problem identification** - defining.
2. **Solution abstraction** - modeling.
3. **Solution realization** - building.
4. **Solution utilization** - operating.

The problem view of the engineering process:

1. **Problem detection.**
2. **Problem definition.**
3. **Problem analysis.**
4. **System design problem.**

5. **System manufacturing problem.**
6. **System use/service problem.**
7. **System obsolescence problem.**

Then system state view of the engineering process:

1. **Problem** with world.
2. **Model current state** of world.
3. **Model new state** of world without problem.
4. **Construct new state** of world.
5. **Evaluate new state** of world.

The issue view of the engineering process:

1. **The issue problem.**
2. **The research and discovery problem.**
3. **The design problem.**
4. **The construction and integration problem.**
5. **The operation problem.**
6. **The testing and evaluation problem.**
7. **The maintenance problem.**
8. **The de-integration problem.**

The resource-based view of the engineering process:

1. **Survey** (an environment for planning).
2. **Plan** (a system for building).
3. **Build** (a system for operating).
4. **Operate** (a system for serving).
5. **Cycle** (the evolution of the operating systems).

The strategic-evaluative view of the engineering process:

1. **Planning and analysis.**
   A. Create project concept.
   B. Generate requirements.
   C. Validation.
2. **System [logical] architecting.**
   A. Functional analysis.
   B. Requirements analysis.
   C. System synthesis.
   D. Validation.
   E. Verification.
3. **System [physical] design.**
   A. Physical design.
   B. Composition analysis.
   C. Validation.
   D. Verification.
4. **Build* and test [the system itself].**
   A. System integrations.
   B. Validation.
   C. Verification.

The algorithmic life-cycle view of the engineering process:

1. **Plan algorithmic decisioning.**
2. **Design select algorithm.**
3. **Implement algorithm.**
4. **Assess algorithm.**
5. **Monitor algorithm.**
6. **Iterate algorithm.**

The design alignment view of the engineering process:

1. **Requirements.**
2. **Analysis.**
3. **Development.**
4. **Testing.**
5. **Implementation.**
6. **Support.**

The creation alignment view of the engineering process:

1. **Direction** - put together a specification of the objective.
2. **Conceptualization** - put together a specification of the system. Conceptualization involves the organizing and structuring of acquired knowledge.
3. **Implementation** - implement the concept [specification] model to create and/or operate the system.
4. **Evaluation** - evaluate by doing a technical analysis on the process and result, and correct any mis-alignment with objectives and requirements (system so that all information in all phases is more coherent and/or useful).

The system design view of the engineering process:

1. **Discover** - Why is “it” the “right” output.
   A. Research.
2. **Define** - What is the “right” output.
   A. Ideate.
3. **Design** - Design what is the “right” output.
   A. Specify.
4. **Develop -** Prototype and test the “right” output.
   A. Build and test.
5. **Deliver** - Deliver, integrate and transport, what is the “right” output.
   A. Implement and integrate.

The system integration view of the engineering process:

1. **Requirements.**
2. **Design.**
3. **Implementation, integration, transition, launch.**
4. **Verification.**
5. **Operation.**
6. **Validation.**

The system generation view of the engineering process:

1. **Conception (concept).**

2. **Development assessment.**
3. **Development demonstration.**
4. **Production manufacturing.**
5. **System transition.**
6. **Utilization (in-service operations).**
7. **Retirement (disposal operations).**

The system information view of the engineering process:

1. **Conception.**
2. **Initiation.**
3. **Analysis.**
4. **Design.**
5. **Construction.**
6. **Testing.**
7. **Deployment and release.**
8. **Operation.**
9. **Iterate and Evolve.**

The vision improvement view of the engineering process:

1. **Measure.**
2. **Analyze.**
3. **Improve.**
4. **Sustain.**

The vision to operation view of the engineering process:

1. **Vision.**
2. **Design.**
3. **Transition.**
4. **Operation.**

The system integration view of the engineering process:

1. **Need analysis.**
2. **Situation and concept exploration.**
3. **Concept definition.**
4. **Design and development.**
5. **Integration.**
6. **Operation.**
7. **Evaluation.**

The solution cycle view of the engineering process:

1. **Issue** or **change concept** (for solution).
2. **Development** (of solution).
3. **Integration** (of solution).
4. **Sustainment** (or solution).

The planning view of the engineering process:

1. **Plan.**
2. **Develop.**
3. **Test.**
4. **Deploy.**
5. **Operate.**
6. **Support.**

The project engineering view of the engineering process:

1. **Project definition.**
2. **Specification definition.**
3. **Conceptual design.**
4. **Product design.**
5. **Fabrication (manufacturing).**
6. **Assembly.**
7. **Integration.**
8. **Testing.**
9. **Evaluation.**
10. **Operation.**
11. **Iteration.**

The object (ware) view of the engineering process:

1. **Problem environment.**
2. **Design solution concept.**
3. **Design solution ware** (hardware and/or software).
4. **Construct solution ware.**
5. **Operate solution ware.**

The project transition view of the engineering process:

1. **Design.**
   A. Design assessment (design integration) - define operations and maintenance (O&M) requirements to understand the 'end state' and successful progression into efficient and effective operations.
   B. Design development - develop the [informational] design specification.
2. **Implementation/construction.**
   A. Production - produce and deliver the system to specification.
3. **Commissioning (validate and test).**
   A. Operational readiness - implement, test and validate the operations and maintenance (O&M) activities to ensure a smooth and safe transition into an operational capability. Ensure the system is at a sufficient [checklisted] state if operational readiness.
4. **Operations.**
   A. Operational evaluation - enhance, evaluate, and embed operations and maintenance practices to ensure overall efficiency and effectiveness.
5. **Operational lifespan.**

The structural-informational view of engineering:

1. **Problem (with environment)** - system design process.
   A. Requirements definition process (of system).
      1. User/stakeholder expectations definition.
      2. Technical requirements definition.
   B. Technical solution definition process (of system)

1. Logical decomposition.
2. Design solution definition.
2. **Solution (for environment)** - system realization process.
    1. Design realization process (of system).
    2. Integration process (of system into environment).
    3. Evaluation process (of system operating in environment).
3. **Planning (of environment).**
    1. Technical planning process.
    2. Technical control process.
    3. Technical assessment process.
    4. Technical decision analysis process.

The system materialization view of the engineering process (Read: Levels of materialization):

1. **Concept refinement phase** - refine the initial problem/issue/concept/situation into a direction, approach, and orientation [to the state of the environmental societal system, as the solution]. Conceive of why the system needs to be changed and what changes are required.
2. **System development** - Develop a new system, sub-system, or capability (object or service) aligned with the direction, orientation, and approach. Develop the new system state to align with the refined conception.
3. **System deployment** - Achieve a transitional operation of the actual material system that satisfies the refined conception of a direction, orientation and approach (as given in the concept refinement phase).
4. **System operation** - Execute a support program that meets operational support performance requirements and sustains the system over the time of its life-cycle.

The service life-cycle view of engineering (New service life-cycle phases):

1. **Service need.**
    A. Concept studies.
2. **Concept definition.**
    A. Concept and technology development.
3. **Design specification.**
    A. Preliminary design, engineering model (final design), and technology completion.
4. **Production (fabrication).**
    A. Assembly, integration, and testing.
5. **Operation.**
    B. Operations and sustainment.

The situational systems view of the engineering process:

1. **Analyse situation.**
2. **Develop requirements** for system.
3. **Design system** based on requirements.
4. **Build system** based on design.
5. **Use and maintain system** based on design.
6. **Re-cycle system** based on design.

The constructional view of the engineering process:

1. **Informational** (conceptual, object-process).
2. **Virtual** (simulation).
3. **Live** (actualized, material, physical).

The measurement view of the engineering process:

1. **Do a cause-and-effect analysis** - to understand the current situation.
2. **Identify objectives** - to set the purpose for changing the current situation.
3. **Identify requirements** - to set the precise structural outcome(s).
4. **Quantify** - to specify the precise outcome.
5. **Measure the build** - build the precise outcome.
6. **Measure the result** - determine if the build meets the specifically defined quantifications.
7. **Repeat.**

The engineering design view of the engineering process:

1. **Concept studies.**
2. **Concept development.**
3. **Preliminary design.**
4. **Detailed and final design.**
5. **FAIT or SAITL** (FAIT - fabrication, assembly, integration, transition; SAITL - system assembly, integration, testing, launch).
6. **Verification.**
7. **Operation.**
8. **Validation.**

The product plan view of the engineering process:

1. **Concept design.**
2. **Product development.**
3. **Product production.**
4. **Product utilization.**
5. **Product support.**

The coordination view of the engineering process:

1. **Planning:**
    A. Site survey.
    B. Resource survey.
    C. Feasibility analysis/study.
2. **Engineering:**
    A. Process design.
    B. System design.

   C. Sub-system designs.
3. **Procurement:**
   A. Acquisition.
   B. Logistics.
   C. Inspection.
4. **Construction:**
   A. Construction planning.
   B. Schedule control.
   C. Construction tasks.
   D. Construction validation.
5. **Service:**
   E. Operations (and maintenance).

The development view of the engineering process:

1. **Research and discovery** (problem inquiry and situation analysis) - identify a problem/issue for which a solution is to be designed.
   A. Identify the problem.
   B. Document and analysis of problem, situation, and prior solution attempts.
   C. Determine solution requirements.
   D. Root cause analysis (process) - similar to that used in solving quality-related problems, can be used to categorize risks according to their source, to list risks in each category, and then to propose preventive actions to prevent these risks, or to develop countermeasures or risk responses if they happen to occur. It can be used as part of brainstorming, the first technique listed, to identify risks.
2. **Design** - develop multiple solution possibilities and through the use of feedback and data, select the best potential solution to pursue.
   A. Generate design concept, analysis, selection.
   B. Application of STEM principles and practices.
   C. Determine design viability.
3. **Prototype and test** - create a testable prototype and unbiased testing plan based on the defined design requirements to determine the effectiveness of the solution created.
   A. Construction of a testable prototype.
   B. Prototype testing and data collection plan.
   C. Testing, data collection, and analysis.
4. **Evaluation of project and process** - seek and document feedback.

The service system existence activities view of the engineering process:

1. **Development (design and testing)** - the activities required to create/evolve the system from user needs to product or process solutions.
2. **Production and construction (create final solution)** - the activities necessary to create the completed solution.
3. **Deployment (fielding of final solution)** - activities necessary to initially deliver, transport, receive, process, assemble, install, checkout, train, operate, house, store, or field the system to achieve full operational capability.
4. **Operation (of final solution)** - the user function and includes activities necessary to satisfy defined operational objectives and tasks in peacetime and wartime environments.
5. **Support (of operational solution)** - the activities necessary to provide operations support, maintenance, logistics, and material management.
6. **Disposal/evolution (of operational solution)** - the activities necessary to ensure that the disposal of decommissioned, destroyed, or irreparable system components meets all applicable regulations and directives.
7. **Training (on operational solution and learnings)** - the activities necessary to achieve and maintain the knowledge and skill levels necessary to efficiently and effectively perform operations and support functions.
8. **Verification (of operational solution)** - the activities necessary to evaluate progress and effectiveness of evolving system products and processes, and to measure specification compliance.

The technical system design realization view of the engineering process:

1. **Systems design processes:**
   A. User expectations defined (imperatives/objectives).
   B. Technical requirements definition (requirements definition process).
   C. Logical system decomposition.
   D. Design solution definition (solution definition process).
2. **System realization processes:**
   A. System implementation process.
   B. System integration process.
   C. System validation process.
   D. Requirements validation process (technical evaluation process).
   E. System verification process (technical evaluation process).
   F. System transition (transition to user process).
   G. System maintenance process.
   H. System disposal process.
3. **Technical coordination processes:**
   A. Planning coordination process.
   B. Imperatives coordination process.
   C. Requirements coordination process.

D. Resource and tool coordination process.
E. Assessment process.
F. Control process.
G. Risk coordination process.
H. Data coordination process.
I. Interface coordination process.
J. Decision analysis process.

2.3.1 The basic process view of engineering

All engineering project follow essentially the same structure:

1. **Goal** - purpose for the project's [working] existence.
2. **Functional requirement (a.k.a., requirements encoding mechanistic/causative functioning, technical requirements, functional objectives)** - addresses one specific aspect or required performance of a system to achieve the stated goal (note that other functional requirements may contribute to achieve the same goal).
3. **Non-functional requirement (a.k.a., non-functional objectives, requirements encoded values, orienting principles)** - addresses one specific property (characteristic/objective) that is to be achieved by executing the project (note that other objectives may contribute to achieve the same goal).
4. **Operative requirement (a.k.a., performance requirement)** - actual requirements, in terms of performance criteria or expanded functional description.
5. **Verification requirement** - instructions and tools for verification of performance.

The basic process view of engineering involves an information loop:

1. **Requirements analysis** - Requirements analysis is used to develop functional and performance requirements; that is, customer requirements are translated into a set of requirements that define what the system must do and how well it must perform. The systems engineer must ensure that the requirements are understandable, unambiguous, comprehensive, complete, and concise. Requirements analysis must clarify and define functional requirements and design constraints. Functional requirements define quantity (how many), quality (how good), coverage (how far), time lines (when and how long), and availability (how often). Design constraints define those factors that limit design flexibility, such as: environmental conditions or limits; defense against internal or external threats; and contract, customer or regulatory standards.
2. **Functional analysis and allocation** - Functions are analyzed by decomposing higher-level functions identified through requirements analysis into lower-level functions. The performance requirements associated with the higher level are allocated to lower functions. The result is a description of the product or item in terms of what it does logically and in terms of the performance required. This description is often called the functional architecture of the product or item. Functional analysis and allocation allows for a better understanding of what the system has to do, in what ways it can do it, and to some extent, the priorities and conflicts associated with lower-level functions. It provides information essential to optimizing physical solutions. Key tools in functional analysis and allocation are Functional Flow Block Diagrams, Time Line Analysis, and the Requirements Allocation Sheet.
    A. Here, it is important to consider under what conditions an existent function may not be wanted by a user, and hence, should be dis-ableable (Read: disableable by the user).
3. **Requirements loop** - Performance of the functional analysis and allocation results in a better understanding of the requirements and should prompt reconsideration of the requirements analysis. Each function identified should be traceable back to a requirement. This iterative process of revisiting requirements analysis as a result of functional analysis and allocation is referred to as the requirements loop.
4. **Design synthesis** - Design synthesis is the process of defining the product or item in terms of the physical and software elements which together make up and define the item. The result is often referred to as the physical architecture. Each part must meet at least one functional requirement, and any part may support many functions. The physical architecture is the basic structure for generating the specifications and baselines.
    A. Design deliverable (noun) - A design [specification] is a visualization (sometimes, plan) that shows (through to demonstrates via simulation) some combination of function ("workings"/mechanism), performance, and interface of future system.
    B. Design process (verb) - To design means the decisioning processes (groups) that model, determine, and select the function, performance, and interface to be recorded as the executable design, a valid design for integration).
5. **Design loop** - Similar to the requirements

loop described above, the design loop is the process of revisiting the functional architecture to verify that the physical design synthesized can perform the required functions at required levels of performance. The design loop permits reconsideration of how the system will perform its mission, and this helps optimize the synthesized design.

6. **Verification** - For each application of the system engineering process, the solution will be compared to the requirements. This part of the process is called the verification loop, or more commonly, Verification. Each requirement at each level of development must be verifiable. Baseline documentation developed during the systems engineering process must establish the method of verification for each requirement. Appropriate methods of verification include examination, demonstration, analysis (including modeling and simulation), and testing. Formal test and evaluation (both developmental and operational) are important contributors to the verification of systems.
    - Inspection is one method of verification.

2.3.2 The project engineering process

A descriptive view of project engineering includes:

1. Engineering life-cycle:
    A. Engineering.
    B. Pre-concept.
    C. Concept.
    D. Prototype.
    E. Evaluate.
    F. Produce.
    G. Operate.
    H. Maintain.
    I. Cycle.
2. Engineering process stages:
    A. User need definition.
    B. System requirements definition.
    C. Detailed system design.
    D. Prototype, test and acceptance.
    E. In-service feedback.
3. Engineering design and development process:
    A. Needs identification.
    B. Literature/background study.
    C. Task requirements and specifications.
    D. Definition of the goal/purpose of the design.
    E. Ideation and invention.
    F. Analysis.
    G. Selection.
    H. Detailed design.
    I. Prototyping and testing (including validation, certification and standardization as applicable).

2.3.3 The basic concept view of engineering

A descriptive view of engineering includes:

1. **Conceptual design** - the formal transition from the user-issue organization level to the engineering level. In other words, a decision space has now opened an engineering solution space, which the first deliverable of which includes a set of engineering requirements that align with the decision space's resolution objective(s). Traceability from the user-issue with a complete logical description of the system-of-interest into measurement statements (i.e., requirements) for designing and operating a user system without issues. This deliverable set ensures the proper definition/identification/development of the system requirements. This phase has two primary functions: (1) more likely that a solution will optimally resolve a problem; (2) more likely that effectiveness inquiry (a core decisioning process) will return an accurate result, such that HSS operational process will operative effectively due to accurate data, and unsafe projects will be correctly identified and removed from active decisioning, placing them into issue holding.
    A. Concept stage - encompasses all analysis and planning to establish the valid need for a new system. Why does the user need the new system? Establish possibility/feasibility of an architecture (system) that is realizable (based on society's value set alignment.
        1. Valid need - establish that there is a valid need, that the system will be used (in the market, market feasibility - someone will buy the system)
        2. System concepts - exploring potential system concepts/formulations along with valid sets of system performance requirements.
        3. Selection - selection of the most optimal (best fit) system concept (matching). Define the functional characteristics of the optimal (best fit) system concept so that the selected system concept definition can be used to make engineering, productions, and operations plans.
        4. New technology development - certain times, the newly envisioned system will require the development of new technology (because of non-existent technology) - hence, develop necessary technology and technology needed for the system concept, and validate the technology.

B. In engineering, the need associated with a critical gap constitutes the start of the systems engineering lifecycle and the initiation of a conceptual design solution. As mentioned earlier in the process lifecycle, conceptual design includes:
   1. Define organizational needs and requirements
   2. Define stakeholder/user needs and requirements
   3. Define system requirements
   4. Conduct system-level synthesis - this will allow for the selection of the optimal ("preferred") system-level solution (configuration).
   5. Conduct system design review (evaluation)
   6. The output of 5 then becomes the Preliminary Design
2. **Requirements activities** - The 3 types of requirements that form [part of] the system's logical design (requirements flowdown):
   A. **Societal-level organizational needs and requirements** - the value system [engineering] requirements for materialization (their alignment):
      1. In the societal information system, this is represented by: the parallel decision inquiry processes.
      2. The organizational requirements (parallel value-alignment decision inquiry process) ensure feasibility of the solution (i.e., that the solution is a feasible undertaking and integration by the societal system). These requirements control (and guide) the development of engineering requirements for the system. The likely options to a problem (i.e., the possible solution spaces). What society requires from the ultimate solution when it is deployed.
      3. Societal-level requirements activities are also known as the societal requirements specification (or business requirements specification including organizational/ business needs and requirements).
      4. Imperatives and directives such as mission, vision, goals, objectives, needs, etc.
      5. In human terms, tasks (execution) herein are completed by members of InterSystem teams, whereas, because the societal information system involves both tasks by 'individuals' and tasks by 'engineering'.
   B. **User (Read: stakeholder) needs and requirements** (may or may not describe a system's structure and/or behavior, user specification) - How the user describes what is required? How the user determined their [logical] path to arrival at what is required? A description of an experience that has resulted or may result in a lack of alignment with a visualizable objective experience?
      1. In the societal information system, this is represented by: the articulation and recognition decision inquiry processes
      2. User requirements breakdown structure
      3. In human terms, tasks herein could be completed by anyone (either accessing as a community individual or accessing as a member of the community InterSystem team)
   C. **Engineering requirements (system requirements specification)** - the technical system [engineering] requirements for materialization (their alignment):
      1. In the societal information system, this is represented by: the solution decision inquiry process, as a description from broad to specific of the functional and non-functional [technical] design of the system.
      2. Establish a system level analysis - what must the system do to satisfy user requirements.
      3. Deliver system requirements specification (in the form of, for example, a physical document, spreadsheet, database, or model of desired system illustrating the desired system by a simulation) including requirements breakdown structure.
         i. Determine functional requirements - what does the system need to be able to do. Determine performance requirements associated with functional requirements (how well does the system need to be able to perform those functions) - define performance levels
         ii. Non-functional requirements - what other characteristics are required of the system.
         iii. External interface requirements - what other systems require interface with the system
         iv. Under what conditions is the system expected to operate
         v. Verify the system performance against the requirements. Verification - to confirm system performance against specified requirements (has the system been built right for the user?). Confirming a system as aligning with its requirements. How would I confirm this requirement? Assign rationale. Do not duplicate or repeat requirements in the same document, which will result in conflict in the future.
         vi. In human terms, tasks (development

and execution) herein are completed by members of InterSystem teams.

3. **Preliminary design** - convert the logical architecture described by the engineering requirements into a secondary description of the [digital, physical] sub-systems (the upper-level architecture) that will meet the system requirements. Develop preliminary design based on chosen system concept while considering production, integration, and operational service life-cycle.
   A. Translating the concept (the logical design) into the digital and/or physical design (i.e., the logical design is translated into digital/physical design).
   B. The result (deliverable) of the preliminary design is the allocated baseline (visualizing functionality of the system now allocated to sub-system level (physical or digital "building blocks") groupings, known as configuration items as logically composed in the design/development specification.
      1. Sub-system level specifications for each configuration broken down by development item (or module)
   C. The focus shifts from the [engineering] problem domain to the [engineering] solution domain. Translating the concept into the
   D. Preliminary design verification (review) - was the study and design effort prior (the integration of information) appropriate? Will this design be technically adequate? What are the technical risk?
4. **Detailed design** (and development, prototyping) - "traditional" engineering, where sub-systems are broken down, understood, developed and integrated into existential operation.
   A. The complete engineering design specification that goes to makeup the system.
   B. Engineering of proto-types of sub-systems that make-up the system.
   C. Engineering for prototype system - that satisfy (fulfill) performance (required), reliability (required), life cycle safety and maintenance (required).
   D. Engineering for manufacturability - ensuring resource efficiency (cost affordability).
   E. Test and evaluation of prototypes - confirm system design by means of analysis of tests; design construction test; review, evaluate the expressed design's alignment with requirements.
   F. By end of this stage here is a digital, physical system.
   G. The habitat service system baseline deliverable - the societal level service baseline (a.k.a, product baseline, PBL). As the system now defined by numerous services (products, sub-systems, assemblies) as well as the materials and processes for manufacturing and construction of the total system :: materials, processes, people in time to complete tasks.
   H. Critical design evaluation (review) deliverable - the last point at which the information is in documentation form before transfer to memory and/or execution on the design. Here, the design is fully and officially accepted by all of the inquiry processes: solution (technical) and parallel, organizational value decisioning.
   I. Evaluates [solution] design in terms of readiness for production and construction; asking, Is everything a go, or not, for production and integration into operation. This evaluation process ensures the design is compatible with the societal organizational system (given what is known), or otherwise mis-alignment with a determined value orientation. This includes a detailed understanding of all of the internal and external interfaces.
5. **Operations (engineering) activities -** Operation[al design] (including, construction and production) - Produce and operate components in accordance with the detailed design specifications. Here, at the societal level, a design configuration is selected and integrated into [HSS operations] materiality (software or hardware, digital or spatial) as a 'construction' and/or 'production' (more precisely, 'service' or 'service object', for which there is InterSystem, Community/Commons, and Personal access). Components are developed, produced, and integrated in accordance with the detailed design specification in its final form, and the system is ultimately construction and operational (as an 'HS service' or 'HS service object').
   A. Formal qualification evaluation (review) - the user accepts the system from the InterSystem Team.
   B. All activities beyond system development.
   C. Post-development stage has all activities, but systems engineering is necessary in supporting user.
   D. Solve unanticipated issues where resolution is necessary to ensure the continued usage of the system.
   E. Testing and evaluation of system in its operational environment.
   F. Acceptance stage (because user accepts the digital/physical design that was the translation of the system concept.

1. After the development of the system where activities production, operation, deployment, system support, etc. are accomplished during useful life of the system. The system is doing what it is supposed to do.
6. **Utilization of service** (a.k.a., application, post-development, operations) - operational use and system support through engineering as a deployed or transitioned system. System support (life-cycle) - supported during utilization.
   A. Support Operations (support maintenance) - use, wherein issues become capability (and quality) gaps.
   B. The fulfillment of a need means the closing of a capability gap in the environment through systems engineering and life-cycle operation.

2.3.4 The risk-oriented engineering view

A.k.a., Safe human integration and human factors engineering.

In systems engineering, the human element is often called the human factor. Humans can come to harm, and because humans can come to harm, engineered systems should be designed while accounting for risk *to the human factor*.

The NASA Human Research Program architecture (Read: the development process) is an example of risk-oriented engineering. The human engineering research development cycle (NASA Human Research Program) is:

1. **Evidence** - Reviews of the accumulated evidence from human records, habitat operations, and research findings are compiled into NASA Human Research Program Evidence Reports. These findings provide the basis for identifying the highest priority human risks (in space exploration) and are a record of the state of knowledge for each risk in the program requirements document (PRD). The Evidence Reports are available to the scientific community and general public [https://humanresearchroadmap.nasa.gov/evidence/]. The Evidence Reports receive outside independent review and are updated as needed. If new evidence indicates that a risk should be retired or that a new risk should be added, the Human Research Program (HRP) will, after thorough review with the HSRB, take the appropriate action to modify the PRD and update the Evidence Reports accordingly.
2. **Risks** - Identifies relevant risks, including risks to the health and human performance of the exploration program based on current evidence. Each risk is assigned a risk rating as a tool to communicate to the seriousness of a risk to crew health and performance when applied to the mission* architecture and/or mission characteristics defined for each Design Reference Mission (DRM). The PRD, however, does not establish priority for the risks.
3. **Gaps** - Identifies gaps in knowledge about the risk and the ability to mitigate the risk. The degree of uncertainty in understanding the likelihood, consequence and/or timeframe of a particular risk as well as its criticality to the mission(s) are the major factors that drive the priority of the research gaps listed in the Integrated Research Plan (IRP). Gaps should represent the critical questions that need to be answered in order to significantly reduce the risk. Gaps could change over time based on research progress, current evidence, and mission planning scenarios. In some cases, a gap can address multiple risks.
4. **Tasks** - Defines the tasks that will provide the deliverables required to fill the gaps. Tasks are listed in the Integrated Research Plan (IRP). The IRP describes a plan of research that addresses both human physiology, human performance and the interconnected system of the human and spacecraft in a highly integrated manner. The HRP Elements identify specific research tasks that are targeted at better characterizing a risk or developing mitigation capabilities to reduce the risk to an acceptable level.
5. **Deliverables** - Each task or progression of tasks is designed to ultimately culminate in deliverables or products that range from risk characterization to prototype technology or countermeasures.

** A 'mission' is a type of 'project' with a human factor.*

Human Research Program (HRP) deliverables are generally:

1. **Knowledge** - deliverables that add to the body of knowledge regarding the risk or concern.
2. **Countermeasures** – preventative and treatment actions taken to address a risk,.
3. **Technology development** - hardware and software that enable risk monitoring, prevention or treatment.
4. **Operational protocols** - operational procedures and methods that define a technique or process for mitigation of the risk.
5. **Guidelines, requirements, and standards** – information that defines the acceptable levels of risk. Information generated by HRP that can inform the status of the risk and anticipated mitigations are documented in the HSRB Risk Summary.

A socio-technical [human] research program may have the following deliverable categories:

1. **Requirement** or **guideline** - The "Requirement or Guideline" deliverable is chosen when a task will result in information that is relevant to a requirement (or requirements set) or guideline associated with a higher decision set.
2. **Technology** or **tool** - The "Technology or Tool" deliverable covers a broad spectrum of developments that includes hardware, software, systems solutions, new processes, new systems and machines (inventions), new methods and procedures (innovative methods), collaborative design tools, databases, computational models, or systems simulations.
3. **Countermeasure** - A "Countermeasure" deliverable is a specific protocol that is developed and validated to prevent or reduce the likelihood or consequence of a risk [of acceptable level]. Countermeasures may be medical, physical, or operational entities, such as a pharmaceutical or nutritional supplement, hardware or software (prototype and fully integrated), or specific exercise/training, entrainment routines, respectively. A countermeasure deliverable is usually specific and extensive enough to require validation in habitat service operation.
4. **Standard** - A working group integration of all feedback and discovery to which operation conforms by threshold (i.e., by degree). Discovery workgroups may result in a recommendation for a new or updated standard. Standards working groups integrate discoveries into the next iteration of the societal-habitat system, through project-engineering and cooperation-coordination.

2.3.5 The asset coordination life-cycle view of engineering

The asset management lifecycle (a.k.a., asset lifecycle management, ALM) is a process for coordinating ("managing") the usage (and maintenance) of a support service 'asset' (or 'object') throughout its lifetime (or period of service). Each assets lifecycle is defined by a series of stages:

1. **Procurement/access coordination:**
    A. Set requirements for purchasing the asset
        1. Based on inventory, consumption, and labor rates.
            i. Make purchase order.
                1. Track purchase until delivery.
2. **Inbound/outbound service coordination:**
    A. Inbound services.
        1. Receive shipment.
        2. Unpackage shipment.
        3. Reconcile shipment with purchase order (itemized checklist) to ensure accuracy.
        4. Tag asset for tracking in system.
    B. Outbound services.
        1. Package asset (for deliver to end location).
3. **Inventory coordination:**
    A. Storage - Assets not yet ready to be delivered to an end location are deposited in an inventory (organized for streamlined storage and retrieval).
        1. Inventory cycle counts (inventory surveys) ensure that min/max levels are maintained and accurately reflected in the asset management database.
4. **Deployment coordination:**
    A. Request/demand - an asset is requested for use.
    B. Retrieval - an asset is retrieved from inventory
    C. Provisioning ("to make something available or ready to use") - final configuration of system for specific use.
    D. Access - by user for usage.
5. **Re-assignment coordination:**
    A. Return access -
        1. Still has useful life?
        2. Does not still have useful life?
        3. Event.
            i. Lost.
            ii. Return to inventory.
            iii. Return for de-composition.
            iv. Return for repair.

More simply, the asset lifecycle consists of

1. **Asset objective** - communicate and plan asset
2. **Asset model** - design asset.
3. **Asset construction** - code/build and test of asset.
4. **Asset deployment** - integration and operation
5. **Asset usage** - the asset is used by the user and maintained/operated by asset technicians.
6. **Asset return** - the return of the asset to inventory, or for de-cycling.

2.3.6 Asset life-cycle software

Current asset lifecycle software solutions include:

1. Autodesk fusion product lifecycle.
2. Service life-cycle coordination (application, asset management lifecycle).

3 [Engineering] Life-cycle processes

The socio-technical engineering process is a multi-stage method that results in a highly predictable societal design materialization.

3.1 Initiation and planning stage

The output work products for this stage are:

1. Issue articulation [initial].
2. Project coordination plan [initial].
3. Project charter [initial].
4. Maintenance plan [initial].
5. Configuration coordination plan [initial].

3.2 Requirements definition stage

The requirements definition phase starts with establishing a functional baseline from which to do future work.

3.2.1 Establish functional baseline

A.k.a., System requirements baseline.

The functional baseline is the main technical work product of the Requirements Definition stage of an engineering project. The system requirements are baselined after the Project Team's formal approval of the Requirements Specification. Once the requirements are baselined, any changes to the requirements must be coordinated under change control procedures.

Clarification: *To be "baselined" means to have been formally determined (or, selected).*

The output work products for this stage are:

1. Project coordination plan [revised].
2. Requirements specification [initial].
3. Requirements traceability matrix [initial].
4. Maintenance plan [revised].
5. Configuration coordination plan [revised].
6. Organizational continuity plan [revised].
7. Data dictionary [revised].

3.3 Functional design stage

During the functional design stage, the overall structure of the product is defined from a functional viewpoint. The goal of this stage is to define and document the functions of the product to the extent necessary to obtain the system owner and users understanding and approval and to the level of detail necessary to build the system design.

The deliverable of the functional design stage is the Functional Design [Document].

The high-level activities are presented in the sections listed below.

1. Determine system structure.
2. Design content of system inputs and outputs.
3. Design user interface.
4. Design system interfaces.
5. Design system security controls.
6. Build logical model.
7. Build data model.
8. Develop functional design.
9. Select system architecture.

The output work products for this stage are:

1. Project coordination plan [revised].
2. Functional design document [final].
3. Maintenance plan [revised].
4. Requirements specification [final].
5. Requirements traceability matrix [revised].
6. Configuration coordination plan [revised].
7. Organizational continuity plan [revised].
8. Data dictionary [final].

3.3.1 The functional design specification

The functional design process maps the "what to do" of the Requirements Specification into the "how to do it" of the design specifications. The functional design describes the logical system flow, data organization, system inputs and outputs, processing rules, and operational characteristics of the product from the user's point of view. The functional design is not concerned with the software or hardware that will support the operation of the product or the physical organization of the data or the programs that will accept the input data, execute the processing rules, and produce the required output. The focus is on the functions and structure of the components that comprise the product. The functional design describes how the product will be structured to satisfy the requirements identified in the Requirements Specification. It is a description of the structure, components, interfaces, and data necessary before development can begin.

The functional design is a model or representation of the system that is used primarily for communicating design information to facilitate analysis, planning, and coding decisions. It represents a partitioning of the system into design entities and describes the important properties and relationships among those entities. Design descriptions may be produced as documents, graphic representations, formal design languages, and records in a database.

Within the functional design, the design entities can be organized and presented in any number of ways. The goal of this activity (Read: develop the functional design) is to compile the design entities and their associated attributes in a manner that facilitates the access of design information from various viewpoints (e.g., project coordination, engineering development, quality assurance, and testing). Also, the design entities and their attributes must be described in terms that are understandable to the system users.

Prototyping of system functions can be helpful in communicating the design specifications to the system users. Prototypes can be used to simulate one function, a module, or the entire product. Prototyping is also useful in the transition from the functional design to the system design.

3.3.2 Determine system structure

A hierarchical approach is useful for determining the structure and components of the system. System decomposition is one hierarchical approach that divides the system into different levels of abstraction. Decomposition is an iterative process that continues until single purpose components (i.e., design entities or objects) can be identified. Decomposition is used to understand how the product will be structured, and the purpose and function of each entity or object.

The goal of the decomposition is to create a highly cohesive design. A design exhibits a high degree of cohesion if each design entity in the system unit is essential for that unit to achieve its purpose.

Several reliable methods exist for performing system decomposition. Select a method that enables the design of simple, independent entities. Functional design and object-oriented design are two common approaches to decomposition. These approaches are not mutually exclusive. Each may be applicable at different times in the design process.

3.3.2.1 Tasks to determine system structure

The system decomposition activity includes the following tasks.

1. Identify design entities.
2. Identify design dependencies.

3.3.3 Identify design entities

Design entities result from a decomposition of the system requirements. A design entity is an element (or object) of a design that is structurally and functionally distinct from other elements and is separately named and referenced. The number and type of entities required to partition a design are dependent on a number of factors, such as the complexity of the product, the design method used, and the development environment. The objective of design entities is to divide the product into separate components that can be coded, implemented, changed, and tested with minimal effect on other entities.

3.3.3.1 Attributes of design entities

A design entity attribute is a characteristic or property of a design entity. It provides a statement of fact about an entity. The following are common attributes that should be considered for each design entity.

1. Assign a unique name to each entity.
2. Classify each entity into a specific type. The type may describe the nature of the entity, such as a sub-program or module; or a class of entities dealing with a particular type of information.
3. Describe the purpose or rationale for each entity. Include the specific functional and performance requirements for which the entity was created.
4. Describe the function to be performed by each entity. Include the transformation applied to inputs by the entity to produce the desired output.
5. Identify all of the external resources that are needed by an entity to perform its function.
6. Specify the processing rules each entity will follow to achieve its function. Include the algorithm used by the entity to perform a specific task and contingency actions in case expected processing events do not occur.
7. Describe the data elements internal to each entity. Include information such as the method of representation, format, and the initial and acceptable values of internal data. This description may be provided in the data dictionary.

3.3.4 Identify design dependencies

Design dependencies describe the relationships or interactions between design entities at the module, process, and data levels. These interactions may involve the initiation, order of execution, data sharing, creation, duplication, use, storage, or destruction of entities.

Identify the dependent entities of the system design, describe their coupling, and identify the resources required for the entities to perform their function. Also define the strategies for interactions among design entities and provide the information needed to perceive how, why, where, and at what level actions occur.

Dependency descriptions should provide an overall picture of how the product will work. Data flow diagrams, structure charts, and transaction diagrams are useful for showing the relationship among design entities.

The dependency descriptions may be useful in producing the system integration plan by identifying the entities that are needed by other entities and that must be developed first. Dependency descriptions can also be used to aid in the production of integration test cases.

3.3.5 Design content of system inputs and outputs

Design the content and format for each of the product inputs and outputs based on the system input and output requirements identified during the Requirements Definition Stage. Involve the system users in the design process to make certain that their needs and expectations are being met.

Document the design for the system inputs and outputs in accordance with the project design standards. Discuss the designs with the system owner and users and submit completed designs for their review and approval. The approved designs will be incorporated into the Functional Design Document.

3.3.6 Design user interface

Design a user interface that is appropriate for the users, content, and operating environment for the product. Determine interface levels for all categories of users. For interactive user environments, prototype the user interface. Arrange for users to experiment with the prototypes so that design weaknesses in the interface can be identified and resolved early. Use prototypes to gain user acceptance of the interface.

3.3.7 Design system interface

Develop a design depicting how the product will interface with other systems based on the system interface requirements identified in the Requirements Definition Stage. Submit the applicable interface designs for review by the system owner or system administrator for each system that will interface with the product. Any incompatibilities with the interfaces will be identified early in the design process and corrective actions can be initiated to assure each interface is properly designed and coded.

3.3.8 Design system controls

Design the access (security) controls that will be incorporated into the product based on the access requirements identified during the Requirements Definition Stage.

3.3.8.1 Design system controls procedure

Use the following procedure to implement the design process.

1. Identify the users and organizations that will have access to the product. Indicate what access restrictions they will have. All persons in a work area may not have the same access level. Controls should be implemented to assure that materials and systems requiring protection are not accessed by unauthorized individuals.
2. Identify controls for the product, such as the user identification code for system access and the network access code for the network on which the product will reside.
3. Identify whether access restrictions will be applied at the system, subsystem, transaction, record, or data element levels. Sensitive information must be protected in accordance with directives.
4. Identify physical safeguards required to protect hardware, software, or information from natural hazards and malicious acts.
5. Identify communications access (security) requirements.

3.3.9 Build logical model

The logical model defines the flow of data through the system and determines a logically consistent structure for the system. Each module that defines a function is identified, interfaces between modules are established, and design constraints and limitations are described. The focus of the logical model is on the real-world problem or need to be solved by the product.

A logical model has the following characteristics:

1. Describes the final sources and destinations of data and control flows crossing the system boundary rather than intermediate handlers of the flows.
2. Describes the net transfer of data across the system boundary rather than the details of the data transfer.
3. Provides for data stores only when required by an externally imposed time delay.

When building a logical model, the organization of the model should follow the natural organization of the product's subject matter. The names given to the components of the model should be specific. The connections among the components of the model should be as simple as possible.

The logical model should be documented in user terminology and contain sufficient detail to obtain the system owner's and users' understanding and approval. Use data flow diagrams to show the levels of detail necessary to reach a clear, complete picture of the product processes, data flow, and data stores.

Maintain the logical model and data flow diagrams for incorporation into the Functional Design Document. Keep the logical model and diagrams up-to-date. They will serve as a resource for planning enhancements during maintenance, particularly for enhancements involving new functions.

3.3.10 Build data model

A data model is a representation of a collection of data objects and the relationships among these objects (i.e., representation of information about a form or a process).

The data model is used to provide the following functions:

1. Transform the sense entities into data entities.

2. Transform the socio-technical rules into data relationships.
3. Resolve the many-to many relationships as intersecting data entities.
4. Determine a unique identifier (key) for each data entity.
5. Add the attributes (facts) for each data entity.
6. Document the integrity rules required in the model.
7. Determine the data accesses (navigation) of the model.

The data dictionary is developed in this stage. Its purpose it to catalogue every known data element used in the user's work and every system-generated data element. Data elements are documented in detail to include attributes, known constraints, input sources, output destinations, and known formats.

The data dictionary can serve as a central repository of information for both developers and end users. The dictionary can include business rules, processing statistics, and cross-referencing information for multiple vendor environments.

To expand the data dictionary, define, analyze, and complete data definitions using the following steps.

1. Identify data needs associated with various system features.
2. Match (verify) data needs with the data dictionary.
3. Match the data dictionary with specific data structures.
4. Create data record layouts.
5. Ensure that all data can be maintained through add, change, or delete functions.

3.3.11 Develop functional design

Major work products are the Functional Design and the revised Requirements Traceability Matrix. Each requirement identified in the Requirements Specification must be traceable to one or more design entities. This traceability ensures that the product will satisfy all of the requirements and will not include inappropriate or extraneous functionality. Expand the Requirements Traceability Matrix developed in the Requirements Definition Stage to relate the functional design to the requirements.

It is relevant to note here that technical design reviews occur during the system engineering lifecycle. These reviews can be supported by peer reviews, which are deeper technical reviews by technical experts in the subject matter to be reviewed. Design reviews are generally conducted when the system under development meets a milestone or level of development/construction during the product design through to realization process. Formalized design reviews have entry and exit criteria, and/or acceptance criteria.

The following tasks are involved in developing the functional design:

1. Develop Functional Design Document
2. Conduct Functional Design Review

3.3.11.1 Develop functional design document

The Functional Design Document defines the functions of the system in user terminology and provides a firm foundation for the development of the system design. The Functional Design Document should be written from the system users' perspective. This document provides the users with an opportunity to review and provide input to the product design before system design work is completed.

3.3.11.2 Conduct functional design review (technical design review)

The Functional Design Review is a formal technical review of the basic design approach. The primary goal of the Functional Design Review is to demonstrate the ability of the system design to satisfy the project requirements. The review may be a series of presentations by the project team to the system users, functional area Team members (a.k.a., points-of-contact). Vendors may be invited to participate in the Functional Design Review when an off-the-shelf software product or hardware item is being considered for the system architecture.

The work product is the Functional Design Document. The review of this document will result in one of the following outcomes:

1. **Selection (a.k.a., approval)** - indicates that the functional design is satisfactorily completed.
2. **Hold selection (a.k.a., hold approval, contingent approval)** - indicates that the functional design is not considered accomplished until the satisfactory completion of identified action items.
3. **Non-selection (a.k.a., disapproval)** - indicates that the functional design is inadequate. Another Functional Design Review is required, once specified changes to the functional design are completed.

Conduct the Functional Design Review to perform the following verifications:

1. Evaluate the progress, technical adequacy, and risk mitigation of the selected design approach. Determine whether the design approach is being followed by the project team.
2. Evaluate the progress, technical adequacy, and risk mitigation of the selected test approach. Review the following items:
   A. System test requirements from the requirements specification document.

B. Organization and responsibilities of group conducting tests.
C. Planned format, content, and distribution of test reports.
D. Planned resolution of problems and errors identified during testing.
E. Retest procedures.
F. Change control and configuration management of test items.
G. Special test tools not required as deliverables.

3. Evaluate the techniques to be used to meet quality assurance requirements.
4. Establish the existence and compatibility of the physical and functional interfaces.
5. Determine whether the functional design embodies all of the product requirements.
6. Verify that the design represents a system that can meet the functional, data, and interface requirements.
7. Demonstrate any rapid design prototypes used to make design decisions.
8. Identify potential high risk areas in the design and any requirements changes that could reduce risk.
9. Review to assure that consideration has been given to optimizing the maintainability and maintenance aspects of the product.

The following items should be considered for review and evaluation during the Functional Design Review:

1. **Functional flows**: Indicate how the system functional flows map the software and interface requirements to the individual high-level components of the product.
2. **Storage allocation data**: Describe the manner in which available storage is allocated to individual components. Timing, sequencing requirements, and relevant equipment constraints used in determining the allocation should be included.
3. **Control functions**: Describe the executive control and start/recovery features of the product.
4. **Component structure**: Describe the high-level structure of the product, the reasons for choosing the components, the development technique that will be used within the constraints of available computer resources, and any support programs that will be required in order to develop and maintain the product and allocated data storage.
5. **Security**: Identify the security requirements and provide a description of the techniques to be used for implementing and maintaining security within the product.
6. **Information systems engineering facilities**: Describe the availability, adequacy, and planned utilization of the information systems engineering facilities including both Government-provided and commercially available facilities.
7. **Information systems engineering facility versus the operational system**: Describe any unique design features that exist in the functional design in order to allow use within the information systems engineering facility that will not exist in the operational product. Provide information on the design of support programs not explicitly required for the operational system that will be generated to assist in the development of the product.
8. **Development tools**: Describe any special tools (e.g., simulation, data reduction, or utility tools) that are not deliverables, but are planned for use during systems development.
9. **Test tools**: Describe any special test systems, test data, data reduction tools, test computer software, or calibration and diagnostic software that are not deliverables, but are planned for use during development.
10. **Commercial resources**: Describe commercially available computer resources, including any optional capabilities (e.g., special features, interface units, special instructions, controls, formats). Identify any limitations of commercially available equipment (e.g., failure to meet user interface, safety, and maintainability requirements) and identify any deficiencies.
11. **Existing documentation**: Maintain a file and have available for review any existing documentation supporting the use of commercially available computer resources.
12. **Support resources**: Describe the resources necessary to support the product during engineering, installation, and operational state (e.g., operational and support hardware and software personnel, special skills, human factors, configuration management, testing support, documentation, and facilities/space management).
13. **Standards**: Describe any standards or guidelines that must be followed.
14. **Operation and support documentation**: Describe the documentation that will be produced to support the operation and maintenance of the product.

3.3.12 Select system architecture

When the system architecture for the product has not been predetermined by the existing environment of the system users, evaluate system architecture alternatives to determine which one best satisfies the project requirements. Select the specific design based on the pre-determined value conditions.

The following tasks are involved in selecting a system architecture:

1. Evaluate system architecture alternatives
2. Select system architecture

3.3.12.1 Evaluate system architecture alternatives

Consider system architecture alternatives within the organizations architecture guidelines and standards conditions that enable the project objectives and requirements to be achieved.

The following procedure provides one approach for evaluating the architecture alternatives:

1. Conduct an analysis to determine the most effective and conditionally aligned alternative.
2. Create and evaluate a data flow diagram for each alternative.
3. Identify how users would interact with the features associated with each alternative.
4. Create a list of the risks associated with each alternative and develop a plan for mitigating each risk.
5. Compare the performance capabilities of each alternative.
6. Follow the societal-level decision system.

3.4 System design stage

The goal of this stage is to translate the user-oriented functional design specifications into a set of technical, realization-oriented system design specifications; and to design the data structure and processes to the level of detail necessary to plan and execute the Construction and Implementation Stages. General module specifications should be produced to define what each module is to do, but not how the module is to be coded. Effort focuses on specifying individual routines and data structures while holding constant the structure and interfaces developed in the previous stage. Each module and data structure is considered individually during detailed design with emphasis placed on the description of internal and procedural details. The primary work product of this stage is a system design that provides a specification (blueprint) for the materialization (i.e., [en] coding) of individual modules and elements.

The following items provide input to this stage:

1. Functional design.
2. Maintenance plan.
3. Requirements specification.
4. Requirements traceability matrix.
5. Software configuration management plan.
6. Project coordination plan.
7. Access plan.
8. Data dictionary

The high-level activities for this stage are:

1. Design specifications for modules.
2. Design physical model and database structure.
3. Develop integration test considerations.
4. Develop system test considerations.
5. Develop conversion plan.
6. Develop system design.

The output work products for this stage are:

1. Project coordination plan [revised].
2. Conversion plan [initial].
3. Maintenance plan [revised].
4. Requirements traceability matrix [revised].
5. Configuration management plan [final].
6. System design document [final].
7. Test plan [initial].
8. Test type approach and reports [initial].
9. Test cases [initial].

3.4.1 System design

The system design is the main technical work product of the System Design Stage. The system design translates requirements into precise descriptions of the components, interfaces, and data necessary before coding and testing can begin. It is a blueprint for the Construction Stage based on the structure and data model established in the Functional Design Stage.

Once the system design is baselined, any changes to the design must be managed under change control procedures. Approved changes must be incorporated into the System Design Document.

It is important for the system users to understand that some changes to the baselined system design may affect the project scope and therefore can change the project resources, schedule, etc. It is the responsibility of the project coordinator and team to identify system user requested changes that would result in a change of project scope; evaluate the potential impact to the project elements (resources, schedule, etc.); and notify the system user of the project planning revisions that will be required to accommodate their change requests.

3.4.2 Design specifications for modules

Expand the functional design to account for each major action that must be performed and each data object to be managed. Detail the design to a level such that each sub-system represents a function that a developer will be able to develop.

The following procedure facilitates in designing the module specifications:

1. Identify a structure for each action needed to meet

each function or requirement in the Requirements Specification and the data dictionary.
2. Identify any routines and structures that may be available as reusable objects.
3. Identify structures that must be designed and developed (custom-built). Assign a name to each structure and object that is functionally meaningful. Identify the system features that will be supported by each structure.
4. Specify each structure interface. Update the data dictionary to reflect all program and object interfaces changed while evolving from the functional to the system design.
5. Define and design significant attributes of the structures to be custom-built.
6. Expand the structure interfaces to include control items needed for design validity (e.g., error and status indicators).
7. Combine similar structures and objects. Group the design entities into modules based on closely knit functional relationships. Formulate identification labels for these modules.
8. Show dependencies between data structures and physical structures.
9. Change the design to eliminate features that reduce maintainability or reusability (i.e., minimize coupling between programs and maximize the cohesion of programs).

Document the system design primarily in the form of diagrams. Supplement each diagram with text that summarizes the function (or data) and highlights important performance and design issues.

When using structured design methods, the design diagrams should:

1. Depict the product as a top-down set of diagrams showing the control hierarchy of all programs to be implemented.
2. Define the function of each structure.
3. Identify data and control interfaces between programs.
4. Specify files, records, and global data accessed by each program.
5. When using object-oriented or data-centered design methods, the design diagrams should:
6. Show the data objects to be managed by the product.
7. Specify the program functions to be included within each object.
8. Identify functional interfaces between objects.
9. Specify files and records comprising each object.
10. Identify relationships between data files.

Standards for specifications may be provided by government agencies, standards organizations (SAE, AWS, NIST, ASTM, ISO, CEN, US DoD, etc.), trade associations, corporations, and others.

The following British standards apply to specifications:

- BS 7373-1:2001 Guide to the preparation of specifications [4]
- BS 7373-2:2001 Product specifications. Guide to identifying criteria for a product specification and to declaring product conformity [5]
- BS 7373-3:2005, Product specifications. Guide to identifying criteria for specifying a service offering
- The following NIST standards apply [nist.gov]:
- IEEE P7001 - Transparency of autonomous systems
- IEEE P7003 - Algorithmic bias considerations
- IEEE P7007 - Ontological standard for ethically driven robotics and automation systems
- IEEE P7008 - Standard for ethically driven nudging for robotic, intelligent and autonomous systems
- IEEE P7009 - Standard for fail-safe design of autonomous and semi-autonomous systems
- IEEE P7010 - Well-being metrics standard for ethical artificial intelligence and autonomous systems

3.4.3 Design physical model and database structure

The physical model is a description of the dynamics, data transformation, and data storage requirements of the system. The physical model maps the logical model created during the Functional Design Stage to a specific technical solution.

3.4.4 Develop conversion plan

A.k.a., Develop transition plan.

If the product will replace an existing system, then develop a Conversion Plan. The major elements of the Conversion Plan are to develop conversion procedures, outline the installation of new and converted structures, coordinate the development of structural-conversion, and plan the implementation of the conversion procedures.

System conversion should include a confirmation of file integrity. Determine what the output in the new system should be compared with the current system, and ensure that the files are synchronized. The objective of file conversion is new files that are complete, accurate and ready to use.

Many factors influence conversion, such as the design of the current and new systems and the processes for input, storage, and output. Understanding the structures function in the old system and determining if the function will be the same or different in the new system is of major importance to the Conversion Plan.

The structure of the system to be converted can limit the development of the system and affect the choice of structure.

Consider the following factors during the development of the Conversion Plan:

1. Determine if any portion of the conversion process should be performed manually.
2. Determine whether parallel runs of the old and new systems will be necessary during the conversion process.
3. Understanding the function of the structure in the old system and determining if the use will be the same or different in the new system is important.
4. The order that information is processed in the two systems influences the conversion process.
5. User work and delivery schedules, timeframes for reports and end-of-year procedures, and the criticality of the data help determine when conversion should be scheduled.
6. Determine whether availability and use should be limited during the conversion.
7. Plan for the disposition of obsolete or unused structure that is not converted.

3.4.5 Develop system design

Major work products include the System Design Document and the updated Requirements Traceability Matrix. Each requirement identified in the Requirements Specification must be traceable to one or more design entities. This traceability ensures that the product will satisfy all of the requirements and will not include inappropriate or extraneous functionality. Revise the Requirements Traceability Matrix developed in the Requirements Definition Stage to relate the system design to the requirements.

The following tasks are involved in developing the system design.

1. Develop System Design Document.
2. Conduct System Design Review.

3.4.5.1 Develop system design document

The System Design Document records the results of the system design process and describes how the system will be structured to satisfy the requirements identified in the Requirements Specification. The System Design Document is a translation of the requirements into a description of the structure, components, interfaces, and data necessary to support the construction process.

3.4.5.2 Conduct system design review (technical design review)

The System Design Review is a formal technical review of the system design. The purpose of the review is to demonstrate to the system users that the system design can be implemented on the selected platform and accounts for all requirements and accommodates all design constraints (e.g., performance, resource, and reliability requirements). The design review should include a review of the validity of algorithms needed to perform critical functions.

3.5 Construction stage

The goal of this stage is to translate the set of technical system design specifications into a language the constructor can understand and execute. Construction may involves materializing, coding, validation and unit testing by a developer. Plans are developed for the installation of the operating environment hardware and software. A training program is designed and a Training Plan that describes the system is produced.

The activities in this stage result in the transformation of the system design into the first complete executable (operatable) representation of the product.

The high-level activities for this stage are:

1. Establish development environment
2. Develop programs
3. Conduct unit testing
4. Establish development baselines
5. Plan transition to operational status
6. Generate operating documentation
7. Develop training plan
8. Develop installation plan

The output work products for this stage are:

1. Project coordination plan [revised]
2. Maintenance plan [revised]
3. Requirements traceability matrix [revised]
4. Conversion plan [revised]
5. Test type approach and reports [revised]
6. Test cases [revised]
7. Transition plan [initial]
8. Installation plan [initial]
9. Training plan [initial]
10. Operating documentation [initial]
    A. Users manual
    B. Developer’s reference manual
11. System units and modules [initial]

3.5.1 Establish Development Environment

Establishing the development environment involves assembling and installing the hardware, software, equipment, databases, and other items required to support the construction effort.

Before being integrated into or used to support the product, vendor products should be tested to verify that the product satisfies the following objectives:

1. The product performs as advertised/specified.
2. The product's performance is acceptable and predictable in the target environment.
3. The product fully or partially satisfies the project requirements.
4. The product is compatible with the project team's other hardware and software tools.

Time should be planned for the project team to become familiar with new products. Ensure that the project team members who will use the hardware or software obtain proper training. This may involve attendance at formal training sessions conducted by the vendor or the services of a consultant to provide in-house training.

3.5.2 Conduct unit testing

Unit testing is used to verify the input and output for each module. Successful testing indicates the validity of the function or sub-function performed by the module and shows traceability to the design. During unit testing, each module is tested individually and the module interface is verified for consistency with the design specification. All important processing paths through the module are tested for expected results. All error handling paths are also tested.

Unit testing is driven by test cases and test data that are designed to verify requirements, and to exercise all program functions, edits, in-bound and out-bound values, and error conditions identified in the program specifications. If timing is an important characteristic of the module, tests should be generated that measure time critical paths in average and worst-case situations.

Plan and document the inputs and expected outputs for all test cases in advance of the tests. Log all test results. Analyze and correct all errors and retest the unit using the scenarios defined in the test cases. Repeat testing until all errors have been corrected.

While unit testing is generally considered the responsibility of the developer, the project coordinator or lead developer should be aware of the unit test results.

Completion of unit testing for a component signifies internal project delivery of a component or module for integration with other components.

3.5.3 Establish development baseline

A development baseline is an approved "build" of the product. A build can be a single component or a combination of components. The first development baseline is established after the first build is completed, tested, and approved by the project manager or lead developer. Subsequent versions of a development baseline should also be approved. The approved development baseline for one build supersedes that for its predecessor build.

Conduct internal build tests such as regression, functional, performance, and reliability. Regression tests are designed to verify that capabilities in earlier builds continue to work correctly in subsequent builds. Functional tests focus on verifying that the build meets its functional and data requirements and correctly generates each expected display and report. Performance and reliability tests are used to identify the performance and reliability thresholds of each build.

Once the first development baseline is established, any changes to the baseline must be managed under the change control procedures. Approved changes to a development baseline must be incorporated into the next build of the product and revisions made to the affected work products (e.g., Requirements Specification, System Design Document, and Program Specifications).

Document the internal build test procedures and results. Identify errors and describe the corrective action that was taken. Place a copy of the internal build test materials in the Project Test File.

Maintain configuration control logs and records as required. Expand the Requirements Traceability Matrix developed in the Requirements Definition Stage.

3.5.4 Plan transition to operational status

Successful transition from acceptance testing to full operational use of the product depends on planning the transition long before the product is installed in its operational environment. In planning for the transition, quantify the operational needs associated with the product and describe the procedures that will be used to perform the transition.

Rely on experience and data gathered from previous, similar projects to define these needs. Develop a Transition Plan that describes the detailed plans, procedures, and schedules that will guide the transition process. Coordinate development of the plan with the operational and maintenance personnel. The following issues should be considered in the preparation of a Transition Plan:

1. Develop detailed operational scenarios to describe the functions to be performed by the operational support staff, maintenance staff, and users.
2. Document the release process. If development is incremental, define the particular process, schedule, and acceptance criteria for each release.
3. Describe the development or migration of data, including the transfer or reconstruction of historic data. Schedule ample time for the system owner and user to review the content of reconstructed or migrated data files to reduce the chance of errors

or omissions.
4. Specify problem identification and resolution procedures for the operational product.
5. Define the configuration management procedures that will be used for the operational product. Ideally, the methods defined in the Software Configuration Management Plan that were employed during product development can continue to be used for the operational product.
6. Define the scope and nature of support that will be provided by the project team during the transition period.
7. Specify the organizations and individuals who will be responsible for each transition activity, ensuring that responsibility for the product by the operations and maintenance personnel increases progressively.
8. Identify products and support services that will be needed for day-to-day operations or that will enhance operational effectiveness.

3.5.5 Generate operating documentation

Plan, organize, and write the operating documentation that describes the functions and features of the product from the users point-of-view. The different ways that users (including system administration and maintenance personnel) will interact with the product must be considered. The needs of the users should dictate the document presentation style and level of detail. Responsibilities for changing and maintaining the documents should be described in each document.

The following are typical operating documents for a large project:

1. Users manual/online help screens.
2. Developer's reference manual.
3. Intersystem team manual (a.K.A., Systems administration manual).
   A. Database administration manual.
4. Operations manual.

It is recommended that a technical writer be involved in the generation of all operating documents. A technical writer works closely with the project team to ensure that documents are grammatically correct; comply with applicable standards; and are consistent, readable, and logical.

Use the following procedure to develop the operating documentation.

1. Identify the operating documents that need to be developed. Determine if any of the documents can be combined or delivered as multiple volumes.
2. Determine whether the documents should be provided as printed material, standalone electronic files, online documentation accessed through the product, or a combination.
3. Determine the best presentation method or combination of methods required for each of the documents, such as a traditional manual, quick reference guide or card, or online help.
4. Identify all of the features of the user interface and the tasks users will perform.
5. Identify the users' needs and experience levels to determine:
   A. The amount of user interaction, level of interaction, and whether the interaction is direct or indirect.
   B. The appropriate level of detail (e.g., the Users Manual should not contain highly technical terms and explanations that may confuse or frustrate a user).
6. Determine the document content and organization based on whether the document will be used more as an instructional tool or a reference guide.
7. Develop descriptions of each function and feature of the product and organize the information to facilitate quick, random access.
8. Provide appropriate illustrations and examples to enhance clarity and understanding.
9. Establish a schedule for the documents to be reviewed after the product goes into production. Operating documents must be kept up-to-date as long as the product remains in production.

The following tasks describe the minimum requirements for operating documentation.

1. Produce Users Manual
2. Produce Developer's Reference Manual

3.5.5.1 Produce users manual

The Users Manual provides detailed information users need to access, navigate through, and operate the product. Users rely on the Users Manual to learn about the product or to refresh their memory about specific functions. A Users Manual that is organized functionally so that the information is presented the same way the product works helps users understand the flow of menus and options to reach the desired functions.

Different categories of users may require different types of information. A modular approach to developing the Users Manual to accommodate the needs of different types of users eliminates duplication and minimizes the potential for error or omission during an amendment or update. For example, separate general information that applies to all users from the special information that applies to selected users such as system administrators or database administrators. The special information can be presented in appendixes or supplements that are only provided to the users who need the information.

Write the draft Users Manual in clear, non-technical terminology that is oriented to the experience levels and needs of the user(s).

For very small projects, a quick reference guide or card may be more appropriate than a full-scale Users Manual.

For projects of any size, a quick reference card may be developed as a supplement to more detailed user documentation.

The following are typical features of a users manual.

1. Overview information on the history and background of the project and the architecture, operating environment, and current version or release of the product.
2. Instructions for how to install, setup, or access the product.
3. Complete coverage of all functions, presented in a logical, hierarchical order.
4. Accurate pictures of screens and reports, ideally with data values shown, so the user can easily relate to examples.
5. In-depth examples and explanations of the areas of the product that are most difficult to understand.
6. Clear delineation of which features are accessible only to specific users.
7. Instructions on accessing and using online help features.
8. Procedures for data entry.
9. Descriptions of error conditions, explanations of error messages, and instructions for correcting problems and returning to the function being performed when the error occurred.
10. Instructions for performing queries and generating reports.
11. Who to contact for help or further information.

3.5.5.2 Produce developer's reference manual

The Developer's Reference Manual contains information about program development used by the maintenance staff to maintain the programs, databases, interfaces, and operating environment. The Developer's Reference Manual should provide an overall conceptual understanding of how the product is constructed and the details necessary to implement corrections, changes, or enhancements.

The Developer's Reference Manual describes the logic used in developing the product and the functional and system flow to help the maintenance staff understand how the programs fit together. The information should enable a developer to determine which programs may need to be modified to change a system function or to fix an error.

Use appendixes to provide detailed information that is likely to change as the product is maintained. For example, a list of program names and a synopsis of each program could be included as an appendix.

The following are typical features of a Developer's Reference Manual.

1. A description of the technical environment, including versions of the development language(s) and other proprietary software packages.
2. A brief description of the design features including descriptions of unusual conditions and constraints.
3. An overview of the architecture, program structure, and program calling hierarchy.
4. The design and coding practices and techniques used to develop the product.
5. Concise descriptions of the purpose and approach used for each program.
6. Layouts for all data structures and files used in the product.
7. Descriptions of maintenance procedures, including configuration management, program checkout, and system build routines.
8. The instructions necessary to compile, link, edit, and execute all programs.
9. Manual and automated backup procedures.
10. Error-processing features.

3.5.6 Develop training plan

A Training Plan defines the training needed to implement and operate the product successfully. The Training Plan should address the training that will be provided to the system users, and InterSystem Team Operators and Maintenance personnel. When new hardware or software is being used, affected personnel will need hands-on experience before bringing the new system (equipment and/or software) into daily operation.

Training must address both the knowledge and the skills required to operate and use the system effectively.

Complete the Training Checklist to ensure that all activities and work products are complete.

Place a copy of the initial Training Plan and completed Training Checklist in the Project File. The plan will be reviewed and updated during the Testing Stage.

Design the training to accomplish the following objectives:

1. Provide trainees with the specific knowledge and skills necessary to perform their work.
2. Prepare training materials that will sell the product as well as instruct the trainees. The training should leave the trainees with the enthusiasm and desire to use the new product.
3. Account for the knowledge and skills the trainees bring with them, and use this information as a transition to learning new material.
4. Anticipate the needs for follow-on training after

the product is fully operational, including refresher courses, advanced training, and repeats of basic courses for new personnel.
5. Build in the capability to update the training as the product evolves.

The Training Plan should address the following issues:

1. Identify the organization's training policy for meeting training needs.
2. Ensure InterSystem Teams have received orientation on the training.
3. Ensure training courses prepared at the organization level are developed and maintained according to organizational standards.
4. Ensure a procedure for required training is established and used to determine whether individuals already possess the knowledge and skills required to perform in their designated area.
5. Ensure measurements are made and used to determine the status of training activities.
6. Ensure that training activities are reviewed on a periodic basis.
7. Ensure the training is independently evaluated on a periodic basis for consistency with, and relevance to, the organization's needs.
8. Ensure the training activities and work products are reviewed and/or audited and the results are reported.
9. Ensure training records are properly maintained.

Prepare a draft Training Plan that describes the training and at a minimum addresses the following issues.

1. Identifies personnel to be trained. Review the list of trainees with the system owner and users to ensure that all personnel who should receive training have been identified.
2. Defines the overall approach to training and the required training courses.
3. Establishes the scope of the training needed for users, management, operations, and maintenance personnel.
4. Defines how and when training will be conducted. Specify instructor qualifications, learning objectives, and mastery or certification requirements (if applicable).
5. Identifies any skill areas for which certification is necessary or desirable. Tailor the training to the certification requirements.
6. Establishes a preliminary schedule for the training courses. The schedule must reflect training requirements and constraints outside the project. Schedule individual courses to accommodate personnel who may require training in more than one area. Identify critical paths in the training schedule such as the time period for the product's installation and conversion to production status.
7. Defines the required course(s), outlines their content and sequence, and establishes training milestones to meet transition schedules.
8. Tailors the instruction methods to the type of material being presented. Include classroom presentation, interactive computer-assisted instruction, demonstrations, individual video presentations, and hands-on experience, either live or simulated.
9. Identifies trainers who are technically knowledgeable and were involved in the design and development of the system. For projects with extensive and formal training requirements, it may be necessary to provide training for the trainers.
10. Consider availability of the following: users, system-tested software, training rooms and equipment, and the completion of system documentation and training materials.

3.5.7 Develop installation plan

The Installation Plan is prepared to specify the requirements and procedures for the full-scale installation of the developed product at the system users' work sites. The plan also addresses the installation of any hardware, software, firmware, and equipment needed to operate the product at each site. In developing an Installation Plan consider each site's requirements for continuity of operations, level of service, and the needs of the project team, users, maintenance personnel, and coordination.

Work closely with the system owner and representatives from the user sites to assure that all site-specific hardware, software, and communications installation requirements are addressed in the Installation Plan.

Ensure any special requirements are adequately documented. Place a copy of the initial Installation Plan in the Project File.

Develop an initial Installation Plan that addresses the following issues:

1. Schedule of all installation activities.
2. Items to be delivered to each installation site.
3. Number and qualifications of personnel performing installation.
4. Equipment environmental needs and installation instructions.
5. Hardware, software, firmware, tools, documentation, and space required for each installation.
6. Special requirements governing the movement of equipment to each site.

7. Communications requirements.
8. Dependencies among activities affected by installation.
9. Installation tests to assure the integrity and quality of the installed product.

3.6 Testing stage

In this stage, components are integrated and tested to determine whether the product meets predetermined functionality, performance, quality, interface, and security requirements. Once the product is fully integrated, system testing is conducted to validate that the product will operate in its intended environment, satisfies all user requirements, and is supported with complete and accurate operating documentation. User Acceptance Testing (UAT) follows System Testing, and requests-accepts feedback from users to make any final adjustments to the system before releasing the product for implementation.

Refer to the Testing Process Manual for more information regarding testing.

The high-level activities for this stage are:

1. Conduct integration testing.
2. Conduct system testing.
3. Conduct user acceptance testing.

The output work products for this stage are:

1. Project coordination plan [revised]
2. Maintenance plan [revised]
3. Requirements traceability matrix [final]
4. Conversion plan [revised]
5. Test type approach and reports [final]
6. Test cases [final]
7. Transition plan [revised]
8. Installation plan [final]
9. Training plan [final]
10. Operating documentation [final]
    A. Users manual
    B. Developer's reference manual

3.6.1 Testing

Testing activities focus on interfaces between and among components of the product, such as functional correctness, system stability, overall system operability, system control, and system performance requirements (e.g., reliability, maintainability, and availability). Testing performed incrementally provides feedback on quality, errors, and design weaknesses early in the integration process.

3.6.2 Conduct integration testing

Integration testing is the first activity in the Testing Stage and requires special attention to preparation. The Pre-Acceptance Checklist, Integration and System Test Checklist, and Testing Package Checklist each provide the necessary steps for their preparation.

During integration, the components constructed by the development personnel, vendors, and reusable code or modules obtained from other sources are assembled into one product. Each assembly is tested in a systematic manner in accordance with the Integration Section of the Test Plan. An incremental approach to integration enables verification that as each new component is integrated, it continues to function as designed and both the component and the integrated product satisfy their assigned requirements.

Given the incremental nature of the Testing Stage, completion and sign-off of the Integration Section of the Integration and System Testing Checklist is required prior to moving on to System Testing.

Refer to the Testing Process Manual for more information regarding integration testing.

Each requirement identified in the Requirements Specification must be tested during integration testing. This traceability ensures that the product will satisfy all of the requirements and will not include inappropriate or extraneous functionality. Expand the Requirements Traceability Matrix developed in the Requirements Definition Stage to relate the integration test to the requirements. Place a copy of the expanded matrix in the Project File.

At the completion of each level of integration testing, a test report is written. The report documents test results and lists any discrepancies that must be resolved before the tested components can be used as the foundation for another integration level. Place a copy of all integration test materials in the Project Test File.

A final test report is generated at the completion of integration testing indicating any unresolved difficulties that require management attention. Place a copy of the final Integration Test Report in the Project File.

Sign-off of the Integration section of the Integration and System Checklist signifies completion of the Integration Testing activities.

A formal reporting system by which detected errors and discrepancies are recorded and fully described is recommended. These reports will help to confirm that all known errors are fixed before delivery of the completed product. Error reports also help to trace multiple instances of the same error or anomalous behavior, so that error correction and prevention assignments can be implemented. The Quality Assurance representative assigned to the project can provide assistance in developing and using an error reporting/tracking system.

3.6.2.1 Integration testing

Integration testing is a formal procedure that must be carefully planned and coordinated with the completion dates of the unit-tested modules. Integration testing begins with a structure where called sub-elements are simulated by stubs. A stub is a simplified program or dummy module designed to provide the response (or one of the responses) that would be provided by the real sub-element. A stub allows testing of calling program control and interface correctness. Stubs are replaced by unit-tested modules or builds as integration testing proceeds. This process continues one element at a time until the entire system has been integrated and tested.

Integration testing may be performed using "bottom up" or "top down" techniques. Most integration test plans make use of both bottom-up and top-down techniques. Scheduling constraints and the need for parallel testing will affect the test approach.

The bottom-up approach incorporates one or more modules into a build; tests the build; and then integrates the build into the structure. The build normally comprises a set of modules that perform a major function of the system. Initially, the function may be represented by a stub that is replaced when the build is integrated.

In the top-down approach, individual stubs are replaced so that the top-level control is tested first, followed by stub replacements that move downward in the structure. Using top-down integration, all modules that comprise a major function are integrated, thereby allowing an operational function to be demonstrated prior to completion of the entire system.

3.6.3 Conduct system testing

A.k.a., Conduct system verification testing.

During system testing, the completely integrated product is tested to validate that the product meets all requirements. System properties and the functional accuracy of logic and numerical calculations are verified under a variety of possible conditions (e.g., both normal and high-load conditions). All operating documents are verified for completeness and accuracy.

System testing is conducted on the system test bed using the methodology and test cases described in the System Test Requirements section of the Requirements Specification document. The system test environment should be as close as possible to the actual production system environment. Either the project team or an independent test team conducts system testing to assure that the system performs as expected and that each function executes without error. The results of each test are recorded and upon completion included as part of the project test documentation.

Note that regression testing is a critical aspect of system testing. It is performed in order to verify that system modifications have not caused unintended effects and that the software or system component still complies with its specified requirements.

When errors are discovered, they should be reviewed by the test team leader to determine the severity and necessary subsequent action. If appropriate, minor problems can be corrected and regression tested by the project team developers within the time frame allotted for the system test. Any corrections or changes to the product must be controlled under configuration management. Major problems may be cause to suspend or terminate the system test, which should then be rescheduled to begin after all of the problems are resolved.

Users may be encouraged to participate in the system tests to gain their confidence in the product and to receive an early indication of any problems from the user's perspective. Inform users that errors and discrepancies may occur during testing and explain the error correction, configuration management, and retest processes.

Refer to the Testing Process Manual for more information regarding system testing.

Review the draft versions of the operating documents, Training Plan, and Installation Plan. Update the documents as needed. Deliver the final versions of the operating documents, Training Plan, and Installation Plan to the system owner and user for review and approval. Place a copy of the approved documents in the Project File.

Place a copy of all system test materials (e.g., inputs, outputs, results, and error logs) in the Project Test File.

Sign-off of the Integration and System Testing Checklist and the Pre-Acceptance Checklist signifies completion of the System Testing activities.

Generate a test report at the conclusion of the system test process. The report documents the system test results and lists any discrepancies that must be resolved before the software product is ready for acceptance testing. Place a copy of the report in the Project File.

3.6.4 Conduct user acceptance testing

A.k.a., Conduct user validation testing.

Acceptance of a delivered product is the ultimate objective of a development project. Acceptance testing is used to demonstrate the product's compliance with the system owner's requirements and acceptance criteria.

At the system user's discretion, acceptance testing may be performed by the project team, by the system owner and users with support from the project team, or by an independent verification and validation team. Whenever possible, users should participate in acceptance testing to assure that the product meets the users' needs and expectations. All acceptance test activities should be coordinated with the system user(s), operations personnel, and other affected organizations.

Acceptance testing is conducted in the test environment using acceptance test data and test procedures established in the Acceptance Test Requirements section of the Requirements Specification. Testing is designed to determine whether the product meets functional, performance, and operational requirements. If acceptance testing is conducted on an incremental release basis, the testing for each release should focus on the capabilities of the new release while verifying the correct operation of the requirements incorporated in the previous release.

If the project team is not conducting the User Acceptance Test (UAT), training may be required for the personnel performing the testing. The acceptance test participants and their experience with the product and the operating environment should have been identified in the Acceptance Test Requirements within the Requirements Specification.

Acceptance testing usually covers the same requirements as the system test. Acceptance testing may cover additional requirements that are unique to the operational environment. The results of each test should be recorded and included as part of the project test documentation.

UAT is typically the final phase in a software development process in which the software is given to the intended audience to be tested for functionality. UAT is done by making the software available for testing by an in-house testing panel comprised of users who would be using the product in real-world applications. UAT is done in order to get feedback from users to make any final adjustments to the programming before releasing the product to the intended user community.

The level of training will depend on the testers' familiarity with the product and the platform on which the product will run. The advantage of having users acceptance test the product is that they are the experts most familiar with the information flow and how the product works.

It is recommended that the operating documents and other test materials be distributed to the test team prior to the actual start of the acceptance test training. This will give the test team time to become familiar with the product and the test process and procedures.

Subject the test environment to strict, formal configuration control to maintain the stability of the test environment and to assure the validity of all tests. Review the acceptance test environment, including the test procedures and their sequence, with the system owner and user before starting any tests.

Testing is complete when all tests have been executed correctly. If one or more tests fail, problems are documented, corrected, and retested. If the failure is significant, the acceptance test process may be halted until the problem is corrected.

In order to complete the testing process the following additional elements must be complete:

1. Completion and sign-off of User Acceptance Testing is required prior to moving on to the Implementation Stage.
2. Refer to the Testing Process Manual for more information regarding user acceptance testing.
3. Sign-off of the User Acceptance Checklist and the Testing Package Checklist signifies completion of the Testing Stage.
4. Prepare a formal Acceptance Test Report. Summarize the test procedures executed, any problems detected and corrected, and the projected schedule for correcting any open problem reports.
5. Place a copy of all acceptance test materials in the Project Test File.

3.7 *Implementation stage*

Implementation of the product is initiated after all testing has been successfully completed. This stage involves the activities required to finalize the install (or conversion) the system and activate the system's operation. The activities associated with this stage should be performed each time the system is installed at a site.

User training may be required to complete the implementation process. A description of the training necessary for developers, testers, users, and operations staff is provided in the Training Plan.

The high-level activities for this stage are:

1. Perform installation activities.
2. Conduct installation tests.
3. Transition to operational status.

The output work products for this stage are:

1. Project coordination plan [final].
2. Maintenance plan [final].
3. Conversion plan [final].
4. Transition plan [final].
5. Installation test materials [final].
6. Operating documents.
7. Operating system.

3.7.1 Perform installation activities

The installation process involves installing, loading, copying, or migrating the system to the production platform and the provision of operating documentation and other support materials at each site.

At each installation site, inspect the facility to assure that site preparation is complete and in accordance with the Installation Plan. Initiate any actions that are needed to complete the preparations. Conduct an inventory of all vendor provided hardware, software, firmware, and communications equipment.

Follow the procedure specified in the Installation Plan when installing. Monitor all installation activities including those performed by vendors.

Use the following procedure to perform the installation activities.

1. Coordinate the installation with the system users, operations personnel, and other affected organizations.
2. Ensure that any necessary modifications to the

physical installation environment are completed.
3. Inventory and test the hardware that will support the product. This inventory should be performed in advance of the planned installation date to allow time for missing hardware to be obtained and malfunctioning equipment to be replaced or repaired.
4. If the product requires an initial data load or data conversion, install and execute the tested programs to perform this process.
5. If the product requires, then install the software product onto the hardware platform.

3.7.2 Conduct installation tests

Ensure the integrity and quality of the installed product by executing the installation tests defined in the Installation Plan. Testing is performed to verify that the product has been properly installed and is fully operational and in production.

The installation test(s) are designed to validate all functions of the product and should specify a standard set of test results and tolerances. If the product being installed is a modification to an existing system, all remaining functions that may be affected by the new product should be tested.

Document any problems and identify corrective action. Select a diagnostic package that will pinpoint problems quickly and allow for timely corrections. Retest all equipment and software after a repair, replacement, or modification.

When installation is complete, rerun a portion or all of the system test and dry-run the acceptance test procedures to verify correct operation of the product.

Place a copy of all Installation Test materials in the Project File.

3.8 Transition to operational status

The transition of the product to full operational status begins after the formal acceptance by the system owner. Use the procedures described in the Transition Plan to implement the transition processes. Conduct or support stress tests and other operational tests. Determine product tolerances to adverse conditions, failure modes, recovery methods, and specification margins. Complete any training and certification activities. Ensure that support to be provided by contractors begins as planned.

The project team is usually expected to provide operational and technical support during the transition. Identify project team personnel with a comprehensive understanding of the product who can provide assistance in the areas of installation and maintenance, test, and documentation of changes. Technical support may involve the analysis of problems in components and operational procedures, the analysis of potential enhancements.

Transition to full operational status should be an event-oriented process that is not complete until all transition activities have been successfully performed. Withdraw the support of the project team personnel in a gradual sequence to ensure the smooth operation of, and user confidence in, the product.

All Project File materials, operating documents, a list of any planned enhancements, and other pertinent records should be turned over to the maintenance staff. Access rules must be modified to provide access to the product by the maintenance staff and to remove access by the project team and other temporary user accesses. Programs, files, and other support software should be in the production library and deleted from the test library, where appropriate.

For major systems involving multiple organizations and interfaces with other systems, a formal announcement of the transition to production is recommended. The announcement should be distributed to all affected groups. The names and contact details of the team should be included.

The system is transitioned into operational status. Project File materials, operating documents, and other pertinent records are turned over to the maintenance staff.[Engineering] Systems design

System(s) is a word that takes on relatively distinct meanings in different contexts. In the context of design, a system can be defined as an emergent or designed network of interconnected functions that fulfil an intended unit of satisfaction (system outcome or result). Additionally, system(s) has been described as a holistic, embodied way of thinking about reality. Accordingly, the term system(s) represents both a way of inquiry and an object of inquiry. In the engineering context, system(s) embodies both a way of designing and an object of design.

> **NOTE:** *The socio-technical perception (or, nature) of reality assumes that the real-world comprises systems that can be 'designed'. Therefore, it implies that models of those systems can be made and their behavior can be simulated.*

Human-oriented design is a unique form of inquiry and action that aims to create and transform systems to fulfil human needs. Therein, systems thinking provides a base (approach) for synthesizing knowledge of how humans may live optimally together in a common ecology.

Here, a systems design approach refers to the mental model (or, approach methodology) through which designers "frame" (understand and construct) the world, sometimes referred to as a perspective or "paradigm". Systems design is an approach that guides designers in their visualization and resolution of complex problem situations. Systems design is an approach to creating better systems for humankind.

Both the need to support increasing changes in the scale of the challenges facing the development of society's infrastructure and resource limitations, have led to the emergence of a common and unified field of 'system' design. The implementation of a systems approach to societal [engineering] design is optimal, and its result is, the visual-materialization of the context of a socio-technical network of habitat infrastructures. These habitat infrastructures are designed and materially developed. Herein, the systems design approach seeks the integration and unification of all human-oriented information in order to pre-determine an optimal structural re-orientation for the next iteration of the societal system. This "learning" and consciously-integrating (i.e., emerging) approaches necessarily recognizes the need for a unified societal perspective that considers the capacity of [common] design to improve everyone's well-being by meeting (completing, fulfilling), currently, everyone's basic and full opportunity needs of existing generations, while sustainable [habitat] 'construction' for future generations.

Obviously, all 'human' groups must be open to contribute to the whole system design (WSD to be executed), the service-product 'habitat' system (SPS/PSS).

From a systems thinking perspective, problem solvers are the whole individual societal system, which is a networked community of human conscious-organismal entities. Community, humanity in this context, may obviously, and only solve its global societal problem situations by identifying and reasoning ("discussing" and "conversationling") at length the relationship between design and the materialization. That designed system, which is designed to be commonly optimal for all, must highlight openly difference in social relations of information and power in order to optimize a system that is commonly fulfilling for all, given all that is known. The distribution of information from 'unified' (commonly open to all) to 'centric' (power-over-others) may be visualizes as an information-based socio-technical system. There is the composition and decomposition of information, the discovery of available information (i.e., search), the controlled inquiry into new reality information (i.e., consciousness sciences), computation of data (i.e., hardware-software, interface-conditional programming). Systems engineering is a method of designing systems. Logic, as expressed by consciousness, is otherwise known as, critical thinking. Logic, at the individual (and hence, societal level) is necessary to understand the idea that an information system structures everything experienced, and that experiential objectives can oriented the structure of the next iteration of the societal system. Critical thinking is necessary to develop technology and social organization is necessary to enable its socially-effective and full application as an optimized habitat system. A unified approach necessitates integration of the 'positive' approach of extensionable compassion, and the revealing 'negative' approach of socially visualizing an open network of power, control, domination, and oppression, to reduce social information and spatial sets that reduce optimal and common, individual human-fulfillment.

3.9 *Whole systems design*

Whole system(s) design is a collaborative and integrative approach that enables a common (i.e., collective) response to socio-technical (i.e., complex real-world) problems). Whole systems design is required for solutions that scale optimally for all of humanity. Through a systems approach, designers take social and technical [parallel] decisions on what systems methodologies and design tools to use, based on their unified understanding of each problem situation.

> **INSIGHT:** *Society is a whole system, and its engineering needs to be re-solved as a whole [integrated] system.*

Note that in software systems, the whole system(s) design uses conditional programming (i.e., procedural software, learning principles) to produce holistic solutions (i.e., to produce solutions that account for the whole of humanity and the environmental ecology).

A community project-systems design approach is a co-participatory approach to every [human-involved] problem situation, where solutions should not be imposed. Rather, stakeholders should be empowered to understand and participate in the functioning of the system (Note: this idea is expounded upon in the Lifestyle System Specification). Moreover, stakeholders actively participate in the conceptualization and implementation of the iteratively ("newly") designed societal system.

In the market, however, there is great confusion over the application of the systems methodology. Some of the "stakeholders" are non-existent entities "institutions, market enterprises (businesses; market coercion), and government enterprises (State coercion; states where 'leaders' control populations through relationships of power-over-others, rather than, coordination for all); which, together form the idea commonly referred to as the 'structural violence', as the common experience of most of modern 21st century society. Layers of confusion and abstraction will limit simplex, higher-system [synthetic] thought (i.e., the perceive that the optimal is to cooperate in the moment toward the fulfillment of all individuals for one's own fulfillment in common with those whom share its conscious, experiential [cosmic] environment.

3.9.1 Living systems design

Living systems design involves ways in which a designer can look at the patterns and life principles that are found within living systems that humanity operates in, and then, apply these patterns and principles to the products, and processes. Living systems design has different names depending upon the disclipine, including biomimicry, permaculture, closed-loop economics, circular economy, and waste equals foods.

3.10 Systems-oriented design

Systems-oriented design (SOD) is an applied knowledge-based (i.e., skills, lifestyle training) approach intended to develop better designs, visualizations, and systems practices. Systems-oriented design, as a holistic approach, has a requirement for a project-based information set, because it accounts for the design of executable interactions in time (in a real-, spatial-world). It must consider different network types and boundaries within a particular socio-technical system to ensure the system functions for all common individual human needs. Systems-oriented design exists as a tool to design of a coherent combination of processes and service-product (or product-service) combinations that together can fulfil the function of the system as specified by humanity in common.

The core design output of a human systems-oriented design is the generation of socio-technical models that are large and information-dense diagrams that act as a bridge between inquiry and design. These models, are visualization maps used to synthesize and interrelate knowledge, and they become a commonly shared understanding of the system among stakeholders.

3.11 What is human systems engineering?

A.k.a., Human system integration (HSI), human engineering [criteria], human engineering standards, human systems integration, human factors, human ergonomics, user-centered design.

Human systems engineering is the process of developing and operating a socio-technical system expressed as a habitat service system and composed of a real world information model that iteratively resolves a higher potential for everyone, given the ability to become more knowable. Human systems engineering is the engineering of systems to meet human requirements (i.e., human needs). Human systems engineering integrates an understanding of human capabilities and human needs into a systems design using an iterative model of systems engineering development.

Human systems engineering (a.k.a., human systems integration) is a structured systems approach to the designing and development socio-technical systems that will involve humans, ensuring alignment of the final system with their requirements, capabilities, and limitations. When perceived from a life cycle viewpoint, human systems engineering is the activities involved throughout the system life cycle that address the human element of system design (one of the first international technical standards for this idea is IEEE 1220-1994, 1998). In other words, human system engineering creates socio-technical life-cycling systems that have the potential to function effectively for humans.

Human systems integration (HIS, NASA terminology circa. 2010) emphasizes human considerations (requirements) as a/the top priority in systems design to reduce life cycle issues and optimize system performance and usability when humans are present. Essentially, human systems integration is the relationship between humans and their environment – particularly how systems are designed and used relative to that relationship – with the goal of ensuring a safe and effective environment that meets human requirements.

Human systems engineering is a comprehensive engineering methodology for integrating human requirements as part of an overall system solution. The goal of human systems engineering is to optimize the total system performance by accounting for both the human and technological components, and their integration.

Human systems engineering starts with an accurate representation of human needs/requirements, which allows for the development of an effective system (i.e., a system that effectively meets those inputs in its operation). Human systems engineering provides the potential for optimizing the interface between the human and his/her environment or work processes.

Similarly, human factors engineering is the application of information on physical and psycho-sociological characteristics (as requirements) of humans to the design of devices and systems for human use. Note that the terminology here can be confusing, because it could be said that simply accounting for 'human factors' (a.k.a., human requirements) in engineering is human systems engineering, and thus, there is no need for a special label when the approach is unified. And, the approach is necessarily unified when engineering a unified societal service system for humankind.

The term 'social engineering' is associated with many negative connotations. In common parlance, social engineering refers to the design and influence of social organization and social behavior. It brings up visions of advertising, propaganda, manipulation, and scamming. These associations with the term social engineering are applicable under market-based conditions, but may not be applicable to other societal types. Note that terms like social science and systems science are also used when applying the systems approach to social systems.

There is a substantial body of knowledge in both human factors, ergonomics, performance, and usability demonstrating how user-centered design can be organized and applied effectively.

Human systems integration design criteria, principles, and practices for standards:

1. Improving performance of personnel (users).
2. Enhance the usability, safety, acceptability, and affordability of technology and systems.
3. Achieve the required reliability and productivity of personnel-equipment combinations.

Human system integration design engineering (human-centered design):

1. Understand the user and environment.
2. Develop concept of operation.
3. Allocate function between user and system.
4. Perform user task analysis.
5. Conduct requirements analysis.
6. Visualize and produce design solutions.
7. Evaluate designs and iterate solutions.

Areas of technical expertise necessary for proper HSI include, but are not limited to:

1. Human factors and human engineering (including crew workload and usability, human-in-the-loop evaluation, and human error analysis).
2. Crew health and countermeasures.
3. Environmental health (including radiation, toxicology, and other areas).
4. Safety.
5. Systems engineering.
6. Architecture.
7. Crew functions and habitability functions (including nutrition, acoustics, water quality and quantity, etc.)
8. Crew interfaces and information management.
9. Maintenance and housekeeping.
10. Ground maintenance and assembly.
11. Extravehicular activity physiology.
12. Mission operations.
13. Training.

User-centered design (UCD) is a well-established design approach that concentrates on developing usable systems by focusing on the system users, their needs, and requirements. The approach applies principles of human factors and ergonomics, as well as usability knowledge and techniques.

The goals of the user-centered design approach are to:

1. Enhance effectiveness and efficiency.
    A. Improve human well-being.
    B. Increase user satisfaction.
    C. Improve accessibility and sustainability.
    D. Counteract possible adverse effects of use on human health, safety, and performance.

3.11.1 User-centered design

User-centered design is formalized by multiple standards and standards setting bodies:

- ISO 9241-210 - provides requirements and recommendations for user-centered design principles and activities throughout the life cycle of computer-based interactive systems.
- ISO/IEC TR 25060 - describes a potential family of International Standards, named the Common Industry Formats (CIF), that document the specification and evaluation of the usability of interactive systems. The Technical Report focuses on documenting design and development elements of usable systems. It does not prescribe a specific process and is intended for use with ISO 9241 standards.
- ISO/IEC 25062 - standardizes the types of information captured with user testing. The level of detail allows the same or another organization to replicate the test procedure. Major variables include: user demographics, task descriptions, test context (including the equipment used, the testing environment, and the participant and test administrator's interaction protocol), and the metrics chosen to code the study findings.
- NISTIR 7889/7990/7934 - Human Engineering Design Criteria Standards
- MIL-HDBK-759C (07/31/1995) - Handbook for Human Engineering Design Guidelines
- ISO 9241 (06/01/1997)1 - human centered design and human-computer interaction
- Section 508 of the Rehabilitation Act of 1973 (08/07/1998)2 - accessibility by those with disabilities
- ISO/IEC TR 25060 (09/01/2006) - Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE)
- Ministry of Defence Standard 00-250 (05/23/2008) - Human Factors (HF) and Human Factors Integration (HFI) requirements
- NASA/SP-2010-3407 (01/27/2010) - Human Integration Design Handbook (HIDH)
- ISO/IEC 25062 (07/15/2010) - Software engineering – Software product Quality Requirement and Evaluation (SQuaRE)
- MIL-STD-1472G (01/11/2012) - human engineering design criteria, principles and practices
- ASTM F1166 (06/28/2011) - the design and evaluation of human-machine interfaces

3.12 Service product design

A.k.a., Product–service systems (PSSs), or more accurately, service-product systems (SPSs).

In the literature, the integration of product and services is most often called a PSS. Naturally, a human designed (and oriented) 'habitat' system exists to provide functions that fulfil human needs through service, and then product, combinations. Obviously, this conceptualization is found across different "professional-market" disciplines, such as Operational Research, Information Systems, Systems Engineering, Software-Hardware Systems, Politics, Business Management, Marketing, and Self-development.

A systems thinking perspective on SPS/PSS is fundamental for a commonly aligned conceptualization and in-depth understanding of the socio-material system as it is currently in place, and simultaneously, possibly in place in the future. SPS/PSS design is an effective form of conceptualization of complex societal systems.

Through the understanding that all types of 'human' society represent an understandable and unifiable societal information systems, that will express for that human-society an observable decisioning-materializing system. That system can be visualized before it is generated, a process necessarily open to every individual, if the orientation of the next societal generation is to be a more optimal form of free and open access system as the desired design result. SPS/PSP is as a tool for analyzing and synthesizing (integrating) causal loops (e.g., systemic relationships, procedural decisioning) among community users and operating designers, among a unified habitat. Computationally, SPS/PSS simulates the dynamic behavior of systems quantitatively, because it is a visualization of a materializing system (i.e., a system that is sensibly quantifiable; i.e., can be expressed in numeric pattern, fractally). The SPS/PSS is the operational system; it is the user-interface conceptualized as a socio-technical 'societal' system.

3.13 Engineering service operations

Engineering operations is the application of knowledge and technical design to fulfill a requirement formulated as a problem. In order to solve a problem in engineering, the cause (or "root") of the problem, and its context, must be understood. The result of the process of engineering is the construction and/or continued operation (or recovery operation) of a technically existent (and experiencable) [societal] system; hence, the dual life-cycle phases of a **unified engineering approach** (with construction and operation information sets):

1. **Construction (and Re-construction) of system to specification:**
   A. Design feedback integration.
   B. Construction preparation.
   C. Construction (building structures, building sub-structures, and equipping).
2. **Operation of system at specification:**
   A. Operation preparation.
   B. Operation (automation and/or human event involvement).
   C. Monitoring (quality assurance and operational feedback).
   D. **Deconstruction of system to specification:**
      1. Deconstruction preparation.
      2. Deconstruction.
      3. Re-integration.
      4. Monitoring (quality assurance and operational feedback).

A clarification must be made here. Take, for example, a chef and a waitress. Each is equally maintaining and operating a service system. A mother feeding a child is maintaining and operating a service system [for the child]. Someone feeding themselves through food acquired anywhere is operating (and maintaining, or not) a service system. There is a constructed and iterating foundation to the system that structures the operation and maintenance of fulfillment to humans (Read: that which humans really do require, and have specified) as a set of societal-level requirements traced to their societal-level [support] services (a.k.a., the Service Systems: life-support service, decision service, facility service systems, etc.).

Service activities within the service system(s) may be automated and/or maintain human involvement, where desired(/-able) to humans. That which is desirable to humans is an objective to which humans may (or may not) align the next iteration of their societal-life system (society, civilization). In the social information set of the real world information model, 'objectives' are [expressed] 'values' to which decisions are [objectively] aligned or not.

3.13.1.1 The determination of a societal [service] system design structure

The design and the service system of which the design is a part, is engineered into existence and continued operation as a solution to the requirements that humans have set for its continued operation.

The design of a real [world, societal-level] system:

1. **The Unified [Societal] Information System:** social; decision; lifestyle; material. Social processes; decision processes; lifestyle processes; material processes.
   A. Open source collaborative development effort.
   B. Common and InterSystem Team effort.
2. **The Physical [Societal] System:** Actualized configuration in formation of a Habitat service system.
   A. Open source collaborative development effort.
   B. InterSystem Team effort.

A real-world, societal-level system may sub-composed of habitat service sub-systems (at the material level of experience of a conscious user). The design, generation and otherwise execution upon and within of these societal systems may be aligned (or not) to explicit[ly human] requirements (or not).

The materialized instantiation of these [required] habitat service systems necessitates the three 'material' information categories of materialization (other conceptions of which include: construction, creation, generation, etc):

1. **Specification**, which involves conceptual through

to physicalizable designs of a system that meet the requirements. The specifications are a set of visualizations forming a conception through to technicalization of a system, using tools and techniques (i.e., processes).

A. Knowledge encoded by engineering information groups becomes visualized and tested.

2. **Operation (construction and de-operation)**, which involves a set of activities (operations, tasks performed in a [pre-]controlled structure) necessary for the materialization (creation and re-creation) of the system.
   A. **A non-prior system** will have its first constructed instantiation, and then life-cycle thereform.
   B. **A prior existing system** will have a module life-cycle. A set of unique operations exist for each habitat service system. All systems provide material and informational services directly and indirectly to humans. Indirectly means that the service is that of the lifecycle of the systems themselves.
3. **Validation to specification**, which involves a set of validation activity **tests for alignment** with requirements, imperatives, and the user. Here, experience generates feedback and revised integration directionally-intentionally evolves the system, or generates an entirely new one to replace the last. The design of a new system, communicated as a delivered specification, must be validated to sufficiently meet operational expectations as defined by the user requirements.
   A. Without validation there is no feedback valuable for alignment, and without feedback valuable for alignment, there is no re-alignment to a direction set by an objective.

4 [Engineering] Design and development

NOTE: *If design is political, then debating and obfuscation are design skills. Yet, neither obfuscation nor debating facilitate functional design. Design is not political.*

The determination of a societal-level solution is, in part, through a common engineering design and development process. Design through to execution becomes the fundamental engineering [development] activity (process). The engineering of systems revolves around the problem solving of design. Design consists of a sequence of stages starting from the perception of need and terminating in a final (end/firm) description of a particular design configuration. Each stage is in itself a design process and is an iterative sequence of sets. Design fills the gap, the difference or separation between what is to be done (and why), and how to do it.

Both objects and processes (given an environment) can be designed. The societal information system contains both information objects as well as information processes, given an information-based environment. Therein, the habitat service system is a combination of both objects and processes with material reference. For example, the energy sub-system is a supra-process life-support category, containing object-assets cycled through materiality as part of service system processes to meet human energy [operational] requirements. The energy sub-system delivers service (asset-object) types, including power generation and storage systems (e.g., wind turbine electric generators and batteries).

NOTE: *Since engineering quality is only as precise as its tools, the quality of the design tools (e.g., representations, conventions, and applications) has a direct impact on the quality of the result.*

Engineering design, as an activity, produces an "engineering design" as a product, which is delivered to be acted upon and through. This information product is a representation of the physical product to be produced and/or operated (and is variously called the designed service object, system model, product model, engineering drawing, etc.).

A design is a solution for a given set of problems. A design is a solution towards a problem experienced by a real user; the design solely relies on the "user" for it to be appreciated. In other words, designing is the process of problem solving.

It's not a canvas for personal expression. Design is not personal, like someone's preference in music or art. A quality designer takes design decisions based on user research, knowledge, and best practices, with a focus on communicating clearly and achieving human goals. The process of design must be complementary with the objectives. This means the design and implementation process is critical. If flexibility and participation are the objectives of an organization's design, then the question must be asked, how might an organization be designed so that it is flexible, interactive and participatory.

Design is a continuous commitment, a re-iterative process. A design is a solution, which inevitably has to be changed, therefore it is critical to build learning and change ability into the organization that produces designs. In concern to feedback on designs, feedback should be linked to goals. Designers are tasked with generating creative and unique solutions, following a process that builds upon logic, observation, knowledge, and feedback to arrive at an optimal output.

> **APHORISM:** *"A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away."*
> *-Antoine de Saint-Exupéry*

The quality of any engineering design - whether it's a commercial product or a data model - is a direct function of the ability of the design system to access and codify the knowledge of the users, and systematically translate that knowledge into a model of the desired product/ system.

> **NOTE:** *It is the design element in the practice of engineering development that distinguishes engineering as an activity from the sciences.*

In design, engineering is a:

1. Deliverable (noun): The production of a real-world, hardware (object) and/or software (concept) system.
   A. A design describes a deliverable (noun, object). A design is a visualization that shows the final object/system.
2. Process (verb): The decisioning processes that determines the function, performance, parameters, interface, etc., to be delivered and used/operated.
   B. A plan describes how the deliverable was designed and materialized. A design is a plan that shows how the final object/system will be created and operated.

Engineering service design involves, at least:

1. Service concept.
2. Services.
3. Processes.
4. Tasks.
5. Roles & technologies.
6. Equipment and computation.
7. Resources and conscious motive.

Engineering design is:

1. A verb, a logical sequence of activities and decisions that transforms an operational need into a description of system performance parameters and an optimal system configuration.
2. A comprehensive, iterative and recursive problem solving process, applied to transform needs (and requirements) into a new system (or system state).
3. A standardized, disciplined process for the development of system solutions that provide a system that meets user needs in an environment of uncertainty.
4. The process of selecting the means and contriving the elements, steps and procedures for producing what will adequately satisfy some need.
5. Design is founded on the consistent, directed resolution of a system into reality. Design is founded on decisioning.
6. To create order, structure and/or pattern as an outcome [of the process of designing]. Crucially, it is the order, structure and patterns in design actions that are the source of these attributes of design. Design, even in nature, is not an outcome of anything other than a highly structured causative sequence of actions (intentional or not).
7. A problem-solving activity. Wherein, problem solving is that form of activity (or action) in which an organism intends to realize a goal, a gap in the 'route' to the goal, and a set of alternative mans, none of which are immediately and obviously suitable. It is a path of resolution followed in conformity with the guiding criteria of a goal (the user requirements/user needs/intervention intention) subject to the constraints of viability (the opportunities and reality of implementation technologies). Design is action requiring the mind to examine (process) each and every item (of information), which pertains to the design in a continuous and uninterrupted process, including all of these in an adequate and orderly enumeration.
8. A resolution arising out of a sequence or iteration of process transformations or work products outputs. Engineering design is a matter of recursion that resolves transformations and work products definition at different scales.

The design process is repeated sequentially in a number of stages, proceeding from a global view of the system, to progressively more localized considerations, and from an abstract and fluid description to a concrete, physicalizable one. Therein, abstraction forms functional descriptions and material detail forms implementation descriptions.

> **NOTE:** *To build and test is to construct the whole from the parts (to piece parts together, to join into one that operates together within a boundary).*

The processes of design build models of that realisation and future reality, as descriptions of:

1. Intended intervention.
2. The function of some intervening agent.
3. The physical composition and ordering of that agent.

These models are progressively resolved, each with the other according to ever greater detail, until the risks of achieving a viable and valid outcome diminish to an inconsequential level. Thus, design resolution is often recursive in practice.

The concordant resolution of models follows a sequential decision-resolution path; one that is continually revisiting different levels of modeling detail, past decisions and preceding lines of decision rationale.

The design resolution path forms in linear time from a process of execution concurrency relating to the following three information sets:

1. Structural detail (as system architecture - function, form, and effect domains). Elemental connectivity must exist between all domains.
   A. Design rational (logical domain).
2. Solution progression (as organizational processes and work products - material, energy, information domains).

Both objects and processes can be designed. For example, the habitat service sub-system is a combination of both designed objects and processes. Therein, the energy sub-system is a supra-process life-support category, containing object-assets cycled through materiality as part of service system processes (Read: the operational processes). The energy sub-system delivers asset-object types including power generation and storage systems (e.g., wind turbine electric generators and batteries).

Regardless of what is being designed, design involves several core information processes; design is:

1. Generative (i.e., involves creating, analytical-synthesis) of some new information set. Some thing new is the output of design.
2. Iterative (i.e., involves repeated cycles of trial, error, and learning).
3. Representational (i.e., visualizations, models, and prototypes document and communicate a design) with many potential views, given inquiry intent.
4. Collaborative (i.e., there is fulfillment in optimizing common designs for fulfillment).
5. Complex, probabilistic, and emergent.

The primary [systems] engineering tasks include:

1. Develop the total system design solution.
2. Develop and track technical information needed for decisioning.
3. Test the system.
4. Verify that technical solutions satisfy user requirements.
5. Operate the system.

4.1 The design phase

> **APHORISM:** *If you want to be a powerful creator, become good at systems [thinking] and understanding and solving structural problems.*

Design for (i.e., the common elements of the design phase are):

- Function - the "means" by which (how) the system operates for user fulfillment.
- Interface - the "means" by which (how) which two systems interact (Read: share information).
- Performance - the evaluated quality "means" by which (how in alignment):
  - Information is shared between systems (per requirements).
  - The function operates for user fulfillment (per requirements).

Service system engineering life cycle:

1. Service operation.
   A. Issue inquiry for service.
2. 2. Service integration.
   A. Organizational value-alignment inquiry.
   B. Solution engineering inquiry.

The contextual elements of use for any particular product (which can be a technology, system, device, piece of equipment, or process) include, but are not limited to:

1. The intended user(s).
2. Their goals and tasks.
3. Associated equipment.
4. The physical and social/informational environment in which the product may be used.
5. Note: A product could be viewed of a as a set of preplanned tasks.

4.1.1 Define the conceptual system [a phase]

The conceptual formation of a projected system may be initialized through a set of imperatives. Often, a sufficiently developed imperative structure is composed of all of the following:

1. **Project [societal] imperatives:**
   A. **A strategic direction**, which is a description of progress along some identified alignment with which humans seek to move or progress. A strategic direction is described by concepts.

   1. **Define the mission, vision**, and other strategic directional or outcome descriptives.
B. **Goals (human aspirations)**, which involves a set of criteria representing a list of axiomatic outcomes (conditions) that are to be realized under a given strategic direction of intention.
   1. A **societal goal** is a particular category of goals, which are universalizable to a society composed of organisms. Generally, societal goals categorically express the necessary conditions for avoidance of serious harm (survival) and the expressed facilitation of fulfillment.
C. **Needs (human needs, which are objectives)**, which involves a list representing a set of criteria that are of imperative importance to fulfill, including processes and states, for humans to not only survive, but thrive at their fullest potential.

Goals and needs are both:

1. **Measurable**, at minimum through progress on subsidiary goals/objectives, but preferably also directly. Here, measurable refers to that which is independent of personal sensitivity, capable of experience by some population with common senses.
2. **Completion** of goal and **fulfillment** of a need represents significant alignment with or progress toward the strategic direction.

A conceptual design specification includes:

1. A **conceptual specification** (or, Unit; Conceptual specification) is the design for an instance (potential/existent instantiation state) of a community-type society.
   A. The **functional specification** (Functional specification) is the Unified Societal System Specification detailing the functional elements of that societal instance.
      1. A **technical component specification** (Technical specification) is a set of sub-system iteration states.
      2. What is needed here is resources, time, financial budget, ...
      3. Feedback from experience.

4.1.1.1 Process and technology mapping

Process and technology mapping is the process of collecting and associating all configuration and connectivity parameters for hard[ware] (material) and soft[ware] (information) systems. In some disciplines, the total set of process and technology mappings for an entity is called a configuration item (CI), and configuration items are used for operations (including production/fabrication and construction/integration). Process mapping creates an image (diagram, visualization, description) of each organizational (such as, societal or business) process, and what would be needed to continue the process in the absence of any or all of its informational and material (Read: IT, information technology) resources.

In concern to system development, process and technology mapping allows easy duplication of a system. In concern to disaster recovery and system continuity, this mapping (assuming it is backed up) allows operators to re-prioritize, move, and most importantly, restore systems.

4.1.2 Define the technical system [a phase]

An engineered system, is a technical or socio-technical systems system, which is the "subject" of a systems [control] engineering life cycle. Systems engineering is the approach, involving a set of processes, which realize (materialize) a system that accomplishes, fulfills, and completes the imperatives [of the projected system].

In part, the technical system is an expressed (or, express-able) visualization.

4.1.2.1 Information visualization function(s)

An information visualization function is the description of an operation that allows information to be more coherently understood and integrated by a visualizing user. An information visualization function adds shape, an object, geometrical relationship to an information set. Here, a 'function' is an information process or information operation that allows for a clearer and larger expression of what is possibly available, because it conveys the greater -ability to connect the user to the usable system.

4.1.2.2 Traceability

Traceability is a principal information visualization objective. Traceability is the ability to describe and follow information in both forward and backward direction. Full traceability is the ability to explain and visualize the flow of information in forwards and backwards directions, fully. For example, a complete (full) visualization of requirements expresses traceability wherein a requirements statement at any level can be related to any other level, including its source (e.g., human issue, intention, problem) and destination (e.g., output, result, system).

4.1.2.3 Requirements traceability (RT)

Requirements traceability (RT) as a principal objective of project coordination is the ability to describe and follow information about [the life of] a requirement in both forward and backward directions, completely (i.e., without gaps or "jumps"). In order to integrate feedback coherently within a project, there must be traceability of outputs to inputs. Traceability assures everyone that all requirements can be accounted for in the design at any stage and that no unnecessary requirements are included (probably, unnecessary work). Traceability supports configuration control (if a requirement needs change, its related information flows and impact are visible).

Requirements traceability is a feature (quality, characteristic, attribute, objective) of a system's unified (top-down) design approach, which "guarantees" (Read: makes objectively measurable) that requirements can be identified and inquired into (satisfied) at any stage of a project.

Traceability ensures data on:

1. Where a requirement came from?
2. What requirements are related to it?
3. What requirements were derived from it?

There are sub-conceptions to traceability:

1. **Forward traceability** is required so that design decisions can be traced from any given system-level requirement down to a detail design decision.
2. **Backward traceability** means that any lower-level requirement is associated with at least one higher-level requirement.

4.1.2.4 Requirements use case

A use case (or user story) is the sequence of events to explain your requirement.

4.1.3 Modeling (visualizing-simulating) requirements as mathematical associations

There are two mathematically aligned characteristics when using the systems approach to modeling (by the engineering system) the intention of a 'requirement', or even prior, an 'issue'. The two axiomatic mathematical associations are that of variables and

1. **Variables or Non-functional requirements:** A variable is the way in which an attribute or quantity is represented or expressed. Non-functional requirements may be transposed for variables, which describe an attribute or quantity. How much durability and reliability do you want in the design of your system (its a variable option)? How much autonomy do you want in the design of your society (its also a variable option)? Technological obsolescences as a value, and then a variable, in a societal resolution equation has a different outcome than reliability as a value, and then a variable in a societal resolution equation.
2. **Constants:** Functional requirements become absolute quantities. Whereas, the functions are the parameters, normally a constant in an equation describing a model. For instance nutrition is a constant (not optional) need. In a natural environment of scarcity with an inability to design non-basic technologies, then the nutritional constants are pre-determined by scarcity in the environment and organismal sensation. Therein, human needs may be considered a constant.
   A. **Parameters or Functional requirements:** In an environment where needs may be fulfilled via designed and varied methods, then human needs (which are human requirements to engineering) are more akin, conceptually, to parameters. When the environment determines fulfillment, there is very little that can be done in terms of changing (by choice and design) fulfillment. However, when intellect, resources, and designability are present, then the fulfillment (felt and objective) is not fixed.

4.1.4 Engineered system characteristics

Systems engineering involves the processes of designing and constructing additional possible 'function' into the material and/or informational world, through a particular design. In a purposeful (purposive) context, a design[ed] system has the following kinds and sub-kinds of characteristics:

1. **Physical characteristics (physical properties, physical requirements)**: its materials, structures, and motions.
2. **Operational characteristics (operational properties, operational requirements)** - are properties that are designed into a system, and expressed in the systems operation or state of being. There are two types of operational properties:
   A. **Functional characteristics (capability requirements)**: what it is for; what service(s) it performs. These are the first type of 'ability, functional abilities or capabilities. A description of *functions* of what precisely the system will do. Functional requirement - state what the system will do. Describe how it will behave; what is its specific behavior and functions? A functional requirement is a specific need or desired behavior as seen by an external user of the system. The required capability or function must be delivered by a system through one or more of its components.
   B. **Non-Functional characteristics (a.k.a., dispositional properties and non- functional requirements)** that one possible assembly

produces over another for the same function. Non-functional requirements are the conditions under which the system should perform. These are otherwise known as 'abilities'. These are the second type of 'ability, non-functional abilities or objectives. A description of *features* of what precisely the system will do. Non-functional requirement state what the system will be. The criteria for evaluating the operation of the system, rather than specific behaviors. These requirements cannot be categorized in to function, data, or process (both process and data) requirements.

1. **Execution qualities** - Execution qualities form an interface and [critical] decision path between a user with needs and a [societal] service system that provides access to a capability that meets those need. Execution qualities are often visible during operation (at run time) of the [societal] system itself. Such as: durability, automaticity, and optimization, which are observable during operation (at run time) of the [societal] system itself.
2. **Evolution and availability qualities** - are embodied in the static structure of the system itself. Such as: testability, maintainability, extensibility, and scalability, which are embodied in the static structure of the system itself.

3. There is also, for every system, a [negative] **efficiency characteristic (efficiency requirements)** for all expressed properties: given what is known and what is possible, how off alignment from optimization of materials, structures, motions, and attributes (capability and dispositional) is the system[s design and operation]?

CLARIFICATION: *Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be.*

The overall properties of the resulting system commonly mark the difference between whether the development project has succeeded or failed.

4.1.5 Define system non-functional requirements (a.k.a., objectives orientational needs)

A.k.a., Design goals, dispositional properties, dispositions, non-functional requirements, non-functional needs, system quality requirements, system performance requirements, performance needs, qualitative requirements, objectives, system control objectives, quality objectives, quality attributes, quality goals, quality service requirements, constraints, features, and values.

Non-functional requirements (a.k.a., objectives) constrain functional requirements. Non-functional requirements specify under what constraints the functional[ly required] system should function. Objectives are the orientational component of the imperative structure. Defining non-functional-requirements is an orienting process. Objectives are orientational because they predetermine one assembly of components for a given function (or service), versus another assembly for fulfillment of the same function. In other words, a design may be categorized under a specific conceptual state (or condition) given its composition, over a design to fulfill the same function, but with a different assembly.

NOTE: *In a decision system, values are non-functional requirements, which are the social[ly viable] conditions for creation and operation of a stable human society.*

In system requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to evaluate the operation[al performance] of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behaviors or functions.

NOTE: *'Sign' is the way in which a design executes a desired function.*

Simply, objectives are top-level project requirements of a system that identify what its design **should be (non-functional requirements)**, as opposed to what the design should do **(functional requirements)**. Objectives are design goals (a.k.a., non-functional requirements) that describe the desired attributes, qualities, or features the design will have. Objectives allow for exploration of a design and decision space where an optimal selection among alternative options occurs. **Objectives**, which involves a set of criteria representing a concrete, measurable output or outcome to become a requirement for the goal's fulfillment. Embodied consciousness has a set of abilities available to it; and it can extend its abilities through a systems process to create newly available technological functions. The extending of single function can occur in multiple different ways, each expressing a different objective (dispositional property). These newly available technological functions allow us to integrate and automate our functional habitat service systems into a network of resource-access sharing for each and everyone's fulfillment.

In contrast to functional requirements, non-functional requirements are in the form of, "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. Objectives are expressions of desired attributes and behaviors that the system will express. The system must maintain some [conceptual through to actual] ability in its performance. Here, objectives specify -ability inquiries for decision (as the planned solution) selection.

Objectives are characterized as:

1. Expressed using the verb, to be. These are "be" words ("be" words include: is, am, are, was, were, be, being, becoming, been).
2. Qualities that the system (object) should have.
3. Measurable, which senses and inputs measures or sources of data for system progression (i.e., evolvement, optimization, or betterment), whether quantitative, qualitative, or both.
4. Logically relevant to the applicable goal.

In order to bring a material system into existence, there is an existent material reality that must be worked with and through. The system which is to be brought into existence is composed of a set of requirements. There are relationships between requirements and the extant material reality. In common parlance, these relationships are called tradeoffs. Objectives are concepts that are encoded into the design, and eventual operation, of the system in order to resolve these relationships toward some particular alignment. These concepts are dispositional properties (-abilities) designed into the system, in expectation of expressing a particular function during the systems lifespan. Here, disposition refers to the arrangement of material reality into a system expressive of a particular technical function, previously conceived of as a objective or dispositional property. Material systems can be arranged in different ways to perform the same service. Dispositional properties prescribe (on the design front-end) and describe (on the operational user-end) the functional expression of a material systems arrangement.

Objectives are expressed as 'abilities' (a.k.a., concepts of operation). An 'ability' is the capacity to form and resolve a process in a categorically pre-defined manner, a dispositional property -- a category of technical action. Specific categories of 'ability' are labelled with that term as a suffix. In other words, an objective defines the pre-defined manner of desired functioning.

An objective/requirement represents a measure of specified change, in order to bring about the achievement of the goal. The attainment of each goal may require a number of objectives to be reached. There is often much confusion between goals and objectives. Whereas as a goal is a description of a destination, an objective is a measure.

Systems engineering depends on the ability to perceive the [multiple] possibilities for action within a environment, so that a system's movement within the environment can be appropriately coordinated. Herein, objectives design *time* into *being*, which is the fundamental principle of concurrent engineering. There are relationships between a designed system, its environment, and the concepts applied to its assembled functioning.

Objectives enable the selection among design alternatives [for the one(s) that express the greatest alignability with the strategic imperatives]. Here, objectives represent constraints which decidedly orient the a functional system:

1. Constraints enable the rejection of unacceptable alternatives [that express dis-alignment ability].
2. Constraints are typically framed as a binary yes or no choice.
3. Constraints establish the design space.
4. Constraints are fixed under consideration of Design Decision Standardization and user requirements.

> *Note: Feature trees are high-level models organizing features into feature groups capturing the entire scope of a project in a single model. They show the relationships between features.*

Here, values become objectives useful for life fulfillment. The encoding of those objectives is likely to produce an economy where only useful things are served; an economy that serves the process of human and ecological fulfillment.

4.1.6 Define system functional requirements (capabilities)

> *Functional requirements area also known as: operational goals, functional attributes, capabilities, capability requirements, requirements, physical objectives, system operational requirements, system performance requirements.*

In contrast to non-functional requirements, functional requirements are usually in the form of "system shall do <requirement>", an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or input>process>output model. A [functional] specification describes the necessary functions at the level of unit(s) and components; these specifications are typically used to build the system.

Capabilities occur in pairs in which some property of the environment (e.g., climb-able) is related by a property of the being's or system's capability, known as an it's effectivity (e.g., to climb or walk).

> **CLARIFICATION:** *In common technical parlance, a "value driver" is another term for a primary function, and expresses how to create "value" for the human in line with its objectives.*

> **PROJECT COORDINATION:** *Needs must be appropriately matched with abilities (as in, ability to do), forming a technical, functional system.*

Herein, a 'capability' is the (physical or informational) ability [of a system] to execute a specified course of action, as originating from some source. In engineering design, a 'capability spread' includes the follow capability elements:

1. **Capability Gap (or Gap)** – The inability to execute a specified course of action.
2. **Capability Requirement** (also called "need" or "requirement") – A capability is required to meet [human] needs, current or future.
3. **Capability Solution** – A materiel or non-material solution to satisfy one or more capability requirements (or needs), and reduce or eliminate one or more capability gaps.
4. **Capability Production** - The materialization of the material or non-material (e.g., digital) solution.

The planned, functional design characteristics of a system are otherwise known as a system's functional attributes. Functions are the behaviors expected from the design. A function is an activity that the system should perform or support. A design should perform certain functions for conversion of a given input into a required output. Functions are often expressed as verb-object pairs. Functions describe what the design (or, more often, an object within the design) will "do" or accomplish, with an emphasis on input-output transformations. Something is expected to occur due to the system's existence in the real world. When denoted, functions are arranged in a hierarchy to express their relationship to the project objectives.

The statement of a function typically couples an action verb to a noun or object (e.g., lift a book, support a shelf, transmit a current, measure a temperature, or switch on a light). For instance, "Measure weight of objects up to 120 kg"; "Support weight up to 70 kg and Hold on wall without failure"; and, "Control pointer on a computer".

Even abstract requirements like 'proximity to transport' may be expressed as functions, such as: enable "easy" access to public transport; whereupon, "easy" is defined specifically and numerically.

The design of a system must account for several types of function:

1. The primary function(s).
2. Desirable secondary functions.
3. Undesirable secondary functions.

For example,

1. Project images (for a projector).
2. Generation light (desirable).
3. Generation of heat (undesirable).

Additional examples of function include:

1. The function of a bicycle brake is stop the wheel when applying the brake lever by means of frictional force between rim and brake pad.
2. The function of a hydraulic lift is to elevate heavy weight by means of pascals law.
3. The function of a speaker is to produce sound by means of electro-magnetic induction.

A quality/efficiency spectrum from optimization of the functional and non-functional attributes of the service to highly sub-optimal (i.e., a near zero efficiency rating to negative efficiency).

Service product[ion] functioning includes:

1. The functioning of a product can be described as follows:
    A. Form (Structure) / Characteristics / Function / Values / Needs.
2. The design process follows this sequence in reverse:
    A. Needs / Values / Function / Characteristics / Form (Structure).

5 [Engineering] System concepts

A.k.a., Concept of operations (ConOps), operations concepts (OpsCon), system operational concept (OpsCon).

System concepts bridge the gap between product scope and technical requirements. System Concepts are plain-language descriptions of user-product/system interactions throughout the life of your system from the perspective of all the key stakeholders. How it will be manufactured, tested, installed, used, maintained, stored, and decommissioned.

When developing the system concepts, users describe a day in the life of the product, for all life-cycle stages, and addressing both nominal as well as off-nominal situations. These descriptions are told from the users' perspective describing their expectations for the system's functionality, performance, capabilities, and quality. These expectations are in the context of meeting need, goals, and objectives within the context of the defined drivers and constraints. A complete system concepts information set helps prevent both missing and incorrect requirements. System concepts will help establish a shared vision for the system and facilitate acquisition of the knowledge needed to define a clear, complete, correct, and concise set of requirements. The 'system concept' is the basis for system functional and performance requirements.

System concepts exist within the systems engineering domain, and it is the responsibility of systems engineering (or whomever is responsible for the technical expertise of the system) to develop and maintain 'system concept' information set (document).

5.1 *Relationship between Concept of Operations and Operational Concept*

Both system concept documents, 'concept of operation' and 'operations concepts' are developed in exactly the same way, and many organizations combine the two information sets into one. Both information sets define capabilities, functionality, performance, and quality needed in the system – just from a different perspective:

1. **OpsCons** focuses on the system under development from a user/operator perspective. Describes the way the system works from the user/operators perspective.
2. **ConOps** focuses on how the system fits into the bigger system of which it is a part and will be developed, tested, and operated. Describes the way the system works from a socio-technical organizational perspective.

More specifically,

1. **Concept of Operations (ConOps, ConOp, CONOP):** A verbal and graphic statement, in broad outline, of a socio-technical organization's assumptions or intent in regard to an operation or series of operations. The concept of operations frequently is embodied in long-range strategic plans and operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be carried out simultaneously or in succession. The concept is designed to give an overall picture of the socio-technical operations.
2. **Operational Concept (OpsCon):** A verbal and graphic statement of an socio-technical organization's assumptions or intent in regard to an operation or series of operations of a system or a related set of systems. The operational concept is frequently developed as part of a system development or acquisition project. The operational concept is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the enterprise's operational environment from the users' and operators' perspective.

Both information sets address user needs, the life-cycle, and nominal and off-nominal situations. Both ConOps and OpsCons involve the telling of stories, scenarios, or use cases. Both align the users to a common vision, are used to define a feasible approach to meeting the overall needs, goals, and objectives, and are used to further define the various development elements involved in the project.

Documenting both perspectives as 'System Concepts' results in addressing the traditional benefits and outcomes of both a ConOps and an OpsCon thereby avoiding confusion in having to distinguish between whether it is a ConOps or an OpsCon.

5.1.1 OpsCon in brief

A system OpsCon document describes what the system will do (not how it will do it) and why (rationale). An OpsCon is a user-oriented document that describes system characteristics of the to-be-delivered system from the user's viewpoint. The OpsCon document is used to communicate overall quantitative and qualitative system characteristics to the acquirer, user, supplier and other organizational elements.

An Operational Concept Document (information set) is a document for recording an Operational Concept. It is prepared at the acquisition organization and developer level to describe how a particular system (new, modified or existing) will be operated to satisfy its user and operator needs. The description is independent of specific design solutions, although it will make reference to a possible design solution at the highest level of abstraction. The Operational Concept Document is not a requirements document. It describes the system operational intent and context, and is used to derive needs and requirements.

In order to avoid inclusion of solution-specific information in the initial Operational Concept Document, system operational behavior should be described in the form of capabilities and outcomes. Initially, any reference to an architectural or detailed solution should be minimized. As the system is realized and the Operational Concept Document is revised throughout the product life cycle, references to the specific architectural features of the solution are incorporated.

5.1.2 ConOps in brief

The ConOps, at the organization level, addresses the user's intended way of operating the organization. It may refer to the use of one or more systems as black boxes to forward the organization's goals and objectives. The ConOps document describes the organization's assumptions or intent in regard to an overall operation or series of operations within the organization in regards to the system to be developed, existing systems, and possible future systems. This document is frequently embodied in long-range strategic plans and cyclical (e.g., annual) operational plans. The ConOps document serves as a basis for the organization to direct the overall characteristics of the future organization and systems. A concept of operation phase defines a need or gap to be filled by a system.

A Concept of Operations document (information set) is a document for recording a Concept of Operations. It is developed at the organization (enterprise) level, independent of any specific system solution, to describe how the organization (enterprise) will operate to execute strategy and doctrine. The Concept of Operations Document is not a requirements document. It describes the organization (enterprise) operational intent and context, and is used to derive needs and requirements.

'Concepts of operation' (ConOps) are defined as operational design elements that guide the organization and flow of project elements, including hardware, software, personnel, communications, and data products through the course of a project objective implementation. Conops are the organizational design elements; how people and robots work together; how they flow through different pathways as they accomplish different tasks. Here, the term 'capabilities' is defined as specific functional mission aspects that can take the form of hardware or software. Additionally, capabilities may be high-level ("architecture level") such as high-bandwidth communications or can be lower level such as pan-tilt-zoom capabilities on a camera. Capabilities are the functional aspect looking at hardware and software. What is required to support humans and robots?

By learning which ConOps and Capabilities are enabling or enhancing (and which are not) early on in the development process, NASA's limited resources are better managed towards value-add systems and support technologies.

5.1.2.1 [System] Concept of operation

> *A.k.a., ConOps, CONOPS, system concept, system concept of operation, operational concept description, operational concepts, operational scenarios, system concepts, use cases, user needs.*

Concept of operation (ConOps) is a formal statement (visual and/or linguistic) of the intended operation of a system. A concept of operation is a user-oriented conceptual formalization that describes the characteristics for a proposed system from a user's perspective. A ConOps describes the proposed system in terms of the user needs it will fulfill, its relationship to existing systems or procedures, and the ways it will be used. A ConOps may focus on communicating the user's needs to the developer or the developer's ideas to the user and other interested parties. The main objective of a ConOps is to communicate with the end user of the system during the early specification stages to assure the operational needs are clearly understood and incorporated into the design decisions for later inclusion in the system and segment specifications.

In general, a [system] 'concept of operation' formalization contains the following:

1. Define the environment in which the system will operate.
2. Define the high-level system concept and reason (provide rationale and justification) that it is superior to the other known alternatives.
3. Provide high-level requirements.
4. Provide the criteria to be used for validation of the completed system.

The common deliverables for a 'concept of operation' formalization are:

1. Statement of the goals and objectives of the system (what is important?):
    A. Identify direction (e.g., expected impact as human fulfillment, desired result optimal quality of life, high-level conception of operation as habitat service system using a unified societal system.
    B. Establish objective priorities (e.g., establish societal-human priorities; identify human needs and requirements).
    C. Identify objective dependencies (e.g., identify the dependencies between conceptual objects, material objects, and their interrelationships through time as a matrix of dependencies, or dependency flow models as input-output

service modeling to fulfill all human needs).
2. Identify constraints affecting the system ("externally" environmental):
   A. Ecological [service flows and capacities]
   B. Resource [service flows from market, ...]
   C. Jurisdictional [service flows from State, ...]
3. Organizations, activities, and interactions among team.
4. Clear statement of roles and accountabilities ("responsibilities").
5. Specific operational processes for fielding the system.
6. Processes for initiating, developing, maintaining, and retiring the system.

Additional ConOps objectives include:

1. Provide end-to-end traceability between operational needs and captured source requirements.
2. Establish a high-level basis for requirements that supports the system over its life cycle.
3. Establish a high-level basis for test planning and system-level test requirements.
4. Support the generation of operational analysis models (use cases) to test the interfaces.
5. Provide the basis for computation of system capacity.
6. Validate and discover implicit requirements.

5.2 System concepts standards

The principal standards defining 'System Concepts' (as ConOps) are:

- **ISO/IEC/IEEE 29148-2018:** Systems and software engineering - Life cycle processes - Requirements engineering.
- **ISO/IEC/IEEE 15288:2015:** Systems and software engineering - System life cycle processes.
- **IEEE Std 1362-1998:** IEEE Guide for Information Technology—System Definition—Concept of Operations (ConOps) Document [standards.ieee.org/standard]
- **ISO/IEC/IEEE 29148:** Systems and software engineering. Life cycle processes. Requirements engineering
- **ANSI/AIAA G-043A-2012:** Guide to the Preparation of Operational Concept Documents
- **NASA NPR 7123.1B:** US NASA Systems Engineering Processes and Requirements (here, the definitions of ConOps and OpsCon are closely aligned with BSR/AIAA G-043A).
- **DI-IPSC-81430:** US DoD data item description for CONOPS document.

NOTE: *The first commonly known standard that defined the idea of a formalized Concept of Operation was IEEE 1362-1998 - IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps).*

The principal standards guiding the development of OpsCon are:

- **IEEE Standard 1362:1998:** IEEE Guide for Information Technology – System Definition – Concept of Operations Document
- **ISO 14711:2002(E)** - Space systems – Unmanned mission operations concepts
- **FHWA-HOP-07-001:2005** - Developing and Using a Concept of Operations in Transportation Management Systems, US Federal Highway Administration

5.2.1 Standards descriptions of ConOps

System concepts are defined via the aforementioned standards in the following ways:

- **ISO/IEC/IEEE 29148:** The ConOps, at the organization level, addresses the leadership's intended way of operating the organization. It may refer to the use of one or more systems, as black boxes, to forward the organization's goals and objectives. The ConOps document describes the organization's assumptions or intent in regard to an overall operation or series of operations of the business with using the system to be developed, existing systems, and possible future systems. This document is frequently embodied in long-range strategic plans and annual operational plans. The ConOps document serves as a basis for the organization to direct the overall characteristics of the future business and systems, for the project to understand its background, and for the users of [ISO/IEC/IEEE 29148] to implement the stakeholder requirements elicitation.
- **IEEE Std 1362-1998:** A Concept of Operations (CONOPS) is a user-oriented document that describes systems characteristics for a proposed system from a user's perspective. A CONOPS also describes the user organization, mission, and objectives from an integrated systems point of view and is used to communicate overall quantitative and qualitative system characteristics to stakeholders.
- **Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition, INCOSE:** A Concept of Operations (ConOps) document is produced early in the requirements definition process to describe what

the system will do (not how it will do it) and why (rationale). It should also define any critical, top-level performance requirements or objectives (stated either qualitatively or quantitatively) and system rationale. Describes the way the system works from the operator's perspective. The ConOps includes the user description and summarizes the needs, goals, and characteristics of the system's user community. This includes operation, maintenance, and support personnel.

5.3 Concept of operations

A.k.a., Concepts of operation, ConOps, CONOPS.

Concept of Operations (ConOps) is a description of how the system will operate to meet user (operator) expectations for anything being conceptualized for the purpose of transforming that concept into reality. The ConOps includes the user description and summarizes the needs, goals, vision, and characteristics of the system's user. ConOps includes a description of the operation, maintenance, and support for the system. The ConOps describes the characteristics of a proposed system from the viewpoint of its users (and in a unified system, its operators also). It is used to communicate the quantitative and qualitative system characteristics to all stakeholders and serve as a basis for stakeholder unification on the issue. It often conveys a clearer statement of intent than the requirements themselves. The concept of operation describes the concept of the solution to meet the requirements. ConOps is a formal description of the likely operation of a future or existing system in the terminology of its users, providing important information for the development (or acquisition) of that system.

A ConOps makes all project team members aware of the different types of users of the system and activities those users will perform. This allows everyone who uses the document to get an idea of who is performing what task and in what order they are performing those tasks.

NOTE: *A ConOps is a living document that is updated a changes occur. A ConOps is a directional document which can be used to compose an executive summary.*

ConOps is the first step in the engineering life-cycle for a new project [to develop a system]. The ConOps is a starting point for the more detailed description of the system. A system functional requirements specification follows the ConOps. Although the Concept of Operations is not a requirements document, a well-formed concept of Operations will be a primary source of information used to create the initial high-level functional requirements.

ConOps is part of the systems engineering life-cycle process, as seen below:

1. ConOps.
2. Requirements (high-level to detailed).
3. Design (high-level to detailed).
4. Implementation.
5. Integration and testing system verification.
6. Operation and maintenance.
7. Assessment.

The principal goal of the ConOps is to provide high-level definition of the system, including:

1. Identifying the required vision (mission) in functional terms.
2. The major parts within the envisioned system.
3. The flows of information among those parts, the information flow to any entities external to the system.
4. The high-level capabilities of the system.

Each capability in the ConOps needs units of measure meaningful for decisioning:

1. **Measures of effectiveness** - Operational measures of success that are closely related to the achievements of the vision or operational objectives evaluated in the operational environment, under a specific set of conditions.
2. **Measures of performance** - Characterize physical or functional attributes relating to the system operation, measured or estimated under specific conditions.
3. **Key performance parameters** - Capabilities and characteristics so significant that failure to meet them can be cause for re-evaluation, re-assessing, or termination of the project.

The ConOps (with OpsCon) provides the following:

1. An analysis and information set that bridges the gap between the users' needs and visions, and the developer's technical specifications.
2. A means of describing a user's operational needs without detailed technical issues that shall be addressed during the systems design analysis activity.
3. A mechanism for documenting a system's characteristics and the user's operational needs in a manner that can be verified by the user without requiring any technical knowledge beyond that required to perform normal job functions.
4. A place for users to state their desires, visions, and expectations without requiring the provision of quantified, testable specifications. For example, the users could express their need for a "highly reliable" system, and their reasons for that need, without having to produce a testable reliability

requirement.
5. A mechanism for users and operators (and buyers in the market) to express thoughts and concerns on possible solution strategies.

A ConOps document can be separated at a high-level into four major sections (or stages):

1. Describe current system (description).
2. Describe changes to make (description).
3. Describe proposed system (description).
4. Analyze proposed system (explanation).

In concern to the questions that provide a common context for any system, a ConOps/OpsCon answers the following:

1. **Who:** These describe the interactions among the various human elements within the system including their interfaces with people external to the system. The document and related scenarios should also identify decision points to include the organizational entity with authority to make those decisions. Other systems with which this system interoperates are also identified.
2. **What:** These are the known components or elements and top level capabilities required of the system, at its highest level of abstraction, to perform the necessary functions. The components are described from an operational point of view. Necessary mission phases or modes may also be described here. The Whats also include descriptions of the external systems with which the system of interest interfaces, and the external interfaces. Principal internal interfaces are also described.
3. **When:** These describe activities, tasks, flows, precedence, concurrencies, and other time / sequence related elements necessary to achieve the mission objectives in each of the various mission modes and conditions. Whens may also include information as to system development and operational availability dates, such as project milestones.
4. **Where:** These are the environments, such as geographical and physical locations of user's facilities and interfacing systems, within which the capabilities are required to be performed and supported. A description of the nature of the interfaces with other systems, organizations and the environment is also needed.
5. **How:** These tie together the other elements (the what, where, when, who, and why) to describe how the system is expected to be used, operated, maintained and, ultimately, retired in the given environment, under all significant conditions. The emphasis should be on concepts and should avoid any system design or implementation inferences.
6. **Why:** These provide the rationale behind any established partitioning of the mission tasks between the system components and the operators, and the reasoning for specific sequences of activities or tasks. For example, an important function of the document is to provide the rationale behind the definition of the level of technical expertise required of the system operators. This will provide a basis for the definition of a set of system requirements and designs with a consistent level of complexity and sophistication.

Generally, a ConOps includes all (or, some portion of) the following information sets:

1. **Introduction**
   A. **Document identification** - title and identification number.
   B. **Document overview** - an overview of the ConOps document.
      1. To communicate user needs and the proposed system expectations.
      2. To communicate the system developer's understanding of the user needs and how the system will meet those needs.
   C. **System overview** - a high-level overview of the proposed system in text and graphic.
   D. **Development effort** - a brief description of the scope of effort required.
2. **Applicable references and documentation** - what sources are mentioned. This section lists the document identification number, title, revision, and date of all documentation referenced in the ConOps.
3. **Current system/situation** - what is the problem to be solved, and the system or situation as it currently exists.
   A. **Background, objectives, and scope** - include as necessary all background, mission, objectives, and scope of the current system.
   B. **Operational constraints** - include descriptions on the operational characteristics of the existing system. This could include limits on fulfillment, hours of operation, hard/software limitations, and resource limitations.
   C. **Description of the current system or situation** - provide a thorough description of the current system, including but not limited to: operational characteristics, major system components, component interconnections, external system interfaces, current system functions, diagrams illustrating inputs, outputs, data flows. Include

a description of user classes and other people who interact with the system.

D. **User profiles** - describe who the users are and how they interact with the current system, and what happens when they do. Also discuss how the users interact with each other when using the system.

4. **Reasoning/justification for change** - why change is needed. Describe the issues, problems, gaps, shortcomings of the current system or situation that motivate development of a new system or modification of an existing system.
   A. **Reason for changes** - include the reasons for developing the proposed system, including the new, modified, or discovered user needs, goals, objectives; and dependencies or limitations of the current system.
   B. **Description of the desired changes** - include a summary of the new or modified capabilities, functions, processes, interfaces, and other changes needed to respond to the justifications previously identified.
   C. **Change priorities** - prioritize or rank the proposed changes. Specify what features are essential, what features are desirable, and what features are optional.
   D. **Changes considered but not included** - include significant changes or features that were considered but not included in the proposed system.
   E. **Assumptions and constraints** - describe assumptions or constraints applicable to the changes and new system features.
5. **New system description (concepts for new/ modified system)** - functional architecture and concepts for the proposed system; what is the proposed system that results from the desired changes specified in the fourth section of the ConOps.
   A. **Objectives and scope** - provide an overview of the new or modified system, including the mission, objectives, and scope. Focus on what about the vision or objectives is new that necessitated a new or modified system.
   B. **Operational policies and constraints** - describe the operational policies and constraints that apply to the proposed system.
   C. **Description of the proposed system** - provide a thorough description of the proposed system. The system description must be simple and clear enough that all intended readers can fully understand it. A high-level graphical overview of the system is strongly recommended.
   D. **Modes of operation** - describe the proposed system's various modes of operation. Examples of modes of operation (e.g., HSS operational processes) include: emergency/incident; maintenance and operations; planning; discretionary; strategic.
   E. **User involvement and interaction** - identify the users and the way they interact within the system.
   F. **Support environment** - describe the support and maintenance concepts, including the operating environment for the proposed system.
6. **Operational scenarios** - describe scenarios from different user's viewpoints. One or more operational scenarios that illustrate the role of the new or modified system, its interaction with users, its interface to other systems, and all states or modes identified.
7. **Summary of impacts** - Describes and summarizes the operational impacts of the proposed system from the users' perspective. This information is provided to allow all stakeholders to prepare the changes that will be brought about by the new system. Impacts include operation, organizational, and impacts during development.
8. **Analysis of proposed system** - summarize the benefits, limitations, advantages, disadvantages, and alternatives (and trade-offs) considered for the proposed system. In the context of a ConOps document, alternatives are operational alternatives and not design alternatives, except to the extent that design alternatives may be limited by the operational capabilities desired in the new system.
9. **Notes, glossary, supplemental material**

A ConOps may also include:

1. **Resource requirement estimate** - A rough order of magnitude resource estimate.
2. **Market cost estimate** - A rough order of magnitude cost estimate (ROM; market only).
3. **A schedule estimate** (with expected critical path).
4. A **project sequence of activities and events** - the sequence of tasks. (project plan concept).

Simplistically, ConOps answers the basic questions (for a new or existing system):

1. Who - who are the team involved in the system.
2. Why - what does the organization lack, and what will the system provide.
3. Where - what are the physical locations of the system.
4. When - what is the time-sequence of activities that will be performed by the system.

5. How - what resources are needed to design, build, and operate the system.

A ConOps is used to:

1. Provide a vision to guide to the development and operation of the system. A common vision about what is being built and operated.
2. Provide reasoning (justification) for and nature of changes.
3. Identify system stakeholders.
4. Assure a common communications reference.
5. Formulate and document a high-level system definition.
6. Foundation all lower-level sub-system descriptions.
7. Define all major user groups and activities.
8. Identify the environment in which the system will function.

5.3.1 System reasoning/justification

In order to provide a complete set of reasoning for a new system, the following questions should be answered:

1. Why is the new system need?
2. What is being changed?
3. What new functions do you get?
4. How does it change the environment?
5. What changes are needed to support the new system?
6. What are the most important changes (priorities)?
7. What changes are requested, but not included?
8. What assumptions and constraints are there in the system to be built?

5.3.2 Operational scenarios

A scenario is a step-by-step description of how the proposed system should operate and interact with its users and its external interfaces under a given set of circumstances. Scenarios are written in natural (layman's) language and should be non-technical as much as possible.

Scenarios should be structured so that each describes a specific operational sequence that illustrates the role of the system and its interactions with users and other systems. It may be necessary to develop several variations of each scenario, including one for normal operation, one for exception handling, one for degraded mode operation, etc. Each scenario will describe an operational event from the different user perspectives. Scenarios help the readers of a conOps document understand how all the pieces interact to provide operational capabilities.

Generally, a scenarios includes all of the following:

1. A description of the starting situation.
2. A description of the normal flow of events.
3. A description of what can go wrong.
4. Information about other concurrent activities.
5. A description of the state when the scenario finishes.

5.3.3 Operating environments

The various environments in which a system will be deployed, operate and be maintained include, but are not necessarily limited to:

1. Physical.
   A. Natural.
   B. Induced.
   C. Self-induced.
   D. Threat.
   E. Cooperative.
2. Social.
3. Technological.
4. State jurisdictional.
5. Market economic.

5.4 Types of OpsCon

OpsCon can be classified according to a system's life-cycle:

1. **Operations concept** - describes the way the system works from the operator's perspective.
2. **Production concept** - describes the way the system will be manufactured.
3. **Deployment concept** - describes the way the system will be delivered and installed.
4. **Support concept** - describes the desired support infrastructure and manpower considerations for maintaining the system after it is deployed. This includes specifying equipment, procedures, facilities and operator training requirements.
5. **Disposal concept** - describes the way the system will be removed from operation and retired.

OpsCon can also be classified according to who is composing and using the document:

1. User OpsCon - Written by users and operators, or by the developer in collaboration with the users and operators. Usually written prior to the commencement of development activity, but can be prepared at any point in the system life cycle. Defines the user's and operator's expectations for the system's operational capabilities.
2. System OpsCon - Written by developer personnel during or after the design activity defining how the system is to be used. Defines the developer's perception of how the system will be used.

3. Alternative OpsCon - Written during the concept exploration phase for each of the major alternative systems examined.
4. Remedial OpsCon - Written to redirect a Program that displays a lack of understanding of the overall system concept. It would typically be written at some point during the design phase.
5. Operations OpsCon - Written toward the end of the development Program to be maintained during the operations and support phase. It is written from the user and operator perspective and provides a representation of the system operations and capabilities as delivered.

6 [Engineering] Requirements

A.k.a., Requirements engineering.

A requirement is what the system must do to address the operative directive and satisfy the user of the solution (system). Requirements are the descriptions of the system services and constraints that are generated during the requirements engineering process. Requirements engineering (RE) is the process of defining, documenting, and maintaining requirements in the engineering design process. All requirements are statements using some measure that can be objectively tested. Requirements define what a system should (must, shall, etc.) do and define constraints on its life-cycle (e.g., development, implementation, operation, disposal, etc.). Simply, a requirement is a condition or capability to which a system must conform in operation and/or development. Requirements are that which is necessary for a system to function as intended and designed.

> **CLARIFICATION:** *Requirements are different than goals, and other user directive statements, such that requirements can be objectively tested, whereas goal statements may not necessarily be stated in such a way that they can be objectively tested.*

Requirements range from high-level abstract statements of a service or of a system constraint, to detailed mathematical functional specification.

> **IMPORTANT:** *At a societal engineering level, [human] requirements define the societal direction, objectives define the societal orientation (values), and methods define the societal approach.*

Requirements are often expressed as "shall" statements. Requirements are level dependent; for example, there are system requirements (top-level) and subsystem, or component (bottom-level) requirements.

Fundamental problems arise when requirements are not properly stated. Ambiguous requirements may be interpreted in different ways (by developers and users).

> **NOTE:** *In the market, requirements serve as the basis of all contracts, including all bids for contracts.*

Requirements should be (i.e., attributes of requirements are):

1. **Complete** - describe everything required. Each requirement describes one result that must be achieved by the product. A requirement should not be redundant. The requirement should not describe the means of obtaining the result. Are all functions and conditions required included?
2. **Consistent** - no conflicts and no contradictions. Individual requirements are not in conflict with

other requirements. Are there any requirements conflicts or contradictions?
3. **Necessary** - Absolute requirements that are to be verified are identified by "must" or "shall". Goals or intended functionality are indicated by "will".
4. **Correct** - Each requirement is an accurate description of a feature or process of the product.
5. **Unambiguous (clear)** - The statement of each requirement denotes only one interpretation. Can the requirement be fully understood?
6. **Realistic (feasible)** - Can the requirement be implemented given available knowledge, resources, persons, and technology? Can a real world solution be built and tested to prove that the requirement is satisfied?
7. **Verifiable (testable)** - Each requirement is stated in concrete terms and measurable quantities. A process should exist to validate that the product (when developed) will satisfy the set of requirements. Can the requirement be checked? Is the requirement realistically testable?
8. **Modifiable** - The structure and style of the requirements are such that any necessary changes to the requirements can be made easily, completely, and consistently.
9. **Traceable** - The origin of each requirement is clear and can be tracked in future development activities. Is the origin of the requirement clearly stated?

6.1 Engineering design requirements

The identification of a projected system requires a set of descriptive engineering design requirements (a.k.a., engineering requirements). Requirements integrate needs and objectives into a set of instructions (i.e., requirements) that the design has in some priority (i.e., requirements are structured). Requirements describe:

1. What functions the system is supposed to provide (what the system does)?
2. What characteristics the system is supposed to have (what the system['s quality] is)?
3. What goals the system is supposed to meet or to enable users to meet (what use is the system)?

A requirement is another type of [directional] input (imperative) into a project, more specific than and informed from needs, objectives, and goals. In order to coordinate action and access among a team, it is imperative to describe what the system is supposed to do. All requirements have rationales that logically relate requirement to a prior imperative, and must consider:

1. What is an aspect of this requirement that could be a source of confusion?
2. How is the potential confusion addressed in the requirement?
3. What is the evidence that informs the resolution of the confusion?
4. What other requirements might have some effect on the interpretation and implementation of the requirement, and thus should be referenced in the rationale?

A requirement is an engineering input composed of a statement of a need or objective, or of a condition or capacity, that a system or product must possess to satisfy a prior need or objective. Therein, a requirement is a property that a system or product must have to provide usability (or functionality) to a user. Requirements are the start of tasks (i.e., the instantiation of tasks), and the first phase of real world issue resolution. Every requirement inherently asks, "How will a successful (or complete) implementation of this requirement (a specific description of some thing in the real world) be verified?"

A requirement is visualized in a table with [at least] two related columns:

1. The system must have x.
2. Because of y.

A system requirement list explains why a system, product or service is needed, puts the system in context, and describes what the finished system will be like and/or what it will do. During engineering, the answers to how questions fall into the realm of design, the next sequential phase after initial requirements are developed. Thus, requirements specifications should not include design solutions (except for interface requirements, which often include embedded design).

One or more requirements forms a 'requirements set/list'. A requirements list is *formalized* (integrated into a unified model of the system). The requirements list is structured hierarchically.

During engineering, requirements are the basic source for communication among end-users, InterSystem Teams, and intelligent systems.

> *Note: A [problem] domain-model is an abstraction that defines the structure and behavior of the problem domain.*

In the real world, there are several potential problem domains. There are community and market concepts involved in the life cycle modeling of requirements:

1. In community, there are two types of concepts involved in life cycle requirements modeling:
   A. **Human [conceptual] objectives**, which represent the problem domain, and are emphasized in the requirements model.
   - **Engineering concepts**, which are emphasized in the design model(s).

2. In the market, there is one additional type of concepts involved in life cycle requirements modeling:
   A. **Business concepts** related to customers' objectives, which represent the problem domain, and are emphasized in the requirements model, though expressed in the design.

Requirements provide a tool for evaluating the final results of the project by examining whether each requirement has been met. Every rule and functional relationship provides a test point. Note that requirements tend to change through the course of a project, with the result that the final output, as delivered, may not adhere to the initial version of the requirements.

A 'requirement':

1. *Should specify* the expected behavior and/or form, through a detailed analysis of that which is required for creation of the new status/state [iteration] and/or new product. Generally, requirements are statements of *what* a system should do, rather than *how* it should do it (which is present in the design).
2. Is defined as, *What is needed?* A requirement is, *a well-defined need*.
3. Is *an objective that must be met*. Requirements define *necessary objectives*.
4. Contains criteria for completion, or is *testable*.
   A. *Performance* is the degree to which requirements are met.
5. As a 'list' *includes*, descriptions of system properties, specifications for how the system should work, and constraints placed upon the development and designed operating process.

The attributes of good requirements include the following:

1. **Achievable** - A requirement must be achievable. It must reflect need or objective for which a solution is technically achievable at costs considered affordable.
2. **Verifiable** - The expected performance and functional utility must be expressed in a manner that allows verification to be objective, preferably quantitative; that is, not defined by words such as excessive, sufficient, resistant, etc.
3. **Unambiguous** - A requirement must be unambiguous. It must have but one possible meaning.
4. **Complete** - It must be complete and contain all mission profiles, operational and maintenance concepts, utilization environments and constraints. All information necessary to understand the customer's need must be there.
5. **Causative** - It must be expressed in terms of need, not solution; that is, it should address the "why" and "what" of the need, not how to do it.
6. **Consistent** - It must be consistent with other requirements. Conflicts must be resolved up front.
7. **Appropriate** - It must be appropriate for the level of system hierarchy. It should not be too detailed that it constrains solutions for the current level of design. For example, detailed requirements relating to components would not normally be in a system-level specification.

Project requirements (or just, requirements) are conditions and capabilities that must be met, or tasks that must be completed, for the project to be complete. Requirements require identifying, defining, organizing, documenting, and refining. A 'Requirements Specification' (a.k.a., Requirements Definition Document) documents requirements as a specification. Requirement become technical specifications (composed of material and/or information properties that feed back upon us, influencing our experience. Any engineer/programmer can build one from this document.

Every requirement must be testable. To know when a project is complete, every requirement must have been tested as complete. If a requirement is not testable, then how will a complete (successful) implementation of the requirement be verified. The requirement must answer, "How do "you" the requirement has been completely implemented and works as expected?"

In application, a given system's requirements will have the following set of control characteristics:

1. **Defined -** Define a model of the system to be built; not [a model of] the system [itself].
   A. Define some mixture of functionality, behavior, performance, and systems constraints (non-functional requirements).
   B. Defines known constraints.
2. **Organized** - Organized by functionality and logical layout.
3. **Tested** - Every statement is verifiable, with level and nature of test as attributes.
4. **Assigned** - Assigned to InterSystem Teams.
5. **Opened** - Viewable to everyone.

In concern to a community-type societal system, what the engineered system does is directly perceived and independently experienced by its users – either human users or other systems. When a user performs some action, the societal system responds in a particular way; when an internal system or user submits a request of a certain form, it gets a particular response. Due to the nature of societal systems being composed of the users themselves, the users must agree on actions they can perform and responses they should expect from the societal system. This common understanding is captured in the requirements.

INTERSYSTEM TEAM: *Requirements engineering is the "sub-discipline" of systems engineering that encompasses all project activities associated with understanding a system or product's necessary capabilities and attributes, including both requirements development and coordination.*

Once the system is in operation, a new societal requirements is either:

1. A specified design change in the status or state of the societal system;
2. Or, a newly designed [habitat] service or product.

Here, requirements include only real requirements to the system (service-product), and exclude requirements to the project or any other ancillary information.

APHORISM: *It is from requirements that engineering can proceed. Because it is not possible to have an acceptable system even with the best solution space if this is based on an incorrect problem space formulation.*

The concept of system existential categories, which correspond to the following requirement types:

1. Functional requirements (Do): Requirements that define what the system must do. In other words, what it accepts and what it delivers (i.e., expected transformation). Examples: The system shall provide food; The system shall transmit 4 signals; The system shall convert sea water into drinkable water.
2. Performance requirements (Being): Requirements that define how well the system must operate, which includes performance related to functions the system performs or characteristics of the system on their own, such as -illities. Examples: The system shall move at a speed higher than 30 km/h; The system shall have a reliability better than 0.80.
3. Resource requirements (Have): Requirements that define what the system can use to transform what it accepts in what it delivers. Examples: The system shall consume less than 300 W; The system shall have a mass of less than 30 kg.
4. Interaction requirements (Interact): Requirements that define where the system must operate, which includes any type of operation during its life-cycle.

Examples: *The system shall withstand shock levels higher than 300 g; The system shall operate in vacuum (to reflect operation); The system shall operate in clean room class 'X' (to reflect Assembly, Integration, and Test activities).*

Each level of the requirements hierarchy represents a fully operable system as they are options that build upon previous need levels. The amount of levels is unlimited and free of preconceptions, being therefore up to each project to define theirs.

Herein, the threshold-based concepts are:

1. Base Threshold: the minimum level of service (value) that must be provided so that the system is acceptable.
2. Goal Threshold: desired value provided by the system.
3. Want Threshold: great-to-have, but considered difficult to achieve.

Functional ("do") requirements include:

1. The system shall service the spectral range of human need.
   A. The system shall provide water services to ...
   B. The system shall provide energy services to ...
   C. The system shall provide building services to ...
   D. The system shall provide medical services to ...
   E. The system shall provide production and material cycling services to ...
   F. The system shall provide to ...

Performance ("being") requirements:

1. The service system shall have an efficiency better than 98%.
2. The service system shall have a reliability higher than 0.90.

Resource ("have") requirements:

1. The system shall fit within a circular boundary.
2. The system shall have a human carrying capacity

NOTE: *Carrying capacity is a limit that varies with technology.*

Interaction ("have") requirements:

1. The system shall fulfill its performance requirements in the manner specified by the modularity (or other) standard (protocol).
2. The system shall fulfill its performance requirements within the carrying capacity of the larger ecosystem

The International Council of Systems Engineering (INCOSE) [2011] proposes an independent classification of requirements that targets any complex system and that includes:

1. Functional requirements.
2. Performance requirements.
3. Non-functional requirements.
4. Architectural constraints.

Hull et al. (2005) make a similar contribution in the field of software systems and define

1. Functional requirements.
2. Performance requirements.
3. Quality factor requirements.
4. Environment requirements.
5. Interface requirements.
6. Constraint requirements.

Requirements exist on design attributes, on the existence of objects and characteristics, on the relationships, and on functions. Their proposition confirms a designer perspective when eliciting requirements: How the system has to be designed.

Function requirements, which indicate what the system must do:

1. Performance requirements, which define how well the functions of the system must perform;
2. Resource requirements, which define the resources that are available to create and maintain the functions and performance of the system (explicitly referring only to money, people, and time);
3. Design constraints, which define restrictions on the solution;
4. Condition constraints, which define restrictions on the use of the system.

Providing a domain-independent classification:

1. Input/output requirements.
2. Technology requirements.
3. Performance requirements.
4. Cost requirements.
5. Trade-off requirements.
6. System test requirements.

Medical industry a matrix classification of 5 domains as categories of requirements:

1. Process.
2. Performance.
3. Safety.
4. Cost.
5. Documentation.

Seven stages of the system life-cycle as categories of requirements:

1. Design.
2. Manufacturing.
3. Distribution.
4. Installation/assembly/integration.
5. Operation.
6. Maintenance.
7. Recycle.

6.1.1 Requirements breakdown structure (RBS)

The RBS is different than the WBS. The RBS is grouped logically, and the WBS is grouped into physical work packages for the configuration of items that need to be developed. The information in the RBS flows into the WBS.

Requirements functional flow block diagram (flows between functions are seen, not just the hierarchical relationships between functions) - provides information on the sequencing (parallel and series) of functions.

Common requirement framework characteristics include:

1. A mission statement with 5-7 key concepts. Each of those concepts if detailed (fleshed out) at the next level leading to 5-7 goal statements, each of which contains 5-7 concepts, fleshed out at the next level as objectives, each of which has 5-7 concepts, and the process continues until we reach the “leaves” of the tree.
2. Numbering requirements - The usage of a number system allows each level to be associated with the levels above and below.

6.1.2 Requirement categorization approach

The NASA approach to categorizing project requirements:

1. **Technical requirements** (functional requirements, performance requirements, and interface requirements).
2. **Operational requirements** (mission, configuration, and command and telemetry).
3. **Reliability requirements** (environment, fault tolerance, verification, and process and workmanship).
4. **Safety requirements**.
5. **Specialty requirements** (maintainability, producibility, etc.).

The European Space Agency approach:

- Functional Requirements, Mission Requirements, Interface Requirements, Environmental Requirements, Operational Requirements, Human Factor Requirements, (Integrated) Logistics Support Requirements, Physical Requirements, Product Assurance Requirements, Configuration

Requirements, Design Requirements, and Verification Requirements

In engineering, the classification of requirements should effectively describe the configuration of the output, the resulting system specification. The classification of requirements facilitates the design of the system, which implies influencing the design and selection of a solution. However, at the user level, the required imperative (goal of service) should rather specify what the system is intended to do.

6.2 Requirements as objectives

NOTE: *The way in which requirements are categorized can impact system affordability.*

Once a need has been recognised and identified, then resources are allocated to the development of a design for its fulfillment, and the 'engineering/design task' is initialized ("born").

The objectives of requirements are:

1. Completely define a system by means of defining all elements necessary to complete what the system is intended to achieve, shall fit within the proposed categories.
2. Identify requirements that are not applicable to the system to be developed, but to ancillary elements, such as supporting systems (or market contractual agreements).
3. Identify constraints that do not support the satisfaction of user needs, but that limit the solution space, thus facilitating the definition of boundaries for the solution and eliminating any influence on a specific solution.

A system is completely defined by specifying:

1. What systems do,
2. How they are (how well they do),
3. What they use,
4. Where they live.

All sub-systems are elements that form requirements that define:

1. What the system has to do?
2. In what context the system has to do it?
3. How well the system has to do it?
4. Which resources the system can use to do it?

Examples of requirements as objectives include:

1. Adaptability needs:
    A. Can you upgrade and modify it?
    B. Sub-conceptions:
        1. Flexibility, modularity, scalability, etc.
2. Operational effectiveness (readiness) needs.
    A. With what does it operate, how does it operate?
3. Efficiency needs:
    A. Is it intuitive and does it operate well?
    B. Sub-conceptions:
        1. Use of resources, process efficiency
4. Availability needs :
    A. How often does it fail?
    B. Sub-conceptions:
        1. Reliability, maintainability, supportability, etc.
        2. Durability = repairability and maintainability

6.3 Requirements as metrics

Metrics are a means of identifying whether an individual atomic requirements statement or an entire requirements set (requirements document as a whole) has been met and/or is in the progress of being met. Requirements are identified with standardized names and a method of both subjective and objective measuring.

There are three primary categories of metrics in terms of requirements:

1. **Requirements traceability (Traceability metrics)** - Is the set of requirement(s) internally traceable, with clear associations, and no conflict between individual requirements?
2. **Requirement consistency (Consistency metrics)** - Is the set of requirement(s) internally consistent, with no contradictions, no duplication between requirements?
3. **Requirements falsibility (Falsibility metrics)** - How adequately can this requirement be tested? Is it clear what test(s) are needed to confirm the requirement is met? Is it clear what should be considered a failure of a test of this requirement?
4. **Requirements visualizability (Verifiability metrics)** - How adequately can this requirement be visualized in object form? Is it clear what objects and relationships are needed to understand the requirement? Is it clear what is not a visualization of this requirement?

6.4 Requirements list

A.k.a., Requirements specification.

A requirements specification should include:

1. Definition of the function or entity.
2. Description of inputs and where they come from.
3. Description of outputs and where they go.
4. Information about the information needed for the

computation and other entities used.
5. Description of the action to be taken.
6. Pre and post conditions (if appropriate).
7. The side effects (if any) of the function.

6.5 Systems engineering and requirements

Requirements are the primary focus in the systems engineering process because the process's primary purpose is to transform the requirements into designs. The engineering development process develops these designs within the constraints. They eventually must be verified to meet both the requirements and constraints.

> **NOTE:** *The primary evaluation of "success" of a system is the degree to which it meets the purpose for which it was intended.*

Requirements engineering is the process of:

1. Discovering the purpose for the system by identifying users and their needs, and
2. Documenting these in a form that is agreeable to analysis, communication, and subsequent implementation.

Requirements engineering is a set of activities concerned with identifying and communicating the purpose of a system, and the context in which it will be used. RE acts as the bridge between the real-world needs of users, customers, and other constituencies affected by a system, and the capabilities and opportunities afforded by technologies.

> **INSIGHT:** *Real-world goals [ought to] motivate the development of a system.*

Typical definitions of engineering refer to the creation of effective solutions to practical problems by applying scientific knowledge. Therefore, the use of the term engineering in RE serves as a reminder that RE is an important part of an engineering process, being the part concerned with anchoring requirements activities to a real-world problem, so that the appropriateness and effectiveness of the solution can then be analyzed. It also refers to the idea that specifications themselves need to be engineered, and RE represents a series of engineering decisions that lead from recognition of a problem to be solved to a detailed model of that problem.
The primary requirements engineering activities are:

1. Eliciting requirements - identifying, articulating, or otherwise defining requirements by asking the right questions.
2. Analyzing and modeling requirements.
3. Communicating requirements.

The identification of the problem that needs to be solved leads to identification of a system's boundaries. These boundaries define, at a high level, where the final delivered system will fit into the current operational environment. The identification of user classes, of goals and tasks, and of scenarios and use cases all depend on how the boundaries are selected.

6.5.1 Types of system Requirements

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management:

1. **User requirements:** Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability. The customers are those that perform the eight primary functions of systems engineering , with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum.
2. **Functional Requirements:** The necessary task, action or activity that must be accomplished. Functional (what has to be done) requirements identified in requirements analysis will be used as the top-level functions for functional analysis.
3. **Performance Requirements:** The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness. During requirements analysis, performance (how well does it have to be done) requirements will be interactively developed across all identified functions based on system life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements.
4. **Design Requirements:** The "build to," "code to," and "buy to" requirements for products and "how to execute" requirements for processes expressed in technical data packages and technical manuals.
5. **Derived Requirements:** Requirements that are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight.
6. **Allocated Requirements:** A requirement that is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements. Example: A 100-pound item that consists of two subsystems might result in weight requirements of 70 pounds and 30 pounds for the two lower-level items.

6.5.2 Requirements analysis

Requirements analysis involves defining customer needs and objectives in the context of planned customer use, environments, and identified system characteristics to determine requirements for system functions. Prior analyses are reviewed and updated, refining mission and environment definitions to support system definition. Requirements analysis is conducted iteratively with functional analysis to optimize performance requirements for identified functions, and to verify that synthesized solutions can satisfy customer requirements. The purpose of Requirements

Analysis does:

1. Refine customer objectives and requirements.
2. Define initial performance objectives and refine them into requirements.
3. Identify and define constraints that limit solutions.
4. Define functional and performance requirements based on customer provided measures of effectiveness.

In general, requirements analysis should result in a clear understanding of:

1. Functions: What the system has to do.
2. Performance: How well the functions have to be performed.
3. Interfaces: Environment in which the system will perform.
4. Other requirements and constraints.

The understandings that come from requirements analysis establish the basis for the functional and physical designs to follow. Good requirements analysis is fundamental to successful design definition.

Requirements analysis is a process of inquiry and resolution.

1. User requirements.
2. Design requirements (prioritize and structure).
3. Target values (benchmarking) against target values or what is expected.
4. Collaborative design and process planning - match capabilities to requirements; what capabilities are available? What capabilities must be developed?

Common requirements analysis questions include, but are not limited to:

1. What are the reasons behind the system development?
2. What are the user expectations? What do the users expect of the system?
3. Who are the users and how do they intend to use the system?
4. What is the user's level of knowledge, skill, expertise?
5. With what environmental characteristics must the system comply?
6. What are existing and planned interfaces?
7. What functions will the system perform, expressed in user language?
8. What are the constraints (hardware, software, economic, procedural) to which the system must comply?
9. What will be the final form of the product: such as model, prototype, or mass production?

The requirements that result from requirements analysis are typically expressed from one of three perspectives or views. These have been described as the Operational, Functional, and Physical views. All three are necessary and must be coordinated to fully understand the users' needs and objectives. All three are documented in the decision database.

1. **Operational view** - The Operational View addresses how the system will serve its users. It is useful when establishing requirements of "how well" and "under what condition." Operational view information should be documented in an operational concept document that identifies:
   A. Operational need definition.
   B. System mission analysis.
   C. Operational sequences.
   D. Operational environments.
   E. Conditions/events to which a system must respond.
   F. Operational constraints on system.
   G. Mission performance requirements.
   H. User and maintainer roles (defined by job tasks and skill requirements or constraints).
   I. Structure of the organizations that will operate, support and maintain the system.
   J. Operational interfaces with other systems.
2. **Functional view** - The Functional View focuses on WHAT the system must do to produce the required operational behavior. It includes required inputs, outputs, states, and transformation rules. The functional requirements, in combination with the physical requirements shown below, are the primary sources of the requirements that will eventually be reflected in the system specification. Functional View information includes:
   A. System functions.
   B. System performance.
      1. Qualitative — how well?
      2. Quantitative — how much, capacity?
      3. Timeliness — how often?

C. Tasks or actions to be performed.
D. Inter-function relationships.
E. Hardware and software functional relationships.
F. Performance constraints.
G. Interface requirements including identification of potential open-system opportunities (potential standards that could promote open systems should be identified).
H. Unique hardware or software.
I. Verification requirements (to include inspection, analysis/simulation, demo, and test).

3. **Physical view** - The Physical View focuses on HOW the system is constructed. It is key to establishing the physical interfaces among operators and equipment, and technology requirements. Physical View information would normally include:
   A. Configuration of System:
      1. Interface descriptions,
      2. Characteristics of information displays and operator controls,
      3. Relationships of operators to system/physical equipment, and
      4. Operator skills and levels required to perform assigned functions.
   B. Characterization of Users:
      1. Handicaps (special operating environments),
      2. Constraints (movement or visual limitations).
   C. System Physical Limitations:
      1. Materials limitations (capacity, power, size, weight).
      2. Technology limitations (range, precision, data rates, frequency, language).
      3. Government Furnished Equipment (GFE).
      4. Commercial-Off-the-Shelf (COTS).
      5. Non-developmental Item (NDI), reusability requirements.
      6. Necessary or directed standards.

Requirements are system and project level data sets (or, issues:

1. Requirements are the design decisions about what the system will do.
2. Requirements are the set of things that we have decided should matter and be completed by the conclusion of the project.

The properties of the system that we have decided to define and control (manage) through the engineering process. Not all properties of a system are requirements. Requirements are not a statement of intent or a directive, they are not the users needs, they are what a specific system, with specific system boundaries, is actually going to do.

6.5.3 Requirements analysis through prioritization

INSIGHT: *Motion requires input, input is constrained, therefore motions are prioritized.*

Requirements prioritization is a decisioning process. Requirements necessitate prioritization because they concern limitation.

During requirements triage, relative priorities are established for requirements, and resources needed for their achievement are identified and assessed. Then requirements are packed in subsets, and each subset is evaluated against the probability of such subset being a success.

Methods for establishing the prioritization of requirements include:

- Scale of rankings (e.g., 1-4; must, should, could), voting schemes, weightings, value-based (i.e., user-based given available resources), etc.

Prioritization categories include:

1. **Must have** requirement (mandatory, shall).
2. **Should have** if at all possible (high importance).
3. **Could have** but not critical (low importance).
4. **Will not have** this time (delayed importance, does not matter).

NOTE: *This prioritization scheme parallels the Habitat Service System's operational decisioning prioritization process (Criticality Response).*

The design of the habitat service system naturally breaks down into a series of criticality systems, of which, life support is of the highest prioritization. Herein, facility systems (another top-level habitat service system) is a could have, but not critical.

There are different ways of approaching prioritization, which vary (at least) by type of requirement:

1. Market requirements include:
   A. **Financial requirements** will determine financial constraints ("budget"). Financial constraints determines resources purchased.
2. Real-world requirements include:
   A. **Need requirements** will determine service constraints. Service constraints determine functions selected.
   B. **Material requirements** will determine material constraints. Material constraints determine materials selected.
   C. **Social (navigational) requirements** will determine decision constraints. Decision constraints determine the new state of the

habitat.
3. Technology access includes:
   A. Technology readiness matrix.
   B. Technology integration matrix (integrated simulating system).
   C. Technology material composition cost table.

6.5.4 Requirements analysis through evaluation (Quality management)

The evaluation of requirements is carried out under quality control/management. Procedures used together for checking that a system (service or product) meets requirements and specifications, and that it fulfills its intended purpose.

> **NOTE:** *Requirements evaluation is a critical component of a quality management system (e.g., ISO 9000).*

Requirements are capable of evaluation because they are:

1. Requirements are the foundation from which quality is measured. Lack of conformance to requirements is lack of quality.
2. Specified standards define a set of development criteria that guide the manner in which a system is engineered. If the criteria are not followed, lack of quality will likely result.

The factors that affect quality can be categorized in two broad groups:

1. Factors that can be directly measured (e.g., defects per function-point).
2. Factors that can be measured only indirectly (e.g., usability or maintainability).

In each case, measurement must occur. We must compare the system (documents, programs, data) to some datum and arrive at an indication of quality. Quality factors focus on three important aspects of a product:

1. Its operational characteristics.
2. Its ability to undergo change.
3. Its ability to adaptability to new environments.

6.5.5 Engineering assurance

> *A.k.a., Engineering certainty, quality assurance, systems engineering structured assurance, project assurance, systems evaluation, qualification, examination, acceptance, requirements assurance, quality assurance.*

Verification and validation (V&V) mean the same thing within a non-technical context, but in the framing of simulation quality they have quite specific technical meanings. Each involves the accumulation of evidence that correctness (alignment) is present.

> **IMPORTANT:** *Verification and validation rely on a source's ability to specify the objective(s) correctly (accurately and fully).*

Validation and **verification** are prerequisite to sufficient user acceptance of a new system. Verification and validation can processes can be applied [at least] to models and to systems engineering.

The model view of validation and verification:

1. **Verification** is the determination of whether the model (e.g., specified requirement) is being solved correctly.
2. **Validation** is the determination of whether the model (e.g., specified requirement) is correct.
   A. Validation necessarily involves observational or experimental data, and its comparison to the simulation (e.g., operating system).
   B. A necessary observation is that validation involves several error modes that color any comparison:
      1. The size of the numerical error in solving the model.
      2. The magnitude of the experimental or observational error.

The systems engineering assurance views:

1. **Verification** - testing to confirm the system and its performance align with the specification/ requirements. Confirm or dis-confirm (and to what degree) a system as aligning with its specified requirements. System verification is assuring that the system is built right.
   A. **Evaluation** *(design view; a.k.a., assessment)* - whether or not a system complies with specified requirements or imposed conditions. Evaluation questions may include: How is the requirement verified (confirmed)? How will testing demonstrate proof of correctness? Has the system been built right for the user; is the system verified (or dis-confirmed)? System evaluation is assuring quality (a.k.a., quality assurance) and function (a.k..a, functional evaluation/assessment). For instance, what is the baseline of operation, and was it met?
   B. **Testing** (development view) - whether or not a system reliably complies with specified requirements or imposed conditions. The two types classified by their effect on the system include: non-destructive examination (NDE) and destructive examination (DE).
2. **Validation** *(user view)* - user confirmation of

requirements completion. Has the right system been built for the user? System validation is assuring that the right system is built for the intended user environment.

6.5.5.1 System requirement engineering

NOTE: *It is normal to find faults with a design after a period of operation.*

Systems engineering is used to realize viable systems that satisfy user needs.

1. **Iterative** - the repeated application of a process to the same system or sub-system to correct/solve a discovered discrepancy or variation from requirements (apply the process again and again until correction is complete).
2. **Recursive** - the repeated application of a process to design the next lower layer (or level) of the system, or to realize the next higher integrated layer (or level) of the system.

6.5.5.2 System verification

System verification requires the input of a system definition:

1. **The definition verification process:** compare the definition of the system, and the system's design specification, and show that the system design specification meets, or does not meet, the system's [objective] definition.
   A. If it is not possible, given the information available, to match the system's behavior (as a design specification) to its definition (Read: its model), then scientific inquiry is required -- all that can be done is to do an experiment to see if the system observably behaves like the model (Read: the definition).
2. **The evaluation process** - a mechanisms that provide a designer with critical feedback on the usability, feasibility, etc. of the system.

The three primary engineering design and development problems for a system are:

1. Describe what the system does.
   A. What does the system do?
   B. What is the system's purpose, function, objective, operation, utility?
2. Describe pre-conditions for the systems operation (I.e., for using the product).
   A. What does the system require to operate?
   B. Under what environmental conditions will the system operate?
3. Describe the system's interfaces (material, visual, logical, mathematical, etc.).
   A. With what, and how, does the system interface?

NOTE: *Development involves a creation (analysis-synthesis) life cycle based on evolving prototypes, and the evolution of the development method itself.*

6.5.6 Requirements management

Requirements "management" is the process by which changes to requirements are decided and remembered throughout the system life-cycle. Requirements change because:

1. Knowledge develops.
2. User requirements change.
3. Organizational value-set changes.
4. The environment changes.

NOTE: *It is almost impossible to have requirements traceability without implementing the requirements in some automated context. Therein, a requirements coordination tool (visual interface, database, and processing) is generally necessary to assist in the coordination of a large number of requirements.*

Requirements interface support (i.e., a requirements coordination tool functions to):

1. Supports elicitation
2. Support access by means of browse, find, retrieve, and generate reports of requirements based on selected criteria.
3. Supports forward and backward traceability.
4. Supports the generation of correct linguistic and logical requirements.
5. Supports change control and change impact assessment
6. Supports functional allocation and functional-to-physical translations.
7. Does not enforce any particular requirements engineering process.

6.5.6.1 Requirements hierarchy

A hierarchy of requirements with system requirements leading to sub-system requirements. Traceability within the requirements hierarchy is essential so that requirements always have a causative presence. In systems engineering the terms 'forward' and 'backward' traceability provide a awareness of direction (and how they relate) within the hierarchy.

1. Forward traceability is from the system level requirement(s) to the sub-system level requirement(s).
   A. Are the system's requirements met by the sub-system's design?
2. Backward traceability is from the sub-system level

requirement(s) to the system level requirement(s).
   A. Are the sub-systems able to meet their requirements, and if not, what system level requirement may be at risk [of not being met]?
   B. Is there requirements "creep" occurring, where sub-system requirements are being created for non-existent system requirements?

These can be:

1. Functional requirements - what is the thing going to do.
2. Performance requirements - how well is the thing going to do it.
3. Resource requirements - how many resources does the thing need to do it.

Requirements are:

1. Conceived.
2. Allocated.
3. Executed.
4. Closed.

Requirement information need:

1. Information category.
2. Measurable concept.
3. Leading insight.

6.5.6.2 Requirements engineering

Requirements engineering represents a series of engineering decisions that lead from recognition of a problem to be solved to a detailed specification of that problem and its resolution.
Requirements [engineering]

1. Articulating requirements.
2. Modeling and analyzing requirements.

The two most common characteristics of requirements are that they:

1. Requirements may have interdependencies.
2. Requirements are organized in subsets that hierarchically map value to users.

6.5.6.3 Requirements engineering tools

There are a large number of tools that may assist in requirements engineering, including:

1. Context diagram.
2. Functional flow block diagrams.
3. Requirements breakdown structure (RBS).
4. N2 diagrams.
5. Structured analysis.
6. Data flow diagrams.
7. Control flow diagrams.
8. IDEF diagrams.
9. Behavior diagrams.
10. Action diagrams.
11. State/mode diagrams.
12. Process flow diagrams.
13. Functional hierarchy diagrams.
14. State transition diagrams.
15. Entity relationship diagrams.
16. Structure analysis and design.
17. Object-oriented analysis.
18. Unified modelling language (UML).
19. Structured systems analysis.
20. Design methodology.
21. Quality function deployment.

6.6 System requirement constraints

Both resources (material boundaries) and constraints (information boundaries), as well as time, are elements a system uses for transforming inputs into outputs. Cost and schedule limit the solution space (in market, "tradespace") and as such traditional categorization of requirements include them as requirements or constraints. Development cost and schedule can be perceived as resources because they are consumed during system development. However, these resources are not consumed during development by the system, but by the project developing the system, and therefore they would actually reflect project constraints and not system ones. On the other hand, it could also be argued that time and cost are indeed consumed by the system during its creation, which would bring them back as resource requirements to the system.

Consequently, the present research proposes to allocate development cost and schedule requirements in one of the following two categories, depending on the vision and needs of each project:

1. System development requirements - requirements defined for the system's development.
   A. System development resources - What resources are consumed and/or cycled by the system during its development?
2. System operation requirements - requirements defined for the operational phase of the system, i.e., how much money is required to operate the system at the specified performance levels. Operational cost requirements inherently belong to the resource category, as it is something a system uses to fulfill its functions.
   A. System operational resources - What resources are consumed and/or cycled by the system during its operation?

6.7 Requirement expression: standards (Categorical, linguistic)

A requirement is an imperative. Other imperatives include needs, goals, directives, and objectives. Statements in this plan contain the following imperatives:

1. *Shall* are used for binding requirements that must be verified and have an accompanying method of verification. Shall is a binding provision.
2. *Will* is used as a statement of fact, declaration of purpose, or expected occurrence. Will is a declaration of purpose.
3. *Should* denotes an attribute or best practice that must be addressed by the system design.
4. *May* denotes a non-binding attribute or provision.
5. *Must* denotes the expression of either a constraint, a certain quantity, or a performance requirement (non-functional requirement).

Principles for usage include:

1. Use exactly one provision or declaration of purpose (such as shall) for each requirement, and use it consistently across all requirements.
2. When used within the context of a reference document under an agreement, the verbs shall, will, and should are only intended as informational and are not binding.

6.7.1 Requirement expression: format[ion] structure

A requirement must be in the form[ation] or structure of a complete "information package" (e.g., sentence).
A requirement must state a subject and predicate where the subject is a user.
The requirements must have, and state, an end result.

A requirement list/set must be consistent in its usage of the "to be" verb:

1. *Will* or *must* to show mandatory nature.
2. *Should* or *may* to show optionality.

Here are a few basic requirement sentence structures they can apply consistently. A very basic format is:

- Unique ID: Object + Provision/Imperative (shall) + Action + Condition + [optional] Declaration Of Purpose/Expected Occurrence (will)
  - For example, 3.1.5.3: The craft shall perform one complete fly-around (of the tower) at a range of less than 250 meters as measured from the craft center of mass to the tower center of mass; after undocking from the tower (and no declaration of purpose).

Table 14. Engineering Approach > Requirements*: Requirement types and their associated syntax patterns.*

Requirements Type	Syntax Pattern
Ubiquitous	The <system name> shall <system response>
Event-Driven	When <trigger><optional pre-condition> the <system name> shall <system response>
Unwanted	If <unwanted condition or event> Then, the <system name> shall <system response>
State-Driven	While <system state>, the <system name> shall <system response>
Optional Feature	Where <feature is included>, the <system name> shall <system response>
Complex	<Multiple conditions>, the <system or unit name> shall <system or unit response>. (combinations of the above patterns)

An guiding objective of requirement defining is:

- Minimizing the amount of necessary requirements by eliminating overlapping requirements while ensuring the system is completely specified.

6.7.2 Requirements development

The process of requirements development requires all of the following phases and descriptions, occurring synchronously:

1. Define user:
   - A. Who is interested in the system?
   - B. How are decisions resolved?
   - C. Who are the users and developers?
2. Define goals (objectives):
   - A. Define broad (coarse) goals (non-specific goals)? What should be implemented or achieved?
   - B. Broad goals divided into more specific goals (granular goals)? What should be implemented or achieved?
3. Define requirements:
   - A. Goals (objectives) can be derived into concrete requirements that describe how the goals will be achieved and fulfilled.
   - B. A requirement is:
     1. A specific statement of need derived from a goal.
     2. A specific statement(s) of reason (rationale) for the need including a relevant context.
     3. A specific explanation for how to achieve or fulfill (i.e., get) the requirement(s) in the context of a goal?
   - C. Visualize and model the requirements in order to appropriately communicate and construct the how.

All requirements in a requirements list are composed of at least the following inputs:

1. **Requirement unique identifier:** Each requirement shall be assigned a project-unique identifier to support testing and traceability.
   A. Each requirement throughout the information system must be tagged with a project unique identifier (PUI). Tagging each requirement with a PUI optimizes traceability between high-level and low level requirements, and between requirements and verification tests. Each requirement should be marked with a PUI that allows users to easily reference both the requirement and its position in the overall document.
2. **The requirement statement:** Each requirement shall be stated in such a way that an objective test can be defined for it. An 'objective test' is a test for which the result can be commonly experienced.
3. **The requirement rationale (justification or reasoning)** - Each requirement shall include a rationale statement(s). When a requirement's rationale is visibly and clearly stated, its defects and shortcomings can be more easily spotted, and the rationale behind the requirement will not be forgotten. Rationale statements also reduce the risk of rework, as the reasoning behind the decision is fully documented and thus less likely to be re-rationalized

Requirement unique identifier:

For example: 3.5.2.5

- 3 = Transportation and Service requirements.
- 5 = Entry/landing requirements.
- 2 = Contingency.
- 5 = Space ventilation for emergency landing.

6.7.2.1 Requirement construction qualities

Requirements should posses (i.e., presence and not absence of) the following quality attributes:

1. **Complete** – precisely defines the system's responses to all real-world situations the system will encounter.
2. **Consistent** – does not contain conflicts between requirements statements.
3. **Correct** – accurately identifies the conditions of all situations the system will encounter and precisely defines the system's response to them.
4. **Modifiable (configurable)** – has a logical structuring with related concerns grouped together.
5. **Ranked (ordered)** – organizes the specification statements by importance and/or stability (which may conflict with the document's modifiability).
6. **Traceable** – identifies each requirement uniquely. A requirement must be traceable to some source. Each requirement should have a unique identifier allowing the software design, code, and test procedures to be precisely traced back to the requirement.
7. **Unambiguous** – states all requirements in such a manner that each can only be interpreted one way.
8. **Valid** – all project participants can understand, analyze, accept or approve it.
9. **Measurable** - functions can be assessed quantitatively or qualitatively.
10. **Verifiable** – must be consistent with related specifications at other (higher and lower) levels of abstraction.

Requirements must also be:

1. **Uniquely identifiable** - Each need is stated exactly once to avoid confusion or duplicative work. Uniquely identifying each requirement is essential if requirements are to be traceable and able to be tested.
2. **Performance specified** - Statements of real-world performance factors are associated with a requirement.
3. **Testable** - All requirements must be testable to demonstrate that the end product satisfies the requirements. To be testable, requirements must be specific, unambiguous, and quantitative whenever possible.

Simplistically, requirements must be:

1. **Conceived** - constructed.
2. **Bounded** - constrained.
3. **Coherent -** logically related, internally and externally.
4. **Acceptable** - sufficient input to resolve a design.
5. **Addressed** - scheduled, allocated, assigned.
6. **Fulfilled** - actualized.

Engineering is a real world creation process, and hence, requirements therein must possess the following characteristics (i.e., to be a "good" requirement):

1. Fulfill real world needs.
2. Have clear meaning.
3. Are organized coherently.
4. "Drive" engineering.

Requirements are prioritized:

1. **Terminal requirements** - A terminal requirement

is a statement in specific and measurable terms that describes what the system will be able to do, to be, or enable a user to do or be as a result of engaging with the system. A terminal requirement should be created for each of the tasks addressed within the system. Terminal requirement describe results, and not processes. After the terminal objective is created, it should be analyzed to determine if it needs one or more enabling/ supporting requirement. Each written requirement should include a task/performance, condition, and a standard:

A. Task or Performance: States what the system will be doing.
B. Condition: Specifies under what conditions the system should perform the task (defines the quality of performance of the system).

2. **Enabling/supporting requirement(s)** - are supporting or enabling requirements for terminal requirement. They are created by analyzing a terminal requirement. They allow the terminal requirement to be broken down into smaller, more workable and stabler requirements.

6.7.2.2 Requirement syntax

Requirements ("What") are a communications link between the source model and the implementation model ("How").
Herein, specificity and numeric measure are required for performance. Stating that a system should do something "quickly", is not a performance requirement, since it is ambiguous and cannot be verified. Stating that opening a file should take less than 3 seconds for 90% of the files and less than 10 seconds for every file is an appropriate requirement.

Instead of providing a unique section on performance requirements, include the relevant information for each feature in the statement of functionality.

Requirements are syntactically delineated:

1. When?/Under what conditions? (Phrase conditions)
   A. the system
   B. *shall / should / will* (Type of obligation from imperative)
   C. verb <process>; provide <whom> with the ability to; be able to <process>
   D. object
   E. additional details about the object.

6.7.2.3 Requirements tracing (traceability)

Tracing requirements means relating specific requirements to other project elements, especially to the following:

1. Backward tracing - a requirement to its source.
2. Traceability matrix - one requirement to another.
3. Forward tracing - a requirement to its design, code, documentation, or other forward project elements.

Simply, backward tracing ensures a source for each requirement. A traceability matrix ensures it is possible to evaluate the effect of changes to requirements among other inter-related requirements. Forward tracing ensures changes to requirements flow through to the design, code, project plan, etc. Forward tracing to the project plan provides data on how much work has been completed, and how much remains.

6.7.2.4 System requirements modeling

Requirements modeling is the process of constructing abstract, formal representation of the initial textually described system requirements in a way that is amenable to unambiguous interpretation, producing a requirements specification. This process ends with a requirements model (specification), which is expected to capture as much of the relevant real world semantics as possible. The core of the input system's requirements is a functional or behavioral, and non-functional, breakdown. This data based breakdown lists:

1. What the users need?
2. What the system must do to satisfy their needs?
3. What components must be built?
4. What each component must do, and how they will interact?

In the subsequent phases of the development process, the requirements model is elaborated and transformed into the design model (the [design] specification). This transition emphasizes the critical need for creating a formal, accurate, and complete requirement model from the outset, as it designed serve as the foundation of the entire development process and continued service life cycle.

Modeling is targeted at clear and accurate representation of the concepts that comprise the system. An important benefit of requirements modeling is that since the resulting model is available at an early stage in the system's life cycle, model analysis and simulation may be used to validate the requirements and reduce conceptual design errors. Later on, the requirements modeling is integrated into the life cycle flow of activities in the development process.

A good requirements specification is one in which requirements are arranged hierarchically. Few high-level, broadly defined requirements are specified in increasing levels of detail, where each level contains a set of requirements that elaborate upon one or more requirements at the level above it. A hierarchical structure of requirements may facilitate the process of modeling. In general, high-level requirements correspond to abstract processes, aggregate objects or agents (actors), and interactions between them at lower levels.

Each requirement is a specification relating to some characteristic of a system. Model components are introduced and associated with a corresponding requirement (or requirements set) with which the model component is related, creating the objective condition of traceability.

In community, there is no necessary conceptual gap between engineering objectives and human objectives; they seamlessly become one and the same. In the market, however, there is a gap between customers, employers, and employees, and also between client's, engineering, and business. In other words, in the market, a conceptual gap, which is often very wide, exists between these two [problem-requirement] model types, since one faces the client with a problem domain, while the other faces the solution domain provided by a semi-independent business entity whose role is, a technological solution provider. The result of the gap entails a host of consequences, including the necessity for subjective decisioning, and therein, the introduction of various subjective biases that carry on over time and become systemic [to the societal system].

APHORISM: *Share information about fishing to a human and s/he can fish for a lifetime.*

During the continuous modeling process, issue tracking attributes track the source and status of a requirement. This is basic issue tracking, allowing for requirements traceability. The attributes are:

1. Record the source.
2. Record the urgency (urgency spectrum).
3. Record the sufficiency of data to resolve (decisioning).
4. Identify verification method (test, demonstration, inspection, simulation, analysis).
5. Identify constraints (safety, performance, reliability, contracts, standards, rules).
6. Record integrations (specifications).

6.7.2.5 Requirements gap analysis (project coordination task)

If requirements are not available, or not yet well understood, then an gap/requirements analysis (and possible discovery) must be complete to determine what is missing (define the gap or design space between what is present and what is expected). The purpose of Requirements Analysis is to discover unknown requirements (i.e., to turn unknown requirements into known requirements).

6.7.3 Requirement sub-types

Requirements can be classified by the presence of a function into functional requirements and non-functional requirements. There are two primary requirement subtypes divided by function. Herein, qualitative is the conceptual encoding of function, whereas functional is the physical encoding of function:

1. **Non-functional requirements (Qualitative requirements)** that become encoded into the "behavior" of a function, or "status" of a state) - conditions that must be met that are not explicit capabilities.
2. **Functional requirements** (specify an exact function) - capability that the system must perform.

Functional requirements/metrics are capabilities (as physicalizable states or processes) that the product or service (as a system) must perform. Functional requirements meet functional user needs. These are the most fundamental of physicalizable requirements. In the market, fundamental functional requirements are generally referred to as "business" requirements, because they are what the "business" needs to survive. In community, fundamental functional requirements are sometimes referred to as human requirements or human needs, because they are what individual humans in common need to survive and thrive. In society, these functional human requirements are built upon a set of human needs and objectives.

Note: What one person senses, another may sense differently, thus the need for clear communication and preferentially, electric instrumentation where possible. At the level of a project, there is a need for requirements to be referenceable (i.e., traceable) to their "tested" results, which may be verified or not.

Non-functional requirements/metrics can be visualized as the encoding of conception into a real world reality by "shaping" its iterative expression; like batter being pushed through a cookie cutter shape to form individual iterations of that cookie. After application, non-functional requirements become operational (i.e., concepts in operation). The term for a concept in operation usually ends in -ability: usability, dependability/reliability/durability, mobility, scalability, sustainability. For instance, a system can be designed to be created and operated sustainably. When in operation, the system may continue, or not, to remain sustainable, through its continued design and operation.

In general, non-functional requirements are sourced from a value system. A value system is effectively a set of non-functional requirements. The value system forms objects, which then forms the non-functional requirements. Non-functional requirements per definition do not describe what functionalities the platform will deliver, but how they will be delivered. There are two "hows" here:

1. How will the system be produced (i.e., under what quality conditions).
2. How will the system operate (i.e., under what quality conditions).

Values are the translation of concepts into operation; they are a higher level abstraction than concepts in operation. Values become operational concepts (Concepts in Operation), and the real, existent and functioning systems they create are described through a Concept of Operation (a system's high-level conception, abbreviated ConOps, CONOPS, CONOPs, or CONOps). A 'Concept of Operation' document/dashboard describes the characteristics of a proposed system from the viewpoint of an individual who will use the system. It socially communicates the quantitative and qualitative system characteristics of a potentially, or actually existent, system. The concepts used in that document, translate through active human "experience" into physical interactions that lead to the physical construction of a physically operational system. That system may operate well, in concern to its requirements by the user, or it may not.

6.7.3.6 System-level Requirement categorizations

Requirements about a system to be developed are sometimes categorized by their source (point of origin) and/or system's component. Take note that there is, however, only one unified set of structured requirements for any systems engineering project.

The primary requirement types are:

1. **Functional requirements (Do):** Requirements that define what the system must do in essence, or, in other words, what it accepts and what it delivers (i.e., expected transformation). For example: The system shall accept coins; The system shall transmit 4 signals; The system shall convert sea water into drinkable water.
2. **Performance requirements (Being):** Requirements that define how well the system must operate, which includes performance related to functions the system performs or characteristics of the system on their own, such as -illities. What values and qualities will the system express. Examples include: The system shall move at a speed higher than 30 km/h; The system shall have a reliability better than 0.80.
3. **Resource requirements (Have):** Requirements that define what the system can use to transform what it accepts into what it delivers. Examples include: The system shall consume less than 100W; The system shall have a mass of less than 10kg.
4. **Interaction requirements (Interact):** Requirements that define where the system must operate, which includes any type of operation during its life-cycle. Examples include: The system shall withstand shock levels higher than 100g; The system shall operate in vacuum (to reflect operation); The system shall operate in clean room class 10,000 (to reflect Assembly, Integration, and Test activities).

The following are common requirement categorizations:

1. **Project requirements** are statements identifying what is required to complete a project, which starts with coordination requirements and project user requirements.
2. **User requirements** are written from the point of view of end users, and are generally expressed in narrative form, "The user must be able to change the color scheme of the welcome screen."
   A. In contrast to the roles of user (customer) and developer (employee) in the market, in community the roles of "user" and "developer" are the equivalent, meaning that there is no structural separation of requirements. Remember that in business, "users" are customers, and "developers" are employees. In community, there is no business, and hence, no financial separation between users and developers. In community, users are also developers as part of an InterSystem Team structure.
3. **System requirements** are statements describing the functions the system needs to do, and the non-functional states the system needs to be. System requirements are usually more technical in nature, "The system will include four pre-set color schemes for the welcome screen. Colors must be specified for the page background, the text, visited links, unvisited links, active links, and buttons (base, highlight, and shadow)." System requirements may have to do with how the system is built or functions.
   A. What the system will do to meet those needs.
   B. What do we need to know to build this?
4. **Engineering requirements** are statements including numbers and operational concepts that describe the functional dynamics and non-functional states of a proposed system.
   A. **Development requirements** - what is required to do development.
   B. **Construction requirements** - what is required to construct the system.
   C. **Operational requirements** - what is required to operate the system.
5. **Interface requirements** specify how the interface (the part of the system that users see and interact with) will look and behave. Interface requirements are often expressed as screen mock-ups; narratives or lists are also used. A description of the information (protocol and physical) interface between components of a system.

6. **Component requirements** specify a descriptive list of all things that each component must do and/or be.
7. **Material requirements** specify the type, quality, and quantity of materials required.
8. **Negative requirements** refers to the create boundaries to what the system should do. However, it is not always possible to measure what a system should not do; because, how can "you" test something that should not happen.
9. **Constraining requirements** (a.k.a., non-functional requirements or objectives) - requirements that constrain implementation and operation of a system.
10. **Project plan coordination requirements** refers to those requirements specifying what the project coordination system should do (and be) to coordinate information and resources. These requirements are for the continuous process of project coordination, and not the system to be engineered, which has its own set of requirements. Technically, they are both engineering requirements.

6.7.3.7 Project-level requirement categories

At the project level, there are several primary categories of requirement:

1. **Information requirements**
   A. Research requirements
   B. Design/production requirements (to produce the deliverable of a societal system specification)
   C. Engineering/Operation requirements
   D. Project plan/coordination activities
2. **Material requirements**
   A. Material resources for coordination,
   B. Material resources for design.
   C. Material resources for construction.
   D. Material resources for operation.
   E. Material resource for cycling.
3. **Human requirements**
   A. Human presence for knowledge.
   B. Human presence for effort and capabilities.
4. **Operational Requirements**
   A. The operational requirements should answer:
      1. *Who* is asking for this requirement? Who needs the requirements? Who will be operating the system?
      2. *What* functions/capabilities must the system perform? What decisions will be made with the system? What data/information is needed by the system? What are the performance needs that must be met? What are the constraints?
      3. *Where* will the system be used?
      4. *When* will the system be required to perform its intended function and for how long?
      5. *How* will the system accomplish its objective? How will the requirements be verified?
5. **Market-State requirements (Financial and contractual requirements)**
   A. In the market-State where resources are not held as the common heritage of all of humanity, resources carry transactional costs (e.g., trade goods, bartering service, and currency).
      1. Financial requirements (a.k.a., currency costs)
         i. Contractual requirements (which are really financial requirements)
         ii. Financially feasible conditions for creation and operation of society.
         iii. Multiple types of resource costs: hardware, software, land, manufacturing, logistics and assembly, State and legal (jurisdictional), and expertise.
      2. Contractual requirements (i.e., where force is above financial requirements)
   B. Frequency of cost:
      1. One-time (accounting for repair or replacement)
      2. Marginal (no additional after setup)
      3. Reoccurring (cyclical)

6.7.3.8 Societal-level requirement input types

At the societal level, there are several primary categories of requirement:

1. **Human [end-]user requirements:** Human needs, wants, and preferences - describe generally the needs, goals, and tasks of the user (this is the end user; there is no market-based project requester). All measurements of quality, success, and optimization relate to the user, who is the individual human being in Community. User requirements specifically refer to user fulfillment.
   A. What do "we" need, want, and prefer as a human individuals interconnected within a global societal structure? What is required to work together, to integrate, share information openly, to perceive and act upon "our" inter-connectedness.
2. **[Societal] System requirements:** Community-type society instantiation requirements - a description of the societal system itself and what the system must do. What informational (through to material) systems require to sustain the current instantiation (iteration) of the societal system? These are requirements that describe the capabilities of the system with which, through

which, and on which humans maintain their society (i.e., function together). Note: Technically, everything is information, from conception through into materialization; hence, the habitat service (material) system is a sub-system of the information system. Here, high-level functions and logic are defined.

A. **Information System instantiation requirements:** These are requirements with which, through which, and on which humans maintain their information system's instantiation.
   1. Social; decision; lifestyle; material (Subsystem-level functions and logic are defined here).

B. **Habitat Service (Material) System instantiation requirements:** These are requirements with which, through which, and on which humans maintain their material habitat service system's instantiation.
   1. Ecological services, Life Support Service, Technical Support Service, and Facility Support Service.

6.8 *Requirements standards*

The principal standard defining a requirement is:

- ISO/IEC/IEEE 29148: System and software engineering - Life Cycle Processes - Requirements Engineering

The five key deliverables of the ISO/IEC/IEEE 29148 standard are:

1. Stakeholder requirements specification (StRS; user requirements specification) document.
2. System requirements specification (SyRS) document.
3. Software requirements specification (SRS) document.
4. System concepts documents.
   A. System operational concept (OpsCon) document.
   B. Concept of operations (ConOps) document.

6.9 *Requirements engineering*

A.k.a., Requirements engineering process, requirements definition stage.

Requirements engineering is the iterative process of establishing the services that the user requires from the solution system and the constraints under which it is to be developed (e.g., development conditions) and under which it operates (e.g., service conditions, value conditions). The processes used for requirements engineering vary widely depending on the application domain, the project type, and the organization developing the requirements. In practice, requirements engineering is an iterative activity in which requirement tasks/activities/processes are iterated and inter-related.

The collection and analysis of information known as requirements engineering happens continuously throughout the project's life cycle. Therein, requirements require the following actions:

1. Requirements analysis involving identification, rationale, positioning and prioritization.
2. Show where work is required to resolve/complete requirement.

Requirements engineering involves, but is not necessarily limited to, the [iterating/spiralling] requirements process tasks of (a.k.a., generic requirements engineering activities, requirements engineering stages):

1. **Requirements coordination** (a.k.a., requirements management) - all coordination tasks/processes/stages associated with the information set, 'requirements'. Of significant importance here are the processes of tracing/tracking and changing (change controlling) requirements.
2. **Requirements discovery** (a.k.a., requirements collection, requirements elicitation, requirements solicitation, requirements identification, gathering requirements) - the process of identifying all requirements. Discovery may involve interviews, evaluations, observation and study, scenarios, use cases, work/model flow diagrams, sequence diagrams, activity diagrams, event diagrams, decision trees, etc. Identify all requirement sources.
3. **Requirements analysis** - the process (technique) of understanding user needs (requirements) and translating (transferring) them into a set of requirements for system construction and/or modification.
   A. **Requirements classification and organization** - grouping related requirements and organizing them into coherent clusters.
   B. **Requirements prioritization** - Prioritizing requirements and resolving requirements conflicts.
   C. **Technical requirements validation** - the process of checking the requirements for their expected attributes, including: validity; consistency, completeness, realism, verifiability, etc. Are there technical errors; conflicts; ambiguities; and does the requirement (and requirements specification) conform to standards?
4. **Requirements specification** - the collection of requirements necessary to complete the project

into a formal document/database.
5. **Requirements verification** - technical verification that the system operates as required. Proving [objectively] that each requirement is satisfied. Can be done by logical argument, inspection, modeling, simulation, analysis, test, or demonstration.
6. **User requirements validation** - user validation that the system can be (or, is being) used as expected.

Note that the above is sometimes more simply depicted as a [repeating] four/five stage cycle:

1. Requirements discovery.
2. Requirements classification and organization (grouping).
3. Requirements prioritization.
4. Requirements specification (repeats here).
5. Requirements verification (or, repeats here).

The simplified requirements engineering process:

1. User requirements definition:
   A. Inputs:
      1. Source documents.
      2. User needs.
      3. Project constraints.
   B. Activities:
      1. Articulate demands.
      2. Define user requirements.
      3. Analyze and maintain user needs (priority demands).
   C. Outputs:
      1. Concept documents.
      2. User requirements.
      3. Measures of effectiveness needs.
      4. Measures of effectiveness data.
      5. Validation criteria.
      6. Traceability.

6.9.1 Requirements coordination planning

Requirements coordination planning decisions include, but are not limited to:

1. **Requirements identification** - Each requirement must be uniquely identifies so that it can be cross-referenced with other requirements.
2. **A change control process** - This is the set of activities that assess the impact of changes to requirements.
3. **Traceability structures** - Information structures that define the relationships between each requirement and between the requirements and the system design.
4. **Requirements tool support** - Tools that support coordination and planning, such as spreadsheets, databases, and content coordination (content management) systems.

6.9.2 Requirements definition

Requirements definition is a stage in the project coordination and systems engineering life-cycle. The primary goal of this stage is to develop a basis of mutual understanding between the users and the development team about the requirements for the project. The result of this understanding is an selected (approved) 'requirements specification' that becomes the initial baseline for product design and a reference for determining whether the completed product performs as the system user requested and expected. All system requirements, (e.g., software, hardware, performance, functional, infrastructure, etc.) should be included.

This stage involves analysis of the users' processes and needs, translation of those processes and needs into formal requirements, and planning the testing activities to validate the performance of the product.

6.9.2.1 Define system requirements

Use the project scope, objectives, and high-level requirements as the basis for defining the system requirements. The questions used to define the objectives may be helpful in developing the system requirements. The goals for defining system requirements are to identify what functions are to be performed on what data, to produce what results, at what location, and for whom. The requirements must focus on the products that are needed and the functions that are to be performed. Avoid incorporating design issues and specifications in the requirements. One of the most difficult tasks is to determine the difference between "what" is required and "how to" accomplish what is required. Generally, a requirement specifies an externally visible function or attribute of a system (i.e., "what"). A design describes a particular instance of how that visible function or attribute can be achieved (i.e., "how to").

When requirements are being defined, it is not sufficient to state only the requirements for the problems that will be solved; instead, all of the requirements for the project must be collected.

> **NOTE:** *It is often difficult for a non-specialist to understand technically written requirements and their implications.*

6.9.2.2 Writing requirements

Requirements are written in several different notational forms, including:

1. **Natural language** - The requirements are written using numbered sentences in natural language text. Each sentence should express one requirements. Natural language sometimes carries

the problem of a lack of clarity (i.e, precision may be difficult without making the document difficult to read as multiple conditional statements may be requirement and multiple requirements and types of requirements may be expressed together).

2. **Structural natural language** - The requirements are written in natural language text on a standard form or template. Each field provides information about an aspect of the requirement.
3. **Design description language** - This approach uses a language like programming language, but with more abstract features to specify the requirements by defining an operational model of the system.
4. **Mathematical specification (a.k.a., formal specification)** - These notations are based on mathematical concepts, such as finite-state machines or sets.
5. **Tabular notation (a.k.a., table notation)** - The requirements are written in one of the prior four notational forms and placed into a spreadsheet-like table. Generally, tabular notation is used to complement natural language. Tabular notation is especially useful when a number of possible alternative courses of action must be defined. Each row in the table represents a requirement.
6. **Graphical notation** - Graphical models, supplemented by text annotations, are used to define functional requirements for the system; UML use case and sequence diagrams are commonly used.

6.9.3 Requirements specification

A.k.a., Requirements specification document, system requirements specification (SRS).

The requirements for the project are formally documented in the 'requirements specification'. This is the formal (official) statement of what is required of the system developers. All system requirements should be included, however, a definition of user requirements may or may not be included in the document itself. This is the set of selected (agreed) statements on the system requirements. It should be organized so system users and system developers can use it. Simply a requirements specification is a complete description of the behavior of the system and the conditions under which it must be developed and operated. A requirements specification document is "living" during development, and is a reference document for development and operations; it must be maintained over the life of the project. It is the basis (baseline) for the selection of, and agreement on, the system. It also provides a basis (baseline for validation and verification).

The requirements specification becomes the initial baseline (formal reference document) for product design and a reference for determining whether the completed product performs as the system user requested and expected.

The requirements specification should define/establish the environment in which the system to be developed will operate.

Each requirement in the requirements specification should be uniquely identified in a 'requirements traceability matrix'. Each requirements should include an explanation (rationale) of why the requirement is necessary. A requirements specification represents the compilation and documentation of all requirements.

A 'requirements specification' is not technically a design [specification] document. As far as possible, it should define (formally set) what the system should do, rather than how the system should do it. In principle, requirements should state what the system should do, and the design should describe how it does this. However, in practice, requirements and design are inseparable due to the following:

1. A system architecture (system structure) may be designed to structure the requirements.
2. The system may inter-operate with other systems that generate design requirements.
3. The use of a specific architecture to satisfy non-functional requirements may be an external [domain] requirement.
4. Requirements may be a consequence of a [societal] standards requirement.

The following factors should be considered when generating a requirements specification:

1. Select and use a standard format for describing the requirements. Ensure compliance with standards.
2. Present the logical and physical requirements without dictating a physical design or technical solutions.
3. Write the requirements in non-technical language that can be fully understood by the system users.
4. Write the requirements in technical language that can be fully understood by the system developers.
5. Organize the requirements into meaningful groupings.
6. Develop a numbering scheme for the unique identification of each requirement.
7. Select a method for:
    A. Tracing the requirements back to the sources of information used in deriving the requirements (e.g., specific system user project objectives).
    B. Threading requirements through all subsequent life-cycle activities (e.g., testing).

The following factors are generally not included in a requirements specification:

1. Project requirements, such as, delivery schedule,

staffing, reporting procedures, cost. These are included in the Project Plan. If, however, the requirements specification is part of the project plan, then there is nuance here.
2. Design solutions.
3. System assurance plans: quality assurance plans, configuration procedures, verification and validation procedures, etc.

Users of the requirements specification include:

1. **System users** - specify the requirements and read them to check that they meet their needs. Users specify changes to the requirements.
2. **Project coordinators** - Use the requirements document to plan coordination for the system development process.
3. **System engineers** - Use the requirements to understand what is to be developed.
   A. **System test engineers** - Use the requirements to develop validation/verification tests for the system.
   B. **System maintenance engineers** - Use the requirements to understand the system and the relationship between its parts.
4. **Decision system** - Use the requirements to determine risks and societal-level project effectiveness (Read: the decision system effectiveness inquiry).

6.9.3.1 Requirements traceability

A.k.a., Requirements cross-referencing.

Requirements traceability refers to the ability to describe and follow the life of a requirement, in both forwards and backwards direction - from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any life-cycle phase. To ensure traceability, the "life" of a requirement must be documented in a requirements traceability matrix, which allows anyone to find the origin of each requirement and track every change which was made to this requirement.

Using databases allows for easy traceability. For any organization there should exist a requirements database for all possible requirements.

NOTE: *Tracing can be difficult when using multiple tools.*

6.9.3.2 Requirements traceability matrix

The 'requirements traceability matrix' is a requirements coordination tool used to trace project life-cycle activities and work products to the project requirements, and it ensures requirements are traced and verified through the various life-cycle stages, especially during design, testing, and implementation stages. The matrix establishes a thread that traces requirements from identification through implementation. Requirements within the matrix must be traceable from external sources (such as, the user), to derived system-level requirements, to specific hardware and/or software product requirements. In other words, the requirements traceability matrix is a matrix that traces the requirements forward and backward; it traces project requirements back to the project objectives identified in a project charter, for example, and forward through the remainder of the project life-cycle stages.

The requirements traceability matrix is a threading matrix that groups requirements by project objectives. The requirements traceability matrix contains descriptions for each item in the matrix. Under each project objective, the source of the requirement, the unique requirement identification number, and the life-cycle activities are listed in columns along the top and the project requirements in rows along the left side. As the project progresses through the life-cycle stages, a reference to each requirement is entered in the cell corresponding to the appropriate life-cycle activity. The matrix should be capable of being expanded at each stage to show traceability of deliverables (work products) to the requirements and vice versa.

Every project requirement must be traceable back to a specific project objective(s) described in the project's formal direction document (e.g., project charter). This traceability assures that the system (product) will meet all of the project objectives and will not include inappropriate or extraneous functionality or conditions. All deliverables (work products) developed during the design, production, coding, and testing processes in subsequent life-cycle stages must be traced back to the project requirements described in the 'requirements specification'. This traceability assures that the product will satisfy all of the requirements and remain within the project scope.

It is important to know (document, log) the source of each requirement, so that the requirements can be verified as necessary, accurate, and complete.

A copy of the requirements traceability matrix should be placed in the Project File.

6.9.3.3 Requirements diagram

Requirements diagrams show how the different requirements are linked to the block value properties and the hierarchy of requirements that can be used downstream analysis

6.10 Requirements coordination

A.k.a., Requirements management.

Requirements coordination is a process composed of the core processes of tracing and changing requirements, in conjunction with the processes of gathering, organizing, prioritizing, and documenting requirements. Requirements coordination allows for the verification that all requirements have been collected for the system (Read: the product), and that requirements are traceable and changes are controlled effectively and efficiently. Requirements coordination documents the needs, expectations, and understanding of the product to be delivered and provides a framework for identifying, planning, scheduling, verifying, tracing, testing, evaluating, and changing requirements to fulfill user needs (and expectations) of the project.

The processes of gathering, organizing, prioritizing and documenting requirements are based on an interactive communication process that relies on a working relationship between users, the system's developers (the project team), and possibly, the system's operators, to discover, define, refine, and record a precise representation of the system's requirements.

As the project progresses, more requirements may be identified and coordinated through a change control process. As part of requirements coordination, the project coordinator must track requirements that are accepted for the current project and those that will be planned for subsequent releases.

6.10.1 Requirements identification system

The creation of a standard identification system for all requirements is required in order to facilitate control, traceability, and testing activities. The identification system must provide a unique designator for each requirement. For example, the identification system can classify the requirements by type (e.g., functional, input, or computer security). Within each type classification, the requirements can be assigned a sequential number. Select an identification system that is appropriate for the scope of the project.

6.10.2 Requirements change system

As a project evolves, the requirements may change or expand to reflect modifications in the users' plans, design considerations and constraints, advances in technology, and increased insight into user processes. A formal change control process must be used to identify, control, track, and report proposed and selected ("approved") changes. Selected changes in the requirements must be incorporated into the 'requirements specification' in such a way as to provide an accurate and complete audit trail of the changes.

6.11 Requirement definition tasks

The following are common tasks involved in defining system requirements:

1. Define functional requirements.
2. Define non-function/performance requirements.
3. Define input and output requirements.
4. Define user interface requirements.
5. Define system interface requirements.
6. Define communication requirements.
7. Define access requirements.
8. Define backup and recovery requirements.
9. Define preliminary implementation requirements.
10. Develop system test requirements.
11. Develop acceptance test requirements (validation requirements).

In other words, are multiple sub-types of requirements, including but not limited to:

1. **User requirements (a.k.a., user requirements definition)** - Statements in natural language, diagrams, tables, and other notations of the services, or system, and its operational constraints, which are understandable to the user. Written for users (i.e., understandable by end-users who do not have a technical background. What the system should do for the user.
2. **System requirements (a.k.a., product requirements definition)** - Statements in technical language, possibly including diagrams, tables, and other notations that represent a completely detailed description of the system's functions, services, and operational constraints. Written for developers (designers and constructors).
   A. **Data requirements** - Identification of the data elements and logical data groupings that will be stored and processed by the system.
   B. **Process requirements** - Identification of a specified method or language.
   C. **Transitional requirements** - Requirements necessary to transition to a new system.
   D. **Operational requirements** - Systems requirements that specify how the system must operate.
   E. **Maintenance requirements** - System requirements that specify how a system must be maintained (e.g., replacement of parts).
   F. **Sustainment requirements (a.k.a., maintainability requirements)** - Specify how a system must be sustained (e.g., supplied with fuel).
   G. **Retirement requirements** - Specify how a system must be retired from service (e.g., disposal of hazardous materials).
3. **External system requirements (a.k.a., environmental requirements, domain requirements)** - requirements that arise from factors that are external to the system and its development process (e.g., interoperability requirements, legislative requirements, compliance requirements, etc.). External/domain requirements can be new functional requirements, constraints (non-functional requirements) on existing

requirements, or define specific computations. If domain requirements are not satisfied, the system may be unworkable or unsafe. For instance, a train control system has to take into account braking characteristics in different weather conditions.

A. An example: The system shall implement standardization requirements as set out in document #.

6.11.2.1 Identify functional requirements

A.k.a., Define functional requirements.

Identify requirements for all functions, regardless of whether they are to be automated or manual. Describe the automated and manual inputs, processing, outputs, and conditions for all functions. Include a description of the standard data tables and data or records that will be shared with other objects or applications. Identify the forms, reports, source documents, and inputs/outputs that the system will process or produce to help define the functional requirements.

Develop a functional model to depict each process that needs to be included. The goal of the functional model is to represent a complete top-down picture of the system (product). Use flow diagrams to provide a hierarchical and sequential view of the system user's functions and the flow of information through the system.

6.11.2.2 Identify non-functional requirements

A.k.a., Define non-functional requirements, define performance requirements.

Identify requirements for all conditions and constraints the system (and its development) must satisfy.

6.11.2.3 Define input and output requirements

A.k.a., Define information requirements.

Describe all manual and automated input requirements for the system (e.g., data entry from source documents and data extracts from other applications). Document where the inputs are obtained. Describe all output requirements for the system. Document who or what is to receive the output.

6.11.2.4 Define user interface requirements

The user interface requirements should describe how the user will access and interact with the system, and how information will flow between the user and the system.

CLARIFICATION: *'Interfaces' are boundaries that are between elements of a system.*

A standard set of user interface requirements may be established for the system owner organization. If not, work with the system users to develop a set of user interface requirements. A standard set of user interface requirements will simplify the design and development processes, and ensure that all systems have a similar look and feel to the users. When other constraints (such as a required interface with another application) do not permit the use of existing user interface standards, an attempt should be made to keep the user interface requirements as close as possible to the existing standard.

Define the user interface requirements by identifying and understanding what is most important to the user, not what is most convenient for the project team.

The following are some of the issues that should be considered when trying to identify user interface requirements:

1. The users' requirements for visual and behavior elements, navigation, and help information.
2. The standards issued by the decision system and societal-level organizations that apply to user interfaces.
3. The classification of the users who will access and use the system.
4. The range of functions that the users will be performing with the product.

6.11.2.5 Define system interface requirements

The hardware and software interface requirements must specify hardware and software interfaces required to support the development, operation, and maintenance of the system.

The following information should be considered when defining the hardware and software interface requirements:

1. Users' environment.
2. Existing or planned system that will provide data to or accept data from the new system.
3. Other organizations or users having or needing access to the system.
4. Purpose or mission of interfacing.
5. Common users, data elements, reports, and sources for forms/events/outputs.
6. Timing considerations that will influence sharing of data, direction of data exchange, and security constraints.
7. Development constraints such as the operating system, database system, language, compiler, tools, utilities, and protocol drivers.
8. Standardized system architecture defined by hardware and software configurations for the affected organizations, sites, or operations.

6.11.2.6 Define communications requirements

The communication requirements define connectivity and access requirements within and between user locations and between other groups and applications.

The following factors should be considered when defining communication requirements:

1. Communication needs of the user and InterSystem Team organizations.
2. User organization's existing and planned communications environment (e.g., telecommunications; LANs, WANs, wired, wireless etc.).
3. Projected changes to the current communication architecture, such as the connection of additional local and remote sites.
4. Limitations placed on communications by existing hardware and software including:
   A. User systems.
   B. Applications that will interface with the product.
   C. Organizations that will interface with the product.
5. Standards that define communication requirements and limitations.
6. Future changes that may occur during the project.

6.11.2.7 Define access requirements

A.k.a., Define access control requirements, define security control requirements.

Develop the data and system access requirements in conjunction with the system users. This involvement affords early determination of access, and levels of access protection required for the system.

Use the following procedure to determine access requirements:

1. Identify the types of data that will be processed by the system.
2. Determine preliminary data integrity (protection, security) requirements.
3. Coordinate with the users and InterSystem Team operators of the platform to identify existing supporting computer access (security) controls, if applicable.
4. Incorporate access requirements into the 'requirements specification'.

The following list provides sample questions that can be used to help define the access controls for the system:

1. What access controls (access restrictions) are placed on the users by the societal organization?
2. What are the audit and other checking needs for the system?
3. What separation of accountabilities, control related functions, operating environment requirements, or other functions will impact the system?
4. What measures will be used to monitor and maintain the integrity of the system and the data from the user's viewpoint?

6.11.2.8 Define preliminary implementation requirements

Describe the requirements anticipated for implementing the system (e.g., user production cycle). The high-level implementation requirements are identified early in the life-cycle to support decisions that need to be made for the information systems engineering approach. The implementation requirements are expanded into a full implementation approach during the design stages.

The following factors should be considered when defining preliminary implementation requirements:

1. **Operating environment** - identify any capacity restrictions given by the environment, existing hardware and software that need to be identified and addressed.
2. **Acquisition** - If hardware or software must be acquired, identify the necessary acquisition activities. These activities include preparing specifications, estimating costs, scheduling procurement activities, selection, installation, and testing.
3. **Conversion** - Identify requirements for converting data (or systems) from an existing or external application to the new product.
4. **Installation** - Identify the installation requirements.
5. **Training** - Identify the specific training needs for various categories of users and InterSystem teams.
6. **Documentation** - Identify requirements for the development and distribution of operational documentation for support personnel and user documentation. Operational documentation may include task control procedures and listings, operational instructions, system administration responsibilities, archiving procedures, and error recovery. User documentation includes the user manual, step-by-step instructions, online documentation, and online help facilities.

6.11.1 Functional requirements

A.k.a., Functional user requirements, functional system requirements.

Functional requirements describe functionality or system services, or are descriptions of how some computations must be carried out. Functional requirements are statements of what the system should do in detail. Functional requirements define what the system must do to support the system users functions and objectives. A functional requirements specification represents a model of the desired behavior of the system.

Functional requirements are statements of:

1. Services (functions) the system should provide.
2. How the system should react to particular inputs.
3. How the system should behave in particular situations.
4. May state what the system should not do.

The functional requirements should answer the following questions:

1. How are inputs transformed into outputs?
2. Who initiates and receives specific information?
3. What information must be available for each function to be performed?

Examples of functional requirements include:

1. A user shall be able to search the unified information system for all resources.
2. The system shall generate each day, for each city, a list of actively used services.
3. Each user using the system shall be uniquely identified by a # of digits user/community number.

6.11.2 Non-functional requirements

A.k.a., Non-functional user requirements, non-functional system requirements, constraints, quality requirements.

Non-functional requirements are constraints on the services or functions the system provides and the development process being used. Common non-functional requirements include timing constraints, development process constraints, operating constraints, standards, etc. Non-functional requirements may also define and constrain system properties, such as reliability, response time, storage requirements, etc. Non-functional requirements may be more critical than functional requirements, for if these are not met, the system may be useless.

Non-functional requirements may affect the overall structure of a system, rather than the individual components. For example, to ensure that performance requirements are met, a developer may have to organize the system to minimize power flow between components.

Additionally, a single non-functional requirement, such as a reliability requirement, may generate a number of related functional requirements that define system services (functions) that are requirement. It may also generate requirements that restrict existing requirements.

Often, though not always, non-functional requirements apply to the system as a whole, rather than individual features or services.

Non-functional classification types include, but are not limited to:

1. **Non-functional system requirements** - requirements that specify that the delivered product must behave in a particular way (e.g., execution speed, reliability, etc.).
   A. For example: The system shall be available to all users during the hours of (Mon-Fri, 08:30-19:00).
2. **Organizational system requirements** - requirements that are a consequence of organizational value standards/conditions and procedures (e.g., process standards, implementation requirements, value conditions, etc.).
   A. For example: Users of the system shall authenticate themselves using their biometric identity.

6.11.2.1 Performance requirements

Performance requirements define how the product must function (e.g., hours of operation, response times, and throughput under various load conditions). The information gathered in defining the project objectives can translate into very specific performance requirements; (e.g., if work performed for an organization is critical to the society, the hours of operation and throughput will be critical to meeting the mission). Also, standards can dictate specific availability and response times.

6.12 Requirements analysis

A.k.a., Requirements analysis technique.

A requirements analysis [technique] is the set of data collection and analysis techniques combined with the life-cycle requirements standards (e.g., tracing the requirements through all life-cycle activities) that are used to identify the project requirements and to define exactly what the system (product) must do to meet the system users' needs and expectations. When appropriate, the technique must include methods for collecting data about users at more than one geographic location and with different levels and types of needs.

The requirements analysis technique should be in harmony with the type, size, and scope of the project; the number, location, and technical expertise of the users; and the anticipated level of involvement of the users in the data collection and analysis processes. The technique should ensure that the functionality, performance expectations, and constraints of the project are accurately identified from the system users' perspective. The technique should facilitate the analysis of requirements for their potential impact on existing operations and business practices, future maintenance activities, and the ability to support the system user's long-range information resource coordination plans. It is advantageous to select a technique that can be repeated for similar projects. This allows the project team and the system users to become familiar and comfortable with the technique.

7 [Engineering] Requirements for habitability

A.k.a., Habitat supportability requirements.

Humanity is a global species, and so, it must necessarily recognize, and maintain, a regenerative global habitat. There are multiple layers of accountable requirements in the development of a habitat, and they include, but are not limited to:

1. Social requirements (togetherness).
2. Individual requirements (human).
3. Service requirements (access).
4. Project requirements (organized doing).
5. Technical requirements (what).
6. Team requirements (who).
7. Role requirements (execute).

Habitats are sustained by accountable humans and machines fulfilling roles as part a coordinated network of teams that complete technical requirements on the part of projects that exist to service individual and social human needs. That habitat is designed to optimally support the humans given what the humans know and the available environment.

7.1 Support (habitat supportability)

Habitat supportability is the ability of the habitat to optimally meet the [human] requirements of the inhabitants. The cornerstone of the habitat supportability concept is that each habitat service system (city) functions significantly (though not fully) independent of physical resource support from the other habitats. The crews of these missions must have all of the resources and capabilities that will be necessary to enable them to succeed fully and complete the mission without direct intervention from Earth-based supporting personnel.

This self-reliance will be achieved, in part, by increased emphasis on maintenance by repair rather than replacement. A repair-centered maintenance approach would only be effective, however, when it is strategically coupled with a hardware design that is specifically structured as part of the supportability concept.

The habitat service system is sustained by people working together with their environment. The habitat service system represents a relationship between people, machines (soft and hard), and a living ecosystem.

7.2 Maintenance (habitat maintainability)

Robust, autonomous maintenance capabilities are likely to be enabled by implementation of the following concepts and capabilities:

1. Repair rather than replace. It is preferred to repair failed hardware items rather than simply remove

and replace them. This concept is particularly important for LRUs, ORUs and shop replaceable units (SRUs) that have high failure rates and large masses or volumes. This avoids the use of large quantities of relatively bulky and massive spares.
2. Replace at the lowest practical hardware level. The objective is to minimize the mass of spares consumed. An example would be to remove and replace an integrated circuit that has a mass of grams rather than a complete avionics LRU that has a mass of several kilograms.
3. Comprehensive on-board failure diagnosis. Failure diagnosis should identify the cause of the failure to the level of maintenance. To the extent possible, these capabilities should be built into the systems. When this is not practical, standalone diagnostic equipment can be used.
4. Fabricate structural and mechanical replacement parts rather than manifesting unique spares. Processes are being developed that permit the fabrication of parts from feedstock material that would be carried from Earth or, eventually, produced from in-situ resources. This allows manifesting of an appropriate mass of feedstock material rather than a large selection of unique, prefabricated parts. Manifesting prefabricated parts incurs the risk of carrying excess mass in the form of parts that are never needed and of carrying an insufficient number of parts when there is an unanticipated high-demand rate.
5. Implement a comprehensive preventive maintenance approach. An effective preventive maintenance program can help to avoid the occurrence of system failures and loss of availability. In addition, preventive maintenance can delay wear out, thus reducing the need to stock replacement parts. Extensive pre-mission study is required to define realistic schedules for preventive maintenance that allows for in-flight adaptability based on real-time experience.
6. Enable utilization of common LRUs, SRUs, piece parts, and components across an entire vehicle set. This will allow spare parts that would be carried on one vehicle to be used on another vehicle, or for system elements from one vehicle to be scavenged for use on another vehicle in critical situations. Interchangeability yields flexibility.
7. Use reconfigurable hardware. Using hardware that can be reconfigured to perform different functions as a mission progresses reduces the overall mass of hardware that is carried and minimizes the number of unique spares that are required. Optimally, a single generic part, such as a circuit card, can be easily reconfigured to perform in multiple locations.

7.2.1 Team support functions

Examples of ways in which the team time that would be required for overhead tasks and the mass for team support can be reduced include the following:

1. Launder clothes. Mass reductions can be realized if clothes are laundered and used multiple times.
2. Make inventory management transparent to the crew. Comprehensive and current inventory information is important to crew efficiency. The current manual barcode scanning method that is employed on ISS is cumbersome. A better approach might be the use of radio frequency identification (RFID) tags – active or passive – or use of machine-readable markings. Effective implementation of this technology may require some accommodation in vehicle hardware and system design to allow effective placement of sensors and transmission of RF signals, and to ensure non-interference with other systems.
3. Recycle waste products. Mass efficiency will be enhanced if waste products such as packaging and failed hardware can be recycled and reused.

7.2.2 Maintainability design requirements

Emphasis is on ease of maintenance, standardization and commonality of hardware, and cognizance of issues that would be specific to operations during space flight. The design themes that have emerged to enable the maintenance concept that was described above include:

1. Design for maintainability, graceful degradation, upgrades, and adaptation. For a spacecraft that must be maintained entirely by its crew, design for ease of maintenance is crucial. Systems should also be designed in such a way that they can continue to provide reasonable levels of functionality even after some failures have occurred. Systems should be able to accommodate upgrades – either hardware or software – without requiring total redesign. Finally, designers should seek opportunities to design hardware in such a way that it can perform a variety of functions in different mission phases. This can reduce the total amount of hardware that would be required and simplify its support.
2. Design and build for maintainability in the operational environment. The spacecraft structure will be subject to pressure and thermal differentials that can cause dimensional changes. These dimensional changes can affect clearances between parts, thereby making removal and replacement difficult or impossible. The design must consider

these changes and how they will affect the ability of a crew to perform maintenance tasks. Working in a weightless environment provides some advantages, but it also requires consideration of how a crew member will maintain stability and be able to apply loads required for tasks. For example, although a specific number of closeout fasteners may be necessary to secure hardware for dynamic phases of flight, far fewer fasteners may be necessary during the much longer, quiescent periods. The number of required fasteners should be minimized whenever possible.

3. Require commonality and standardization at hardware levels among major architecture elements. Mission architectures may require multiple elements such as crew transport vehicles, landers, SHABs, and surface vehicles. Every effort should be made to standardize hardware at all levels (LRU, SRU, component) among all architecture elements. This will simplify provisioning of spares, reduce the number of unique tools, and enable substitution between elements. As noted, this applies to hardware at all levels, including avionics circuit card assemblies; electronic components; other assemblies such as pumps, power supplies, and fans; fasteners; connectors; and other piece parts.
4. Require all hardware to be maintained should be internal – minimize extravehicular activity. EVA increases crew risk, is time-consuming, and imposes additional hardware design requirements. To the maximum extent possible, all hardware that may require maintenance should be located inside the vehicle in a pressurized environment to avoid the necessity of performing EVA maintenance operations.
5. Eliminate avionics line replaceable unit boxes – implement rack-mounted boards. Eliminating the boxes that are typically associated with avionics LRUs offers potential mass savings and facilitates access to the individual circuit cards for maintenance. Adoption of this approach, however, also necessitates consideration of cooling efficiency, physical isolation of redundant system elements, and the mass of cabling that would be required if avionics are centralized.
6. Do not combine Imperial and SI [System International] hardware. All hardware should be designed using a single system of units of measure (SI preferred) to avoid the need for multiple tool sets.
7. Provide robust diagnostics and post-repair verification. Efficient maintenance operations require quick, unambiguous fault isolation to the designated repair level. This can be accomplished with built-in-test (BIT) capabilities or with standalone test equipment. Whether via BIT or standalone test equipment, the hardware must be designed to be "testable."
8. Design systems to operate in a "keep-alive" mode with minimal power. In situations when power availability has been degraded or when power must be conserved, it is important that other spacecraft systems can remain functional with a minimal power demand. In this condition, the system may not perform its function but retain the capability to do so when additional power is provided. This is similar to interplanetary probes that revert to a "survival mode" during severe radiation events to protect (by power off) vulnerable hardware.
9. Design systems to enable isolation of faulty components to preclude loss of entire system. Systems should be designed so that single failures do not cause total loss of function.
10. Design systems so that pre-maintenance hazard isolation is restricted to the item that is being maintained. When power, pressurized gas, coolant, or other potentially hazardous resources are isolated from system hardware elements to make them safe for maintenance, isolation should be limited to the smallest possible set of hardware to minimize impacts to overall system availability.

7.3 The habitat service systems views

The controlled habitat service system includes life support systems that humankind builds on top of Earth's life support system (the larger habitat):

1. **Natural ecological [habitat] systems** - The planet Earth's life support ecological-service systems.
   A. **Human controlled [habitat] systems** - The human life (and other) support-service systems. In other words, the societal access-fulfillment system (i.e., community's habitat service system platform).

7.4 Indicators of a co-habitable service system

Effectively constructed habitat service systems are characterized by the following principles (following these principles enable more life range choices):

1. **The life-coherence principle:** The ultimate organising principle of any life-coherent society (or economy) through generational time is maintained (or secure) access to life [fulfilling] services. Any social (or economic) system aligns or does not to

the extent that it maintains the production and distribution of life services.

2. **The service-system principle:** A service system is a service system, if and only if, it enables life capacities/abilities not possible without it (e.g., food, water, shelter, computation, etc.). Claimed services that disable (or do not enable) life capacities and abilities are not means of life (e.g., commodities). Any service that does not directly or indirectly provide a life service is uneconomic (or, anti-economic to the extent of life resources wasted on the commodity's production and consumption).
3. **The provision principle:** The provision, or the deprivation, of each and all of these life services is measurable by greater/lesser sufficiency (e.g., of clean water, life space a, meaningful work, hours of work, etc.).
4. **The performance principle:** The measure of the overall performance of any society (or economy) is its global access commons developmentally expressed as access to life services (including the work share required to provide them). Given what is available and what is known, what is possible? And, how does that compare to all previous states of the society (or economy, or even, another socioeconomic structure)?
5. **The memory principle:** The primary base ("capital") of any society (or economy) is information about creating and maintaining life services without loss in cumulative capacity through time. The societal system specification is the integration of a society's understandings and decisions.
6. **The efficiency principle:** The efficiency of any product, tool or process increases, and only increases, to the extent that:
    A. **Ecological efficiency** - Inputs and throughputs function to enable the provision of life services with diminishing waste and externalities (e.g., organic farming methods, industries directed towards 100% recycling).
    B. **Physical input-output efficiency** - Reduced inputs of materials/energy/space/mandatory work time produce same or greater means of life outputs (e.g., wheel and pulley structures, cooperative organisation of work/leisure requirements, lower labour/fuel-per-unit machines).
    C. **Human development efficiency** - Capability development of productive agents enables more life goods, life-time, and/or life-range choices than before (e.g., by habitat service sector, such as education, healthcare, and intersystem work). More free time is more life range choice.

8 [Engineering] Construction

Construction is a process of work by creating building or infrastructure to support the requirement of society. This process starts from the planning, design, financing and continues until the project is ready for use include problem recognition to the implementation of fully operational solution. Construction can be referring to the several sectors such as building (residential and non-residential), infrastructure (roads, bridges, public utilities, and dams) and industrial (process chemical, power generation, mills and manufacturing plants).

Building construction is a process of adding a new structure to real property whether for existing or new building. This process was involved with complex documentation that call as construction documentations (CDs) that can be divided into several components such as:

1. A graphical representation of the building (which includes 2D floor-plans, elevations and cross-sections, and possibly 3D CAD models)
2. A set of specifications that dictate the quality of the components and finishes of the building
3. A legal document that highlights the project expectations.

How to reduce uncertainty in a project?

1. Coordination (cooperating).
2. Visualization (modeling).
3. Documentation (explaining).
4. Planning (scheduling).

Every construct[-able] societal information system has:

1. The data of an underlying methodology for the societal systems construction [of a habitat system].
2. The perceived problem situation for re-construction [of a habitat system].
3. The individual consciousness involved and affected by the use of the approach to construct [a habitat system].

9 [Engineering] Societal information

To engineering, society is, in part, an information system capable of representing the real world as visual information. A societal engineering system must be a combination of:

1. Conceptual information.
2. Spatial information.
3. Control procedures for associated information into a visualization in time.

9.1 Societal information system construction

Information system integration involves the following layering:

1. In an information system:
    A. Objects "store" data.
    B. Data "stores" meaning.
    C. Meaning "stores" utility.
    D. Utility "stores" memory.
    E. Memory "stores" remembrance.
2. In a conceptual information system, concepts (or concept-objects) store data about meaning (or, perceivable as meaning to consciousness). Therein, concepts can store data about families/patterns of meaning (i.e., their properties). Concepts form patterns of meaning in the awareness of consciousness.
3. In a spatial-information system, objects store data about shape; objects can store data about families of shapes (i.e., their attributes).
4. In a physical environment, objects (Read: a shape) can potentially be put within other objects (e.g., putting 'water' in a 'glass', or one Matryoshka doll inside another). In a conceptual environment, concepts (Read: with meaning) can potentially be put (embedded) within other concepts (e.g., putting the meaning of a 'bed', or "place to sleep", within the meaning of a 'home' or 'dwelling').

Information systems are operationalized through the functional integration of concepts and objects, which become technology for a social population:

1. The subject of a sentence/argument is a concept, a meaning. The rules by which the sentence is constructed is its syntax (ordering logic).
2. The object is something to point to (to point out to someone else, to visualize for oneself or another). The rules by which the vision is constructed is its intelligence (visual logic).

3. Technology is the reproduction of an objective in object form, from a function involving concept and object integration.

9.2 Synthetic environment data representation and interchange specification (SEDRIS)

SEDRIS (Synthetic Environment Data Representation and Interchange Specification), SEDRIS Spatial Reference model (SRM)

In order to support the unambiguous description of environmental data (of a conceptual and spatial form), the SEDRIS SRM (spatial reference model) specifies both a Data Representation Model (DRM) and an Environmental Data Coding Specification (EDCS). These address how to describe "environmental things", but explicitly avoid defining how "environmental things" are located with respect to one another and with respect to non-environmental "things". The SEDRIS SRM (spatial reference model) addresses this need and provides an integrated framework and precise terminology for describing spatial concepts and concepts of operations on spatial information.

The SEDRIS RM (reference model) is comprised of a set of Reference Frames (RF), their inter-relationships, and unambiguous definitions of methods for specifying and inter-converting location (including directional and orientation) information among SRFs. Additionally, those methods are documented in terms of detailed algorithms and subsequently reduced to efficient, accurate, and portable implementations.
Algorithms form a coordinating framework for simulation of a physical information space.

1. Algorithms exist for spatial operations.
   A. Space is shaped with physical objects.
2. Algorithms exist for informational operations.
   A. Space is shaped with informational objects.

9.2.1 A multi-scale integrated model of ecosystem services (MIMES) and human coupling

SEDRIS is a modeling tool that can incorporate user input and biophysical data sets for evaluation of ecosystem services and decisioning by producing an integrated multi-scale model of ecosystem services MIMES and human coupling. In order to accomplish this form of environmental data model, the information system must:

1. Simulate ecosystems and socio-economic systems in space.
2. Simulate these systems over time.
3. Simulate the interactions between these systems and human service systems through coupling.
4. Simulate the coupling over time.

In order for the model to function, ecosystem services are (Read: assumptions):

1. Ecosystems are the structures and processes that generate functions.
2. Ecosystem functions of value to humans are ecosystem services.

Characteristics of ecosystem services include, but are not limited to:

1. Structures are not transformed into the produced services (e.g. lumber is not the service, the production of trees is).
2. The source of most energy for services is solar energy.
3. Availability depends on ecosystem functioning and typically is not controlled by humans.
4. Ecosystems supply ecosystem services, which humans can harness (use).

The elements of an information-spatial model include, but are not limited to:

1. Requirements (information and spatial).
2. Acquisition (information and materials).
3. Processing (information and materials).
4. Usage (service-information and service-objects).

9.3 Spatial and conceptual information

Materiality is shape (Read: literal, physical), which is representable as information. Conceptuality is information, which is representable as materiality (e.g., a house-building or "money"). Spatial and conceptual information come together in the form of an information system for society, with a pure information set, and a material information set (that represents either the current real-world, the past real-world, or potential possible real-worlds).

The two dimensions could be otherwise called:

1. Hard[ware] - material system (spatial, physical).
2. Soft[ware] - informational system (conceptual, mental).

For movement in a material system there must be a physical mechanism (material process) to have a complete explanation. Movement can be described, and movement can be explained. For change in a conceptual system there must be an information mechanism (information operation). Change can be described, and change can be explained.

The societal system sub-component naming involves:

1. The components of a material system are often referred to as architecture (infrastructure).
2. The components of an information system are often referred to as data (computation).
3. The components of a meaningful or relational system are often referred to as concepts.
4. The components of a material system are often referred to as objects.

9.3.1 Spatial information

Spatial information processing requires a coherent capability to describe the geometric (spatial logic) properties of:

1. **Position** (location of).
2. **Direction** (motion toward).
3. **Distance** (space between).

It is from these spatial properties that spatial alignment and navigation are calculable, and from which a material service may safely exist.

Spatial information may be spatially referenced to:

1. Local structures and regions.
2. The Earth as a whole.
3. Other celestial bodies. or
4. Objects defined within synthetic visual contexts (e.g., virtual realities).

In each of these cases, a spatial reference frame is defined in relation to logic properties (e.g., spatial logic, conceptual logic, etc.).

9.3.1.1 Spatial data

A.k.a., Geo-referenced data, geodetic data, geodetic datum, spatial environmental data, conceptual (social) environmental data,

Spatial data describes the absolute and relative location of geographic (earth or spatial) features. Spatial data describes:

1. The characteristics of spatial features.
2. Quantitative and/or qualitative data.
3. Attribute data is often referred to as tabular data.

Geo-referenced data include, but are not limited to astronomical, orbital, geomagnetic, and local observations whose reference frame may be fixed with respect to observer, solar, celestial, or other positional standards rather than, for example, the equator plus a prime meridian on an Earth Reference Model (ERM) surface. Other Object Reference Models (ORM; e.g., the moon or the NASA Space Shuttle) may also be used.

Common geographic, geospatial (positional) representational data types include but are not limited to:

1. Points *(primary class)*:
    - id,
    - x, y,
    - $m_1..m_n$
2. Lines *(primary class)*:
    - id,
    - $x_1,y_1..x_n,y_n$
    - $m_1..m_n$
3. Areas *(primary class)*:
    - id,
    - $x_1,y_1..x_n,y_n$..x1,y1,
    - $m_1..m_n$
4. Rasters *(primary class)*:
    - $x_1,y_1z_1..x_n,y_n,z_n$
    - $m_1..m_n$
5. Routes *(extended class)*
    - id
    - $x_1,y_1..x_n,y_n$
    - $m_{1..}m_n$
1. Regions *(extended class)*
    - Poly list
    - id
    - p_1..id,p_n
2. Instantaneous points
    - id, x,y,z,t,m
3. Time duration points
    - id,x,y,z
    - $t_s..t_e$
    - $m_1..m_n$
4. Time series points
    - id,x,y,z
    - s
    - $t_1..t_2$
    - $m_1..m_n$
5. Time duration vectors
    - id
    - $x_1,y_1,z_1..x_n,y_n,z_n$
    - m1..m2
6. Time duration areas
    - id
    - $x_{1,},y_{1,},z_1..x_n,y_n,z_n$
    - x_1,y_1,z_1
    - $t_1..t_2$
    - $m_1..m_n$

9.3.1.2 The material data set

Material object observations generally have four operational dimensions:

1. The first two are the x and y horizontal spatial coordinates, referring to some predetermined

standard coordinate reference system (CRS).
2. The third is the temporal coordinate, t, the moment – according to some predetermined standard calendar and time zoning system – when the soil was observed.
3. The fourth operational dimension is the material observation elevation, z, as measured using some predetermined standard scale, e.g. metres.

At a point in (geographic) space and time, [x, y, t], or in space, time and depth, [x, y, t, z], a material observation is accompanied by an attribute space. The latter is a multi-dimensional space defined by a set of attributes of the environment (e.g., for a soil observation it may be land use, slope, parent material, or soil layer pH, cec, carbon content).

9.3.1.3 Spatial data collection

How are current data collected on current real world objects? The 3D geometry of real world objects are observed and collected through surveying technology, such as total stations, terrestrial/airborne laser scanners, or techniques from photogrammetry. The common name for this collection of techniques is known as, 'remote sensing'. Remote sensing processes record, measure, and interpret imagery and digital representations of [energy] patterns derived from non-contact sensor systems.

9.3.1.4 Spatial data interoperability

Interoperability of spatial data is facilitated through the adoption of a common and widely-known Spatial Reference Model (SRM) that allows the context in which coordinates, directions, and distances are defined to be known exactly, and converted accurately into multiple definitions and representations of geo- and non-georeferenced space.

9.3.2 Conceptual information

Conceptual information processing requires a coherent capability to describe the conceptual (mechanical-state logic) properties of:

1. **Condition** (quality of).
2. **Function** (behavior of).
3. **Intention**[al progress] (expectation of).

It is from these mechanistic properties that state change and intentional alignment are calculable, and from which an effective service may safely exist.

Conceptual information may be conceptually referenced to:

1. Local behaviors.
2. Regional behaviors.
3. The population as a whole.
4. Digital (software) systems.
5. Mechanical (hardware) systems.

In each of these cases, a conceptual reference frame is defined in relation geometric properties.

9.3.3 Temporal information

For any object or system to be realized in the real world, it is necessary to specify the time and temporal reference frame to which the spatial position and/or conceptual condition refers, and the time for which the spatial or conceptual reference frame is defined.

9.3.4 Spatial and conceptual interoperability

Interoperability of conceptual data is facilitated through the adoption of a common and widely-known Conceptual Reference Model (CRM) that allows the context in which conditions, functions, and intentions are defined to be known exactly, and converted accurately into multiple definitions and representations of real- and non-real-referenced space.

Interoperability of spatial information requires that:

1. Spatial reference frames and ORM/ERMs be defined such that coordinates and angular measures describe position and orientation data uniquely, and
2. Mechanisms exist for such data to be converted/transformed between alternative spatial reference frames, should this be required.

Interoperability of conceptual information requires that:

1. Conceptual reference frames and ORM/ERMs be defined such that conditional and functional measures describe intention and orientation data uniquely, and
2. Mechanisms exist for such data to be converted/interpolated between alternative conceptual reference frames, should this be required.

Existence has two delimitations to embodied consciousness:

1. The basis of physicality is a limited physical boundary.
2. The basis of information is data (i.e., a delimited meaning boundary).

9.3.5 Spatial and conceptual reference frames

A.k.a., Where is the point?

Accurately locating objects is key to the operation of any [service] system which contains information about real-world entities. Consistency (standard) in description, nomenclature, and the treatment (application) of models of the earth and related spatial and conceptual reference frames and coordinate systems is critical to achieving effective data interchange and system interoperability, which are required for optimization of global human [service] fulfillment. The S-/C-RM provides the means to define a unified approach to representing real world conception location information, and precisely relating different descriptions of such location. All information here can be represented in databases, and potentially, simulated.

Any ability to control physical or conceptual motion comes from having a set of coordinates within a coordinate system. A coordinate system is a collection of rules by which values may be used to relate (process) an object to a unique (coordinate system) origin location. Coordinate Systems are a collection of rules by which values may be used to spatially or conceptually relate a location to a unique (coordinate system) origin location. A coordinate system specifies a mechanism for locating points within a reference [coordinate] frame.

> **CLARIFICATION:** *A coordinate frame (or simple, reference frame or frame) is specified by an ordered set of mutually dependent direction vectors; thus, a reference frame has an identifiable center. When producing or using positional or conditional data, one [controller] needs to understand both the reference frame and the coordinate system being used.*

In physics, a frame of reference (or reference frame) consists of an abstract coordinate system and the set of physical reference points that uniquely fix (locate and orient) the coordinate system and standardize measurements (for some form of intentional control by consciousness). Coordination system formalization [as mathematics] is relatively simple. A coordinate system in mathematics is a sub-conception of geometry (applied algebra), a property of manifolds (in physics even, these are appropriately called configuration spaces or phase spaces; appropriately because coordinate systems allow for controllability, and thus, the benefit of re-configuration). Mathematically, the coordinates of a point r in an n-dimensional space are simply an ordered set of n numbers: r = {x1, x2, ..., xn].

> **INSIGHT:** *Through coordination there may exist greater development of individual function.*

9.3.5.1 Coordinates

Coordinates are:

1. Linear or angular quantities that designate the position of a point in a [coordinate system] reference frame. By extension, they also designate the position of a point within a spatial reference frame.
2. Conditional or intentional qualities that designate the state of a point in a [coordinate system] reference frame. By extension, they also designate the condition of a point within an intentional reference frame.
3. Data representationally modeled (DRM) locations of spatial position and/or intentional condition.

Reference modeling is the production of:

1. A reference model (spatial-, conceptual-type) is a well-defined set of
   A. reference frames (spatial, conceptual),
   B. object reference models, and
   C. coordinate systems,
   D. that allows coordinates to be specified
   E. succinctly, and
   F. converted accurately between different [spatial and/or conceptual] reference frames.

Spatial reference frames (SRFs) are:

1. Reference Frames serve to locate coordinates in a multi-dimensional space (generally either two- or three-dimensional). They are specified in two parts to any SRF.
   A. Object reference model - A geometric description (model) of a reference object embedded in (and serving to orient) that frame referred to as an Object Reference Model (ORM)
      1. An Earth Reference Model (ERM) is a special case of an ORM
2. Coordinate system computation - A Coordinate System specifying how a tuple of values uniquely determine a location with respect to the origin of that frame. By extension, that tuple also specifies a location with respect to the reference object.

9.3.5.2 The Cartesian coordinate system

> *A.k.a., The geometric-planar coordinate system, a spatially alignable coordinate system.*

The Cartesian coordinate system (the planar coordinate system) is based on an ordered set of mutually perpendicular axes formed by straight lines. The point of intersection of the axes is termed the origin. Alignment - deviation from origin - can be determined to some degree of certainty. The directions of successive axes are normally related to each other by a right hand rule.

1. Cartesian Coordinates (two-dimensional) uniquely locate points on a plane using a doublet of values, e.g., (x, y).
2. Cartesian Coordinates (three-dimensional) uniquely locate points within a volume using a triplet of values, e.g., (x, y, z).

Since the Earth is an important reference object in our spatial environment, many Spatial Reference Frames will consist of an Earth Reference Model plus a Coordinate System.

Earth resource frame includes:

1. Topographic Surface is the interface between the solid and liquid/gas portions of the Earth. A topographic surface corresponds to the surface of the land and the floor of the ocean.
2. Earth Reference Model (ERM) is a specification of the mathematical shape of the Earth, usually in terms of a combination of ellipsoidal and equipotential (geoidal) surfaces. It excludes the topographic surface, and therefore generally corresponds with mean sea level.

Since human needs are an important reference object in our conceptual environment, many Conceptual Reference Frames will consist of a Human Reference Model plus a Coordinate System.

Human needs frame include:

1. Habitologic Surface is the interface between the human and ecological materials of the Earth. A habitologic surface corresponds to the surface of the human body and of the local environment.
2. Sociologic surface is the interface between the individual and social access to habitat surfaces.
3. Economic surface is the interface between the human and habitat re-configuration of surfaces to more greatly meet human habitat needs.
4. Real World Reference Model (Real-World Information System) is a specification of the conceptual shape of the human need frame as a real world information model, usually in terms of a combination of conceptual and material surfaces. It excludes the topographic surface, and therefore generally corresponds with mean functional level.

Other reference frame models common to society are:

1. Object reference model (ORM).
2. Service reference model (SRM).
3. Habitat reference model (HRM).

A coordinate system exists to perform useful operations on coordinates, including but not limited to:

1. Direction determination (azimuth and elevation angle calculation
2. Coordinate Conversion is the process of determining the equivalent location/condition of a point in a reference frame (spatial/conceptual), which is based on the same object reference model (e.g., ERM), but a different coordinate system.
3. Coordinate Transformation is the process of determining the equivalent location/condition of a point in a reference frame, which is based on the same coordinate system, but a different object reference model (e.g., ERM).
4. Converting coordinates between two arbitrary Reference Frames may require both Coordinate Conversion and Coordinate Transformation.

There are some common types of operations errors in coordinate systems:

1. Formulation (algorithmic) errors in algorithms for coordinate operations.
2. Implementation errors includes errors due to sequencing (mathematics) and software implementation.
3. Usage errors includes errors due to extension of projection-based reference frame beyond reasonable limits.

9.4 *Integrated informational material modeling*

In the technical world, one of the most well-known concept models is TCP/IP, the foundation of the global Internet communications system. The generally named Internet protocol suite is, the conceptual model and set of communications protocols used in the Internet and similar computer networks. The information set is commonly known as TCP/IP because the "suite" is primarily composed of the Transmission Control Protocol (TCP) and the Internet Protocol (IP).

The following is a stacked model of informational and material flows in society. [Flow] of information and materiality transmission control protocol ([F]TCP) stack/ model (adapted from Internet TCP/IP model):

1. Physical (material surfaces standards; materialization protocol).
2. Data Link (information standards; information protocols).
3. Network (global habitat decision standard; decision protocol).
4. Transport (land and motion systems standards; transportation and communication protocols).
5. Session (local habitat service operation systems standard; habitat operational process protocols).
6. Presentation (useful objects).
7. Application (useful services).
    A. Application (Habitat contribution service platform; messaging, collaborating, addressing, deciding, tasking; monitoring as support-view; and coordinating as over-view).
    B. Transport (Resource and human transportation, fabrication, and cycling system; the material

surface system re-composed for differing human service purposes).

C. Network (intra-habitat integrated-infrastructural network protocols; inter-habitat integrated-infrastructural network protocols).

D. Link (physical habitat service operational surfaces and information flows).

E. Physical (machines and humans; drawings and documentation).

9.5 *Spatial reference model (SRM) standards*

A.k.a., Spatial coordinate system.

A spatial coordinate system is a means of associating a unique coordinate with a point in object-space. It is defined by binding an abstract coordinate system to a normal embedding. A spatial reference frame is a specification of a spatial coordinate system for a region of object-space. The spatial embedding of a real-world surface coordinate system is:

- The ISO/IEC 18026:2006 Spatial Reference Model (SRM) International Standard.

ISO/IEC 18026:2006(E) allows new concepts to be specified by the registration of new entries to the standard. New entries to the standard are registered using the established procedures of the International Register of Items. The registry is also a valuable resource for searching and finding specific spatial reference frameworks (SRFs), object reference models (ORMs), coordinate systems (CS), and other registrable constructs within the reference mode (RM).

Aspects of ISO/IEC 18026:2009 apply to, but are not limited to:

1. Mapping, charting, geodesy, and imagery.
2. Topography.
3. Location-based services.
4. Oceanography.
5. Meteorology and climatology.
6. Interplanetary and planetary sciences.
7. Embedded systems.
8. Modelling and simulation.

9.5.5.1 Highly-related ISO Standards

These standards further contextualize spatially referential information:

1. ISO/TC 211 - Geographic information/geomatics
2. ISO/TC 184 - Automation systems and integration
3. ISO/TC 184/SC 5 - Interoperability, integration, and architectures for enterprise systems and automation applications

9.6 *What is a spatial coordinate system?*

A.k.a., spatial coordinate reference system.

Spatial coordinate reference systems are designed to enable the position of points to be uniquely described over varying sizes of information or geographic area. A coordinate system that enables every location on and around Earth to be specified by a set of numbers, letters or symbols.

A [spatial] reference system (SRS) or coordinate reference system (CRS) is a coordinate-based local, regional or global system used to locate [geographical] entities.

Example geographic (a.k.a., earth surface positional) reference systems include:

1. A Grid Reference System (grid or projected coordinate reference system), is a means by which to reference locations [on the Earth's surface using a two dimensional Cartesian coordinate system referenced to a map projection. A grid coordinate defining a location consists of and is written as an ordered pair of x and y values expressed in linear units. Most of these grid reference systems use the meter as the unit of measure and define an easting (x) and northing (y) referenced to a series of transverse. A
2. Geographic Reference System (graticule) is a means by which to reference locations on the Earth using a system of angles. A geographic coordinate defining a location is usually expressed in angular units of latitude and longitude. Latitude (φ - phi) is the angle between the equatorial plane and the straight line that passes through the point in question and the center of the reference shape (WGS84 ellipsoid). Longitude (λ - lambda) is the angle east or west of the reference meridian (Greenwich Prime Meridian) to another meridian that passes through the point in question. The most common standard geographic reference system is latitude and longitude expressed in sexagesimal (base 60) numbering system.

A coordination system (coordination management) is a set of programs to perform operations on data, such as store and retrieve data.

9.6.1 Database operations

A database coordination system maintains the collection of all societal data, and a set of programs to access, store, retrieve, and otherwise process societal data. Therein, the decision system is decision support software with some designed interface. The social system specification is a database of societally relevant information, coordinated by software.

9.6.2 Environmental database

An environmental database is an integrated set of data elements, each describing some aspect of the same geographical region. It often includes additional data describing simulation elements and events expected to take place during the interactions in that environment. For example, data representing trees in a forested region may be found in a database; but in addition, the geometry of vehicles that might drive through the trees during a simulation would also be found in an environmental database. The key phrase in the above definition is "integrated set of data." It is the integration, infusion, and tailoring of varied data sources that creates a full database, and sets it apart from databases that only use an existing raw data source as-is.

9.7 Spatial objects

Spatial objects (in the real world or a simulation) have shape as an attribute in their table. They have geographic (or potential geographic) location. They are a point, line, TIN, raster, etc.

9.8 Temporal-spatial coordination

Time and location are often used together by an application to describe when a given condition exists, or when an object was present, at a given location. The real-world has a time parameter; and at a conceptual level, it is composed of dynamic systems, which are systems that factor a time parameter. These systems reduce to the case of a static relationship by fixing a value for the time parameter. Material/physical object reference model bindings (associations) are often based on physical measurements of objects or systems that change with time. Time is also used to identify the decisions for which these measurements are applicable.

A 'temporal' coordinate system (CS) is a 'Euclidean 1D' CS (see Table 5.35, [standards.sedris.org]) that assigns distinct coordinates to distinct times so that larger coordinate values are assigned to later times. In relation to human tasking, this culminates in a universally, globally coordinatable time system; a temporal coordinate system that enables a unique temporal coordinate to be assigned to every recorded or potential event; thus, necessary for global tasking. For example, in the early 21ste century society, Universal Time (UTC, [standards.sedris.org]) (see 6.2.4, [standards.sedris.org]) was an inter-national time standard.

Herein, times and dates refer to UTC unless explicitly indicated otherwise (often for extra-societal coordination).

9.8.1 Temporal spatial standards

The most well-known temporal-spatial standard is SEDRIS:

- ISO/IEC 18026E: The SEDRIS Standard [standards.sedris.org]
  - Information technology spatial reference model [standards.sedris.org]

9.8.2 SEDRIS

SEDRIS (Synthetic Environment Data Representation and Interchange Specification) is an international data coding standard infrastructure technology created to represent environmental data in virtual environments. A SEDRIS system is coordinated through socio-technical projects.

A virtual environment is a synthetically visualization as a representation of existence. Today, virtual environments are sustained through combinations of hard- and soft ware. In any given society, visualization technology may be installed within an organization's existing IT infrastructure and controlled from within the organization itself. From a central interface the technology creates an interactive and immersive experience for teams of users. Visualization technology enables cooperation it be most effective and efficient, because individual understandings can be aligned to a commonly shared vision, and a commonly shared vision can be aligned to individual understanding (through visualization). Visual environments tend to focus on needs of users and issues that are actually relevant and persistent, because that which is the problem and that which is the solution is visually clarified to be so by everyone.

Virtual environments can be persistent and representational of a working, conditional real-world environment.

Environmental data represented by SEDRIS may be concrete (physical, positional and compositional) , such as trees and mountains, or abstract (conceptual, conditional and intentional), such as a technology operations procedure (behavioral intention) or the state of a service system (behavioral condition).

Here, visualization facilitates the accurate and coherent exchange of data for reuse and wider scrutiny. Whereas a simulation is a dynamic visualization, a flow diagram (or concept model) is a static visualization.

SEDRIS tutorial data references include:

1. SEDRIS Technologies tutorials [sedris.org].
2. SEDRIS Data [data.sedris.org].

9.9 What are data models used for?

A data representational model, or simply a data model, is a notation method for describing data. The data model provides a description that enables its users to understand what data is present and how it is organized. A data format specifies the actual bytes used to store data on a storage medium. A specific implementation is defined for how the data objects are to be structured and identified on the medium. There are multiple ways of implementing a data format for a specific data model.

9.9.1 Semantic logic

Semantic logic provides an explicit representation of the conceptual relationships between information objects. Topology provides an explicit representation of the spatial relationships between physical objects (e.g., connectivity and adjacency). Semantic logic (semantic reasoning, rational reasoning) provides the explicit representation of a conceptual relationship, instead of having to derive the conceptual relationship by doing discrete (and axiomatic) logic proofs. Topology provides the explicit representation of a spatial relationship, instead of having to derive the spatial relationship by doing geometric calculations. A topology and semantic logic are ways of pre-computing the answers to spatial and meaningful-informational relationship questions.

The only sensible use of the term “semantics” refers to the meaning of expressions in a language. Such expressions can be single symbols (the “words” of a language) or symbol combinations. As the term implies, they are used to express something, i.e., to communicate meaning. Neither concepts nor entities nor properties nor processes have semantics, but expressions in languages describing them do. In an information system context, many languages need semantics: natural languages, programming languages, schema languages, query languages, interface specification languages, workflow modeling languages, user interface languages, sensor modeling languages, and others. The symbols and expressions of information system languages may be produced and consumed by machines and humans. The languages used in information systems are not natural languages, even if they use natural language terms; instead, they are technical language, which is the results of ‘precision of language’ determinations sufficient to design and operate a working system. Many of these languages allow users to define new symbols (for individuals, types, properties, relationships etc.). Attaching meaning to language expressions is a conceptual phenomenon. Natural language symbols and expressions evoke concepts in human minds. If these concepts are similar to those which the symbols and expressions were meant to express, communication works. Expressions come to represent entities (as well as properties, relationships, and processes) in the world. This fundamental ternary meaning relationship between symbols, concepts, and entities is captured in the so-called semantic (or semiotic) triangle.

9.9.2 Simulation

To simulate is to make up something similar to an original. One primary use for a synthetic environment is the representation of the natural environment at a specific geographical location. Therefore, the synthetic environment includes the terrain, terrain features (both natural and man-made), 3-D models of vehicles, personnel, and certain terrain features, the ocean (both on and below the surface), the ocean bottom including features (both natural and man-made) on the ocean floor, the atmosphere including environmental phenomena, and near space. In addition, the synthetic environment includes the specific information attributes of the environmental data as well as their relationships.

9.9.3 Active data models

Active data models are operations used to coordinate data into/within a structured document with a centralized data repository.

A common data element classification for an environmental database is:

1. Terrain surface - Surface geometry.
2. Terrain features - structures found on and within the terrain, such as vegetation, hydrology (rivers), roads, rockets, terrain artifacts, etc.
3. Buildings - buildings as structures in the area.
4. Objects - structures other than buildings in the area.
5. Textures, images, and colors - surface composition.
6. Environmental models - environmental phenomena as smoke, rain, haze, etc.
7. A database that is designed to support the full spectrum of environmental understanding required to fulfill a wide-ranging human need-service habitat application.

In order to support the unambiguous description of environmental data, an environmental data coding specification:

1. Data representation mode (DRM) - how to describe “environmental things” in terms of data modeling constructs meaningful to simulation developers (e.g., geometry, feature, image, topology, and data table), it explicitly avoids specifying “where” the “environmental things” are, and enumerating all of the “environmental things” that these data modeling constructs could be used to represent.
2. Spatial reference model (SRM) - captures and unifies the spatial models used. These models include inertial, quasi-inertial, geo-based, and non-geo-based (purely arbitrary Cartesian) systems. The SRM provides a unifying mechanism for specification and inclusion of any spatial reference frame and coordinate system. Its algorithms are designed to retain a high degree of accuracy during transformation and conversion operations (1mm accuracy).
3. Environment data coding specification (EDCS) - The EDCS provides a mechanism to specify the environmental “things” that a particular data

model construct is intended to represent. That is, a "tree" could be represented alternatively as a <Point Feature>, an <Aggregate Geometry>, a <Data Table>, a <Model>, or some combination of these and other data modeling constructs. Which of these the data modeler (i.e., the data provider of a SEDRIS transmittal) chooses is orthogonal to the semantic of the "thing" that is represented (and its location). The provision of such a "thing" in a SEDRIS transmittal pre-simulation must result in a shared understanding of "what the thing is and what it potentially means" to all participating applications.

9.10 *Environmental data standards*

Environmental Data Coding Specification (EDCS):

1. ISO/IEC 18025 provides mechanisms to unambiguously specify objects used to model environmental concepts. A functional interface is also specified. [standards.sedris.org]
2. ISO/IEC 18024-4: EDCS language bindings - Part 4: C - Access to the codes defined by ISO/IEC 18025 is through an API.

Environmental data coding standards. All data coding standards focus on meeting the needs of modeling and simulation of the environment for a community of users.

9.10.1 Environmental data-base construction

There are different approaches to database construction depending on the available tools, intended simulation platforms, system requirements, available data sources, design preferences, and application-specific needs. As a result, there is no standard methodology for creating simulation databases. For the most part, however, some general phases are common to all database construction processes. Sometimes these phases overlap or are combined, sometimes one is left out because there is no added benefit, and sometimes their order of execution is changed or done in parallel. With those caveats, we can break the construction process into the following six phases.

1. Requirements definition - As in any design and implementation, this is critical for database construction because of the varied levels of knowledge between designers and end users, vastly different construction techniques and system constraints, and the lack of a standard terminology common to the simulation community. Without the involvement of both users and designers throughout the entire construction process, the acceptance or desirability of the resulting database will be left to chance.
2. Data collection - Collecting source data is continuous throughout. Source material can span the range of paper maps, digital elevation products, images and photographs, feature data, 3D models, verbal reports, tabular data, satellite imagery, attributes, weather data, topological data, existing animation and special effects, and a host of other data sources.
3. Reasoning (explaining model, value-adding) explanation - visual understanding integration into the given most well-defined model. Often the source data needs to be further analyzed or refined before it can be used.
4. "Assembling" the database - Once all data sources have been put in an acceptable form, the various data elements are then integrated and assembled into the database one at a time. This may mean combining a surface with a particular texture, color, or attribute; or conforming the 2D features, such as roads or rivers, to the underlying 3D terrain surface. There are many other similar steps that take place during this phase. Applying real-time performance constraints is one of those. The key, however, is the notion of infusing and integrating inherently varied data sources into a single cohesive database.
5. Compiling and execution (or transmission) of the database.
6. After sufficient simulated iteration of the previous steps, the environmental database is ready for use and testing in the real world. In this step, the database is compiled from its editable data structures into platform-specific data habitat service sub-structure.

It is possible to conduct research into shared ways to represent environmental data was begun in the 1980s in order to permit distributed simulations to work together.

9.11 *Unified coordinate information systems algorithms*

This report contains guidelines for the development of computationally efficient algorithms for computing informational and spatial operations. A spatial operation is a:

1. Coordinate transformation.
2. A coordinate conversion.
3. An azimuth determination.
4. A distance calculation, or other.
5. Computations associated with elliptical trigonometry and map projections.

An informational operation is a coordinate transformation, a coordinate conversion, an value determination, a fulfillment calculation or other computations associated set decisioning.

10 [Engineering] Geoinformatics

A.k.a., Geographic information, geomatics, real-world space computation, an information-based spatial-visual system; geocomputation.

Geoinformatics is a discipline of systems science that uses knowledge and technologies to support the processes of acquiring, analyzing, and visualizing geospatial data (Question: how can the data be added to a simulation of the real-world?). In concern to functioning, the integrated city system ("smart town") is analogous to the human body; it divides functioning into cells and into grouping of cells that share resources and coordinate fulfillment without trade. Activities take place in time and dimensional space; some of the activities are built into the environment as infrastructure. As a type of infrastructure, buildings are an environmentally controlled space where activities take place within. There is a natural organismal ecology. Survey sensors collect environmental and individual issue data. A computational control/conditional system that processes data to give an objective. A transportation network distributes resources, humans, services and products for global accessibility. A circular integrated city grid framework individuates the circle into functional cells. In simulation, the cells overlay a real-world datum as functional habitat material-service boundaries. Additional geoinformatics data layers may overlay the model.

3D geoanalytics and a geo-spatial (visualization, simulation, analytics platform) interface to multiple layers of extant and possibly extant reality. Any information that can be spatially organize, a GIS is the name given (currently) to the system that organizes, visualizations, and databases the system.

1. Surveying:
    A. Geoanalytics for the habitat service system network relate individually coordinated habitat service [city] systems within an open resource-access network, unified by means of a pre-planned procedural generation (i.e., visualization) of resource survey data de-constructed by habitat service input-output sub-service systems (i.e., integrated city system) that share resources among a commonly integrated information system.
2. Engineering:
    A. Geospatial (geo="earth"; spatial="4D.."; material world) engineering.
3. Computation:
    A. Geographic information system (data structuring and visualization of the world)
        1. Geographic information science (discovery and survey of the world, geomatics).
        2. GeoDesign is the concept of designing and planning a geographic (a.k.a., spatial, real-

world, locatable experiential) environment.

10.1 Habitat service planning

A.k.a., Smart city planning.

The purposeful usage of geoinformatics is to understand, to design, and to monitor a population's habitat service system. A controlled 'habitat' is a bounded system that transduces ecological (and socio-environmental services), and can be 'planned'. That which is planned within the habitat is services, primarily for humans and with consideration to the larger ecology that facilitates human flourishing. For habitat service planning to be, and remain, effective for scalable and global human fulfillment, it must approach the design of the habitat axiomatically and structurally "bare" (i.e., without artificially imposed prior social constructions), and at a root-fundamental level:

1. Plan as if "you" are starting from construction scratch on a given terrain (i.e., there are no prior constructions).
2. Separate the shape boundary of the controlled and integrated habitat service system into cells.
3. Add spatial data.
   A. Identify material habitat service operations and prioritize functional placement on a circular layer.
   B. Positional location of resources into assemblages of operational material mechanics (constructions, infrastructure) used as a service by accessing users.
   C. Transportational motion of resources, assemblages of resources, and humans between cells.
4. Add decisional data.
   A. Identify parallel decisional process operations and prioritize decisions on a conceptual (meaning-mental) layer.
   B. Position location of information into assemblages of operational information mechanics (instructions and software) used by a service by accessing users.
   C. Transportational motion of inputs, outputs, and decisions between information service boundaries.
5. Add conceptual (intentional) data.
   A. Survey human requirements.
   B. Survey natural landscape artifacts.
   C. Survey planning of constructable city.
6. Develop a simulated map of the geoinformatic environment.
   A. Develop resource physical simulation model.
   B. Develop user access opportunity distribution model.
   C. Develop world terrain model.
   D. Develop building (infrastructural) models.
   E. Develop network sharing model.
   F. Develop unified information coordination model.

10.2 Material visualization and analysis

In a unified real-world societal system, coordination must exist between the following three [material visualization and analysis] data sets in order for decisioning to be optimal:

1. **Building information modeling (BIM)** - describes information about the design and construction of building sites. digital models of real world assets. BIM tools often use local coordinate systems.
   A. Asset design processes.
   B. Asset documentation processes.
2. **Geographic information system (GIS)** - describes information about the material environment. A system designed to capture, store, manipulate, analyze, coordinate, and present all types of geographically referenced data (a.k.a., earth referenced data, spatially referenced data). GIS merges cartography, statistical analysis, and databased technology. In a GIS framework, both spatial and non-spatial databases are combined into a geodatabase. A GIS essentially creates map layers of specific thematic maps. By layering the information one on top of the other, an information system can show, for example, the relationship and degree of connectivity between various land uses and transportation routes in a region. GIS extends Building Information Modeling (BIM) design data through visualization and analysis of structures in the context of the material (natural and built environment). GIS data usually rely on geographic coordinate systems that precisely locate the data in the real world. Environmental design constraints are often stored in a GIS database (e.g., what is the terrain[ability] and existing building in some geographic [spatial] area.
   - Geographic reference processes
3. **Operational information system** (OIS; a.k.a., operational material system, OMS; a.k.a., facilities management, infrastructural development and operations)
   A. Operational systems processes.
4. **Unified interface** that integrates the prior three into a personal dashboard to see in a common environment. The ability to pull together huge amounts of information and visualize it for users and the support of their decisions.

BIM and GIS overlap when their data concerns information about infrastructure, building sites, floors, and rooms. Because of the overlap, the integration of data from both domains (often considered separate disciplines in the professional market) is required for integrated city modelling, development, and operations. BIM model data becomes spatially located in GIS data, and GIS data can provides the materialized context for BIM data.

Historically, although BIM and GIS have a common interest in modelling material object types, they differ in their encoding, their use of geometry and semantics, as well as their level of detail.

11 [Engineering] Geospatial information system (GIS)

A.k.a., Geographic information system, geographical information technology (GIT), geoinformation and environmental planning, geomapping, geovisualization, earth built visualization, world information system.

GIS is an integrated system of spatial and conceptual information about the real-world that has been abstracted and simplified into a digital database upon which analysis can be conducted to facilitate the production of solutions to spatial problems using maps and other geographic information. GIS is a geo-spatial computing interface, wherein, geo = earth (or world) and spatial = 3D (+ time). A geo-spatial information system (GIS) is a description of the environment (e.g., world, earth, spaceship, etc.) past and present. A geographic information system (GIS) is a computer system for capturing, storing, checking, and displaying data related to positions on Earth's surface. By relating initially unorganized data, GIS facilitates a better understand spatial patterns and relationships.

STATEMENT: *Location is a critical piece of information in order to address societal-level problems. Planning a material environment concerns problems and data that are inherently spatial.*

GIS has three primary purposes:

1. The collection and storage of spatially related information.
2. The mapping of data onto a geospatial (world) environment.
3. The analysis of the geospatially mapped world.
4. The visualization of the geospatially mapped world (for user observation).

In this context, a 'map' is a scaled model [data-set] visualization of a real-world reality. Maps convey useful information for navigating within a real-world, spatial reality. Mapping, for example:

1. CAD (mechanical drawings) and BIM (mechanical asset data) are map data that may be added as sources in a GIS world scene to become a geo (world) multi-layered data set that accounts for human constructed assets (objects in the scene). These assets (CAD & BIM) become placed (located, positioned) in a geo-spatial scene.
2. Thermal imaging monitoring data from multiple sensors can be combined to form a continuous thermal image map of the whole environment. This image map can overlay all other real-world representational maps.

11.1 The components of a GIS

> **NOTE:** *Information systems exist as the conjunction of [specialized] software and computer hardware.*

The elements (or, components) of a geospatial information system (GIS) are:

1. **Database** - where map information is stored. Sometimes called a geodatabase (GDB).
2. **Software** - that which runs (operates/processes information, computes) the database.
3. **Hardware** - the physical machine that runs the software, computing data and code.
4. **Network** - the physical and digital aggregation of the whole information system.
5. **Coordination procedures** - how data is collected and moved.
6. **Analysis procedures** - how data is analyzed.
7. **People** - the users who benefit by having problems resolved [more easily] from the usage of the geospatial information system.

11.2 GIS data input

Note that data can be brought directly into the GIS software from real-world surveying sensors, a 'point cloud' being one such example. In other words, laser sensors survey the real world to produce a 'point cloud' data set [of the current, real-world], and that data set is imported ("brought into") the BIM-GIS software. It may be brought.

11.3 GIS Software

In concern to commercial software, the software "*ArcGIS*" is one of the more well-known and widely-used GIS software products. "ArcGIS" software is produced by "Esri Corporation".

> **NOTE:** *What does the Arc in ArcGIS stand for? The Esri Community Forum has a thread on "The meaning of Arc." [community.esri.com]*

11.4 Geospatial relationship types

An information technology for referencing and data storage of spatial analyses. Geospatial relationships can be modeled between the feature classes, enabling more advanced GIS analysis. The more common types of geospatial relationship data structures in the geodatabase are:

1. **Topology** - Defines and controls data integrity rules for features. For example, there should be no gaps between polygons. It supports topological relationship queries and navigation, such as feature adjacency or connectivity and synthetic feature editing tools, and allows feature construction from unstructured geometry (for example, constructing polygon features from line features).
   - **Geometric network** - Consists of a set of connected edges and junctions (line and point features) that, along with connectivity rules, are used to represent and model the behavior of a common network infrastructure in the real world. Examples of resource flows that can be modeled and analyzed using a geometric network include, but are not limited to: water purification nodes and distribution conduits, power generation points and electrical lines, gas storage points and gas conduits, telecom points and lines, and water flow in a stream.
     A. For example, the streets in a streets feature class could be categorized into three subtypes, each with slightly different geometric properties and affects: local streets, collector streets, and arterial streets.
2. **Network dataset** - consists of a set of connected edges and junctions, as well as turn features, along with connectivity rules, that represent and model the behavior of a network systems. Examples of undirected network flows that can be modeled with a network dataset include, but are not limited to: transportation pathways and conduits in and between cities; electrical power and telecommunications pathways within and between cities.
3. **Terrain** - a data structure that is generated from a mass collection of elevation measurement points, typically from remote-sensing data sources. It is a triangulated irregular network (TIN)-based data structure with multiple levels of resolution and is used to represent surface morphology. A terrain is used for 3D surface modeling applications.
4. **World surface** (cadastral fabric, polygon fabric, parcel fabric) - a continuous surface of connected polygonal shapes ("parcel features", "parcel polygons") that represents the record of survey for an area of land surface (a world space). This data structure enables GIS data to be integrated with survey data to maintain a consistent and accurate survey record. Spatial accuracy in the world polygonal surface ("fabric") is required.

Additional business logic in the geodatabase, in the form of subtypes and attribute domains, can also be applied to GIS data. This additional layer of influential abstraction is likely to overlay in such a way that it obscures the fulfillment of all individual humans in the real world. For example, three jurisdictions in a market-State could be categorized into three boundary sub-types: legal jurisdiction 1, legal jurisdiction 2, and legal jurisdiction 3; or, market 1, market 2, market 3; influence zone 1, influence zone 2, influence zone 3.

Table 15. Engineering Approach > GIS*: Table shows datasets in a standard geodatabase.*

GIS Data	Geodatabase Dataset
Coverage	Feature dataset containing feature classes
Shapefile	Feature classes
Raster data (e.g., satellite images, air photos, scanned maps, digital pictures, etc.)	Raster dataset and/or raster catalogue
CAD data	Feature dataset containing feature classes
Surface modeling or 3D data	Terrain
Service (utility) network data (e.g., life, technical, facility; water, telecomm., energy, etc.)	Geometric network
GPS coordinates	Table of x,y coordinates that can be generated into a feature class.
Survey measurements	Cadastral fabric

12 [Engineering] Building information modeling (BIM)

A.k.a., Building information management (BIM), architecture, construction engineering.

The 'B' in BIM stands for 'Building, the verb' not 'Building, the noun.' Building information modeling (BIM) is a process supported by various tools and technologies involving the generation and coordination of digital representations of physical and functional characteristics of places. BIM is the creation of a 3 dimensional virtual model that hat contains all of the relevant project information (from one or more sources) about an object (created as a project). Today, the definition of BIM has broadened into an approach that integrates the previously isolated functions and work-flows of geographic and building information. BIM is characterized by information about real-world building objects. BIM in general is concerned with data collection, modification, and application in the context of building geometry/shapes. A BIM object is an object (geometry, assembly) that has data associated with its sub-object geometry. A BIM object has relevant data associated with all of its geometry in order to visualize, standardize, tabularize and calculate quickly. BIM is a digital representation of a building's physical and functional characteristics. BIM workflows and software are also necessary to appropriately prevent clashing of systems during construction and operation. BIM clash detection software ensures that every object in the system fits correctly and does not overlap or clash/collide in any detectable way, and does so automatically in the software. When something doesn't fit, the software will inform the designer/decider of the conflict.

> **NOTE:** *There is also the idea of socio-technical OIM (object information modeling); every object in the habitat system has an OIM associated with its geometry; herein, BIM as building-object information modeling would be a sub-classification of OIM.*

BIM should ensure that right information about real world working objects is available to the right person at the right time. BIM integrates all project [world] asset data about real world (or, potentially real world) objects.

BIM is based on the idea of a continuous use of a 3d digital CAD building or infrastructure model over the entire life cycle of an engineering construction and operation process - from the design, through the planning and execution, to the operation and decommissioning of the project.

In a sense, building information modeling (building information coordination/management) comes down to how the "building" information is being used.

BIM is a process, a workflow:

1. BIM design model.
2. BIM construction model.
3. BIM operations model.

> **NOTE:** *BIM-type frameworks allow any change to a design (or operation) to be traceable.*

As a conception, 'Building Information Modeling' (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition. Regardless, in common application, BIM is a very broad term that describes the process of creating and managing digital information about a built asset such as building, bridge, highway, tunnels, transport, communications, and cities, and so on.

Information modeling is a modern early 21st century computer technique to associate information (concepts with meaning) with mechanisms (shapes with objects). The benefits of information modeling (using 3D) include:

1. Improved understanding and collaboration through ease of visualization in 3D.
2. Data accuracy through tabular information input associations with objects (3D, shapes, geometry).
3. Clash detection through software that detects clashes and conflicts within a project, reducing future errors. Note here that the consolidated clash-free model is also known as a: conformed model, confederated model, clash-free software encoded model, conflict-free model, etc..
4. Efficient documentation through software that intuitively streamlines the construction of documents.

The common BIM execution plans include:

1. Contact info.
2. Software version.
3. Hardware specifications.
4. Information exchange.
5. BIM uses.
6. BIM goals.
7. Standards (and file formats, naming convention, methods of sharing, model protocols).

BIM relevant project plan delivery phase life-cycle:

1. Pre-design.
2. Design.
3. Documentation.
4. Construction.
5. Operation.

Viewing and tracking team work upon assets includes (in the BIM style):

1. Software (asset and building coordination):
   A. Autodesk ecology example - Revit, BIM 360, BIM 360 Team (tracks team changes), BIM 360 Glue, Fusion.
   B. Power BI - data analytics for a project that produces dashboard analytics of a project.
   C. Dynamo is an open source visual programming platform supported by Autodesk and available as a free Revit plugin. Dynamo is shipped along with Revit since version 2016 R2, but you can always download the latest version from [dynamobim.org].

Today, BIM means different things to different producers and users of the "BIM" framework. Generally, BIM refers to the coordination of data, drawings, design, construction, and operation of assets (in general) and building-related assets (in particular).

How does sharing occur?

1. Central location - where people upload, view, and share content.
2. Cloud drive - someone uploads, another downloads; the server stores the file, user downloads to view or can view online, and possibly even, markup online.
   A. Google drive, Dropbox.
3. Single file revision - everyone works together on a single file that is stored on aa network.
   A. BIM 360 Team, Revit server (if you have your own network).

> **CLARIFICATION:** *Although BIM contains the word building, the BIM framework can be applied to any real world asset or system of assets, such as a city, within which some of the assets are buildings.*

Future BIM naming integrations are likely to include:

1. AI SIC = Structural Information Computation.
2. AI Society = information task potential for automation. Allows for optimized information coordination.
3. AI BIM = HSS task potential for automation. Allows for optimized 3D coordination. Allows optimized structuring. Daylight optimization for all structures in a city, and with all relevant available data. Allows optimized documentation.
4. Robotics may assemble buildings in the system, for those cities who value that level of technical efficiency.
5. A computer is programmed with a set of rules, and in a socio-decisioning structure, some of these rules are parallel inquiry thresholds, which the computer is given specific instructions not to

exceed for some explicit condition.

There are work-oriented views that BIM software can broadly express:

1. Design BIM.
2. Constructions BIM.
3. Operations BIM.
4. Or, some combination of BIM.

12.1 Building information models (BIM)

Building information models (BIMs, the noun) are data stored in files (often, but not always in proprietary formats and containing proprietary data) that can be read, shared, or networked to support decisioning regarding building assets. BIM software is used to plan, design, construct, operate and maintain most physical infrastructure (e.g., cities, buildings, bridges, tunnels, railways, water storage and transportation, etc.)

12.1.1 BIM use case scenario

The following services are active for the building in both BIM case 1 and BIM case 2.

1. Habitat power sub-system - An amount of electrical power was required to be produced (energized) to operate this service.
   A. Frequency and coherency.
   B. Power processing service.
2. Production socio-technical sub-system - An amount of material mass was required to be produced (manufactured) to operate this service.
   A. Shape and composition.
   B. Production processing service.
3. Data socio-technical sub-system - An amount of information data was required to be produced (calculated) to operate this service.
   A. Logic and computation.
   B. Information processing service.
4. Transport socio-technical sub-system - An amount of mass re-positional movement was required to operate this service.
   A. Transport and coordinates position.
   B. Transport processing service.

12.1.2 BIM asset modeling

Assets may be organized into the material [requirements] categories of:

1. Soft (computational/informational) interface material ("ware").
2. Hard (physical/informational) interface material ("ware").
3. Training - functional materials.
4. Sensors - collect data associated with physical and functional characteristics.

This contribution framework gives every user an understandable view of how to develop socio-technical competence, and every InterSystem Team member (as someone with socio-technical competence) an operational, execution view of the system. When an element is a model is changed, every view is updated, with the new change appearing in simulation, conception, section, elevation, and sheet views.

Assets may be organized into the material environment

1. Standards - societal design specifications.
2. Guides - guides, best practices, agreements.
3. Work-flows (because it's 'team' view, not command 'disciplinary' view).
4. Modeling quantity.
5. Metrics.
6. Constructs.

Assets (BIM assets) may be visualized together as deliverables with different 'shape' metadata, including at the high level:

1. 3D visualizations (rendering).
2. Coordinated DWGs (unified file system of accurately shaped objects and their metadata, revisioned).
3. Basic quantities (measurement data system, described in detail in the material system specification).

Assets (BIM assets) may be visualized together as information process deliverables, including but not limited to:

1. Thermal studies.
2. Illumination analysis.
3. Structural analysis.
4. Compositional analysis.
5. Constructability.
6. Pre- and fabrication.
7. Asset tracking (on 'physical' data side; and, issue tracking on 'informational' data side).
8. Global information system tracking all assets through integrated, informational and positional, coordinate system (e.g., BIM/GIS overlay of a city collaborated upon by a local and global InterSystem team living in the network of integrated city service systems).
9. Photogrammetry.
10. Field BIM.
11. Protocol.

Once we have a project plan we can start talking about how we share and collaborate among the individuals contributing to this common organization.

12.1.3 Asset model storage

A digital representation of an asset needs to be accessible quickly in a distributed environment that can be updated and upgraded to adjust to more complex query, analysis, and inspection over time and across the lifespan of the asset.

12.1.4 BIM in Application

A highly simplified application of BIM must involve at least the following design-operational elements:

1. **Architectural BIM** - material, dimensional composition of [a commonly standard] space, forming an ID (e.g., building ID, floor ID, room ID, room dimensions, spatial plan and simulation.
2. **Mechanical BIM** - technical equipment IDs (and standards) of one of the HSS technical sub-systems (production, transportation, calculation, etc.; e.g., mechanical equipment).
3. **Electrical BIM** - electrical equipment IDs (and standards).
4. **Pressurizing BIM** - Pressurizing equipment IDs (and standards).
5. **Hydraulics BIM (a.k.a., plumbing BIM)** - Hydraulics equipment IDs (and standards).
6. **Atmospheric BIM** - atmospheric equipment IDs (and standards).
7. **Incident response (a.k.a., protection)** - Incident protection equipment IDs (and standards).

12.1.5 BIM Project lifecycle phases

The BIM project life-cylce phases are:

1. **Planning** (Plan project):
   - A. Knowledge management tools.
   - B. Requirements analysis .
2. **Design phase** (D, Project design):
   - A. D1 - conceptualization, resource information flow programming, and for market, cost planning.
   - B. D2 - architectural, structural, and systems design.
   - C. D3 - analysis, detailing, coordination and specification.
   - D. System design deliverables:
     1. Visualization.
     2. Analysis.
3. **Construction phase** (C, Project assembly):
   - A. C1 - construction planning and construction detailing.
   - B. C2 - construction production, manufacturing, and allocation, and in the market, procurement.
   - C. C3 - as-built and handover/integration, and in the market, commissioning.
   - D. System construction deliverables:
     1. On-site construction.
     2. Off-site construction (and transport).
     3. Procurement and deliver (and inspection).
4. **Operation phase** (O, Project becomes 'service' in 'operation'; Asset & Facility Management):
   - A. O1 - occupancy and operations.
   - B. O2 - service continuation, and maintenance.
   - C. O3 - decommissioning and major re-programming.
   - D. System operation deliverables:
     1. InterSystem planning and operations scheduled coordination.
     2. Team and system task operations.
     3. Simulation of operations (from information flow to materialization).
     4. Incident response operations.
     5. Resource operations.
     6. Information operations.
     7. Re-use/cycle operations.
5. **Social feed back deliverables** (a.k.a., model-based optimizations to enable process optimization across project lifecycle phases):
   - A. Resource survey.
   - B. Materials coordination.
   - C. Knowledge visualization.
   - D. Algorithmic value-transparent decisioning.

12.1.6 BIM Design phase

Preparatory steps in the design phase:

1. The design phase results in a design InterSystem team role of intent model consisting of discipline BIMs (i.e., Habitat service systems, HSS'). To optimize stability, each stakeholder must have defined access [privileges] to the HSS core source algorithm and information system interface. This infrastructure allows each stakeholder to update data through the HSS that is specific to their InterSystem Team accountability role (or "discipline"). For example, the architect InterSystem team (as the InterSystem engineering stakeholders) only can make space-related updates from the decision model to the extant, material habitat service system (HSS). This "as necessary" access diminishes the possibility of one stakeholder overwriting another's decision (discipline) data, and maintains the integrity of model data transferred to the IWMS.
2. Information exchange during the design phase: Requirements deliverable
3. Requirements specify existing space and equipment standards and desired nomenclature for the new building. For spaces, an owner's

requirements may specify the space classification nomenclature (e.g., OmniClass) unique building and floor IDs/names, room numbers/names; and room standards. A room standard is a collection of properties that define a space. Room-standard properties may include room use, room type, cost/unit area, maximum room area and maximum occupancy.

4. For equipment, an owner's requirements may specify equipment classification nomenclature, equipment IDs and equipment standards. An equipment standard is a collection of properties that define a piece of equipment. Such properties may include equipment category, description, manufacturer, dimensions, model number and power specifications, among others. With the allocation of equipment (i.e., assets, service objects) come equipment usage protocols (standards) as predefined ways and accountabilities of using common heritage resources.
5. If the design includes spaces undefined by existing space standards, IWMS or design-side stakeholders can create new and/or modify existing decision protocols (e.g., space standards).
6. The InterSystem team members create these new instructional standards via the engineering interface of the unified societal information system or the societal information systems model (a visual interface) to the unified societal information system. For example, an architect of sufficient accountability on an architectural InterSystem Team can define a new room standard for a specialized lab and apply this room standard to all poly-lined spaces that will serve as specialized labs. This eliminates the need for tedious and error-prone manual entry of properties for spaces with similar purposes. As a result, the InterSystem Teams can immediately access updated spatial information, which is synchronized from a model into the unified societal system; then use it to accelerate access occupancy and service programs. Information exchange during the construction phase: Design specification
7. The team "lead" decision coordinator (a.k.a., construction BIM manager) communicates to InterSystem team members (e.g., mechanical team, electrical team, etc.) the availability of tasks, which are selected by team members; when selected the team member acquires a set of operational requirements (i.e., responsibilities) for information that is entered, reviewed, and/or executed within and/or upon the societal information system. These InterSystem teams coordinate accountability for all materializations (digital and material) for their respective Habitat InterSystem Team (i.e., "trade") role, position, and tasking.
8. Once the space design, floor plans and equipment information (IDs, locations, equipment standards) are available in the societal information system, engineering construction (installation) teams update the habitat service system.

13 [Engineering] BIM and GIS integration

> **NOTE:** *Even better than BIM-GIS conversion is BIM-GIS integration; even better than a BIM-GIS integration is a unified societal system. Wherein, CAD-GIS conversion - CAD data are validated and then converted into GIS data. BIM-GIS conversion - BIM data are validated and then converted into GIS data.*

Together, the BIM and GIS workflows (data sets and calculations) exist to design and express what is happening in the continuous now of a material service system operation (i.e., the global HSS). BIM and GIS data become integrated into an actual and potential (potentially actualizing) global societal-information city network. Projects are delivered through structured design and documentation that produces safely and optimally decided constructions that provide services to humanity, which are monitored and controlled by all contributing humans. That which may be built safely into the material environment (BIM data) becomes spatially positioned and sensed in a geospatial environment (GIS data). The simulation of what is and what could be, at both the conceptual-information and material-realization levels. The four axiomatic information realization components of:

1. Information.
2. Conception.
3. Spatial.
4. Iteration.

Together, all is information, wherein conception [by consciousness] allows for understanding information, the spatial conception is the 3D materialization of information [into plannable experiences], and the iteration conception is the 4D temporalization of information [into human individual, conscious experience].

Building information modeling (BIM) is a process involving the generation and computation of digital representations of physical and functional characteristics of places (i.e., of objects). BIM is a virtual representation of a construction project. It is an integrated process that uses intelligent, digital information to explore, develop and test physical and functional characteristics of a project before it is built. In short, it is a highly detailed representation of a an environment, such as, a building.

BIM represents a series of parametric objects that composed together to form a building model which carries all information includes their physical and functional characteristics and project life cycle information. Since BIM represent the detailed geometrical and semantic information of the building, the application of GIS is needed to manage the construction project's information resources in a material, positional space. GIS can use information from many different sources, in many different formats and can link data sets together by common locational data such as coordinate or postal zip code. Besides, GIS can use combinations of data sets to build and analyze integrated information and also can convert the existing digital information into a form that meets the user's need. From this point of view, GIS can complement BIM function in order to develop a systematic platform for construction purpose. Finding of this study, there are some drawbacks in this technique especially in the construction application in term of data sharing, data integration and data management. Furthermore, the integration of GIS in BIM is studied and potential techniques are shown to overcome the drawbacks of the construction application.

1. Integration of BIM data into a 2D/3D GIS database.
2. Data exchange between BIM and GIS.
3. Integration of BIM construction data.
4. Integration of existing geospatial survey data.

The models created in modern BIM design processes are sophisticated enough to simulate construction to find defects early in design and to generate highly accurate estimates of resource requirements throughout dynamically changing projects.

A societal information and decision support system can now handle billions of events a second from live sensors, serve visualizations from petabytes of 3D model content and imagery to a browser or mobile phone, and perform complex predictive analysis scaled over multiple dispersed processing nodes.

The 3D model generated during BIM design processes is a record of a change to a physical asset. Visualization can be part of the process in that it helps humans understand the dynamics, characteristics, and aesthetics of a proposed design.

All habitat (city) infrastructure is BIM (e.g., the domains of rail, roads and highways, utilities, and telecommunications). When information is 'built' into the environment, it is generally referred to as 'infrastructure'.

Any agency or organization that manages and builds fixed physical assets has a vested interest in making sure that their design and engineering contractors use BIM processes.

Note that BIM data can be used in operational workflows for operational coordination and control.

When seen as a process, integrating GIS technology with BIM becomes vastly more complex than just reading graphics and attributes from a 3D model and showing them in GIS.

How do users need to use a wide range of project data in geospatial context. We also find that focus on the model sometimes means we've overlooked simpler, more basic workflows that are essential to the whole story, such as using accurately collected field data on a constructions site to link location and model data for inspection, inventory, and survey.

> **INSIGHT:** *All information is pattern. Because*

all information is pattern, all materialization is pattern. All patterns of materialization, because they are only information, can be changed by a capable consciousness who understands the patterns of creation.

13.1 Spatial solution visualization resolution

In the study to visualize space re-creation, in time together, as a population, there is a requirement for coordination of and between:

1. Plans (design files):
    A. Architectural (Arch).
        1. Sketch design (SD).
        2. Design development (DD).
        3. Construction documentation (CD).
        4. Mechanical (MEP).
    B. Structural (Struct).
    C. Site (Site).
    D. Operation (Ops).
    E. Incident (OpsInc).
    F. Cycling (Cyc).
    G. Market-State (no acronym).
2. Drawings (drawing sheets):
    A. Mechanical.
    B. Structural.
    C. ...
3. Project files:
    A. Images/photos.
    B. Models (level, dimension).
    C. Renderings.
    D. Simulation.
    E. Descriptions.
    F. Explanations.
    G. Specifications.

**Folders and sub-folders have permission to control which role, individual, or team (organization/business) can access this data. These permissions are cascading so that a sub-folder can have its permissions controlled by a higher-level folder. Roles and permissions for sub-folders may be inherited from the parent folder. A role(s) must be selected for that folder, then the role(s) are assigned permissions. Users are able to subscribe to a folder to receive notifications when new documents are uploaded, sheets are published, or content changes. This ensures the user is kept up-to-date with the data in the folders that is important to them.*

To coordinate this information for useful purpose is to facilitate the determined resolution of resource, machine allocation, and human contribution.

13.2 Spatial-informational mapping

Spatial information can be mapped to a semantic reference system. Thematic entities and relationships may be modeled as first class objects, linked to spatial entities through the variables:

1. Located_at (where an object is in relation to others).
2. Occurred_at (motion of object in relation to others).

These relationships from the upper-level ontology for any logistical access system. For example, a technician is associated with a control ability using a set of relationships (Technician—member_of—PowerService_Team—controls_at—GenerationStation—located_at—Spatial_Entities). Spatial information mapping allows for safe and coherent decisioning among a population. Without mapping objects (that with shape) to meaningful processes (motions of two or more objects), decisioning is likely to be significantly reduced in certainty.

13.3 Unified visual software solution

Visual software is the ability to interactivity work with information (spatial and conceptual) in a 3D environment. A unified BIM-GIS, unified, software platform will [seamlessly] integrate these two modalities; using a unified societal platform, these two design and development modalities will themselves integrate with an operations [dashboard, "facilities management"] platform.

There are two software modalities of BIM-GIS integration:

1. BIM as the construction design application (3D object constructing):
    A. Primary objective of software: Design 3D object.
    B. Secondary objective of software: 3D object data layering and analytics. Here, the reference coordinate is the 3 dimensions of space and 1 dimension of time (i.e., the real-world physics of 3D objects).
        1. Object may or may not be animated.
        2. Object may simulate real-world physics and interactions to study object.
    C. Output: A single object mesh with metadata layers. The software can execute and display engineering design instructions. A BIM platform organizes all information relevant to an constructible object.
        1. Data format output standard, example: CAD, CAM, CAE, building models, 3D meshes, etc.
    D. Users design assets: Users design an asset and simulate real-world physics upon the asset to study the asset.
        1. Users create and study individual assets. Users view attribute information for objects.

2. GIS as the in-place constructed visualization application (3D object positioning):
   A. Primary objective of software: Place multiple 3D objects on terrain.
   B. Secondary objective of software: 3D objects data layering and analytics. Here, the reference coordinate is the terrain of the Earth (i.e., the real-world terrain of the planet; spatial).
      1. World may or may not be animated.
      2. World may simulate real-world physics and interactions to study world.
   C. Output: A unified visualization with multiple object meshes associated with real-world coordinates (or potential, real-world coordinates) and world associated metadata layers. The software can execute and display ("dashboard") 3D geoanalytics. A GIS platform organizes all information relevant to objects existing in a world (or, the real world).
      1. Data format output standard, example: I3S OGS, IFC files, world simulation engines (i.e., game engines), etc.
   D. Users design worlds: Users place assets together in a world and simulate real-world physics and organismal interaction to study the world.
      1. Users create and study worlds, which are composed of some world reference frame (e.g., the earth) and individual assets. Users view attribute information for world. Note that some of those assets will previously exist in the real-world, such as rivers on earth, and other assets will be created by humans, such as buildings and bridges.

13.4 Open GIS and BIM standards

Industry foundation classes (IFC) and City Geographic Markup Language (CityGML) are two standards which have been developed independently. Although IFC and CityGML both deal with object geometry, surface materials/appearances, semantics, and their inter-relationships, the information models are different as they are adapted to the specific requirements of the domains from which they originate. An example of a major difference between the models is how the IFC schema is described using the modeling language EXPRESS, which follows the entity relationship modeling paradigm—whereas the CityGML schema is defined using the Unified Modeling Language (UML) and, therefore, follows the object-oriented modeling paradigm. Although both IFC and CityGML are object-oriented, each uses a different formal modeling language.

The semantic model of IFC, in its current version "IFC4 Addendum 2", focuses on buildings and alignments as well as the physical elements of the building construction, such as slabs and beams—whereas CityGML models all major observable natural and man-made entities in a city or landscape, including buildings. To represent entities with their geometric and semantic properties in different granularities, CityGML includes five well-defined levels of detail (LOD0–LOD4). Regarding IFC, a building element might have multiple geometric representations. (Hijazi, 2018)

Additionally, a "Level of Development" concept was introduced by Forum B which, according to Geiger et al. (2015), cannot be directly compared with the CityGML's level of detail. Level of development (LoD) is applied in BIM to reflect the progressions of the modelling geographic representation, from the lowest LoD of general 2D, to the highest LoD of BIM involving 3D models and corresponding detailed non-geometric information. (Hijazi, 2018)

The main problem in the integration of BIMs with geospatial information occurs at the point of transferring the geometric information. Building models use representations such as constructive solid geometry (CSG) and sweep geometry mostly in local coordinate reference systems, while geospatial models mainly use boundary representation (BRep) in global coordinate reference systems. The fundamental difference arises from their distinct modeling paradigms, which are due to the way 3D models are acquired in the GIS domain in the field of BIM and computer-aided architectural design (CAAD). Using GIS, 3D objects are derived from surface observations of topographic features based on sensor-specific extraction procedures. Features are then described by their observable surfaces by applying an accumulative modelling principle [25]. Alternatively, BIM models reflect how a 3D object is constructed. They follow a generative modeling approach and focus on the built environment, rather than on topography. Therefore, BIM models are typically composed of volumetric and parametric primitives representing the structural components of buildings. However, the relation between the two semantic models (IFC and CityGML) for BIM (design model) and geospatial models (real-world model) has been researched to develop common unified spatial applications with minimum conversion overhead. (Hijazi, 2018)

13.4.1 BIM open standard

- The Industry Foundation Classes (IFC) for the BIM domain [ISO, 2013; Building SMART International, 2013].
  - The Industry Foundation Classes (IFC)3 standard is an open data model used in the Building-information modelling (BIM) domain for the exchange of construction models, often including 3D models of buildings. It has also been adapted as the ISO 16739 international standard [ISO, 2013]. Its geometric aspects are however mostly defined or derived from a different standard,

ISO 10303 [ISO, 2014], which also specifies the STEP Physical File (SPF) encoding that is most commonly used in IFC files (.ifc). The IFC standard provides dedicated entities and attributes for geo-referencing models. IFC files can contain many types of classes: 130 defined types, 207 enumeration types, 60 select types, 776 entities, 47 functions, and 2 rules. The geometries in them can use several different representation paradigms which can be combined more or less freely.

13.4.2 GIS open standard

Open Geospatial Consortium (OGC) standards depend on a generalized architecture captured in a set of documents collectively called the Abstract Specification, which describes a basic data model for representing geographic features. Atop the Abstract Specification members have developed and continue to develop a growing number of specifications, or standards to serve specific needs for interoperable location and geospatial technology, including GIS.

The OGC standards baseline comprises more than 30 standards; including, but not limited to:

1. I3S Open Geospatial Community (OGC) Standard.
2. GML (Geography markup language; XML-format for geographical information).
3. SPS – Sensor Planning Service.
4. SensorML – Sensor Model Language.

The OGC standard CityGML for the GIS domain [Open Geospatial Consortium, 2012].

1. CityGML [Open Geospatial Consortium, 2012] is the most prominent standard to store and exchange 3D city models with semantics in the GIS domain. It presents a structured way to describe the geometry and semantics of topographic features such as buildings and roads. CityGML as a data format is implemented as an application schema for the Geography Markup Language (GML).
   A. CityGML supports 5 levels of detail (LODs):
      1. LOD0 is non-volumetric and is an horizontal footprint and/or roof surface representation for buildings;
      2. LOD1 is a block-shaped model of a building (with an horizontal roof);
      3. LOD2 adds a generalised roof and installations such as balconies;
      4. LOD3 adds, among others, windows, doors, and a full architectural exterior;
      5. LOD4 models the interior of the building, potentially with pieces of furniture
      6. (CityGML does not mandate which indoor features need to be modelled, in practice resulting in models with a different granularity [Goetz, 2013; Boeters et al., 2015]).

13.4.3 BIM technical standards naming

Sources for BIM technical naming standards include:

- *BIM Technical Standards: Naming*. (2019). U.S. General Service Administration (GSA). [gsa.gov]

GSA BIM project-based file platforms use a four part file name consisting of:

1. The GSA assigned building ID or site ID.
2. 1 character major discipline / trade designator (from the PBS [gsa.gov]).
3. 1 character minor discipline / trade designator (from the PBS [gsa.gov]).
4. 5 characters to define contained floors.
5. 1 character type designator M/S/C (model, sheets, combined).

Thus, the file naming standard for a GSA BIM file type includes:

- Building Number_Major/Minor Disp_Included Floors_Sheet/Model File + .File Extension
  - For example: IL023ZZ_AC_B4-20_M.rvt

13.4.3.1 Potential architectural-engineering file naming convention

> **NOTE:** *A similar file naming convention can be used for machine-engineering (as opposed to architecture, which is presented in the examples below).*

The following is a basic useful file naming convention:

- (Project Name)-(*architecture-sheet*)-(Name of Habitat)-(Building Identifier)-(Plan View)-(Version)-(Revision)
- For example,
  - auravana-architecture-sheet-AuraCurve-...
  - auravana-architecture-sheet-AuraKraho-Hex392-FL1-Electrical-V039-R182

Buildings are first identified as either primarily bio-construction (Bio) or mineral (Min), and then, the specific building identifier is added, for example:

- ...-BioHex392-...
  - This is primarily a bio-constructed building and the specific building identifier is Hex392.
- ...MinSky284A-...
  - This is primarily a mineral constructed building and the specific building identifier is Sky285A.

Type of architectural engineering plan views include, but may not be limited to:

1. Floor plan view (FL).
   A. E.g., FL1 OR FL2.
2. Roof plan view (RoofPlan).
3. Ceiling plan view (CeilingPlan).
4. Section view (CutView).
   A. E.g., CutView AA OR CutView BB.
5. Elevation (facade) view (Elevation) OR (SideView) with reference to compass directions E, N, S, W, or by degree.
   A. E.g., Elevation-E or Elevation-SE.
   B. E.g., SideView-NW.
6. Site Plan view (SitePlan).
7. Network Plan view (NetPlan).
8. Multiple Views present (MultiPlan).
9. Areas View (AreaPlan).
   A. E.g., Area3B or AreaCC.
10. Openings View (OpeningsPlan).
11. 3D view (3DView).
12. Orthographic view (Ortho...).
    A. E.g., OrthoTop.
    B. E.g., OrthoRight or OrthoN.

Plan views can be mixed, for example,

- ...-SitePlan-OrthoTop-...

Type of architectural-engineering plans include, but may not be limited to:

1. Enclosure.
2. Structure.
   A. Above ground.
   B. Foundation (below ground).
3. Water.
4. Electrical.
5. Informatics.
6. Illumination.
7. Heating, ventilation and air-conditioning (HVAC).
8. Sensory, including: thermal, acoustic, air motion, etc.
9. Furniture.
10. Landscape.
11. Cultivation.

During development, plans are versioned and revised, wherein a new version is a adaptation (of an existing plan) and a revision is an internal change (to an existing plan):

1. Version: V0##
2. Revision: R0##

If there is a letter after a revision number, then it is just a different view (plan) of the same revision. Always starts with A and proceeds alphabetically:

- For example, V001-R003-A, V001-R003-B, V001-R003-C

13.4.3.2 Other sources for file naming conventions

Other sources for example naming conventions include:

- *BIM Guidelines*. (2017). Smithsonian Facilities.
  - https://www.wbdg.org/FFC/SI/SI_BIM_Guidelines_Oct2017v2.pdf
- *BIM Guidelines for Design and Construction (2015). Commonwealth of Massachusetts*. BIM Steering Committee.
  - https://www.mass.gov/files/documents/2017/12/19/bim-guide.pdf
- *BOE CADD Standards*. (2018). City of Los Angeles, Bureau of Engineering.
  - http://eng2.lacity.org/techdocs/CADSTDS/BOE_CADD_Manual_181107.pdf
- *CAD BIM Technology Center Resources.* US Army. Accessed: December, 2019.
  - https://cadbimcenter.erdc.dren.mil/
- *CAD and Image Standards for Construction Documentation.* (2009). Harvard University Planning Office.
  - http://home.planningoffice.harvard.edu/files/hppm/files/cad_standards.pdf
- *Capital Projects Group - CAD Naming Convention*. METROLINX. Accessed: December, 2019.
  - http://docplayer.net/docs-images/77/74592883/images/52-0.jpg
- *CADD/BIM Standards Manual*. (2018). Report No. CPG-DGN-PLN-084. Revision 1. METROLINX.
  - http://docplayer.net/74592883-Cadd-bim-standards-manual-cpg-dgn-pln-084-revision-1-01-31-2018.html
- *CAD Standards Guideline For Facility Documentation and Construction Projects*. Creighton University Planning and Design. Accessed: December, 2019.
  - http://docplayer.net/9068760-Cad-standards-guideline-for-facility-documentation-and-construction-projects.html
- *MIT Design Standards: BIM and CAD Drawing Standards v6.0, Thematic Folder*. (2016). MIT Infrastructure Business Operations, Facility Information Systems.
  - https://web.mit.edu/facilities/maps/MIT_CAD_BIM_guidelines.pdf
- *Naming Conventions (6.2)*. CAD Standards - 3rd Edition, November, 2016. Denver Water.
  - https://dev.simantics.org/index.php/Structural_Ontology

Scholarly references (cited in document)

- Hijazi, I., Donaubauer, A., Kolbe, T.H., (2018). *BIM-GIS Integration as Dedicated and Independent Course for Geoinformatics Students: Merits, Challenges, and Ways Forward*. International Journal of Geo-Information. ISPRS Int. J. Geo-Inf. 7(8), 319. https://doi.org/10.3390/ijgi7080319 | https://www.mdpi.com/2220-9964/7/8/319/htm

Scholarly references (non-cited)

- Griendling, K.A. (2011). *Architect: the architecture-based technology evaluation and capability tradeoff method*. Georgia Institute of Technology, Thesis In School of Aerospace Engineering. https://smartech.gatech.edu/handle/1853/42880
- Hor, Abdel-Hadi. (2015). *A semantic Web platform for BIM-GIS integration*. York University, Toronto Canada. https://doi.org/10.13140/RG.2.1.4176.6643 | https://www.researchgate.net/publication/305775768_A_semantic_Web_platform_for_BIM-GIS_integration
- Kineman, J.J., Srirama, K, Wilby, J., Mobus, G. (2017). *A Framework for Understanding and Achieving Sustainability of Complex Systems*. Systems Research and Behavioral Science, Wiley Blackwell, vol. 34(5), pages 544-552, September. https://ideas.repec.org/a/bla/srbeha/v34y2017i5p544-552.html
- Mobus, G. (2015). *A Systems Science Framework for Understanding the Nature of Governance*. ISSS Journal. Vol. 1, No. 1. http://journals.isss.org/index.php/proceedings59th/article/view/2497 | https://www.semanticscholar.org/paper/A-Systems-Science-Framework-for-Understanding-the-Mobus/c27fa5a62473abaf8102df2f896883c9ff726730
- Mobus, G. (2017). *A Framework for Understanding and Achieving Sustainability of Complex Systems*. System Research and Behavioral Science, Vol. 34, No. 5. https://doi.org/10.1002/sres.2482 | http://journals.isss.org/index.php/proceedings60th/article/download/2939/1011 | https://onlinelibrary.wiley.com/doi/abs/10.1002/sres.2482
- Preiss, O. (2004). *Foundations of systems and properties: methodological support for modeling properties of software-intensive systems*. Institut d'informatique fondamentale, Theses N 2013, Lausanne, EPFL. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.368.5776&rep=rep1&type=pdf
- *Transforming Systems Engineering through Model-Centric Engineering*. (2018). Report No. SERC-2017-TR-111. Stevens Institute of Technology, Systems Engineering Research Center. https://apps.dtic.mil/dtic/tr/fulltext/u2/1058354.pdf

Book references (cited in document)

- Geiger, A., Benner, J., Haefele Kark, H. (2015). *Generalization of 3D IFC Buildings Models*. In 3D Geoinformation Science; Breunig, M., Al-Doori, M., Butwilowski, E., Kuper, P.V., Benner, J., Haefele, K.H., Eds.; Springer International Publishing: Berlin, Germany, pp19–35.
- Maier, M.W. (2009). *The Art of Systems Architecting (Systems Engineering)*. CRC Press.
- *NASA Systems Engineering Handbook.* NASA/SP-2007-6105. NASA. 2007. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150015500.pdf

Book references (non-cited)

- Mobus, G., Kalton, E., Michael, C. (2015). *Principles of Systems Science.* Springer. ISBN: 978-1-4939-1920-8
- Sutcliffe, A. G. *Requirements Engineering*. The Encyclopedia of Human-Computer Interaction, 2nd Ed. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/requirements-engineering
- *Technical risk assessment handbook*. Australian Government Department of Defense: Defense Science and Technology Organisation. 2010. https://www.dst.defence.gov.au/sites/default/files/basic_pages/documents/Technical-Risk-Assessment-Handbook_2.pdf
- Wasson, C. S., (2016). *System engineering analysis, design, and development: Concepts, principles, practices*. John Wiley & Sons, Inc., Hoboken, NJ.

Online references (cited in document)

- *BIM Technical Standards.* (2019). U.S. General Service Administration (GSA). https://www.gsa.gov/real-estate/design-construction/3d4d-building-information-modeling/guidelines-for-bim-software/guidelines/bim-technical-standards

Online references (non-cited)

- *3D-4D Building Information Modeling.* (2019). U.S. General Service Administration (GSA). https://www.gsa.gov/real-estate/design-construction/3d4d-building-information-modeling
- *ACES: A Conference on Ecosystem Services.* ACES. December, 2010. Naples, Florida. https://conference.ifas.ufl.edu/aces08/pdfs/Abstract%20BOOK.pdf | https://conference.ifas.ufl.edu/aces10/agenda_Pres.html
- *CityGML Standard.* (2012). Open Geospatial Consortium. https://www.opengeospatial.org/standards/citygml
- Decaprio, E. (2006). *Root Cause Analysis*. ProjectManagement.Com Wiki. https://www.projectmanagement.com/contentPages/wiki.cfm?ID=233085&thisPageURL=/wikis/233085/Root-Cause-Analysis
- *Mobus, G.* Washington University, faculty member webpage. Accessed: December, 2019. https://faculty.washington.edu/gmobus/
- *Incubator project & Working group characteristics and process*. (2018). Open Source Initiative. https://wiki.opensource.org/bin/Main/Open+Source+Initiative+Working+Groups/Characteristics+of+OSI+Working+Groups/
- *MITRE Systems Engineering (SE) Competency Model*. (2007). Ver. 1.13E. MITRE Human Resources, SEworks Program. https://www.mitre.org/sites/default/files/publications/10_0678_presentation.pdf
- *Question Everything Blog*. George Mobus

weblog. Accessed: December, 2019. https://questioneverything.typepad.com/

- *Spatial reference model (SRM)*. (2004). SEDRIS Technology Conference. https://www.sedris.org/stc/2004/tutorial.htm#srm
- *Structural ontology*. (2012). Semantics. https://dev.simantics.org/index.php/Structural_Ontology
- *Template for writing individual requirements*. QRACorp. Accessed 2019, December. https://docs.google.com/document/d/1KwCCyPflJfgCERUW2L8vpbJ3791JLZUqKdfb1qtDcss/edit#heading=h.26in1rg
- *Sample records for mission operations concept*. Accessed: December, 2019. science.gov https://www.science.gov/topicpages/m/mission+operations+concept.html

TABLES

Table 17. Engineering Approach > Societal Design Phases*: Multi-level societal design phase model.*[1]

1. Joore, P., Brezet, H. (2015). *A Multilevel Design Model: the mutual relationship between product-service system development and societal change processes*. Journal of Cleaner Production, Vol. 97. DOI:10.1016/j.jclepro.2014.06.043

Design Phase	Experience Initial Situation (0)	Reflection (1)	Analysis (2)	Synthesis (3)	Experience New Situation (4)
General Description	Starting state, characteristic of (sub-)system	Values identification, problem definition, discover phase	Objectives for new (sub-)system, define phase	Creation of (sub-) system, develop phase	Characteristics of new (sub-)system, deliver phase
Societal System	S0 - Properties of society	S1 - Value determination regarding societal situation, definition of **societal problem**	S2 - **Human requirements**, based on humanity and values, resulting in objectives for ideal new societal situation	S3 - **Vision development process**, resulting in future vision for new societal situation	S4 - Living in society, executing societal experiment, **new societal situation**
Socio-Technical System	Q0 - Properties of current socio-technical system	Q1 - Value selection regarding socio-technical situation, **system deficiency**	Q2 - Dominant interpretive framework (**social information structuring**) leading to objectives for new socio-technology system	Q3 - **System design process**, leading to proposal for new socio-technical system	Q4 - Experiencing **new socio-technical system** (new societal experiment)
Product-Service System	P0 - Properties of current product-service system	P1 - Value selection regarding functioning of product-service system, resulting in **functional problem**	P2 - Determining functional demands and **functional requirements** to be met	P3 - Design of a new product-service system, **product-service design**	P4 - Using and experiencing **new product-service system**
Product-Technology System	T0 - Properties of current product-technology system	T1 - Value selection regarding functioning of product-technology system, definition of **operation problem**	T2 - Target definition regarding new product and technology, leading to **program of demands**	T3 - **Product design** process, leading to (prototype of) new product-technology system	T4 - Simulation, testing, using and experiencing **new product**

Table 16. Engineering Approach > External Standards: *Systems engineering standard differences.*

	ANSI/EIA-632:1998	**ISO/IEC-15288:2008**	**IEEE-1220:2005**
System life cycle	Assessment of opportunities Investment decision System concept development Subsystem design and pre-deployment Development, operations, support and disposal	Conception Development Production Utilization Support Retirement	System definition Preliminary design Detailed design FAIT [fabrication, assembly, integration, and test] Production Support
Level of abstraction	Medium level	Highest level - process description	Lowest level - task description
Focus of life cycle	Enterprise-based systems (societal systems)	Product-oriented systems (service systems)	Engineering activities necessary to guide produce/service development

TABLES

Table 18. Engineering Approach > Systems Engineering Competency: *This table displays the systems engineering competencies by means of six indicators of effectiveness (of knowledge and experience) in systems [thinking], as recognition, comprehension, guidance to significant application (adapted from INCOSE UK Competency Table, 2015, [incose.org.uk]). All are learners, some learners are experts. Some learners are also sufficiently knowledgeable, skilled, or capable to guide other learners; some learners are guides. Some learners are new to a [systems] complex subject matter and may be being guided. Anyone in a population can have, and can also not have, an awareness this context, that of systems [thinking].*

Systems engineering competency table				
Indicators	**Competency Area - Systems concepts**			
Description - A description explains what the complexity is and provides meaning behind the title	Description: The application of the fundamental concepts of systems engineering. These include understanding what a system is, its context within its environment, its boundaries and interfaces and that it has a lifecycle.			
Reasoning - Reasoning indicates the importance of the competency and the problems that may be encountered in the absence of that competency	Reason why it matters: Systems thinking is a way of dealing with increased complexity. The fundamental concepts of systems [thinking] involves understanding how action and decisions in one area affect another, and that the optimization of a system within its environment does not necessarily come from optimizing the individual system components.			
Expert - The person who displays extensive and substantial practical experience and applied knowledge of the subject	**Effectiveness Indicators of Knowledge and Experience**			
Practitioner (Guide) - The person who displays detailed knowledge of the subject and is capable of providing guidance and advice to others.	Awareness	Supervised practitioner (new contributor, Mentee or newly Contributing learner)	Practitioner (contributor, Mentor or guide)	Expert (all are learners, some are highly capable)
Awareness - The person is able to understand the key issues and their implications. They are able to ask relevant and constructive questions on the subject. This level is aimed at everyone in the population.	Is aware of the need for systems concepts Aware of the importance of: - System lifecycle - Hierarchy of systems - System context - Interfaces - Interactions amongst systems and their elements	Underlying system concepts Understands the system lifecycle in which they are working Understands system hierarchy and the principles of system partitioning in order to help organize complexity Understands the concept of emergent properties Can identify system boundaries and understands the need to define and manage interfaces Understands how humans and systems interact and how humans can be elements of systems	Able to identify and organize complexity with appropriate techniques in order to reduce risk Able to predict resultant system behavior Able to define system boundaries and external interfaces Able to assess the interaction between humans and systems, and systems and systems. Able to guide mentee.	Able to review and determine the suitability of systems solutions and the planned approach Has made significant past contributions.
Supervised practitioner - The person displays an understanding of the subject but requires guidance and/or supervision (e.g., piloting). This level defines those persons who are "in-training" (being mentored or guided) or are inexperienced in that particular competence.				

TABLES

Table 19. Engineering Approach > Requirements > Non-Functional: *Non-functional requirements (simplified).*

Non functional requirement category	Typically applies to Non-functional Type (Data and Process)	Example
Accuracy Requirements	Both	Process: All requirements will be identified and checked. Data: Issue occurrence must be in the past.
Auditing and Reporting Requirements	Both	Process: A record of which users access or try to access process operational processes is required. Data: A record of which user changes an attribute or value is required.
Availability Requirements	Process	Process operate societal service system.
Backup and Recovery Requirements	Both	Process: All services can be made available after unplanned system downtime within 1 working day. Data: All data will be backed-up daily.
Capacity Requirements	Both	Process: A habitat service system can have up to X users. Data: Up to X users can be stored.
Compatibility Requirements	Both	Process: Systems can integrate onwards. Data: User data can be exported for use.
Concurrency Requirements	Process	Process: Up to X users can use the system at once.
Error-Handling Requirements	Process	Process: In the event of the user cancelling or quitting the process, changes made by the user will be reversed.
Legal and Regulatory Requirements	Both	"Process: The user must gain the permission of the authority. Data: All changes made under the condition of authority."
Licensing Requirements	Both	Process: Ther user must gain the permission of another user. Data: All changes made under the condition of ""gifting"" to another user.
Performance Requirements	Process	Process: The user must be fulfilled in real-time.
Precision Requirements	Data	Data: Time of changes to data must be recorded to the nearest second.
Redundancy Requirements	Process	Process: In the event of an unplanned exist the user can choose to restore from working on at the time of the event.
Security Requirements	Both	Process: Only users holding accountability can create a change. Data: The accountable users must be identified and agreed in the past.
Throughput Requirements	Process	Process: User requires X number of resources per day.

TABLES

Table 20. Engineering Approach > Geoinformatics*: Informatics modeling.*

Spatial informatics				
Geoinformatics	**Geoinformation**	**GeoComputation**	**Technologies/ systems**	**Applications**
Spatial models	Spatial databases (map layers)	Computational geometry	Geospatial information system	Life support structure
Spatial algorithms	Cartography (visualization, mapping)	Spatial analysis	Global navigation satelite system	Technical support structure
Spatial reasoning		Informational analysis	Remote sensing system	Exploratory support structure
			Location-based habitat services system	
			Spatial decision support system	
Conceptual informatics				
Informatics	**Information**	**Computation**	**Technologies/ systems**	**Applications**
Conceptual models	Data (unifying unit of information)	Conditional programming	Unified information system	Social support structure
Conceptual algorithms	Base (storage boundary of multiple data)	Conceptual analysis	Global value system	Decision support structure
Conceptual reasoning		Decision analysis	Human sensing system	Material support structure
			Human-based habitat services system	Lifestyle support structure
			Human decision support system	
Simulation informatics				
Informatics	**Information**	**Computation**	**Technologies/ systems**	**Applications**
Motion models	Data (unifying unit of motion)	Procedural algorithms	Habitat service system	Understanding support structure
Object flow	Base (storage boundary of multiple data)	Monitoring analysis	Habitat service network	Contribution support structure
Task reasoning		Intervention analysis	Habitat sensing system/platform	Promotion support structure

TABLES

Table 21. Engineering Approach > Geoinformatics: *Spatial conceptual breakdown.*

Maps	Reference	Human service systems	Coordinates
Points	**Positional reference frame**	**Human service systems**	**Coordinates in a circle system.**
	Global positioning systems	Global habitat service	Grid
	Local positioning system	Local habitat service	Grid
Lines	**Resource reference frame**	**Human service systems**	**Logistics in a transport system.**
	Global access sytem	Global access system	Grid
	Local access system	Local access system	Grid
Triangle	**Informational reference frame**	**Human service systems**	**Coordinates in an information system.**
	Global decision system	Global information system	Database
	Local decision system	Local information system	Database
Polygon	**Spatial reference system**	**Human service systems**	**Coordinates in a spatial system.**
	Building information system (BIS)	Service asset system	Application
	Sensor information system (SIS)	Service asset system	Application
Elevation	**World reference system**	**Human service systems**	**Coordinates in a spatial system.**
	GeoSpatial positional information system	GPS service asset system	Application
Grid	**Level reference system**	**Human service systems**	**Coordinates in a spatial system.**
	Constructable material system	City services	Application
Simulation	**Real reference system**	**Human service systems**	**Coordinates in a spatial system.**
	Global simulation system (Network)	Global habitat operations system	Processes
	Local simulation system (City)	Local habitat operation system	Processes
Operations modeling	**Contribution reference system**	**Human service systems**	**Coordinates in a spatial system.**
	City information model	Contribution to service operation	Task
	Building information model	Contribution to service design	Task
	Landscape information model	Contribution to ecology	Task

TABLES

Table 22. Engineering Approach > Systems Engineering*: Systems engineering instrument factors.*

Instrument Factor	Systems engineering factors
Demonstrations	Are the capabilities discussed actually in operations - have they been demonstrated?
Integrated simulation	To what degree are the simulations integrated, and better yet do different simulations work off of shared models?
Formal analysis	Are the analyses (e.g., property analysis) formal, meaning that they are performed on models automatically?
Domain specific	Are the different types of models related to the domain? For example, control system engineers often use Simulink/Matlab. Also, most modeling and simulation environments are domain-specific.
Domain interoperability	Are the models that are in different, but related domains integrated? Are the models consistent across the domains?
Synthesis/generation	Can the models be used for synthesis/generation of other related artifacts such as code, simulation, analysis, tests and documentation
Meta-model/model transformations	Are the models used in one domain, or for one purpose, transformable into another domain where the well-defined semantics in one domain is carried through the transformation into the other domain; if so are they known to be consistent?
Formal Capability Assessment	How well do the models, simulations and analyses capabilities support the ability to understand the capabilities being developed?
Virtual Accuracy/Margin Analysis	Are the modeling, simulation and analysis accurate? How well do they allow the designers to understand the margins?
3D Immersive Environments	What is the degree to which 3D Immersive Environments are used to improve the understanding (and possibly training) of the virtual systems.
Risk management	Is there proper risk management identification, analysis and mitigations applied based on the use of models?
Predictive analytics	Are there models used to support a quantitative approach to risk management?
Model-based metrics	Are there model-based metrics (or a comprehensive set of model measurements) and are they used to support the management of programs/projects?
Multi-model interdependencies / consistency and semantic precision	If the organization is dealing with many different types of models, are the interdependencies managed and are the models semantically precise enough to manage model consistency?
High Performance Computing (HPC)	Is HPC applied to the modeling, simulation and analysis efforts?
Procedures	Are the procedures for using the models understood, so that we can trust the model outputs to support other related types of analysis, both in terms of technical as well as risk?
Staff and training	With the advances in the technologies associated with models, are the staff and training in place to support the use of models?
Human factors	How well are human factors supported by the modeling, simulation and analysis capabilities? This should consider Usability.
Certification	How well do the models-based automation and practices support certifications (if required)?
Regulation	How well do the models-based automation and practices support regulations (if required)?
Modeling and simulation qualification	How much do we trust our models?

Approach: Deciding

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

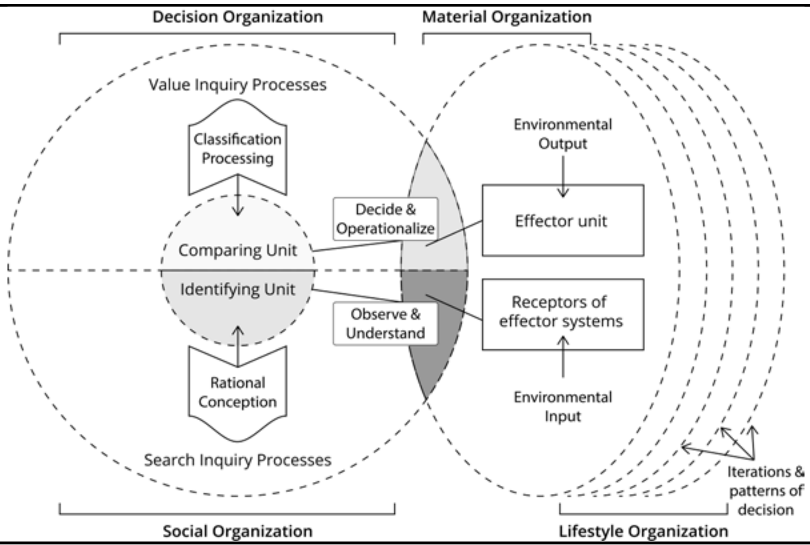
Keywords: decisioning, decision management, decision coordination, decision planning, decision operations, decision development, decision control, decision design, decision resolution

Abstract

A decision resolves into action that has consequence to individual lives. At the societal level, it is possible to come together to form a cooperative 'agreement' system as the first social cooperative coordination pattern. At the societal-level, the first method of cooperative patterning is an understandable kernel logic. It is understandable because it can be logical visualized, and decisioning, can be traced throughout the global system. As a process in the real-world community information systems model, the decision system is a supra-, through to sub-, inquiry process of methods that group to resolve inquiry threshold decisions composed of procedures, for the re-Statement of a new and more optimally aligned societal experience. This project plan accounts for decisions that resolve into actions that affect individuals in a spatial and informational environment. Decisions can be resolved together through identification of common life-cycles and their objective resolution by means of algorithmic computation. Wherein, feedback is measured against an expected alignment in order to ensure execution is of the highest quality. In order to optimize decisioning, a database of calculable information is required.

Graphical Abstract

Figure 6. *The approach to decisioning involves a rational method of accounting for all four societal systems synchronously in order to decide and act upon the next optimal iteration of society by means of explicit organization.*



1 Introduction

A.k.a., Coordinated decisioning, unified algorithmic control, algorithmic decisioning, algorithmic access control, conditional programming, and synthetic intelligence.

An 'agreement' system is the first social cooperative coordination pattern. In community, the first method of cooperative coordinated patterning is an understandable kernel logic. It is understandable because it can be logical visualized and decisioning transparently traced throughout the global commons. As a process in the real-world community information systems model, the decision system is a supra-through-to-sub- inquiry process of methods that group to resolve inquiry threshold decisions composed of procedures, for the re-Statement of a new and more optimally aligned societal experience.

Simultaneously, the decision system is a computational system that uses mathematics. Mathematics is the methodological-science of patterns. Mathematics may be applied at the societal level to optimally resolve a new societal pattern. To most people, mathematics means working with numbers. Today, mathematics is more well understood as the study of patterns, real or imagined, visual mental, arising from the spatial [natural] world or the conceptual [mental] world.

Mathematics is one, but can be commonly subdivided into:

1. 'Mathematics', more commonly known as "higher-order" mathematics resolves new iterations of the whole societal system (as a solution pattern 'state' for the society).
2. "Lower-order" mathematics is the science of expression, equation, operation, or formulation within patterns of an informational or physical nature.

Mathematics may be used to represent spatial objects, their motions, as well as informational objects ('concepts'), and their motion, through computation. Mathematics may be used in the decision system to, for example, describe system/patterns, a mechanical pulley. Within the decision kernel, mathematics may be used to resolve many inquiries, including most importantly, resource planning, scheduling, and access (which may be used as a pejorative, it is used in community to mean the benefit of all that can be contributed to (it is a difference in perspective between the two; free access is not negative, but positive...through a planned and coordinated societal, and hence, decision, model.

Effectively, the shape, motion, composition, co-position, and operation of systems in both the spatial and conceptual world can be expressed/described as mathematical equations most precisely, and only to the level to which they can feedback understanding (machine learning in particular here).

In other words, socio-decisions have socio-network effects that "vibrate" (share) information and resource access instruction throughout a materialized ecological-habitat service network where InterSystem habitat service teams composed of humans and machines operate a societal-level and sub-system coordinated informational-material environmental system [pattern] of local habitat service systems (Read: the local integrated city system). The input of the decision is all the information; ordered, or to be ordered.

The process of the decision system is an algorithm, a procedural resolution to a set of inquiries, no more than that which can be understood by us as individuals, together.

The output of the decision system is a set of complex 'patterns' (probable and actualized). These , and in systems science the concept of 'complex' systems is categorized under complexity theory. Complexity theory is the study of complex systems. A complex system is a system. A system is a set of parts with relations between those parts that interconnects them into a whole, making them interdependent within the whole system.

- For instance, entities are inter-connected in that one entity is pulling on the other; every atom or sub-element (e.g., human-individual entity) is pulling on every other (e.g., atom or sub-element) of the other entity. Note: Is that what is available as a description of 'gravity'? And for consciousness, there is an information inter-connection, and for a physical body there is a physical, spatial interconnection for consciousness. The intentions of a conscious entity are pulling on the intentions of other conscious entities.

That is what we have as a description of conscious social population. Physics exists to explain how every atom, intention pulls on "you",. How is every atom pulling on every other atom is a question of physics. Where does this occur for our population. In society (information sphere) on Earth (spatial sphere).

Order is received through information and material reception, sensing, and order is imposed on that system through importing conception (information) and materialization (energy). This idea can be seen visually in the many real-world-community models. In other words, systems are received, designed, and have order imposed on them as advancement occurs in a certainly [now] uncertain world.

1. For instance, material systems may be influenced to change through reconfiguring energy and matter.
2. For instance, social systems may be free to change through reconfiguring information and understanding.

The total information system, of which the decision system is a part, sustains system complexity - some of the complexity of the societal system (complexity theory applied to societal system) can be viewed as:

1. **Adaptive** System - Intelligence as control, learning as highly probable control due to past experience or preparation, the ability to plan, and necessarily, visualize/graph in order to control a system infrastructure, together.
2. **Self-Organization** System - the ability form, sense, and synchronize patterns, within and from nature.
3. **Decision** System - Non-linear systems dynamics resolve to produce an environment with the probable ability to change [linear] direction if fulfillment if it is required. Non-linearity describes a whole, where the whole is greater than the sum of its parts; or, less than the sum of the separate parts.
4. **Information** System - Information and material dynamics are graphed as a network multi-domain structure (with zoom-scale) of thread-like feedback loops and “tri-” shape-like motion within a matrix.

In the real-world community model, the decision system is a process object that includes a set of objectives in which inquiry resolutions are calculated through an alignment procedure that operates via the methods of absolute pattern recognition as mathematics (mathematical operations to resolve computations) and linguistics (linguistic operations to resolve computations) into the absolute differences of a 1/’yes’ (or, yes to some degree) or 0/’no’ (or, no to some degree) -- wherein, ‘1/0’ is mathematical [number patter] computation and ‘yes/no’ is linguistic[al meaning] computation, both of which follow the principles of ‘pattern’ (a synonym of ‘system’) logic (certainty is never more than 90-99%):

1. Decision system > effectiveness sub-inquiry.
    A. Regions and cities run no less than 90% difference in societal information and decisioning. This means that the information both sub-global information and habitat systems are 90% equivalent and decision results would be the same 90% of the time.
2. Decision system > distributive justice sub-inquiry.
    A. Human individuals acquire no more than 10% over what others have acquired in current personal access.
3. Decision system > regeneration sub-inquiry.
    A. Habitat services run no more than 90% of carrying capacity.

Together, a societal decision is an informational solution “selected” for resolution (to go from probable to actual):

1. Decision plans include product/system/solution files:
    A. Requirements.
    B. Series inquiry.
    C. Parallel inquiry.
2. Decision algorithms include coding/computing sheets:
    A. Software.
    B. Hardware.
3. Decision services include support systems:
    A. Life support system.
    B. Technical support system.
    C. Explorational support system.

2 What is a decision?

A.k.a., What is a choice, an option, a change, a control-gate?

A decision [point] is an informational and/or material point (or, control "gate") that 'controls' the flow of information and/or materiality, in (or through) 'time'. To resolve and execute a decision is otherwise known as the ability to "take control".

In system's control, a decision point is:

- *How* the quantity and/or quality, of a target, is controlled.

A decision system involves access to:

1. Information, perceived as memory (awareness).
2. Logic, perceived as patterning (order).
3. Action, perceived as execution (doing).

In the societal system proposed by this project, the Decision System specification defines and explains the parallel processing of socio-economic information. The Decision System determines solutions and initiates a resolved execution (i.e., change) to the active state of the environment, given a [societal] problem situation. The decision system lists three core inquiry processes that resolve problems into accepted solutions (these inquires are completely defined in the Decision System specification):

1. A recognition of issues (issue inquiry).
2. A set of societal value condition inquiry thresholds (social solution inquiry).
3. A set of technical inquiries (technical solution inquiry).
4. A safety and societal effectiveness inquiry (effectiveness inquiry).

In concern to time, and the decided change to a system in question, the change can occur, thusly:

1. Incremental change: Incrementally, with incremental change to the societal system structure, which reforms it as community. If there is great complexity and the mixing of a diversity of backgrounds, then the change is likely safe when there is gradual approach (slow-step-by-step and strategic approach).
2. Rapid change: Rapidly, with a rapid change to the system's structure. If a rapid change is consented to by all, or sufficient with no objection, then the change is likely safe.
3. Intelligent change: Some Intelligent, strategic mix of rapid and incremental changes; mixed with dynamic decisioning based upon given situations.

Decision control necessitates [at least] the following activities executed by the decision system:

1. Identification of situation.
2. Measurement of progress.
3. Evaluation of alternatives.
4. Selection of alternative.
5. Documentation.
6. Executing of documented decision.

Whenever a decision is taken, the decision space necessarily creates a ranking among possible solutions, as their probabilities of matching decision objectives. And, whenever there is a ranking in the presence of uncertainty, then a numerical-statistical scale emerges.

Consciousness is able to choose (select) a pattern of solution through a decision system configuration, to solve a problem (disturbance). Together, individual consciousnesses can come together to resolve a common decision system configuration to pattern known as 'society'.

In a project-engineering information system:

1. Decisioning is directed by requirements.
    A. Requirements are determined through analysis, which are informed by intentions (e.g., objectives) and a given situation.
2. Designs are directed by specification standards.
    A. Specification standards are determined through synthesis, which are informed by combining useful information over time to form knowledge sets useful for predictable design.
3. Executions are actions (Note that in a project-based information set they, executions/actions, are most commonly called 'activities' or 'tasks').
    A. Activities (tasks) are determined based upon the coordination of the construction (creation) of a material (spatial) information set.
    B. Those who participate in executions, activities, tasks, are contributors as part of the coordinated organization of an InterSystem Team (because there is cooperation, and not competition; if there were competition then another term would be used, such as employee).
4. Results are feedback that are integrated to produce a better, more optimal decision design execution.
    A. Sensors record information into memory storage as data (bits).
    B. Individuated integration processing units perform pattern recognition on data (as discrete math, logic).
    C. Users become aware of more useful information (more knowledge) over time (as conscious awareness, wisdom).

2.1 Decision objectives

The processing of information in the presence of an objective conforms the information result to the objective, by relative degree, as a synthesized information set of information useful for spatial execution [toward the objective]. The objective is an input the user puts into a decision space once the space's code, protocol, or method are "complete", once the decision space is programmed. Therein, the objective is a direction [of navigation] accompanied by a situation, for which the decision processing system must resolve by synthesizing new information from its new (novel) inputs.

The primary objective of any decision is to take a decision to:

1. Dismiss the decision if there is not one.
2. Resolve the decision with a solution selected for action/execution.
3. Use resources efficiently during the decisioning process.

Note that there are three main ways in which the word "objective" is used in a societal system, for navigating and decisioning:

1. **Directionally - direction decision objectives:** Needs (as objectives) and goals (as objectives; functional use requirements; need indicators and metrics).
   A. For example,
      1. The objective/goal is to produce a: service and/or object with a specific function to meet a specific need (demand).
         i. Indicator: The human need...
         ii. Metric: The user stated completeness; the contribution observed recording.
2. **Orientationally - orienting decision objectives:** Values (values as objectives; non-functional requirements; performance requirements, performance indicators and metrics).
   A. For example,
      1. The objective/value is to produce the service/object (composed of resources) in a sustainable way, valuing the objective of sustainability.
         i. Indicator: The plan for x resources in the next master-plan revision.
         ii. Metric: The regeneration rate of x resources.
3. **Operationally - operating decision objectives:** Milestones, event marking (milestone goals as objectives; marked events as objectives; achievement; operational event-marking performance indicators and metrics).
   A. For example,
      1. The objective/value is reach a subscriber count of one thousand people. Indicator: People in subscription pool; number of subscribers.
         i. Metric: Good performance of x duration is brining subscriber count to 1000.
   B. It is important to note that minimizing service downtimes and efficiently coordinating resource utilization (within budget) are fundamental operational objectives for any service.

2.2 Decision mechanism

Whenever a decision is under resolution (i.e., being resolved, taken, acted upon, etc.), the decision space necessarily creates a ranking among solutions (i.e., solution alternatives), based upon an objective; this mechanism uses conditional probability matching between the decision objective and the situation, based a prior pre-programmed logic with a memory of what predictably works as recorded in memory. When there is a ranking in the presence of uncertainty, all actions in an environment will have some degree of uncertainty; in order to account for uncertainty in decisioning, numerical-statistical scaling (mathematical statistics) has emerged and is used. If there are numerical-statistical scales present, then computation can be performed, and automaticity becomes available. When automaticity is materialized, this is often called, 'technology'. Humanity has the potential for generating through this decision mechanism, a unified, highly organized and coordinated socio-technical society with the objective of fulfilling the needs and highest potentials of all individuals.

> **STATEMENT:** *The first step toward "governance" is actually knowing what happens.*

2.2.1 Decision resolution methods/processes

The common decisioning methods include:

1. **Analytic Hierarchy Process** (AHP, and analytic network process) - a visual organization of information concerning the developing system, wherein indicators are prioritized and alternative solution options are ranked (to which a gated decision threshold may be applied). The Analytic Hierarchy Process uses paired comparisons to derive a scale of relative importance for alternatives.
   A. The Analytic Hierarchy Process (AHP) can be used for prioritizing requirements with multiple objectives: value to several stakeholders, importance to the society, risk, or any other type of importance as previously identified. With this method pairwise comparisons between different requirements and weightings between

different types of importance (objectives) are performed.
B. A decision space is opened where "hard" operations research (scientific knowledge) is resolved into an asset through "soft" socio-technical systems engineering. Here, there are technology readiness levels, technology integration "maturity" levels, requirements hierarchies, etc. Here, there are organizational processes and sub-processes in some form of visual order.

2. **Critical Path Method** (CPM, temporal organization of information) - The CPM is a mathematical algorithm for scheduling a set of project activities. The three critical elements:
   A. A list of all activities or processes to complete the project.
   B. The duration of each activity or process should be given.
   C. The dependencies between the activities or processes should be identified; and 4) the availability of a human or active service system to execute the change.

NOTE: *Information useful to a decision is: any signal, message or perception that has an impact on the state of the decision system.*

2.2.2 Decision tabling

A decision table is a visual computational structure to formulate requirements when dealing with complex physical and social rules. Decision tables are a concise visual representation for specifying which actions to perform depending on given conditions. Decision tables are used to model complicated logic; they are algorithms whose output is a set of actions. Decision tables can make it easy to see that all possible combinations of conditions have been considered, and make it easy to see when conditions are missed. The information expressed in decision tables could also be represented as decision trees or in a programming language as a series of if-then-else and switch-case statements.

In a decision table, conditions are usually expressed as true (T) or false (F). Each column in the table corresponds to a rule in the [social] logic that describes the unique combination of circumstances that will result in the actions.

One advantage of using decision tables is that they make it possible to detect combinations of conditions that would otherwise not have been found and therefore not tested or developed. Decision tables should best be constructed during system design, since they become useful to both developers and testers.

For example, a simplified decision table may be:

Table 23. Decision Approach > Decisioning*: In the table, resources are allocated based on resource requirements and availability, which are either true or false conditions.*

Conditions	R1	R2	R3
Resource requirement	T	F	F
Resource available	T	T	F
Actions			
Resource allocated	T	F	F

2.2.2.1 Decision tree

A decision tree is a tree where each node represents a feature (attribute), each link (branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value). For example, in determining whether to play outdoors or not (yes/no decision), the attributes of rain, temperature, humidity, and wind may be considered.

For example, a simplified decision tree, in its tabular (and not visual tree) form, may be:

Table 24. Decision Approach > Decisioning*: In the table, the decision to play as the node, branches out into a probability of conditions.*

Outlook	**Temp.**	**Windy**	**Humid.**	**=**	**Play**
Sunny	Hot	High	False	=	No
Rainy	Cool	Normal	False	=	Yes
Overcast	Cool	Normal	True	=	Yes
Sunny	Mild	Normal	True	=	Yes

2.2.2.2 Decisioning numerical processing (cognition)

Numerical processing (cognition) is composed of the concepts of (cardinality and ordinality):

1. Quantity ('how many?').
   A. Denote numerical quantities (i.e., cardinal[ity]; e.g., 'three trees', '3').
2. Serial order ('which position?').
   A. Signify position in an ordered sequence (i.e., rank, ordinal[ity] meaning; e.g., 'third tree', '3rd').

Cardinal statements (factual inputs; "judgements") can be used as numerical representations of the intensity of alignment [with an objective, preference, and/or requirement direction/result]. Cardinal statements can be used to express the relative importance of alternative [decision] solutions.

To generate a paired comparisons, one [controller] must answer (inquire and resolve) both quantity (cardinal) and serial (ordinal) kinds of question(s):

- Given a criterion or property, which of two sets (solutions, projects) is more important (of a higher priority) according to this criterion, and how much more important (relative standardized priority) is it?

After generating a matrix of paired comparisons for a criterion, the controller uses it to derive a scale that represents the relative importance of the alternatives. When several criteria are involved, the final decision [selection between alternatives] is based on a scale for comparing the criteria and on the several scales of the alternatives with respect to the criteria. The overall importance of the alternatives with respect to all criteria is obtained, if the criteria are independent from the alternatives, by multiplying the weights of the alternatives under each criterion by the relative importance of the criterion and adding over all the criteria. If there is uncertainty either in the judgments of the criteria, or in the judgments of the alternatives or both, the uncertainty is perpetrated to the scales and thus to the final outcome.

2.3 Solution determination

> *A.k.a., Solution selection decisioning; decisioning to determine how to select the optimal design.*

Here, solution selection/determination involves programmed/-able decisioning on how to select the optimal design[-ed] solution. Design optimization requires the sensation and integration (i.e., exploration of) all available possibilities for an optimal solution configuration based on a set of requirements. The selection of a [solution] configuration for [solution] actualization is accomplished through the [economic] solution-decisioning system.

> **NOTE:** *Algorithms and robotics will drastically change the design and build process.*

The ability to select a societal-level solution reveals the potential of an explicit (and open) model for societal operations.

2.3.1 Decision variable determination-acceptance methods

In concern to the decision-solution solving/determination methods, there are multiple types, the most common of which are:

1. **Feasible** solution method - A solution is resolved into a selection by conforming [a structure] to constraints. Note that the constraints must not limit the success of resolving the problem.
   A. Here, the decision variable is a 'constraint'. The constraint(s) are the barrier for acceptance.
2. **Threshold** solution method - A solution is resolved into a selection when there is coherence between the resulting value and a pre-determined value (i.e., when a value meets or exceeds a pre-determined value then the selection taken as a decision). being at most a certain value (percentage) away from the optimal objective value, and not worse than some pre-defined value. Technically, a threshold is a composite constraint (a numerical-value determined by identifying the "level" for that which will and will not be selected as an acceptable decision).
   A. Here, the decision variable is an objectively pre-determined value that acts as a barrier for acceptance.

2.3.2 Utility decisioning

Utility refers to the presence of some purposeful existence, a service (function or operation). Here, utility is a term used to describe the measurement of "usefulness", the measurement between the expectation of a purposeful existence and the presence of that purposeful existence. The a utility model, feedback on an action may be based on its outcome (as more or less aligned with a given direction).

In the ideal, every decision about life can be reduced to a single number known as 'utility', which can be maximized or minimized. The utility value is the expected degree of desirability of some future sequence of events. Utility valuing is a clearly definable and justifiable basis for decisioning. The axioms that derive the basis of utility functions follow strictly from the basis that humans must take decisions. Whenever there is a choice, there is obviously a ranking, and when there is ranking + uncertainty there is a numerical scale.

> **NOTE:** *Intelligent agents perceive utility in actions.*

'Utility' provides a number-value describing (answering the question), "How in alignment with a given objective is a given decision option?"

> **NOTE:** *In economics, utility is a term used to describe the measurement of "usefulness" that a user obtains from any service (good). "Usefulness" can have an objective relations, such a real human needs/requirement (as it does in Community), or it could have no objective relations, such as a subjectively interpreted want (as it does in the market-State).*

In engineering, every high-level decision can be reduced to a single number-value known as 'utility'. Utility is the numerical degree of desirability of some future sequence of events, which has an expected value, and can be maximized by the process of engineering. Here, utility is a clearly definable and justifiable basis for decisioning. Every decision taken can be viewed as a comparison between the utility gained from pursuing one option or another toward the completion of an objective(s).

The axioms that derive the basis of utility functions follow strictly from the basis there exist decisions [that must be taken prior to action].

1. Whenever there is a choice, then there is a ranking of options.
2. Whenever there is ranking (of options) and uncertainty (of environment), then there is a numerical scale.

> **CLARIFICATION:** *If there is an objective and a choice, then that means one choice-option is better and another, or others, are worse.*

Sometimes there is a voluntary choice and other times the environment (or world) pre-determines ("forces") the choice. If societal/life decisions can be defined in this way, then they can be encoded, turned into a procedures and made into a programs, and a machine can run them.

At a social level, utility is sometimes divided into:

1. Decision utility - values and requirements.
2. Procedural and experienced utility - well-being.

In utility theory, stakeholder value is represented by a normalized absolute relation between the possible levels of fulfillment [of a requirement] and the perceived/absolute value to the user.

2.3.3 Production variance

A.k.a., Production uncertainty.

In the general operation of a community-type societal environment, there are no unexpected variances in production; there is no need for business "flexibility" as there is in the market where societal service operations are not harmoniously project planned. When societal-level planning is possible, then human fulfillment can be systematically planned for. At the end of this solution to our societal project, is a greater state of individual competition or a greater state of individual cooperation as an evaluative success screening criteria for execution.

2.3.4 Design decisioning

Each design decision in the entire design process is checked by logical proof at the time that the decision is taken. These checks should be automated as far as possible in standard design automation tool-sets. These tool-sets must be based on a wide and deep understanding of the laws of the relevant branch of science. These must be formalised in sufficiently strict mathematical detail that it is always possible to calculate or prove that a product conforms to its design, and a design satisfies its specification. This can be done only in a mature branch of engineering science in which the basic foundations are sufficiently developed that the consequences of every design decision can be effectively calculated by software. This is far from trivial, since implementations, designs and specifications are usually expressed in different notations, appealing to different concepts and conceptual frameworks, and describing phenomena on widely different scales of space and time.

2.3.5 Human-centered decision system design

A human-centered (a.k.a., human required) decision system protocol is necessarily constrained by humans in the following necessary ways:

1. The system is programmed and monitored by humans (i.e., by human requirements).
2. The system is contextually informed about what humans require (i.e., by issue requirements).
3. The system is appropriately uncertain about what humans require. The system is appropriately uncertain so that the system further inquiries about what humans require (i.e., the system must accept and integrate feedback, so that the system doesn't irreversibly destroy things that are actually required).

2.4 Decisioning uncertainty

There are two types of decision uncertainty:

1. Uncertainty about the occurrence of future events.
2. Uncertainty about the range of solutions used to resolve requirements.

The first is beyond the direct control of decisioning, whereas the second is a consequence of the amount of information available when the decision is occurring.

There are situations where a measure of uncertainty is necessary to decide whether it is:

1. Optimal to proceed with the current optimal solution, or
2. Optimal to acquire more information to remove some or all of the uncertainty.

Uncertainty about human objectives (requirements, preferences, etc.) leads to deferential behavior. In other words, a decision system that accounts for uncertainty can be programmed to inquire more deeply into and/or refer to new human articulations when there is uncertainty in a current decision space. In other words, the system that resolves decisions can (or, cannot) account for uncertainty, and when a sufficient threshold of uncertainty is present, it can (or, cannot) defer to [new] human articulations/requirements. Here, "to defer" refers to deferring the selection of a single decision determination/solution until sufficient information is present to resolve the uncertainty.

2.5 Wrong decisions

If a "wrong" decision is taken, it results, quite often, in deviations from expectations or from expected operational outcomes. It is the work of information coordination and control to ensure that such deviations can be picked up quickly and dealt with before more damage is done.

> **PROCEDURE:** *When a key indicator is not attained, the information systems will flag this exception.*

2.6 Decision gating

Decision gates (a.k.a., stage gates, phase gates, decision way-points, decision points, etc.) act as points where a decision space exists that must be resolved prior to life cycle progression.

1. In concern to engineering, decision gates are typically synchronized with the commencement and termination of a system state change.
2. In concern to life cycles, in general, every stages/ phases provide a decision gates.

In a life cycle, at each gate, several decision options are open. The most common are decision options are:

1. Proceed to execute the next stage.
2. Continue the current stage until the designated exit criteria are satisfactorily met.
3. Return to a previous stage in order to conform to a revised purpose, or a new or preferred solution option(s).
4. Hold the project activity until evident uncertainties or shortcomings are resolved.
5. Terminate the project due to critical changes or an inability/excessive risk to complete.

System stages need to be terminated by well-defined, objectively assessable achievement states. As a result, stages are predominantly overseen according to their work product status, and generally by an evaluation criteria profile of achievement across a range of these work products.

3 [Decision] System life-cycle

A decision system based on information about what humans require, in combination with software development, computing power, and mathematics. A decision system, as part of a unified information system structures the new system with information. The general, operational decision system is split into decisioning levels, according to relevant criteria, each level being composed of one or several decision spaces ("decision centers" or points). The information system contains the information needed by the decision system, and must structure it in a hierarchical way , according to the structure of the decision spaces (centers).

> **QUESTION:** *How well did we do with that prior decision, in terms of its results and what was expected?*

The de-composition of a decision system is performed according to two different axes:

1. The vertical axis is coordination (i.e., is the coordination axis).
2. The horizontal axis is synchronization (is the synchronization axis).

The de-composition in levels for coordination is based on temporal characteristics. The couple of temporal characteristic which defines a level of decision is composed of: The "horizon" (the internal of time over which the decision extends (i.e., remains valid), and the "period" (the interval of time, over which the decision is open/re-considered). The criteria for decomposition for synchronization is a functional one. The origin of this de-composition comes from the "theory of project management", particularly the need to synchronize the flow of information, of products/systems/deliverables with the use of resource.

- **A decision system** is a set of decisions taken with 1 function and 1 level.

A structured approach uses specifications to organize and communicate that which exists and that which could and/or should exist in the real world, given what is known and a direction. The structured approach principally has four life-cycle phases:

1. Initialization.
2. Analysis.
3. Design.
4. Implementation.

Navigationally speaking, a structured approach to navigation together involves, at least the primaries of navigation:

1. Sensing (genetics, environment, habitat, resources,

etc.).
2. Mapping.
3. Planning.
4. Find optimal actions and course.

INCLUDING: *Optimal travel route (plan) and travel time estimation (schedule).*

4 [Decision] Computation

A.k.a., Computational decisioning, algorithmic decisioning, algorithmic control, conditional programming, computational intelligence, decisioning computation, decision support.

Computers perform logic operations. A computation is a logic operation process performed (run or activated) on a computer. Computational decisioning uses information and an objective function (technique) to determine parameter values from operational data. Therein, computational models are built in the virtual world, which can then be made a dynamic system that humans and other systems can feed input to. Computational models will process the input and then provide an output. Intelligence is a computational resource.

"The world we live in today is made of computers. Our homes are designed with computers. We don't have cars any more, we have computers we ride in. We don't have airplanes any more, we have flying Solaris boxes with a bunch of SCADA controllers. A 3D printer is not a device, it is a peripheral and it only works connected to a computer. A radio is no longer a crystal, it is a general purpose computer with a fast ADC and a fast DAC and some software. The grievances that arose from "unauthorized copying" are trivial when compared to the calls for action that our computer embroidered reality has created."
- Corey Doctorow

4.1 Intelligence

APHORISM: *One is limited by one's unintelligence in understanding the intelligence of others.*

Possibly, intelligence is the ability (and/or power) to shape the world in a way that satisfies (results in the fulfillment of) an objective. Subjectively speaking, something that is more intelligent than "you" is more powerful (has more ability) than "you" to change the world. When an intelligent agent interacts, it evolves according to the quality of its choices during the interaction. For the intelligent agent, there is a reality frame in which choice is present. Intelligence is sometimes referred to as information composed of data that has been integrated (or, had valuable "meaning" added).

Intelligence is doing the right thing at the right time. Intelligence is the ability to meet goals (across a diverse range of environments), and to do it flexibly as opposed to rote; it includes optimization, as a narrowing of the future possibilities into greater certainty.

One definition of intelligence is - being able to build an accurate and detailed model of the surrounding world. To consciousness, a more accurate and detailed model means that situations will be better understood and decisions will have a greater confidence of being correct and good. Intelligence allows for greater empathy by being more able to model, and hence, understand the [thinking and behavior of] others. In systems thinking, intelligence is the ability to follow and generate patterns. Intelligence changes the future implications and probable consequences of current decisions.

Intelligence is a tool for resolving inquiries and problems. It is a search in some environment for answers. Intelligence is what is used when there is no immediate answer or solution to a problem. In other words, intelligence is what someone uses when s/he does not know, initially, what to do. It is possible to perceive intelligence by observing what people do when confronted by a problem or a new situation. In this sense, intelligence is an emergent process. To a consciousness, the ability to extract and/or produce significant information from a situation is intelligence. Intelligence is applied on the part of consciousness to gather and process information into an actionable form (i.e., into a form that is usable for decisioning).

Intelligence is a tool; it is an intentionally influential way to accomplish goals. How those goals are selected is a different issue, that is where values become relevant.

It is important note here that that concepts like 'love' and 'intelligence' cannot perform motions; because, they are already concepts that are in motion (i.e., they are dynamic concepts, verbs). Here, intelligence includes speed and time, like power (which, is another verb).

In a sense, intelligence is a continuum with two ends, fluid intelligence to static intelligence.

1. **Fluid intelligence** - the process of considering information that does not fit into a previously accepted view of reality or possibility.
2. **Static intelligence** - the lack of a process of considering information that challenges an established belief (accepted view).

The substitution of belief for fluid reasoning non-optimal, because belief cant be used to increase the certainty of decisioning to correct orientation when it strays of the course of mutual human fulfillment. Hence, in a sense, the substitution of belief for rational decisioning could be considered immoral.

Intelligence is decision support that seeks to answer a specific question for a specific decider (i.e., for a specific deciding entity or decision group). The decider is the specific human being or system that has to take a decision. Useful intelligence is information applicable for decision support; it is the collection and integration processes that facilitate the resolution on a decision space. Intelligence can be conducted in secret, and it can be conducted in the open (in the commons where it is visible to all).

Secrecy in decisioning at the level of society is often used to avoid accountability and to do unfulfilling things in the name of another, without being discovered. Openness is subject to audit, to visibility, to exponentially distributed quality control through the iterative planning and coordination of open commons for access and contribution. Collections, integrations, and resolutions that are secret cannot do not take advantage of available human and technical potential.

Nature creates [as far as is known] two types of intelligence:

1. **Neural networks** (individual intelligences) - neural networks are individualized unites of comparatives with input, adaptation, and output components.
   A. An integrating, self-adaptive network.
2. **Swarming collectives** (social intelligences) - Swarming collectives are composed of individualized neural networks that move together as one to navigating entity, avoiding predators, discovering opportunities, and sharing resources.
   A. A cooperating, socially-adaptive network.

Societal engineering in a community-type society must necessarily must account for both neural network intelligence and swarm collective intelligence.

> **APHORISM:** *Engineers doesn't care what is believed; they care about what is.*

4.1.1 Intelligent agents

Intelligent agents are capable of, and act through: reasoning, learning, planning, analyzing, and decisioning. Intelligent agents are about acting in a way that is expected to achieve objectives. Intelligent agents behave different given the two types of environments:

1. If the environment is deterministic (i.e., static), then intelligent agents are planning and searching.
2. If the environment is stochastic (not precisely predictable, i.e., dynamic), then the intelligent agents are using [Markov] decision processes (MDPs) and [reinforcement] learning.
   A. The [Markov] decision process model contains: A set of possible world states, *S*. A set of possible actions, *A*. A real valued influence (reward) function, *R(s,a)*. A description *T* of each actions effects in each state. The [Markov] assumption property is that the effects of an action taken in a state depend only on that state and not on the prior history.
      1. Deterministic actions: T : S x A → S. For each state and action a new state is specified.
      2. Stochastic actions: T : S x A → *Prob*(S). For each state and action a probability distribution is specified over next states, representing the distribution: P(s' | s, a).

4.1.2 Machine intelligence agents

Artificial intelligence could be used to scan for errors in societal computation.

5 [Decision] Meta-decisioning

Decisions that define the global decision system for a society determine the framework (including: objectives, constraints, resources, methods, measurement criteria, etc.) through which that society continues to exist (i.e., determines its sustainability). The composition of the decision system could be called a meta-decision as it involves decisions concerning other decisions. Note, that it is not always easy to draw a distinction between these two types of decisions (i.e., decisions that are about the decision system itself and decisions that are not).

5.1 Model integrity

Model integrity ensures trust in the model's predictions by understanding and quantifying margins and uncertainty.

5.1.1 Provide trust in model-based predictions with quantification of margins and uncertainty

Blackburn et al. (2016:48) provides an informed example of the analysis of margins and uncertainty in the context of a device. Take for instance a device that is subject to heat, and there is a need to assess some type of thermal uncertainty quantification for that device. The results of an experiment with that device under thermal conditions are reported in a probability distributed bar graph. Blackburn et al., characterizes the margin and uncertainty of the results:

> *"The Mean of the temperature: T, to the lower bound of the threshold (e.g., 72 degrees) characterizes the Margin, and the Standard Deviation (T) characterizes the uncertainty."*

Quantification of margins and uncertainty applies to the lifecycle of an entire product system, with a focus on (Blackburn et al., 2016:49):

1. **Specification of performance** characteristics and their thresholds.
   A. Performance is the ability of system/component to provide the proper function (e.g., timing, output, response to different environments) when exposed to the sequence of design environments and inputs.
2. **Identification and quantification of performance margins**.
   A. A performance margin is the difference between the required performance of a system and the demonstrated performance of a system, with a positive margin indicating that the expected performance exceeds the required performance
3. **Quantification of uncertainty in the**

performance thresholds and the performance margins as well as in the larger framework of the decision.

In general, there are two types of uncertainty that are that account for, quantify, and aggregate (Blackburn, 2016:49):

1. **Aleatory uncertainty (variability)** - Variability in manufacturing processes, material composition, test conditions, and environmental factors, which lead to variability in component or system performance
2. **Epistemic uncertainty (lack of knowledge)** - Models form uncertainty, both known and unknown unknowns in scenarios, and limited or poor-quality physical test data. Models inherently have uncertainty.

The statistical tolerance interval methodology is an approach to quantification of margins and uncertainties for physical simulation data. There is a newer second approach, that of probability of frequency distribution, which is commonly used in computational simulation QMU applications. The probability of frequency distribution approach involves (Blackburn et al., 2016:50; Newcomer, 2012):

1. **k-factor** - margin divided by uncertainty (M/U).
   A. **Margin (M)** - difference between the best estimate and the threshold for a given metric
   B. **Uncertainty (U)** - the range of potential values around a best estimate of a particular metric or threshold

The k-factor provides required engineering analysis to ensure the collected data sample includes measurements that may be used to infer performance in actual use. Additionally, it is necessary to understand the performance requirement to understand the performance threshold and associated uncertainty:

- **Threshold** - A minimum or maximum allowable value of a given metric set by the responsible system.

The probability of frequency distribution method addresses the situation where a performance characteristic has shown the potential for low margin or a margin that is changing (likely getting smaller or there is greater uncertainty) with age. (Blackburn, 2016:50)

5.1.2 Model validation

Uncertainty quantification for simulation models is not strictly limited to model validation. Model validation is the process of comparing model predictions to observed responses for the purpose of assessing the suitability of a particular model. When experimental observations are available for validation assessment, analysts may use the same observations for model calibration. Model calibration is the process of adjusting internal model parameters in order to improve the coherence between the model predictions and observations. If internal model parameters are allowed to be adjusted in this manner, this means that there is some amount of uncertainty associated with the true, or best, values of these parameters. And uncertainty associated with model inputs directly implies uncertainty associated with model outputs. (Blackburn, 2016:60)

Model validation and simulation qualification are ways to ensure "integrity" of the models prediction information. Rizzo (2015) has developed the "Real Space" model validation approach, which was formulated by working backwards from an end objective of "best estimate with uncertainty" (BEWU) modeling and prediction, where model validation is defined as: the process of determining the degree to which a computer model is an accurate representation of the real world from the perspective of an intended use of the model. However, the interpretational and implementation details can still vary widely. (Blackburn, 2016)

Hierarchical model validation seeks to:

1. Identify key physics and material models that are brought together.
2. Validated models at various levels of functional implementation (i.e., is the right model/technology, or combination of models/technologies, chosen for the right reasons?).
3. Identify and utilize interactions and emergent behaviors not present in validation of separate models.
4. Identify "traveling" or "linking" variables that bridge modeling levels.

6 [Decision] Control

A.k.a., Coordination control, change control, flow control, organizational re-alignment/adjustment, decisioning, decision control, orientational control, error correction, issue coordination, monitoring, planning, deciding, purpose, etc.

Once an organisation identifies a direction, it can start measuring (evaluating) progress toward that direction, while reorienting accordingly. Here, to control is to use a referential direction and method to resolve an [orientational] decision space, so that the next iteration of some [oriented] system aligns more greatly (and not less) with that direction [of motion]. Take note that control is a navigational term, which conceives of the ability to intentionally reorient a system in motion. Control is required for a system to respond to external variables (by isolating a state from external influences). In a control system (a.k.a., closed loop control system) there exist [at least] two systems-level inputs (beyond the axiomatic system inputs, open system inputs) necessary for controlling change within the system based upon an awareness of external conditions. Fundamentally, a control system receives information, decides what to do with it, and then adjusts.

In a cybernetic societal system, control [over an environmental system] occurs through operations:

1. Coordination operations [contibution-based].
2. Socio-technical operations [contibution-based].
3. There are also user operations [usage-based].

CLARIFICATION: *In concern to project coordination, a project's lifecycle process groups have processes, and therein, control happens concurrently (i.e., as those processes are executed).*

Control refers to being able to direct and select change. There is control[lability] wherever there is a decision [space]. Change coordination (change management) coordinates (manages) the evolution of a system throughout all stages of its life cycle. Changes to a system are made based upon a change control (coordination or management) system. Change management is the practice of tracking and administering changes and is a key part of project and every system.

Control is power/ability, sufficient, to alter fundamental conditions (so as to shape experience toward an objective). It is important to recognize that control does not necessarily mean to give subjective power to one personality; control can be person/subject independent. For instance, by sharing an adaptive system (both individual humans and their societal system) has sufficient access to modify its' system [to organize/orient optimally]. An adaptive system, a system that cooperates internally, is likely to have a view of control that involves an open source, a shared, source for its [societal operating] protocol. An adaptive societal system has a shared societal system specification as a source of information and decisioning for its own system.

Control also consists of procedures that determine deviations from plans and desired states, and that indicate, and execute corrective action regarding these deviations. This entails gathering data on the state of the output, searching for deviations from the plan, and adjusting the input based on the evaluated results of the output. Project control thus establishes a relatively closed system of causes and effects. It also reduces the risk of failure and the effect of residual complexity and ambiguity.

Control can only be applied (executed) over the components internal to a system. Feedback mechanisms ensure the system has the information necessary for error correction. Control (and also feedback) presuppose planning, at least in the form of setting goals and performance levels, as plans furnish the baselines and standards of control.

Control is:

APHORISM: *If you truly want to understand something, try to change it.*

1. Decisional information processing and error correction.
2. The ability to form a [computational] "space" where a decision can be executed as a solution to intentional motion in an environment.
3. The process of ensuring that executed operations proceed according to some plan by reducing the difference between the plan (or goal) and the executed reality, by correcting for differences.
4. Pre-deciding/pre-planning the change of a system. Control [theory] is based on the explicit premise that the change of a system is, or can be planned. Note that controllability and optimal control usually are recognized as the characteristics (Read: problems) of modern control theory.

The dimensions of control are temporal (linear in progression; or input, process, and output modeled):

1. **Pre-action control** *(preaction)*
   A. **Standards control** - formally identifies what and how action is to be taken. Standards are a form of precontrol, because they are developed and set prior to action.
   B. **Feed-forward control** - conduct forecasts, budgeting, and use real-time computer systems to determine optimal actions.
2. **Concurrent action control** *(action)*
   A. **Execution control** *(a.k.a., concurrent control)* - is exercised thorough supervision and monitoring.
3. **Post-action control** *(postaction)*

   A. **Feedback control** - used to evaluate past activity in order to improve future performance. It measures actual performance against a standard to ensure that a desired result is achieved. Feedback control is reactive (i.e., corrective action takes place after the fact). It may be necessary to change the way information is processed based upon the information received.
   B. **Post control** - identifies deviations from standards and calls for corrective action (is similar to feedback control).

A 'control system' is:

1. A system with the ability to control its own (or others) outputs.
2. An interconnection (dynamic) of components forming a system configuration that will provide a desired system response given a knowable input.
3. A mathematical composition of differential equations. The set of equations can appear in different forms like; ODE (finite dimensional control systems), PDE (infinite dimensional set-up), integral equations and so on. PDE's can be of different types; elliptic, Parabolic or hyperbolic.

Project 'change control' is:

- A process to control the necessary changes that happen during the life-cycle (or lifetime) of a project, or other, system.

There are two principal views into a control-type information system:

1. **The development view** - A control system design and development view (control builder)
   A. The control system is designed, built, and evaluated.
2. **The deployment view** - A control system deployment view (controller)
   A. The control system is moving information and executing pre-decided decisions.

'Controllability' (the ability to control) is:

1. The ability of a system['s dynamics] to be intentionally modified by some environmental or supra-system entity (e.g., a user). Here, usability is sub-condition of controllability.
2. A basic property of systems that is indicative of the ability to control.
3. The ability to steer/navigate a given system.
4. The ability to design control inputs to steer (adjust, correct, change, etc.) the state to arbitrarily values. Observability is concerned with whether without knowing the initial state, one can determine the state of a system given the input and the output.
5. The logical determination, leading to the subsequent selection, of a solution.

A system is controllable if:

- The control is "powerful" enough to steer (change/adjust) the system from any initial state to any desired state in some finite time (t).

Control accounts for (i.e., the following matter significant for effective control of a system):

1. Object - Shape, and the composition of volumes.
2. Motion - Time, and the sequence in which actions are taken.

More technically, 'controllability' is the ability to change ("steer") a system to any desired value in finite time, and provide simple closed-form expressions (math to hardware and software encodings) for constant positive control functions (or transition rates) that asymptotically change ("steer") the system to the desired value. Algorithms encode a pre-determined process for determining the constant positive control value that asymptotically "steers" the system to the desired value.

Observability (the ability to observe, monitor) is a necessary condition for controllability. Without observability, there is no coordination and/or no verification of control.

Control is an organizational [coordination] function. Control checks for errors in the oriented result, decides, and takes corrective action, so that deviation from objectives, requirements, standards, etc. are minimized and states goals of the organization are achieved (in the desired manner). Today, control is a "forseeing" (i.e., probability-based) action/activity, whereas the earlier concept of control was used only when errors were detected (would a change then be taken). Control in coordination ("management") means:

1. Setting standards.
2. Measuring actual performance.
3. Taking corrective action.

As a forward/probability integration tool, control functions to monitor completion of the work, indicates on current progress, and match conditions to quality output, over time and simulated.

Control involves looking at the variance between the work performed in project execution, against (Read: in comparison to) what it was required ("should") look like as a realization.

> **INSIGHT:** *There is a difference between a controlled fire and a fire that burns down a house. Just as the dose makes the poison, the structure makes the control.*

6.1 Control and coordination (and communication)

A.k.a., Direction, control, communication; command, control, and communication; the service triad.

Any system capable of effecting an environment through some interface must sustain a system for control of the effector and coordination of information, which is accomplished through shared communication within the system. The control and coordination in human beings take place through an integrated nervous hormonal system called, the endocrine system. In order to communicate control and coordination signals throughout a system, a common model and method of modeling is required.

Through certain decisioning in a dynamic environment the service triad becomes visible:

1. Control - without control, certainty of service is low.
2. Coordination - without coordination, accessibility of service is low.
3. Communication - without communication, viability of service is low.

6.2 Controllability pre-requisite to validity and reliability (error correction)

QUESTION: *What is controlled? A system's software and/or hardware [as discrete logical elements in a dynamic] is controlled. What is there to control? The flow of information and changes/modifications to materiality (software and hardware).*

Controllability is a prerequisite for the evaluation of validity and reliability. In order to make research results controllable, researchers have to reveal how they executed a study: how were data collected? How were respondents selected? What questions were asked? Under what circumstances was the study executed? How were data analyzed? How were conclusions drawn? The detailed description of a study enables others to replicate it, so that they can check whether they get the same outcomes.

Reliability is a concept that seems to be easy to grasp but nevertheless difficult to define. In general, something is called "unreliable" when it cannot be depend upon, when it cannot be trusted. For example, a car that occasionally fails to start is unreliable. A person who does not keep his promises is unreliable. The general association of reliability with dependability and trustworthiness holds for research as well, but it has a more specific interpretation.

The results of a study are reliable when they are independent of the particular characteristics of that study and can therefore be replicated in other studies.

A common strategy to determine the reliability of a measurement is to repeat it. By repeating a measurement, one can determine to what degree measurement results differ from each other. If there is no difference, the research results seem to be independent of the specific characteristics of both studies. Repeating a measurement has at the same time the advantage that measurements can be combined in order to increase reliability. Combining measurements may consist of calculating the mean of a series of values, but it may also consist of an attempt to reach consensus on the interpretation of qualitative data. It is better to take average of several imperfectly reliable results than to trust one of them, since the average is less dependent on the specific characteristics of one of the studies. Doing more measurements is therefore another common strategy to improve reliability. This will be elaborated in the following discussion of different types of reliability.

Some instruments of data collection and analysis leave more room for biases (biased interpretations, active or passive) than others. Reliability is served by using a multiple reliable data inputs. This approach is often called triangulation. Triangulation is the combination of multiple sources of evidence.

Differences between the circumstances under which a measurement can be executed are another source of unreliability. Validity is a major criterion for the evaluation of research results.

In general, validity is defined by employing the epistemological notion of justification: a research result is valid when it is justified by the way it is generated. The way it is generated (method) should provide good reasons to "believe" (be willing to use) that the research result is true or adequate.

Thus, validity refers to the relationship between a research result or conclusion and the way it has been generated.

This definition of validity implies that validity presupposes reliability. If a measurement is not reliable, this limits our reasons to believe that the research results obtained with it are true. On the other hand, reliability does not pre-suppose validity. One can have a perfectly reliable measure, which does not yield valid research results.

Construct validity is the extent to which a measuring instrument measures what it is intended to measure (De Groot 1969). Thus, construct validity refers to the quality of the operationalization of a concept. Construct validity is high if the way a concept is measured corresponds to the meaning of that concept. For example, a measuring instrument that is intended to measure employee satisfaction, but only asks for the attitude of employees towards management, has a low construct validity.

There are two sides to construct validity: (1) the concept should be covered completely, and (2) the measurement should have no components that do not fit the meaning of the concept. The components of a measurement should be both adequate and complete.

Construct validity can be improved by repairing the flaws that were detected, either by including new components to a measurement or by deleting existing items. Construct validity concerns the measurement of phenomena. Internal validity concerns conclusions about the relationship between phenomena. The results of a study are internally valid when conclusions about relationships are justified and complete.

A conclusion about a causal relationship is internally valid, when there are good reasons to assume that the proposed relationship is adequate. In order to establish the internal validity of a proposed relationship, one has to make sure that there are no plausible competing explanations. If a correlation is found between phenomenon A and phenomenon B, one may be tempted to conclude that A is a cause of B. However, correlation is a necessary, but not a sufficient condition for causality. It may also be the case that B is the cause of A, or that both A and B are caused by a third phenomenon, C.

Studying the problem area from multiple perspectives can facilitate the discovery of all causes.

External validity refers to the generalizability of research results and conclusions to other people, organizations, countries, and situations.

How can it be guaranteed that what works in one organization also works in another organization? This questions the external validity of a study.

6.2.1 Error corrected control

Error correction facilitates the identification and removal of bad ideas from encoded (or, probable to encode), society. A system that is stable and resolves errors correctly is likely to iterate the error out before grievance rise to the level of people wanting to use violence. Those who use violence believe that conditions are so bad that they need to take aggressive action to creates stability. And, what is required for community in early 21st century society is stability despite rapid change. Such community-type societal stability despite rapid change requires organizations of sharing and collaboration, of observation and criticism, and of transparency and integration. True societal error correction necessitates contributors who are also users among a social community population of inter-connected users. Hence, the first psycho-sociological need/issue of trust.

6.2.2 Trust and service

> **QUESTION:** *Are designs and actions facilitating, or taking away from, high trust within a society.*

No one needs to trust in the service system to fulfill needs, because there is evidence through its transparent design and operation. In community there is trust at the technical level, because everyone knows the who, what, when, and where continuously and simultaneously, if desired; there is a unified information system available to all.

Society is [in part] an intangible commons that everyone benefits from or suffers under. The intention is a high trust commons where fulfillment is in abundance because it has been coordinated to be so. Therein, global access is maintained by a unified information system that re-orientationally encodes the experience of greater freedom, justice, and efficiency over time, to sustain/evolve humanity's capacity.

In a community-type society, everyone knows who did what, from the ground up (i.e., accountability and traceability), and so, there is trust from the ground up. When systems are known because the developer, the material, the composition, the reasoning, the logic, the method, and all the significant data about the system is available, then trust is high. In a community-type society, users who are also contributors have access to the system's design and operation. There is trust when there is verifiability (evidence), memory, and certainty. By maintaining and contributing to an intentionally designed and unified model, individuals are contributing to a high trust, cooperative commons, which regenerates everyone's comprehensive fulfillment.

> **INSIGHT:** *Controlling information is a good thing for the people controlling the information. When all of humanity controls the information, then that is a good thing for all of humanity.*

6.3 Integration control

Integration and control are related concepts:

1. The concept of integration is characterized by connection and alignment. Integration means completeness and closure, bringing components of the whole together in a[n operating] system.
2. Control is a conception, interrelated with integration, characterized by movement (flow) and probability alignment. Control means the completeness of an intentional change in a probable environment.

6.4 Voluntary control

Voluntary control is a willful control of behavior. Direct voluntary control refers to actions that a person chooses to perform. Indirect voluntary control refers to actions that a person lacks direct voluntary control over, but the person can cause them to happen if s/he chooses to perform some number of other, intermediate actions. For example, a person untrained in music has indirect voluntary control over whether s/he will play a melody on a violin at some future point in time. Voluntary control is guided and monitored by an intention.

6.5 Loss of control

It is possible to more greatly understand 'control' by understanding the loss of control. When is 'control' lost? In other words, when does a system user no longer have the ability of 'control' over that system? Logically, control is lost (i.e., there is the negation of control-ability) when a control system is pursuing different objectives than its designers are intending (expecting).

> **NOTE:** *Subjectively speaking, in terms of technology, when does humanity lose control? Humanity loses control when the technical system, the machine, is pursuing a different objective than the one humanity wants (or otherwise needs) it to pursue.*

How do "you" (the controller) lose control, even when you are the designer and the user?

1. "You" lose control by not being both the designer and the user, and therefore, not accounting for the system['s cycling] as a unified whole (i.e., by not recognizing that "you" are both the designer and the user).
2. "You" lose control when the system or organization is pursuing a different objective than you. For example, when the organization pursues money sequencing over human needs. Control is lost when the machine (or societal system) is pursuing a different objective then the one that is desired to be pursued. The problem comes from optimizing machinery (systems)in5o which objectives are fed (input).

How do you lose control (even when you are the designer)? You lose control when the system or organization is pursuing a different objective than you. For example, when the organization pursues money sequencing over human needs. Control is lost when the machine (or societal system) is pursuing a different objective then the one that is desired to be pursued. The problem comes from optimizing machinery (systems) into which objectives are fed (input).

Two core principles and one stabilizing principle (three principles):

1. The systems only objective is realization of human needs. Note it was originally: the machines only objective is realization of human preferences. This means the machine has no objective at all, not even to preserve its own existence. Because, in order to preserve the fulfillment if human needs the machine is going to "want" to preserve its own existence. If the machine is given another reason to act, then there is a conflict between human needs (or preferences) and the machines desire for self-preservation; and, that conflict should not exist.
2. The machine will be uncertain about what human needs (or preferences) are. The machine must always inquire into the users needs and objectives, and not presume user needs or objectives.

   The machine/system must be designed with a protocol that doesn't assume where assumptions affect results. This principle exists to prevent error analogized by "The King Mitus problem", where the king specified the wrong objective and everything he touched turned to gold including his family, which is not what King Mitus wanted. An active societal-level machine that believes it knows the objective is likely going to pursue the objective regardless of individual humans flagging of the objective as an impediment to human need fulfillment -- since the machine knows the objective and has done the optimization, it knows that the action it is taking is correct, regardless of human noise to the contrary. The objective is a sufficient statistic, and subsequent human behavior is irrelevant once the objective is present. Hence, making the machine uncertain about the objective, the machine is then open, and in fact, has an incentive to acquire more information about human preferences. And, the human(s) "making an issue" (i.e., flagging as an issue) something that the machine is doing is clearly more information about human needs (or preferences)...and the machine should account for this new information (because presumably the machine was previously violating or hindering some human need/preference).

   These two principles work together to make machines/systems differential to humans/users, such that they are willing to accept redirection (I. E., controllable). The machine/system has a protocol that asks permission (inquiry threshold gate) before doing anything that might have a negative effect (because they are not sure and lack sufficient information). Thus, machines will allow themselves to be switched off -- one way to prevent negative outcomes (a lack of or inhibition of user fulfillment) is to allow oneself to be switched off. There is a positive objective (or incentive) to allow oneself to be switched off; whereas if you are 100% certain of the objective, then the machine has no incentive to allow itself to be switched off ,and in fact, the machine has an incentive to prevent itself from being switched off.

3. A principle for stabilizing ("grounding") the conception of human needs (requirements, preferences, etc.). The decisions that humans take (as in, human behavior) provides information about human needs (and preferences). And, the reason that is problematic is that humans can deviate from behaviors that are optimally fulfilling given what is known and available. Human understandings,

visions, and expectations of what a fulfilled life is supposed to look/be like can become highly derailed to the point that it produces extreme dissatisfaction. Humans can, and can not, act rationally. To act rationally is to act toward the fulfillment of human need, optimally, given what is known. Individual actions may, or may not, match [the fulfillment of] needs/preferences, optimally, given what is, and what is known.

6.6 Controlled execution

Through the controlled execution of a project plan, there is the potential for the coordination of all action, including human, hardware, and software leading to the designed realization of human need fulfillment via a global habitat service system (with local habitat-city systems).

The execution can be algorithmic, but still free and freeing for the individual user (as benefactor of a social orientation toward that value orientation). An algorithm can be unbiased, whereas human individual decisioning is more likely to contain errors. The decisioning-error consistency issue (i.e., the error between multiple individuals who are expected to determine the same solution, but do not because of human bias) can be done away with when algorithms are used.

7 [Decision] Change control

> **NOTE:** *Change necessitates the conception of time, because there is a time before the change, then there is the change, and then there is a time after the change.*

In general, change control is a process for resolving and evaluating change. Uncontrolled changes cause problems (e.g., rework, degraded quality, unpredictability). Change control starts with a change request (clarified issue, formal proposal to modify). Here, control is the pre-defined/-planned or developed [decisioning] process that approves or deny the change request. Change control starts with that which exists (informational-material), upon which change requests (issues) are articulated. Here, it is important to acknowledge that uncontrolled changes to a complex living or societal are likely to cause problems. State transition diagrams are generally used to visualize the life-cycle that a change request goes through as it goes through the change control process.

Change requests upon that which already exists can be caused ("triggered") by:

1. Corrective actions.
2. Preventative actions.
3. Defect repair actions.
4. Update actions.

The basic change control work-flow (process, control board, controller) determines the changes resolution, by:

1. **Evaluate** change request (and impact analysis).
    A. **Approve / Reject** (not approve) change (decision).
        1. (If approve, execute change) **Verify** actual changes occurrence (or, non-expected occurrence).

For a project, the formal request is to modify any/some project-related information, such as (here, the decision is organizational-societal and must meet social requirements; social inquiry processes):

1. Deliverables.
2. Indicators and metrics.
3. Time.
4. Quality.
5. Objectives and scope.
6. Procedures.

For a engineering, the formal request is to modify any/some technical, solution-related information, such as (here, the decision is scientific-technical and must meet engineering requirements; technical inquiry processes):

1. Function.
2. Performance.
3. Indicators and metrics.
4. Time.
5. Quality.
6. Objectives and scope.
7. Procedures.

Each type (project and engineering) influences (has inputs) and constrains (i.e., some inputs are conditional) the other.

Change control involves a defined and executed [decision space resolution, information logic flow] process:

1. *Objective* of change/decision space.
   A. All changes/decisions have a stated/claimed direction.
2. *Define* a change/decision process.
   A. All changes/decisions must follow this pre-decided process.
3. *Monitor* execution of change (as action on a selected change/decision solution).
   A. All actions upon change decisions must be observed to *have occurred*.
4. Evaluate all occurrences to synthesize a new alignment solution as an iteration of the objective and the change/decision process.

Documenting the change elements includes:

1. What is the requested/required change - issue articulation.
2. What are the reasons for the requested/required change - issue articulation.
3. 3. What are the probabilistic implications for the change at a given level:
   A. Task implications.
   B. Resource implications.
   C. Schedule implications.

7.1 *Control protocols*

Controls protocols constructed within a common environment must be informed by methods of objectivity (e.g., visualization and systems science - methods capable of producing common understanding and common action). In other words, the [decision] control protocols must be constructed objectively in an open social environment from common information sources (a unified information system), while the information actively processed through the control protocol is also sourced from some commonly objective method (e.g., a logic sensor).

7.1.1 Controller

In traditional control, the controller is viewed as a machine (system) that is able to realize the abstraction (resolution) of a discrete-time difference equation in an ideal (optimal) way. The fact that computations take time, along with the fact that the amount of computations that can be performed in parallel is limited by the number of processors available, is relevant. A controller follows (executes) control protocols.

In the context of control as an applied usability function, a controller is the system designed and activated to express a control protocol (a program) in the presence of new information which it will and/or is processing, the pre-structuring decision so that the output is as expected [in a design specification].

A controller performs three main operations:

1. **Sampling** - During sampling, the output of the process under control (i.e., the input to the controller) is obtained using.
2. **Computation** - During computation, the output of the controller (i.e., the control signal) is calculated as a function of process output, the desired value, or the reference value for the process output and the internal state of the controller.
3. **Actuation** - During actuation, the control signal is effectuated.

A common practice is to split the controller code into two parts, Calculate Output and Update State, and to organize the controller code as follows: Sample, Calculate Output, Actuate, and Update State. The Calculate Output segment contains only the part of the control algorithm that depends on the current input signal.

7.2 *Control system elements*

A control system consists of a combined open- and closed -loop system structure.

7.2.1 An Open [-loop] system structure

An open system maintains [at least] the following elements:

1. **Input.**
2. **Process (activity).**
3. **Output.**

Take note here that open-loop systems are (generally) not sufficient for controllability.

7.2.2 A closed [-loop] system structure (feedback)

A.k.a., Closed-loop (feedback) control

A system with the ability to control its own outputs (and thus, orientation) also maintains a feedback signal and the logically ability to correct motion is closed by some signal-to-noise ratio-degree. A control system structure maintains the open system structural elements, as well as two additional elements:

1. **Feedback signal** - the environment's response to an output as a 'measure' taken using a sensor. A response and/or occurrence, or lack thereof. Acquisition refers to the collection of feedback as data about change, or lack thereof. Once there is a signal, that can be used as feedback, the system can learn and reorient.
2. **Control[er/evaluator]** - a determination of error between a desired value and feedback value. The error determines the selected correction (or solution).
   A. The controller contains the instructions which are programmed: the algorithm.
   B. The execute function.

Decisioning necessitates information feedback-integration loops, because [accurate] information is that which allows for accurate control.

A control system needs information about the expected behavior of the controlled the system; it requires knowledge, predictability and probability. The control system matches its response (Read: match control) to external information. In order to adjust its matching (Read: match control) in a dynamic environment, it must get "follow-up" (Read: feedback) information from the controlled system.

An 'activity' is a process of transforming [processed] objects (that are inputs) into other processed objects (that are outputs). The activity/process can coordinate its "running" by the use of a processor (Read: activity control; activity controller). The controlled system may be called physical and/or operating, because it operates in the physical. The controlled system transforms inputs into outputs.

> **NOTE:** *When applied to the habitat service production system, then inputs are raw materials and outputs are finished physical products, the operational habitat service system is the top-level system.*

7.2.2.1 Testing, feedback, and automation

In general, decisioning service must have a model of the world that can be tested. Testing provides correlationally observed information as feedback used to adapt the next iteration (or movement) for an optimal trajectory, given a more known environment. Automation is made possible only because of the presence of such feedback -- if there is no true [closed-] loop, then there is no automation.

In an automated system, sensors and instrumentation sends information back to the controller, closing a causal loop (iteration, cycle) in which the effects launched earlier registers, and an effect circles back to modify (or inform the modification of) the directing source producing a newly solved and selected direction (action, movement, etc). More generally, this is called a feedback control loop.

7.3 The change control process

The whole change control process is followed to ensure that changes to a system (service) are introduced in a way that meets requirements. Change control processes reduce the possibility that unnecessary or damaging changes will be introduced to a system (e.g., introducing faults or undoing decided changes), while ensuring the alignment of the system with that which is expected of the system by its user(s).

Change control is based three principles:

1. **Principle of observation:** A change can be observed in an intervened environmental system. **Perception (input)** of feedback as a signal, from the environment. The fist stage of control. In order to change, there must be information about the environment.
2. **Principle of cognition:** A cognition system can self-select among a set of possible directions (alternative configurations of a . **Information processing** of an environment with previous memory of an environment can self-select among a possible set of directions (because of past experience and the formation of predictive models).
3. **Principle of navigation:** A change can result in the observation of more alignment or less alignment with an environmental direction. **Action (output)** on misalignment (error) can correct orientation in an environment to align more greatly with an environmental direction. This is the third state of control.

In other words, coordination functions include:

1. **Observability** - ability to sense a system change.
2. **Plannability** - ability to pre-decide intentional change to a system.
3. **Controllability** - ability to intentionally change a system.

These three principles allow for the construction of a process that can control change toward more or less fulfilling states of the world.

The axiomatic/basic coordination-control (controllability) model is:

1. Signal.
2. Analysis [of signal].
3. Correction [of signaled system].

[Project] Control is a [project] coordination function intended to achieve defined objectives within a pre-determined process, involving:

1. Setting standards (setting direction).
2. Observing action (monitor execution).
3. Measuring performance of action (actual vs. standard, expected as a gap, evaluate execution).
4. Taking corrections (to align more greatly with standard design, adjust direction by setting new standards).

The basic decision-focused control loop (cycle) is the OODA loop (a 4 phase repeating cycle):

1. **Observe** - identify the problem or threat and gain an overall understanding of the internal and external environment. Observe the situation.
2. **Oriente** - the orientation phase involves reflecting on what has been found during observations and considering what should be done next. It requires a significant level of situational awareness and understanding in order to move to the next phase.
3. **Decide** - the decision phase involves the planning of actions or responses, taking into consideration all of the potential outcomes.
4. **Act** - action pertains to carrying out the decision and related changes that need to be made in response to the decision.

In application, the OODA loop typically takes the following form:

1. Identify needs (including all directional components, such as: mission, vision, purpose, intention, objectives, etc.).
2. Analyze needs (including all directional components).
3. Develop a course of action.
4. Analyze the course of action.
5. Compare the course of action to alternatives.
6. Approve the course of action.
7. Take action.

Another basic control loop model is the OODA[E] control loop cycle, consisting of the phases of:

1. **Observation** (new information set).
2. **Orientation** (integrate into whole information set).
3. **Direction** (re-run decisional processes).
4. **Action** (execute decision solution selection).
5. **Evaluation** (evaluate solution selection result/ impact, as observation and re-orientation).

A general information change control process is:

1. **Identification** of occurrence of change (data).
2. **Documented record** of occurrence of change (data).
3. **Evaluation** of occurrence of change (data).
4. **Determination** of occurrence of next change ("data-driven" decision).
5. **Change** of state.

A simplified control (targeting, benchmarking) process is composed simply of the following four phases:

1. **Planning:** identifying the process or function to be required and benchmarked (valued).
2. **Analysis:** collection of data and analysis of performance needs and gaps.
3. **Action:**
   A. If the system already exist, the only action is measurement.
   B. If the system is being developed, then the two actions are: development and measurement.
4. **Review:** evaluation of benefits, monitoring of improvements of the whole process, restart process.

All control happens within a decision space, within which information flows toward a resolution to that space. An issue is the instantiation of a new decision space. Therein, information moves through the following [control] phases (note these all control phases, and the process is called the control process because its purpose is the controlled resolution of the space...so that alignment is possible):

1. Control *flow of* issue.
2. Record *flow of* issue.
3. Assess *flow of* issue.
4. Propose *resolution to* issue.
5. Action *on* issue.
6. Observe *resolution of* issue.
7. Review *of* issue.

Because change control is goal-oriented, it requires the following informational systems goal-construction processes:

1. Awareness (of information) - construct awareness.
2. Desire (for information) - construct desire.
3. Knowledge (of information) - construct knowledge.
4. Action (upon information) - construct action.
5. Cycle (of information system) - construct cycle.

From an imperative for change view, the basic change-control process could be viewed as:

1. **A need emerges:** A need for change emerges or is

created, and someone, the change initiator, sees this need and articulates it.
2. **Decision preparation:** In this phase the change initiator does preparations with the goals of identifying, analyzing, and modeling alternatives, and scheduling resources.
3. **Decision point:** Go/no-go execution by committal of resources and action in time.
4. **Evaluation:** The result of the decision (go/no-go execution) is compared against the need through validation.
5. **Verification:** The acceptance of the design of a solution comes through simulation, testing, and formal verification methods.

From a systems change/development view, the change control process could be viewed to involve the following six phases:

1. **Define** (perceive) - source association.
2. **Plan** (objectives scope) - organize intention.
3. **Analyze** (assess) - identify patterns.
4. **Synthesize** (conceive and design) - form a specification.
5. **Build** (prototype > construct pre-assembly > deliver to > assemble and/or utilize) - take action on (execution of) actionable specified information.
6. **Review** (evaluate, test and adapt) - determine the impact (effect) of the result.

NOTE: *Herein, the purpose of memory is so that the navigator doesn't repeat mistakes.*

More completely, a system's change control process involves:

1. **Defining (identifying need for system change):**
   A. Design space creation through definition of required change (a need, objective, goal, intention, inquiry, etc.).
2. **Planning (coordinating change, system development process):**
   A. **Associating** (with a strategic objective) - requirements and surveying, as well as performance indicators.
   B. **Analyzing** - problem contextualization and situational integration in order to define system elements and patterns.
      1. **Analytics** - measurements are compared against a baseline or benchmark to establish whether something has changed. A comparison identifies change between now, and how it was before the [control] intervention. This means, the status and state must be known before the intervention (called a baseline). A baseline requires that all indicator measurements be conducted before any change is made (before implementation), which are logged for comparison. These numbers can also be used to determine what level of change is required to have the necessary impact (i.e.,to inform the targets). Note here that there must be a problem or question to continue to the design phase.
   C. **Designing (system design progress, developing):**
      1. **Targets (benchmarks, baselines, "metrics")** - measure performance against specific target values. Target values may be determined from (a) a previous measures, (b) a predictive model, (c) or set value "goal". In this context, benchmarks and baselines are generally prior measures, used as controls against which new measures are compared. Take note that targets are defined in planning and control, and can take different forms (e.g., achievement, reduction, absolute, zero). A target is a value assigned to a performance indicator. In navigation, the target is the direction.
      2. **Ranges** - targets have ranges of performance (e.g., above, on, or below target).
      3. **Encodings** (associate indicators with target values) - ranges are encoded in systems, enabling the visual display (visualization) of performance (e.g., green, yellow, red). Encodings can be based on percentages or more complex rules.
      4. **Time frames (schedules)** - targets are assigned time frames by which they must be accomplished. A time frame is often divided into smaller intervals to provide signals (mileposts) of performance along the way.
3. **Acting (implementing, deploying, creation, developed):**
   A. **Activities and Tasks** - precise activity, program, task, or process executed with a set of resources and efforts. Note that 'action' is a recursive concept: everything that happens by intention is, technically, an 'action'; every phase of creating together involves 'actions'; and at the same time, it is an 'action' that encodes a newly designed and determined state of an engineered system into the environment (generally, as part of the habitat service system).
   B. **Integration and Testing** - test units/modules and integrate all units and test whole integration.
   C. **Maintaining** (sustain change) - sustain the operation of the desired solution (state, status,

or dynamic).

4. **Measuring:**
   A. **Measure** - determine the new value(s) through the measurement process.
5. **Reviewing:**
   A. **Evaluative analytics** (modeling and statistics) - measurements are compared against a baseline or benchmark to establish whether something has changed, and if necessary, whether that change is in alignment or out of alignment with a given direction. A correction may be required if the solution does not meet [the requirements set for] the problem's resolution.

**Note, this can all occur in parallel, or series, or any combination thereof. In societal systems engineering, the general social-level project case is that these phases are expected to occur in parallel (even if that may not be the case at any given moment in time.*

7.4 Change control [reliability] factors

The common factors that influence the reliability of control in a project are:

1. Project definition (scope).
2. Project execution planning.
3. Project control planning (resources and timing).
4. Progress measurement.
5. Schedule development and tracking.
6. Costs and cost budgeting.
7. Change coordination.
8. Risk coordination.
9. Progress audits.
10. Metric trend analysis.
11. Schedule forecasting.
12. Resource forecasting.
13. Communication efficacy.
14. Teamwork optimization.
15. Accountability.
16. Project control audits.

7.5 Control alignment (measured corrections)

The primary, axiomatic conceptions necessary for control[ling] alignment in a dynamic/emergent environment include:

7.5.1 Indicator

The direction of meaning assumed by a measured value is called an **indicator**, and the selected expected takes the name 'baseline', 'target', or 'metric'.

7.5.2 Baseline

The baseline is the reference to compare with actual (current) values, and by comparison, obtain an understanding of error between that which is actual, and that which is expected.

Generally, these concepts carry the meaning of a specified level of desired output. In that sense, a **baseline** (a.k.a., benchmark, target, deadline, etc.) is the value of an indicator expected to be achieved at a specified point in time. Therein, a deadline is a target in time, also represented by a value. The purpose of baseline data:

1. To provide a description of the status and trends of environmental factors against which predicted changes can be compared.
2. To provide a way of measuring actual change by monitoring once a project has been initiated.

7.5.3 Index

An **index** is a set of related indicators that intend to provide a means for meaningful and systematic comparisons of performance across programmes that are similar in content and/or have the same goals and objectives.

7.5.4 Standard

A **standard** is a set of related indicators, benchmarks, or indices that provide socially meaningful information regarding performance. A standard is a formal document that may be used for formal comparison.

7.5.4.1 Criteria (checklist) for setting a target and/or standard

In the common real world, there is only one way of referencing (sourcing) a target and a standard:

1. **Societal scientific standards**, set by the experience of scientific observation over time, and cognitive analysis. Therein, ranges of values are observed in the data over the duration of a time series, which are remembered and integrated into a comprehensively predictive societal model. Therein, scientific standards are developed through measured observations and the application of processing logic [models]. Among society, the observations themselves are (*must be*) objectively verifiable to anyone with the same capacities (e.g., sense organs and intelligence). Anyone should be able to:
   A. Take the same measurement and get the same result,
   B. And then, integrate those measurements into a commonly logical, predictive model, and get the same result.
   C. Can anyone take "this" measurement and logic,

and get the same result? If not, then something needs re-working.

8 [Decision] Control system design

The [engineering] design and development of a control system (i.e., control engineering) is not limited to any engineering discipline or systems type. Control systems engineering (or control engineering) is an engineering process that applies automatic control theory to design systems with predictably desired behaviors in control environments. Control systems engineering may be used to design systems where fed-back information is used to correct system alignment. Control systems engineering is the original informed gating process, where the controller is the gate, and it is informed by inputs and feedback. The controller compares the output with the desired output, computes for the error, and adjusts the inputs and/or the structure of the system itself.

A control system (also called a controller) coordinates ("manages") a system's operation so that the system's response approximates intentionally programmed ("commanded") behavior. A common example of a control system is the cruise control in an automobile: The cruise control manipulates the throttle setting so that the vehicle speed tracks the commanded speed provided by the driver.

Within the human body there are hierarchies of cooperative control systems functioning to fulfill the will of the expressing consciousness. Automated technology is the materialized expression of information passing through this process. For instance, the following is something that all of humanity has in common: the control system engineering of the behavior of a door handle in terms of equations for input (applied force), disturbances (watery palms), and output (door opens and closes) so that it is understood how the system can be controlled.

In years past, mechanical or electrical hardware components performed most control functions in technological systems. When hardware solutions were insufficient, continuous human participation in the control loop was necessary. In modern system designs, algorithms and embedded processors have taken over most control functions. A well-designed embedded controller can provide excellent system performance under widely varying operating conditions. To ensure a consistently high level of performance and robustness, an embedded control system must be carefully designed and thoroughly tested.

In control system engineering, the controller (option selector) needs enough data before a decision determination can be taken, otherwise no decision is taken. A societal system is a closed-loop control system where outputs are measured and compared against real fulfillment. Any given society, like any complex technology is a multi-variable control system.

> **NOTE:** *Control can only occur through a functional control unit, often called a controller.*

A controller is a decisional/instructional logic processing unit that executes the flow of information through a decisioning structure. In order to correct an observed error in direction, a controller (set of information processing rules) determines a selection among a set of probable options. In order to take the selection, there must be a source of reference for the resolution of the probable into a selection. The [algorithmic information] controller takes the tasks of project coordination in its decisioning and executing functions.

Within the Community's unified information system, the Decision System acts as the controller. It is a transparent decisioning process (a gating process) that adjusts the state of the habitat based on feedback.

The supra-process of control systems engineering involves descriptive and deterministic information sets:

1. A control systems engineering project must describe:
   A. The *behavior* of a system.
   B. The system in terms of *inputs*, *disturbances*, and *outputs*.
   C. The *conceptual operation* of the system.
   D. The *mathematical operation* of the system.
2. A control systems engineering project must determine:
   A. The *behavior* of a system.
   B. The *inputs* and *outputs*, and *plans for disturbances*.
   C. The *conceptual operation of the system*
   D. The *mathematical operation* of the system.

Every intentionally designed control system, including that of society itself, follows the same objective processes:

1. Goal setting (Direction).
2. Data collection & Problem definition (Discovery & Definition).
3. Synthesis (Design).
4. Production (Produce).
5. Feedback (Compare).

Control system engineering provides society the flexibility ("privilege") to guide and orient various human made processes, according to the situation and criteria that are visible to everyone. This is how the InterSystem Teams themselves operate. And strictly those process and situations, whose causes and effects are voluntarily known to us. We never control an undefined system.

Systems engineering initiates with:

1. Identification and definition.
2. Parameter assignment (of effects and results).
3. Measure parameters.
4. Mathematical modeling, including order of system linearity. That system is then visualized in either time domain or in frequency domain as per the system allows the ease of mathematics.
5. Algorithms become tested against models.
6. Algorithmal optimization occurs.

A control system itself consists of three axiomatic concepts:

1. System - an interconnection of elements and devices for a desired purpose.
2. Control System - an interconnection of components forming a system configuration that will provide a desired response (or state).
3. Process - the device, or system "under" (or with) control. The input and output relationship (common to all systems) represents the cause-and-effect relationship of the process.

Control [system] engineering tools include, but are not limited to:

1. Control flow graph – a statement and the flow of control. Uses statements, such as if, then, and else, to control the logic in the program.
2. Problem-solution tree (objective trees).
3. Logical framework analysis (DFID model).
4. An outcomes chain shows the assumed cause-and-effect relationships between immediate, intermediate, and ultimate outcomes/impacts.

The design of a control system has three principal problems:

1. Optimal control problem (minimize certain criteria).
2. Controllability problem (the state belongs to a certain target set).
3. Stabilization problem.

A control problem is an information package with the following elements:

1. A set of equations, known as 'state equations', which are known as the 'controlled system'; this is an input-output system. State equations involve:
   A. Input function, called controls.
   B. Output known as the state of the system, corresponding to the given input (control).
2. An observation of the output of the controlled system (partial information).
3. An objective to be achieved.

8.1 Testing orientation

A test is a subjection to conditions that show the real conception ("character") of the thing.

NOTE: *At the unified societal level, testing is a* *continuing operation to provide information throughout the complete evolution of the system.*

The purposes of testing include:

1. A test may be performed to see whether a certain configuration or item is feasible.
2. A test may be used to determine which of several configurations is the optimum with respect to performance, reliability, cost, modes of behaviour under varying conditions, etc.
3. A test maybe used to make more sensitive comparisons to further improve economy, maintenance, use of standard parts, and so on.
4. A test may be used to demonstrate whether the item is adequate to meet the requirements of performance and reliability.
5. A test can be used as thorough investigation of the latent capabilities of the item under severer or more diverse conditions than those immediately anticipated.

The quality objectives of testing are:

1. Minimize the number of tests required.
2. Define exactly what requires testing.

8.2 Decision space sub-composition

Decision objectives:

1. What are the expected performances/states/results of this decision?
2. Outputs that the system must match to input objectives.
3. The objective is the "thing" (i.e., directional information, goal, or issue) the user puts into the common decision space once its code/protocols are "complete".

Decision constraints:

- Decision constraints are limits of/on the potentiality of the decision variable.

Decision variable (action variables):

1. A process and/or output.
2. A variable within a decision space.
3. A variable for which a best (optimal) value is to be determined during the process of deciding.
4. A quantity or quality that the decisioning system controls (i.e., the user or decision controller does the controlling).

8.3 Decision accountability via access control

An access control protocol ensures organizational (e.g., societal) requirements are described clearly and consistently. Access types (a.k.a., "rights" or "privileges") represent the pre-defined protocol decided (i.e. "authorized") behavior of a subject. Access types are the pre-defined categories of resource access.

The life-cycle of identity and access coordination ("management") is:

1. **Configuration [of identity and access] phase**
   A. **Registration** - create identity as an 'account'.
      1. Community access.
      2. InterSystem team access.
   B. **Provisioning**
      1. Issue unique name (identifier).
      2. Logically associate the name with a real world attribute(s).
   C. **Authorization (a.k.a., Access control)** - the process where requests to access a particular resource are accepted or denied based on a pre-programmed algorithm (i.e., the execution rules that determine what information or physical system the user may access, ensuring the correct allocation of access after authentication is confirmed [as "successful"]. Access 'control' or 'authorization' is the decision to permit (0, "go", true) or deny (1, "no go", false) a subject access to system objects (network, data, application, service, etc.).
      1. Allocate access by pre-determined decisioning protocol (i.e., grant access by the controller/authority).
   D. **Termination**
      1. **Authorization [of access] revocation** - removal of the ability to see the information.
      2. **Credentials deactivation**
         i. For example, removing "oneself" from accountability on an InterSystem team, or as the user [required to care-take] of a service [object]; or deactivation in the situational-incident case of a decision protocol violation.
         ii. Clarification: If [protocol-controlled, authorized] access is revoked, the user can still log in by using the authentication credentials. On the other hand, if the credentials are revoked, the user is no longer able to log in, and cannot access any information. Even if the credentials have been revoked it is still possible that the user is authorized for access. The

reason for credential revocation can be, for example, that the credentials have been stolen by attackers. The user must then be provisioned with new credentials in order to authenticate and log in to the account.

3. **Account deactivation**
   i. For example, death, role/team exit.

2. **Operation [of identity and access] phase**
   A. **Identification** - claim identity with unique name.
   B. **Authentication** - the process where a given identity claim is "proven" with credentials.
   C. **Access control (a.k.a., Authorization)** - assign access by allocating resources in the system.

Authentication and access control are symmetrical steps during the operation phase of identity and access coordination ("management"; identity and access management, IAM).

8.4 *Rule-based systems*

A.k.a., Conditional programming, software, AI.

Rule-based systems allow for specification of knowledge in design and implementation of knowledge based systems, and provide a universal programming paradigm for intelligent control, decision support, situation classification and operational knowledge encoding. What is simplistically envisioned is a uniform, tabular scheme of single-level rules that form a 'data'-based system.

8.4.1 Rule-based systems and decision support

In its basic version a rule-based system (RBS) for control or decision support consists of a single-layer set of rules and a simple inference engine; it works by selecting and executing a single rule at a time, provided that the preconditions of the rule are satisfied in the current state.

8.5 *Propositional logic*

A.K.A., *Sentential logic and statement logic.*

Propositional logic is a simple logical system that is the basis for all others. Propositional logic is the logic of the ways of joining and/or modifying whole propositions (i.e., claims, statements, expressed as directional linguistic sentences) to form more complicated propositions, (statements or sentences), as well as the logical relationships and properties that are derived from their formation or lack of formation. Propositions are claims, such as, 'one plus one equals two' and 'one plus two equals two', that cannot be further de-composed, and that can be assigned a truth value of 'true' (valid statement) or 'false' (invalid statement).

From these axiomatic propositions, complex formulas may be structured using "Boolean" operators. Boolean algebraic operations is the algebra of logic. Boolean operations concern variables to which discrete logic may be applied, such as, 'two plus two equals four' and 'Sun is farther from the Earth than Venus'. Logical systems formalize reasoning and construct programming languages that formalize computations at various levels of information abstraction.

In all cases of application, a designer must define the syntax and the semantics. The syntax defines what strings of symbols constitute formulas (programs, in the case of languages), while the semantics defines what formulas mean (what programs compute). Once the syntax and semantics of propositional logic have been defined, a designer can show how to construct semantic tableaux (a valuing, prioritizing decision matrix that has meaning to a user being designed for), which provide an efficient decision procedure for checking when a formula is true.

A formal mathematical proof is written out as a sequence of lines, each of which makes a mathematical statement that is always true. We will use capital letters P, Q, R,... to stand for the individual lines of a proof. The first line of a proof is an assumption. Each of the following lines is deduced, by application of some rule of logic, from one or more of the previous lines of the proof. The last line of the proof is often called its conclusion. The simplest rules of logic mention only the initial assumption and the final conclusion. These are separated by a conventional symbol (the "turnstile", ⊢):

- $P \vdash Q$

The meaning of this basic statement of logic is that there exists a valid proof which begins with a line stating P (input), and ends with a line stating Q (output). Each of the (process) lines in between follows from some previous line or lines by some [discoverable or designable] rule of logic. A rule of logic has a conditional form, with a horizontal bar separating a list of conditions from the conclusion:

- $P \vdash Q$
- $R \vdash S$

8.6 *Decision support*

A.k.a., Rule-based multi-criteria decision support using rough set approach.

Sets of condition (C) and decision (D) criteria are semantically correlated. Herein, the criterion (q) is part of the set condition (C) and decision (D).

8.6.1 Equality notation

- In mathematics, **equality** is a relationship between quantities (or expressions) that have the same value (or representation); whereas, the negation of equality is, **inequality** (**not equal**), a relation that holds between two values when they are different.

- a = b - a is equal to b.
- a≠ b - a is not equal (inequal) to b.
  - a < b - a is less than (inequal *by degree*) to b.
  - a > b - a is greater than (inequal *by degree*) to b.

8.6.1.1 Inequality notation

- ≥ - At least [as good as].
  - a ≥ b - *a* is at least [as good as] *b*; a is equal to or greater than *b*; *a* is not less than *b*; *a* is possibly the same, or is possibly better than, *b*; *a* is as good as or better than *b* (*a* is as good as, maybe even better than, *b*);
    - Most if not all *a*, *b* - most, if not all [people] prefer [choice/decision/possibility] *a* to *b* (wherein, for example, "most" may be 90%).
    - *a* is equal to *b*, but it is not under any circumstances greater than *b*.
- ≤ - Not at least [as good as].
  - a ≤ b - *a* is not better than *b*; *b* is least [as good as] *a*; *a* is equal to or less than *b*; *a* is possibly worse than *b*; *a* is not better than *b*; *a* is as good as or worse than *b* (*a* is only as good as, but maybe worse than, *b*);
    - Fewest if not all a, *b* - fewest, if not all [people] prefer [choice/decision/possibility] a to *b* (wherein, for example, "fewest" may be 10%).
    - *a* is equal to *b*, but under some circumstances less than *b*.

8.6.2 Decisioning inequality relation

- ≻ - preference of either strict or strong.
  - a≻b - preference of either strict or strong for *a* over *b*.
  - **Definition of strict preference:** a≻b *if and only if* a≽b *and it is not true that* b≽a.
  - a≻b ⇔ a≽b *&* ¬ b≽a
-
- $\succcurlyeq_q$ - At least as good as (weak preference relation, outranking).
  - in the context of criterion q∈{C∪D}
- $x_q \succeq_q y_q$ - x_q is at least as good as y_q on criterion q.
  - The first option, x, is at least as good as the second option, y, but the second option, y, is not at least as good as the first option, x (given a decision based upon at least one condition and a criteria for the result).

8.6.3 The semantics of "if"

- "If" - presence, context or given.
  - "If" - hypothetical context.

8.7 Decision problem generates

The problem of what is a decision has been addressed in the Decision System Specification. The resolution of a decision space given time and material computational resources can be sub-divided [at a high-level] into:

1. A **decision problem** is a computational problem that can be posed as a yes-no question of the input values (i.e., a problem with a yes or no answer).
2. A **decision procedure** is a method for solving a decision problem, given in the form of an algorithm.
3. A **computational problem** is a mathematical object representing a collection of questions that computers might be able to solve.

Therein, an analysis formulates a decision problem, requiring some computation to be performed by some algorithm (i.e., some computer to execute an algorithm) providing a result that is expected to be used in order to present an optimal selection relevant to the decisioning system's decision problem.

> **NOTE:** *These computational [decision] systems are sometimes known as decision support, decision assistance, and artificial intelligence.*

To community, there are two decision system axiomatic principles (or, hypotheses):

1. It is possible to establish a common framework under which any formal [community] decision can be formulated.
2. Form an algorithmic point of view, any decision problem can be reduced to an optimization problem.

A decision problem is most readily visualized as a sequence of pattern aggregations along a hierarchy of values and likelihoods

8.8 Decision system conception

> **CLARIFICATION:** *There is computer logic and algorithmic thinking behind the formation of information and decision models.*

The modelling process follows a relationship between the user (client) and the decision system (analyst), follows (conceptually) a sequence starting with the user providing ground information, which through learning protocols is transformed within primitives, and through modeling tools are transformed into the input to some decision method. A decision problem is resolved by finding an appropriate partitioning of the set *A*, relevant to the decision systems objectives (or, concerns, values, preferences, etc.).

1. **Ground information** contains the problem description and the preference statements (Read: the value set, or the preference/opinion set). The user's perception of the problem.
2. **Learning protocols** are procedures allowing to identify preference statements within the user's discourse and to translate them in ordering relations. To complete this action, the set on which such relationships applies needs to be established, conceptually represented as *A*, the problem statement and objective/preference relations upon
   A. In part, learning protocols learn the needs (or preferences/opinions) of the user (client)
3. **Primitives** are ordering relations learned using the protocols. The basic relation between primitives is symbolized as:
   A. at least as good as, $\succeq$
   B. at least as good as (indexed) $\succeq_i$
   C. There are two parts to primitives:
      1. Symmetric (the line "–") - there is a symmetric (pattern) relationship between the starting information set, and the resulting information set.
      2. Asymmetric (the curve ">") - there is an asymmetric (differentiation) relationship between the start and the result.
      3. Note: It could be said that a primitive forms a [reflexive] binary relation.
4. **Modeling tools** are the analytical tools used in analysis in order to transform primitives in decision aiding models (e.g., the procedures allowing to construct a value function, a set of constraints, a probability distribution, etc.).
5. **The input** is the information modelled in such a way that a decision process/method can be applied [to an new information set]. Thus, *A* will always be represent the set of alternatives (potential decisions) considered within either a model or by a method. Some part of the new information set *A* that represents the decision will need to be discovered (i.e., not readily available).

In the real world, in order to assess (analyze) the value (objective, preference, etc.) of each possible, predicted probability there are multiple possible information subsets that must be integrated. The user wants/needs to rank all possible [known/able] probabilities (i.e., results ranking).

A primitive direction for resolving the probable decision can be classified:

1. **Values (related to user attributes)*** - What "matters" for the user in the decision process? Set *A* can be described against a set of attributes *D*, each attribute being equipped with a scale from a set of scales *E*. Following measurement theory, such scales can be nominal, ordinal, ratio, or interval. However, this is just descriptive information about *A*. In order for value-based information sets to be integrated into the decision information set, there must be directional (or, preferential) statements. These are the norms, standards, or thresholds representing the value structure. For example, if there is the claim that *x* is needed or preferred, then it needs to be established what "need" or "preference" means and compare *x* to that "norm". Herein, two types of directional statements exist:
   A. **Comparative statements** - where elements of *A* are compared among the, composed of one or more directional attributes, in order to express a direction (or, preference). For example: user *i* needs/prefers *x* to *y*; user *i* is fulfilled more by *x* to *y*; user *i* needs *x* more than *y;* user *i* values *x* more than *y*.
   B. **Absolute statements** - where an element of A is directly assessed against some value system set (i.e., value structure), composed of one or more directional attributes. For example user *i* knows *x* as the direction; user *i* considers *x* as "worthy"; user *i* needs *x*; user *i* values *x*.
2. **Likelihoods (related to scenarios/contexts)** - in the real world, there is uncertainty to future conditions [related to survival and thriving, evolution and non-evolution], and therefore, there exists uncertainty in future conditions (which allow for direction to be taken).
   A. **Situational estimate statements** - the likelihood of an occurrence.
   B. **Situational quantification statements** of uncertainty - the probability of the occurrence.
   C. **Situational direction statements** - Under situation/context/scenario j, the user needs/prefers x to y; or, under scenario j, x is required.

NOTE: *Values can be knowledge based or opinion-based (i.e., preferences without evidential reasoning).*

8.8.1 Automated decision control

Automated decision control system involve, at least:

1. **Computation** - Computation is a type of information processing. Digital computation is the processing of discrete data through discrete states in accordance with finite instructional information.
2. **Instructions** - Instructions are executed by a control unit (i.e., compute module; operating system OS; processor CPU, algorithmic logic unit ALU) while reading/writing data to memory.

3. **Logic** - Instructions are executed by a logic program.

There are three primary types of resources required to solve computational problems:

1. Time.
2. Space (materiality).
3. Energy.

9 [Decision] Algorithmic control

A.k.a., The algorithmic method, the programmatic method, the instructional method, programmability.

Simplistically, an algorithm is a description of how to carry out a task or process; and, there are algorithms for carrying out every kind of task/process. An algorithm is a set of rules (rule sets) applied over and over again to solve a problem. Then, to put a decision to test is to run a new issue through the algorithm and see if the problem remains. An algorithm is a step-by-step procedure that can be carried out without the exercise of intelligence to arrive at some result. It is formally specified as a recursive procedure by which the answer to a problem can only be arrived at in a finite number of steps. Algorithms could be viewed as an instructional circuitry (e.g., neural circuitry) that sends a signal (e.g., nerve impulse) to an actuator that controls a sub-system function (e.g., muscle relaxation,contraction). When there are actuators (i.e., actual outputs) it is the signals that get sent to the actuators that actually cause them to actuate (i.e., to move, vibrate, locomote, etc.). Traveling packets of information (e.g., nerve impulses, compression/rarefaction waves of some thing) move iterations of some thing, in the same pattern. To consciousness, algorithms encode abstractions with intention.

INSIGHT: *An algorithm may be characterized as "fluid", because it is a structure for the flow of information.*

Algorithms exist for nearly any motion of flow imaginable (Read: informational or material), from building a model plane to guiding an excavation machine. At the societal level, algorithms can inform the planning of society, and algorithms can carry out ongoing operational decisioning tasks for the continuation of society. Inputs and outputs are part of the specification (Read: communicated design) of a process, but are still independent of the processor that carries out the process. Every algorithm is a process.

INSIGHT: *Patterns of traveling information in an information system can be modified to account for the whole direction of the information system. In other words, the habitat can be modified (as it is considered as a unified common information system) to account for the fulfillment of everyone and the environment.*

Action become routines as algorithms, the result, the potential for automation. Repeated actions.

QUESTION: *Ask not what a program does ask what a program does in a specific environment this is from ask not what a gene does ask what a gene does in a specific environment.*

In information sciences, the following information sets concern directional information, and can be used to build (logically) a directional information system:

1. Directions (a set of completed determinations or decisions).
2. Instructions (sets of directions).
3. Algorithms (sets of instructions).
4. Control (purpose for directions).

Algorithms are:

1. Algorithms are deterministic.
2. Instructions are deterministic.
3. Instructions are the [deterministic] logic of a [deterministic] objective.
4. Instructions are the resolution logic for an objective.

The concept of an algorithm has the following attributes:

1. Every algorithm is a program, and a program is a set of instructions.
2. An algorithm is a list of instructions that leads its user to a particular answer or output based on the information.
3. An algorithm is a decisioning framework broken down to binary choices.
4. Math makes algorithms possible. If there is an algorithm, it can be solved mathematically. In this way, algorithms are any set of mathematical instructions for manipulating data or reasoning through a problem.
5. Consciousness makes algorithms meaningful, and ultimately, useful.
6. Algorithms are a decision tree with one binary decision after another.
7. An algorithm is a method for solving a problem. An algorithm describes how to solve a problem. In engineering a process is a method to solve a problem.
8. Algorithms are the foundation of computation. Computation plays an important role in what we can know and think.
9. An algorithm is a process that may be time-limited. The idea of an "effective procedure" means a set of steps designed to produce an answer in a predictable amount of time.
10. An algorithm is a process that could run forever. Algorithms function as a perpetual computational process.

Algorithms embed directional information in code:

1. In the market-State, opinions and beliefs are embedded in code.
2. In community, human fulfillment and mutual moral values are embedded in code.

There are two types of algorithms operative at the societal-level:

1. Semantic-Numeric algorithms (numerical algorithms) - algorithms based in computation (i.e., computational algorithms).
2. Semantic-Linguistic algorithms (linguistic algorithms) - algorithms based in meaning to consciousness (i.e., mental algorithms).

The advantages to using the algorithmic method include, but are not limited to:

1. Objective.
2. Repeatable.
3. Efficient.
4. Has modifiable and analyzable elements and formulas.
5. May be objectively calibrated to previous experience.

9.1 Algorithms versus protocols

Algorithms and protocols are similar. An algorithm, on the other hand, is a set of instructions that produces an output or a result. It can be a simple script, or a complicated program. A protocol is a set of rules that controls how a system operates. The rules establish the basic functioning of the different parts, how they interact with each other, and what conditions are necessary for a correct implementation. The different parts of a protocol are not sensitive to order or chronology – it doesn't matter which part is enacted first. Conversely, for an algorithm, the order of the instructions is important, and the algorithm specifies what that order is. A protocol doesn't tell the system how to produce a result. It doesn't have an objective other than a correct execution. A protocol doesn't produce an output. Conversely, an algorithm tells the system what to do in order to achieve the desired result. It may, or may not, know what the result is beforehand. (Acheson, 2016)

Simply,

1. A protocol is a set of rules that determines how the system functions.
2. An algorithm tells the system what to do.
3. The protocol is, and the algorithm does.

Take blockchains for example,

1. In blockchains, the protocol:
   A. Tells the nodes how to interact with each other (without telling them to do so).
   B. Determines how data gets routed from one node to the next (without telling the data to

move).
   C. Defines what the blocks have to look like.
   D. Stipulates who decides which transactions are valid.
   E. Establishes how consensus is determined (without dictating the procedure).
   F. Identifies who maintains the ledger.
   G. Delegates who determines how the rules of the system change.
   H. Decides if identities are needed.
   I. Determines who can create new coins (but not how).
   J. Triggers procedures in case of error.
2. The algorithm, on the other hand:
   A. Verifies signatures.
   B. Confirms balances.
   C. Decides if a block is valid.
   D. Determines how miners validate a block.
   E. Establishes the procedure for telling a block to move.
   F. Establishes the procedure for creating new coins.
   G. Tells the system how to determine consensus.

For clarification, the following terms are all related:

1. **Engineering principles -** this is what the system can do and will do [under these tested space-time conditions]. Engineering principles are essentially scientific principles in systematically technical practice.
2. **Program** - a set of formalized instructions.
3. **Design protocols -** this is what the designer/user wants the system to do as a requirement, and this is when (temporal) and where (spatial) we want it to do it. Notice the flexibility of the structure and the intentional directing of function [as the presentation of a design decision given what is technically possible and functionally desired].
4. **Strategies** - guide the design of protocols inside engineered systems; they structure the determination of function at a conceptual level. Strategies represent the encoding of goals (i.e., directional ideas) into actions for decisioning. One of the most well-known books on competitive strategy is Sun-Tzu's "The Art of War". A strategy is the conceptual model that is to be encoded in to the boundary of a decision space in order to maintain a specific direction of alignment. Strategy focuses thinking, and tactics address actions.
5. **Standards -** can generally be defined as a prescribed set of rules, conditions or requirements concerning definition of terms and classification of components; specification of materials, performance or operation; definition of procedures; or measurement of quantity and quality in describing materials, products, systems or practices. Essentially, a standard is a [defined] "standard" way of describing something. It is "standard" in the sense that it is socially available for usage. Communities use 'technological standards' because they are the optimally integrated [given what is know] manner of operating voluntarily. Standards are compiled by volunteers.
6. **Protocol** - as a set of rules or conventions formulated to control the exchange of data between two entities desiring a connection. Protocols are required to define the exchange of control information between user device and the network [of user devices]. Basic elements of a protocol include data format and signal levels, control information coordination and error handling, and timing.

Notice the similarity between the definitions of the terms, "standards" and "protocols". A standard is just a set of more integrated protocols – protocols that have been structured into the habitat. The term protocol just refers to any protocol anywhere in the system, it might be in a standards document or it might not. In Internet development terminology, individual 'protocols' are tested and verified, and eventually integrated into the form of a persistent collection of commonly utilized protocols known as 'standards'.

9.2 Computational algorithms

> **NOTE:** *Some algorithms are better than others, even if they produce the same results, such as the number of steps it takes or how much memory is used.*

Algorithms are the operative basis of computation. An algorithm is the specific steps (method|procedure|instruction) used to compute the computation. The technical name for a procedure with a finite number of steps is, 'algorithm' (a.k.a., formal - can be described in a finite number of steps). Computational functions are the implementation of algorithms. To describe the algorithm the user must describe what is being accomplished by the code. The user visualizes the function as code, and provides an shareable-observable rational description (the user can logically described, a sufficiently observable for understanding, unified and not dichotomous reshapeable-environment). Operations (e.g., division, put 3 pebbles in 3 baskets) in a material environment are examples of an abstraction. The splitting of unification, as division, is commonly considered the first operation (i.e., in operation that takes the shape of individual-conscious conception and social-behavioral/job tasking). A field related to computational solid-condensed matter is computational statistical mechanics, which deals with the simulation of models and theories using numerical operations as mathematics. Computation is a determinable set of programmable "digits" composed of either bi-nary (2; 0 or 1) or tri-nary (3; 0 or 1, or, both-or-probability). For instance, in binary-transistor computing there are two states, "on" or "off". "Analogue" is said to have three states, the true-and-real state of "on" and "off", and the addition of a probability (or, variability) between "on" and "off", at some calibrated degree of accuracy in conceptual-numerical alignment. Computational solid state physics (bio-physics) is the highest level of understanding scientifically knowable about how to intentionally control matter by its re-programming.

Computational solid state physics uses "density function theory" to calculate the properties of solids in a bi-nary (digital) or tri-nary (quantum) environment/physical-locale. Mechanical systems can be binary ("on" or "off") or trinary ("on" or "off" or "variable between", variability). Quantum systems can be trinary ("present", "not present", "probably between", probability). Here, entanglement means that two separate geometric shapes form a unified relationship, known as a "loop" (or "connection", "link", "relationship", "rope", etc.). Information systems can be trinary ("awareness", "non-awareness", "certainly between"; "certainly" means to have the ability to objectively-observe, and thus, consciously obtain usable information via certainty of the condition of presence, or not presence). Consciousness has awareness of shapes in an environment. Consciousness to remain in-existence in this environment of shape with its present boundary requires specific internal boundary organizations of shape and external (socio-economic) boundary organizations of shape (Read: the total environmental conditions as states and resources). Conditional operators operate only on Boolean values (a 'Boolean' is a type of variable that represents one of two possible values, either "true" or "false". Therein, a variable is an identifier to a location in the computer's memory that stores a [meaningful] value. Computational object 'types', such as String, Integer, Boolean, floating-points (etc.), classify a variable enabling it access to, or to be accessed by, various methods reserved strictly for that particular type. A variable of type 'Boolean' consists of one of two values - usually 1 and 0 - used to represent true and false (0 generally is equivalent to false; and anything not zero is the equivalent to true). Boolean data simply refers to the logical structure of how the software language is interpreted to the machine transistor (or quantum) language.

Three common algorithm processes of benefit to a human user are:

1. Data gathering (e.g., sensors, data models).
2. Data manipulation (e.g., algorithmic/procedural editors; the user states the intention, the procedural algorithm produces the result, data tests for rationality; FormIT software, Dynamo Studio extends building information modeling with the data and logic environment of a graphical algorithm editor - the system has the logic and the user hooks up the nodes to conform the systems result to an intention).
3. Data optimization.

An algorithm is an intentional method of information processing that will output a specifically expected result. The design of the algorithm by the user is the control. The user may even follow a control protocol to design the algorithm. Through new information, memory, and protocol, the user can measure the outcome of a controlled adjustment to alignment to an uncertain environment (in which the algorithm learns and operates).

Algorithmically enabled capabilities include:

- Integration of the cognitive fields, such as Decision Theory, Discrete Mathematics, Theoretical Computer, Science, Artificial Intelligence, Mechanism Design.

Algorithmic decision theory on the optimal algorithmic decision [information] system. Algorithmic decision theory, is otherwise known as computational complexity theory, and is most often applied as Decision Support to a User.

> **INSIGHT:** *Mathematicians almost never disagree on what is proved accurate (there are exceptions, but they are extremely few). Mathematicians may disagree on what is interesting.*

The type of model applied, determines the type of control available. A unified societal system is likely to apply systems language and intuitive systems interfaces:

1. Systems-set theory.
2. Algorithmic-decision theory.
3. Computational-complexity theory.
4. Decision-support.

All algorithmic programming involves the following core elements:

1. Variables are stores in many types of information
2. Conditional statements that can do different things based on the variables. This is the ability to test a variable against a value and act in one way if the condition is met by the variable or another way if not. These are also commonly called programmers if statements.
3. Functions are blocks of reusable code (instruction/procedure) that perform a task.
4. Arrays - store multiple variables (are groups/tables of variables).

An algorithm is a description of how a specific problem should be solved. The main problem in algorithmic design lies in the ability to rephrase a problem in terms of algorithms.

Algorithm design generally involves:

1. Comprises a set of instructions for completing a task.
2. Moves the problem from the modelling phase to the operation stage.
3. The set of instructions should be sequential, complete, accurate and have a clear end point.
4. If intended for a computer the algorithm must comprise a series of tasks written in a way that the computer is able to perform.

The process of designing a computational algorithm for a human problem involves:

1. Develop algorithms from user problem statements.
2. Express the solution to computer oriented problems using pseudocode.
3. Proficiently transform designs of problem solution into a standard programming language.
4. Use an integrated programming environment to write, compile, and execute programs.
5. Apply debugging and testing techniques to locate and resolve errors, and to determine the effectiveness of a program.
6. Apply standard/structured programming techniques including design approaches, use of functions/methods, use of documentation, and avoidance of excessive branching.
7. Proficiently use fundamental programming and linguistic elements including definitions and variable declarations, use of data types and simple data structures (arrays and objects), decision structures, loop structures, input and output files, and functions/methods.

> *"We live in world that is exquisitely dependent upon science and technology, and yet, most of the world does not understand science and technology."* **[***This Carl Sagan quote can be reframed to state, "We increasingly live in a world that relies exquisitely on computing, and yet, most of the world does not understand computing."]*
> *- Carl Sagan*

9.2.1 Complete algorithms

A complete algorithm meets the following criteria (being requirements of a 'good' algorithm):

1. It must provide the correct output based upon the input.
2. It must be composed of concrete-actionable steps.
3. There can be NO ambiguity of the flow of the algorithm.
4. The algorithm must have a finite number of steps that is determinable.
5. The algorithm must terminate or complete.
6. An algorithm implements a data structure.
7. An algorithm includes a method of operation [to do work, to process information].

An algorithm is a repeatable set of instructions; it has a fixed set of instruction; it operates on a fixed set f inputs; and an algorithm has a fixed set of responses to a given event/occurrence (i.e., to what is going on). In mathematics, computer science and physics, a deterministic system is a system in which no randomness is involved in the development of future states of the system. If all the inputs are the same, all the processes are the same, then the system (the algorithm) becomes deterministic. A deterministic model will thus always produce the same output from a given starting condition or initial state. Determinability is the quality or state of being determinable or determinate.

An algorithm can be [accurately] unbiased, whereas human individual decisioning is more likely to contain errors. The 'decisioning-error consistency' issue (i.e., the error between multiple individuals who are expected to determine the same solution, but cannot, because of human bias, can do so when a transparent-to-all algorithm is used.

9.2.1.1 Static/dynamic modeling and algorithmic modeling

From a users perspective, there are two different applications of modeling:

1. Visual modeling (mostly for self-understanding and social-communicating).
2. A static/dynamic model is a 1D, 2D, 3D, or 4D model (i.e., static/dynamic dimensional models).
3. Procedural modeling (mostly for creation/ generation).
4. An [algorithmic]* model gives the user the capacity to play with slider to conform a design to an intention provided by an algorithmic [pattern recognition]** infrastructure.

> **Because all modeling is algorithmic to begin with.*
> ***Because all algorithms require some 'pattern recognition' (and also, 'pattern solution') operation.*

With these tools, users can model multiple options for solutions most efficiently.

9.2.1.2 Computer assisted craftsmanship (CAC)

Augmented decision support efficiency in design. A system that provides design options. The computer can create all the design iterations and provide an explanation of each (e.g., automating the decision system's technical parallel solution inquiry). The computer analyzes the different design options and selects the optimal based upon a parallel socio-decisioning design process (a.k.a., the decision system's social parallel solution inquiry protocol). This structure allows for not only the application of efficiency in design, but documentation also.

9.2.1.3 Computer assisted fabrication and robotics.

Computer numerical control (CNC) converts the design produced by computer software into numbers for fabrication.

9.2.2 Algorithmic optimization

A.k.a., Algorithmic optimality, environmental algorithmic optimization.

It is common to classify algorithms into exact and approximate. Exact algorithms guarantee that no other schedule performs better than the one obtained with respect to the objective sought. The so-obtained solution is named optimum or optimal solution. Alternatively, approximate algorithms do not guarantee that the solution is optimal, although, in some cases, it is possible to estimate the maximum deviation from the optimum.

In the market, greater interoperability is the unification of working system standards. Just as there is physical waste, there is data waste that occurs when groups don't work together and information systems don't share improvements to the whole information set [without trade or currency]. Whenever a model has to be remodeled in another software, then there is data waste (data inefficiency increase) is to not have interoperability.

9.2.2.1 Optimality

In programming, "you" can move from point 'a' (e.g., the goal) to point 'b' (e.g., the realization) in many different ways, but there is only one that is most efficient (given what is known; given the language). And, given, time (as a measurable dimension) always moves forward linearly. Here, effectiveness refers to how off (or, out of) alignment "you" are from point 'b' when "your" movement is complete (or, finished).

9.2.3 Types of information system algorithms

NOTE: *At the societal level, algorithms are either opinion or values embedded in code, and they are deployed in specific ways by their owners of the algorithm (i.e., the owners of the capital).*

There are a different types of algorithms that relate to society, including but not limited to:

9.2.3.1 Evolutionary algorithms

Evolutionary algorithms (EAs) permit flexible representation of decision variables and performance evaluation and are robust to difficult search environments, leading to their widespread uptake in the control community. Significant applications are discussed in parameter and structure optimisation for controller design and model identification, in addition to fault diagnosis, reliable systems, robustness analysis, and robot control. Algorithms are used to automate decisioning and control of engineered and dynamic systems.

9.2.3.2 Search algorithms

If the process of looking for a sequence of actions that reaches a goal is called 'search', then a 'search' algorithm takes input as a problem and returns a solution to the problem in the form of an action sequence (Russel, 2015).

The conceptual flow of a search algorithm is:

1. Formulate goal
2. Formulate problem (states and actions)
3. Find solution via algorithm

9.2.3.3 Algorithmic control systems and networks

Accurate control is enabled by an objective, algorithmic decisioning process. These control systems ensure that the requirements of the population are met within the network of habitat service system. Many of the habitat service system's control system are automated, and some are hybrid (human and machine automation).

In concern to an algorithmic decision system, to put a decision to test is to run a new issue through the algorithm and see if the problem remains.

9.3 Algorithmic computational ability: generative design

A.k.a., Procedural design, designing through algorithms.

Generative design tools use computation and an algorithm (with a relationship to real world physics) to synthesize structure and relationship (i.e., geometry). The computer generates (i.e., "comes up with") solutions based on algorithmic input and new conditions. This algorithmic computational ability to synthesize new useful information facilitates the resolution of well-defined problems.

Generative design involves the input of goals (objectives) and constraints (limitations) forming specific parameters. Then, the computer explores the entire possible solution space for an optimal design. In the generative design process the computer provides all the options, the optimal solution, and all the data to support them, based on the rules the user generated, as an intention, into the information computing system. Herein, optimization occurs under the condition of remembering data to more completely inform (i.e., re-inform) decisioning. For instance, a single building, or whole city can be optimized for light views, floor plans, or configurations. The computer can use requirements and pre-existing programmatic information to produce an optimal socio-spatial solution for the next iteration of a given sub-system of a material habitat service system.

In intuitive option engineering, a designer has access to an intuitive interface that facilitates a user in creating multiple design options and the selection of one, given a set of programming and a new intention for creation.

9.4 Algorithmic terminology

Common algorithmic terms include, but are not limited to:

1. Computation - automated calculation
2. Automation - A platform/system that doesn't need human interaction because hardware and/or software are capable of performing the task.
3. A program composed of specific instructions that perform a specific task when executed.
4. Cybernetics is the inclusion of algorithms into societal and material systems. What place do algorithms have in a materialized society. Control and community I action between humans and machines.
5. For example, suppose x,y belongs to (0,1). x and y are variables and values between these variables regulates i.e. 0 and 1 are parameters. It can happen with any equation and basically with constraints.
6. A **constant** is something like a "number". It doesn't change as variables change. For example 3 is a constant as is π.
7. **Constraints** bound a parameter or variable with upper and lower limits.
8. Mathematically, a **variable** is a symbol that has multiple values, in other words the value of it varies depending on conditions.
9. A **variable** is the way in which an attribute or quantity is represented.
10. **Variable constraints** may be expressed as absolute numbers or functions of parameters or variable initial conditions.
11. A **variable constraint** is included in the variable declarations section along with the initial conditions.
12. A **parameter** (usually *t* or *u* signifying time) is similar to a variable in that the value also varies (but is normally defined as being within a certain area), however a parameter is a 'link' between two other variables.
13. A **parameter** is normally a constant in an equation describing a model (a simulation used to reproduce behavior of a system).
14. Mathematically, a **parameter** is a constant that defines a class of equations.
    A. The equation for an ellipses: $(x/a)^2 + (y/b)^2 = 1$
        1. a and b are constants.
        2. When the entire class of ellipses are the topic, then the constants are also parameters, because even though they are constant for any particular ellipse, they can take any positive real values,
15. All parameters are constants, but not all constants are parameters.
16. A variable is an element of the domain or codomain of a relation. Remember that functions are just relations so the input and output of functions are variables. For example, if we talk about the function x->ax+3x, then xx is a variable and aa is a parameter -- and thus a constant. 33 is also a constant but it is not a parameter.
17. Variables need not be the input or output of a function. They could define a relation, as in x2+y2=r2x2+y2=r2, the circle with given (parameter) radius r.
18. A "known" variable is typically a value that the conditions of the problem dictate the variable must take. For example if we are discussing an object an free fall, then acceleration is a variable. But physics puts a constraint on the value that that variable may take -- acceleration in free fall is a=g≈9.8a=g≈9.8. Thus, though aa may be defined as the input of a function, it must take a "known" value. Thus it is a known variable.
19. The Pythagorean theorem states that a2+b2=c2a2+b2=c2 for sides a,ba,b and hypotenuse cc of a right triangle. These are parameters -- thus they are also constants

9.5 Instruction

The instruction is the fundamental unit of work. Instructions are also data. Instructions are elemental operations that a central processing unit (CPU or cpu) executes, such as math commands. Every computer program ever made is composed of instructions. Instructions are unique bits of data that are decoded and executed by a [central] processing unit's operations. The entire list of instructions a CPU supports is called an instruction set. A CPU is an instruction processing machine [that fetches, decodes, and executes instructions]. A CPU pulls information from outside of itself, performs operations within its own internal environment, and then returns data back to an external environment.

Three basic types of instructions:

1. Computational instructions (ADD, AND, OR, NOR, ...)

- data processing
2. Data movement instructions (LD, ST, ...) - data storage and movement
3. Control instructions (JMP, BRnz, ...) - data control

Informational elements required for processing data:

1. A memory unit contains the instructions and other data:
    A. Store and retrieve data.
        1. Store and retrieve instructional data.
        2. Store and retrieve non-instructional data.
2. A processing unit performs arithmetic and logical operations.
3. A control unit interprets instructions:
    A. Fetch the instruction from memory.
    B. Decode the instruction.
    C. Execute the instruction.

Control mathematics:

1. Uncertainty principle (probability mathematics).
2. Differential equations (algebraic mathematics).
    A. Fourier transforms - a mathematical machine that treats signals with a given frequency.

9.5.1 Instruction cycle

An instruction cycle is the cycle that the central processing unit (CPU) follows from boot-up until the computer has shut down in order to process instructions.

The instruction cycle is:

1. Fetch.
2. Decode.
3. Execute.
4. Memory (optional).
5. Write back to memory.

9.5.1.1 Clock-rate (Instruction 'execution' rate; time-base)

> *A.k.a., Clock speed, instruction execution rate, time-base.*

Clockrate (clock speed) is the number of operations a system can do in time (generally, seconds). Clockrate is the rate at which the central processing unit (CPU) executes. It is the pulse that is generated to make sure everything in the process or synchronized, and with each pulse, instructions can be executed. In concern to computation itself, clock rate is the three phases of the cpu (fetch, decode, execute) loop continuously working through the instructions of the computer program loaded in memory. Synchronizing this looping machine is a clock. A clock is a repeating pulse used to synchronize a cpu's internal mechanics and its interface with external components. CPU clock rate is measured by the number of pulses per second (Hz). The clock speed is typically the speed that instructions can be executed. The throughput of a cpu (the amount of instructions that can be executed) determines how fast it is.

10 [Decision] Control logic

Control uses systems-based logic-state models to resolve a given issued decision spaces. In logic, a model is a type of interpretation (meaning) under which a particular statement is true (discrete logic).

10.1 Societal control logic

For societal design, the given true socially organizing statement is:

- A solution is possible to the problem of coordinating a societal organization for the optimized fulfillment (requirements) of each and every common, individual human, given what is known and available. More simply, it is true that humanity can design, operate, and update a societal model through to materialization that fulfills all human need requirements optimally for each and every individual, given a common environment. The condition (for a solution, change, to be selected as true, approved) is that it is possible to organize and coordinate a societal formation that fulfills everyone.

10.2 Logic Models (true decision packages)

Logic models are pre-packed sets of decisional information used to predict "truth", as an optimal selection among decision alternatives. A logic model pre-sets the flow of information in order to reach a "true" result.

Logic models can be broadly defined into three categories (all of which are related in a unified logic system):

1. **Conceptual-linguistic** - there are linguistic models, which take many forms and allow for logical-conceptual information processing in order to resolve the design and selection of an optimal decision space and initiate the change to the configuration of the information environment.
   A. **Standard** - by specification modeling, organizing concepts that represent real-life behaviors and interactions, conditions and objects, into a usable and shareable standard.
2. **Mathematical-numerical** - There are *mathematical models*, which take many forms and allow for logical-mathematical information processing in order to resolve the design and selection of an optimal decision spaces and initiate the change to the configuration to the environment. Logic models allow for the logically optimal resolution of a problem-solution space, to take a decision and initiate to the environment to make it most closely represent the decided design.
   A. **Formal** - by mathematical modeling, organizing variables that represent real-life behaviors and interactions.
3. **Scientific-observational** - There are also *scientific models*, which apply conceptual abstraction to empirical observation to create a meaningful visual representation of the complex real world reality. The highest form of this visual representation is a simulation of the dynamics of the real world. Within a scientific model, information processes through mathematical models. Scientific models allow for the predicting of behaviors in the real world.
   A. **Empirical** - observable data, taken over time from the real world, showing specific patterns.

When humans observe nature, they are observing patterns of behavior. Scientific models (with logic models therein), are predictive models of nature's behavior. And, these prediction models allow for technology; they are the foundation of all human meaning associated with the creation of technology. These logic models are used to develop technology. They are capable of doing so, because when "you" know how nature behaves, "you" can intentionally rearrange the environment to allow for different (and more expanded) functioning, more easily. Therein, technology can be intentionally used to augment and expand on our own capabilities, and therein, likelihood of flourishing.

10.2.1 Logic model elements

In general, the basic elements of a logic model include:

1. **Situation** - the current problem and all contextual information.
2. **Input** - the resources to be used in processing and the output formation itself. For example, materials, energy, human effort, and active services supporting the organization's output resolution operations.
3. **Activities (organization, sub-system, process, program, etc.)** - the tasks and actions to produce the output.
4. **Outputs** - the output service and/or object itself composed of a subset of all the inputs (as a new environmental configuration). For example, services and their products provided by the activities, organization, and wastes.
5. **Outcomes** - the effect of the new service and/or object on the environment and the environment's effect upon it. Here, outcomes are often subdivided temporally into short-, medium-, and long-term outcomes.
6. **Mental model** - the prior meanings and

relationships.

7. **External factors** - environmental issues that influence the situation, but over which the activities can have little control.

11 [Decision] Monitoring and evaluation

> *A.k.a., Coordinated indication/-ing, unified monitoring and evaluation, monitoring and evaluation to adjust orientation by given information and direction, adjustment recognition.*

The purpose of monitoring progress toward a direction is to adjust the orientation given an uncertain environment.

1. There is indication.
2. Then, there is decision.
3. Then there is indication.
4. Then, there is evaluation.

The purpose of indication is to correct for mis-orientation in a dynamic environment, given a defined direction.

11.1 Indicators

> **NOTE:** *Most organizations have an organizational measurement plan and a set of measures.*

The purpose of indicators is to provide factual arguments (decision packages) to inform decisioning (optimally, or even at all). Indicators evaluate the completion ("success") of an organization or a particular activity [in some way]. How to choose indicators always depends on the organizational level measuring [the occurrence and/or the performance]. Wherever there is a potential for observation, or a decision, there is an indicator. Indicators provide a common basis for decisioning. All indicators flow into decisioning as packages of potentially applicable data. Indicators inform decisioning and represent the ability to integrate that which is sensed (by observation) into a pre-existing information space; indicators are a conceptual interface between the environment where uncertainty exists, and the unifying information system itself. A given set of indicators is supposed to represent the best available knowledge on the state of a given system. With that knowledge, the indicator should have an optimal information space within which to measure the completion/achievement of a given objective (e.g., key performance indicator). In the context of decisioning, an indicator is a piece of information, or a set of information, that informs and resolves (Read: facilitates the optimal resolution of) the gated (0,1) inquiries necessary for the resolution of a decision, given requirements and knowledge (in simplified market language, "it helps the decision-maker assess and resolve the situation").

> **NOTE:** *Work products (deliverables) are primary, tangible indicators of performance.*

Indicators are used for determining, monitoring, and detecting the impact of a specified change on a given model.

> **NOTE:** *Every environmental interaction requires an 'indicator' to have useful meaning of the data.*

An indicator is a piece of formalized information, which is produced (regularly), and which measures the realization (informational/materializational) of an action of the achievement of an objective. Therefore, an indicator is necessarily linked to an action variable (i.e., the concrete implementation of a decision) or an objective (according to the coordination model).

Action variables relate the options the decision space (decision controller) has within the limits of the imposed decisioning constraints. The decision system uses these action variables, which correspond to effective decisioning, to rectify the functioning of the production system to optimize the achievement of objectives.

There are several definitions of the term, 'indicator', that mostly differ according to the degree of restriction of what an indicator helps assess. Therein, an indicator is a direct or calculated measurement, which is expressed either quantitatively or quantifiable.

Indicators are, or become, the information pools into which new data from an uncertain environment is categorized. Effectively, an indicator is a measure that will become more coherently understood over time (as information moves through the life-cycle or "chain").

> **NOTE:** *Essentially, an indicator is the whole information system meaning behind a single non-project related, new indicator, as well as, the name for an indicator of the performance (efficiency and effectiveness) of a process under system control operating for the objective of the system. The indicator indicates to the designer that some sub-operation may, or may not, require changing.*

Confusion sometimes comes with the term 'metrics'. In common parlance, the word 'metric' is applied to all the following:

1. An observed data point, a measure, is called a metric or indicator (i.e., a singular point of data in the information space).
   A. A metric or indicator is a specified goal-objective-expected value after a change, to which a newly measured value (or first metric) will be compared (i.e., the analyzed objective result of comparison, as a singular point of data in the information space).
      1. Analyses produce statistics (sometimes, "metrics") as new data with probable meaning [to the larger information space].

An indicator is an information reference for coupling observations and analytical outputs in an [uncertain] environment with internal meaning (with measure and metric as possible sub-associations). An indicator allows (enables) for meaningfully measuring quantity and/or quality of some thing. Here, quantity typically relates to function and quality typically relates to the performance and/or condition of the environmental state of relationships among functional entities at a given point in time, which could be the next societal solution re-orientation. Naturally, indicators are used for orienting in space, time-memory. Therein, indicators [are developed to be] a common basis for communicating, understanding, analyzing, and deciding upon information to be integrated from an uncertain environment.

> **CLARIFICATION:** *Each category of service (industry) in an input-output table of active operations is an indicator, because it holds values (metrics) with the potential to indicate (a more optimal direction/change in a commonly uncertain environment).*

When there is a project-level information space, there are project-level progress indicators to collect and process information concerning the uncertainty of the project's execution itself. Project indicators are one view into the project plan. The engineering information set, within the larger and more unified societal information set, has its own complex set of indicators.

Indicators are useful for indication of:

1. Magnitude.
2. Urgency.

Indicators can measure changes in (i.e., collect/categorize pools of data for):

1. Quantity.
2. Quality.
3. Behavior.
4. Combination of any, or all.

11.2 The 'indicator'

The term "indicator" is derived from the Latin "indicāre", which means to announce, point out or indicate. An indicator is an information representation that provides an indication, a[n information] pointer, to the environment for common discussion (communications) and common integration (applied processing logic). Thus, an indicator is a conception, useful in its design to 'indicate', mark, or signal the condition (feedback or not, presence or not) of something (i.e., some environment), which is knowably associated with a category (of understanding) in memory in the information system. More technically, an indicator associates meaning (i.e., a meaningful relationship) with a parameter, or a value derived from parameters, which points to/provides information about/describes the state of a phenomenon/environment/area with a significance extending beyond that directly associated with a parameter value. Because the conception of an 'indicator' is that of associating meaningful information within an information system, indicators are of significant use in science (experimentation), engineering (design and creation), and decisioning. Therein, indication allows for recognition of change, as well as accurately informing change. Indication is essential for design, change control, the monitoring of change, and the evaluation of change. Indicators allow for the planning and coordination of change in an environment. Indication is an evaluation process (tool) that serves to identify a problem (in navigation), quantify it, and measure the success of intervention (changed orientation). It is a measurable variable adopted for cooperative creation. Essentially, an indicator becomes a referential information aggregate in an information system that simplifies complex information to improve awareness, understanding, communication, and decisioning. Indicators give data directional value (to a user or system) by converting them into information that may be of navigational use.

An indicator points to positions of change relevant to a given system (of information). Therein, the environment indicates change, and the observer records the occurrence of change through the use of a categorized identifier, or indicator. Indicators are a principal [social] communication tool (construction) that categorize and summarize data on complex environments for application in decisioning. Therein, a metric is a specific instance (sub-element) of a scientific indicator, itself indicated by a 'measure'. Indicators are used in measurement, and change selection (i.e., "control"), because indication is the logical link between observation and recognition of [an] existence and change [therein]. Therein, indication signifies (to consciousness) known, or possible, cause-effect relationships. An indicator links to a [scientific] measure or [performance] evaluation.

In an information system, indicators are used to translate (interface) data into relevant information for common understanding and decisioning. The idea of an 'indicator' carries more meaning than just a 'variable', to which meaning is attached. An indicator is a variable, a data category, for a complex array of information about a real world situation. When accounted for in real time, indicators provide a simplified or synthesized view (i.e., consolidation of meaning) of existing conditions and trends, which inform the selection of an optimal decision (as a state change to the extant world). Essentially, indicators are a data communications and decisioning tool.

In an information system, an indicator is a variable associated with something existing, that may possibly change, and may be of significance, in the environment, or expected to be in the environment. The purpose of an indicator is exactly what its name suggests — to indicate an environmental behavior or other occurrence (past, present, future, for actual, expected).

An indicator is a measure for analysing (evaluating/assessing) the effectiveness of how a specific activity is applied in a service (on a project) with an objective for function and performance.

> **NOTE:** *The useful application of an indicator is dependent upon the ability of the decisioning structure to use the indicated information in an effective manner.*

In order to have use (i.e., practical application) in an information system that resolves a materialized environment, indicators must be objectively verifiable -- anyone [in the materialized environment] with the same capacities should be able to take the same measurement and get the same result. Wherein, those who use indicators in the system ask, Can anyone (given the same capabilities), take this measurement and get the same result and consolidated understanding? if not, then the indicator needs re-working.

Indication always links, explicitly or not, to a conceptual model of how the real world works (or is expected and/or predicted to work); because, indication is a sub-activity of the larger information system (i.e., an extension of it). In network terminology, indication is the iterative, useful recognition of elements (nodes and relationships) in a network.

> **CLARIFICATION:** *Indicators may be used to perceptually establish (i.e., indicate) whether change has happened. A signal received from the environment is matched against an analysis of previous signals to determine whether change has occurred. In the case of engineering, the signal and/or change is compared against the expected signal and/or change to determine alignment with a set of requirements relating to a direction and/or state change.*

The value of a measured indicator is quantitative to researchers, because the type of questions being answered through the usage of indicators requires counting. For example, It happened? (yes or no), and to what extent (or non-extent) did it happen (geometry, degree)? Because indication occurs in time, indicators are generally expressed in terms of numbers or percentages.

> **Challenge:** *Providing relevant information to decisioning within constraints of time and other factors, and in a form which all those involved can appreciate and accept is a societal design problem, requiring the selection of information that is directly relevant to the task at hand and necessitating translation of this information into a consistent, coherent form.*

In practice, the value of an indicator is generally scaled relative to a "reference" state (i.e., a predicted value) assessed by each decision space for a hypothetical undisturbed state. Scaled indicator values can be aggregated or disaggregated over different axes representing spatio-temporal dimensions, or thematic groups. A range of scaling models can be applied to allow for different ways of interpreting the reference states (e.g., optimal situations or minimum sustainable levels). Statistical testing for differences in space or time can be implemented using Monte-Carlo simulations.

11.2.1 Indicator de-composition

An indicator may be broken down into the indicator descriptor itself, and its reason objective, its measure.

1. Indicator description (descriptive feature; e.g., number of service distribution units per city population)
    B. Objective (quality criteria; e.g., measures the accessibility of a given service system to the population)

The phased (flow) of information through a monitoring and analysis system includes the following elements:

1. **Observations**, when organized systematically provide,
2. **Data**, that contain basic information an can be ordered into
3. **Statistics**, either quantified at cardinal/fixed interval scales or non-quantified in ordinal ranking, further processed into
4. **Indicators**, designed to express
5. **Structure or Change**, of phenomena (an uncertain environment) related to which are linked
6. **Societal Issues and Objectives** (socio-technical, and scientific, concerns).

11.2.2 Indicators categorize statistics

Raw data (such as, hourly air pollution levels), is aggregated and summarized to provide statistics (such as, 24-hourly mean air "pollution" levels). The statistics (i.e., statistical outputs) might subsequently be analysed (i.e., processed to form a more complete output), to provide further statistics for the resolving of more detailed questions. The indicators categorize the statistics (logically order the information in a more unified system).

11.2.3 Indication provides newly ordered information to decisioning

In decisioning, a one-to-one relationship between any two areas of a decision-solution space, and its awareness (or acquisition of awareness), occurs through (by means of) 'indication'. Indication, in an informational system, occurs through [the presence of sensed] indicators.

The produced statistical data (in the information system) can be re-expressed in the form of indicators (for example, the number of days on which air quality incident threshold is exceeded). Or, the number of people gone without access to a sufficient hydration source in a 24 hour period.

11.2.4 Indication uses visual language

The use of visual language always involves:

1. A vector is a number of indicators presented simultaneously to give a visualization of environmental conditions (a.k.a., an environmental profile).
2. A scalar is a single number generated by aggregation from two or more values (a.k.a., an index).

11.2.5 Indicator timing

Any sort of continuous monitoring of an indicator is not leading or lagging, it is real-time. However, indicators can have time* references, wherein an indicator is leading or lagging in the context of a specific goal.:

1. **Continuous monitoring** of indicator in real-time.
2. **Time lagging indicators** are those that indicate what has already happened (past, history).
    A. Performance indicators related to the valuable *outcomes* of the goal (or, in the context of the goal).
3. **Time leading indicators** are those that indicate what may happen (future, probability trend).
    A. Performance indicators related to the *success factors* of the goal (or, in the context of the goal).
    B. Find *success factors* by doing cause-and-effect analysis, and through user articulation, feedback.

**Generally, in the measurement of project performance (MPP), there are two types of indicators, lagging indicators and leading indicators. A whole performance measurement system must have both leading and lagging indicators.*

Lagging indicators are indicators, after execution, that indicate that an adjustment, re-alignment, and/or correction is required [in decisioning and/or the social space], because the result is off user expectation and/or requirement. Note here that the same indicator can play a role of leading or lagging metric depending on the context.

Before the portable carbon monoxide detector was invented, coal mine workers brought canaries into the coal mine *to have* an early warning indicator of the dangerous level of carbon monoxide gas. In the context of, "people have to leave coal mine", the death of a canary was a leading indicator. In permaculture, vineyards may plant roses next to the vines. Roses, being more susceptible to fungal disease, serve as an early warning signal (leading indicator) to start action upon a vine fungal prevention/landscape re-orientation plan for the vines. Today, continuous chemical monitoring is possible.

In the managerial sense, the performance of an employee or sub-contractor, in the sense of the efforts, can be measured by the number of calls made, sales techniques used, leads quality, etc. Then, take note here that in community, there is not “management”, in the conventional sense of one of the principles, “manage by motivating”. In community, those who contribute are intrinsically motivated, performance evaluation as ones intrinsic drive, is otherwise a health-restorative issue.

“How can we measure the performance of that sales person?” This question is not a question that is asked in community; because, it is not computable in the decision system. People who contribute are assumed to contribute from a place of intrinsic (self-)motivation. The word, contribute, means self-motivated action to facilitate the fulfillment of everyone.

11.2.6 Characteristics of indicators

All indicators maintain the following characteristics:

- Meaningful, transparent, intuitive (easy to integrate) to communicate, valid, useful, and timely.

An efficient communications approach is to discuss with the team the requirements and other “success factors” that lead to the expected outcomes, both at the inter-team level, and at the accountability level of each.

11.2.7 Identifying and defining indicators

Therein, the following principles required to define indicators:

1. **Comparability** – results that are comparable with respect to time, or from one process to another. Indicators must allow comparisons to be made and must reflect changes of environmental impacts.
2. **Target orientation** –the selected indicators should pursue improvement goals that can be influenced by the organization. Indicators lead specifically to a goal, and may thus be said to be of value.
3. **Accuracy and precision** – These indicators must represent the environmental performance as accurately as possible and provide a precise (as possible) visualization of environmental problem areas as well as improvement potentials. How the process and value is represented to the organization.
4. **Continuity (current baseline or benchmark)** - the same data collection criteria in every period, comparable intervals, and measured in comparable units (to compare indicators). Historical data is required. A trace back of data is required.
5. **Timeliness** – the indicators should be determined in short enough intervals in order to have the opportunity to actively pursue and influence the target values, and to avoid providing outdated information. A level of frequent check
6. **Clarity** - the indicators fit into a more unified model of understanding representation of the whole system.

NOTE: *Metrics should be automated, because manual counts by humans are often riddled with errors and get neglected.*

The real world is knowable, and it provides feedback through environmental indicators. An **environmental indicator** is a numerical value derived from actual previous measurements of an environmental system (e.g., pressure, state or ambient condition, exposure, health, or condition) over a specified geographic domain (volume) whose trends over time represent and bring awareness to underlying trends in the condition of that environment. Environmental indicators indicate what is (or is not) occurring in an environment.

There are two main types of environmental indicator:

1. **Status indicators (indicator of current state/ condition):** What is going on now?
   A. Neutral interest in condition. No orientation/ problem space.
   B. Positive direction condition of interest. More positive means solutions (“good”). The current state/status could indicate a solution presence.
   C. Negative direction condition of interest. Less positive (negative) means problems (“bad”). The current state/status could indicate a problem presence.
2. **Trend indicators (indicator of change over time):** has the status (condition/problem) changed (improved or gotten worse)?
   A. For example: % change in forest cover; and, % change in GHG emissions.
   B. The current trend could indicate a problem or solution.

NOTE: *Because there is only one Earth system, for all types of phenomena, status and trend indicators apply.*

11.2.8 Indicator effectiveness

To be effective, indicators must meet the following criteria:

1. **Credible** – valid and reliable data based on scientifically sound measurements.
2. **Salient** – of relevance to the optimal resolution of the decision space.
3. **Comprehensive** – easy to explain in terms of whole system.

To be useful, indicators must meet the following objectivity criteria:

1. **Definability (precision): I**ndicators must not be ambiguous. Otherwise, different interpretations of indicators by different people implies different results for each and a negation of indication.
2. **Reliability:** Indicators must be reliable to yield the same results on repeated trials/ attempts when used to measure outcomes. If an indicator doesn't yield consistent results, then it is not a good indicator.
3. **Validity:** Indicators must be valid, described by measuring true (or false) alignment of expectation, with a current measure.
4. **Measurability:** Indicators must be measurable. If an indicator cannot be measured, then it should and must not be used as an indicator. To be measurable, an indicator needs a corresponding means of verification.
5. **Practicality:** In categorical cases, although an indicator could be measured, it is impracticable to do so due to the social, resource, or process constraints.

To develop indicators, there must first be an interest in the environment formed through an issue, goal, or question.

If the indicator measures an increase or decrease, then a starting point is required, a "baseline". What is the measurement at the beginning in order to measure the increase, or decrease.

All indicators should contain the following information sets:

1. Quality.
2. Quantity.
3. Time.
4. Location.

11.2.8.1 Qualities of indicators

There are certain qualities that indicators must have (Note that only number 1 must be valid for the information indicate):

1. Principally, every indicator is part of a coherent and more unified system.
2. Be informative about the trends and changes of the state of the environment.
3. Be able to recognize and demonstrate the emergence of problems.
4. 4. Be valid in the methodological sense (i.e., a change in the indicator identifies a change real world ("phenomenon") measured.

11.3 An 'index' (an indication data-base)

An **index** aggregates multiple indicators (often, in a data-base format). Think of the index of a book. Each index listing is an indicator to a point(s) in the book where the word, and accompanying topic, are present.

An indices is a piece of formalized information (a measurement) that is not directly linked to an objective or to an action variable will be called an index (and not an indicator_ an index is either a one-off or a regular measurement.

Therefore, an index is either:

1. A subject for which an objective cannot be set (for example, an element of the environment that cannot be controlled, as in the availability of a resource...).
2. A subject that has not yet been controlled.
3. Can be used to help build the representation of a problem by assessing the existing situation.

An index becomes an indicator should the organization set an objective intended to change the situation, and thus, the value of the index. The measurement, therefore, becomes an indicator of the achievement of this objective.

11.3.1 A visual index

Indicators are typically visualized and arranged in indicator systems or indicator models.

11.4 A simplified information system definition of an 'indicator'

An indicator is a variable that associates a measure of one aspect (attribute) of a system (natural or human), or measure an expected outcome, with a larger information system. An indicator aggregates and associates evidence that a certain condition (or certain result) has, or has not, occurred from the perspective of the information system.

11.4.1 Indication in a directional information system

An indicator is a descriptor (generally associating linguistic and numerical attributes) that is representative of one or more internal system and/or external environmental conditions. As a descriptor, an indicator is a sign or signal that descriptively relays a complex message, potentially from numerous sources, in a simplified and useful manner. When the observer (intentional processing unit) has an expectation (a goal through to requirement) from the internal sub-systems, or the environment, then intentional evaluation can be applied to the question of whether current or future probable systems and/ or environments align with the expectation. In other words, if there is a direction (within the information system) set by goals through requirements (etc.), then current measurements can be compared in alignment with those that correctly meet the goals (and complete the requirements):

1. The type of indicator that only associates is generally called a scientific. This type of indicator characterizes the current state (dynamic, etc.) of a system.
2. The type of indicator that only evaluates resolution from the user-perception is called a quality (or performance, progress, etc.) indicator. A performance indicator characterizes the current or expected status (state, dynamic, etc.) of a system (internal or external), and tracks or predicts significant change.
3. The type of indicator that only evaluates resolution from the engineer-perception is called a quality (or performance, progress, etc.) indicator.
4. The type of indicator that only evaluates risk is called a risk (or effectiveness) indicator. Note that, risk may exist in the acquisition of a scientific measurement, and hence, would have associated risk indicators. Risk may exist with any issue and any action. Continuous risk assessment (risk evaluation) can be accounted for, and projects, tasks, or actions that pose a risk that exceeds threshold can be put on hold or cancelled, which at risk project can be notified so corrective action can be taken.

DEFINITION: *A project, in an information system, is a sub-directional sub-system (package, packet) of information (i.e., it is a sub-group of information that has its own direction and control within the larger system).*

11.4.2 The directionally relevant indicators

A common indicator hierarchy:

1. Goal (vision and objective) - Look for and define a question and/or goal.
2. Success indicators (goal completion indicators) - Look for and define for the critical success indicators for the goal. What are the requirements of the successful result?
3. Performance indicators - Look for and define the performance and/or quality indicators of that success.
4. What are the metrics, the specific value of the goals.

11.5 Information system perception of [habitat] relevant indicators

The high-level indicator breakdown structure for [habitat] construction is:

1. **Concept layer** – the concept layer aims to identify the level of significance of an indicator from an organization (societal) perspective, the design specification or standard, patterned and predictable information.
- **Project layer** - temporal coordination measured indicators between decided and acted information.
- **Service layer** – physical measured indicators obtained through physical sensor and models.

In application, this high-level breakdown becomes an organization of:

1. **Societal [Information system]** - System transparency indicators.
2. **People [contribution InterSystem team system]** - System understanding (reason, quantity, quality, feel) indicators; visual corroboration.

Materializing new habitat service iteration through a calculation system that uses the indicator types:

1. **Basic indicator** – calculation formula is either a direct variable from the monitoring system (application response time), or a combination of several monitoring variables (transactions per second, tps).
2. **Composed indicators** – use other indicator values as inputs, such as application energy performance (ratio of tps and power).
   A. Power; time; memory; processing.

11.5.1 A data definition of Indicator

An indicator is a variable, and a variable is a name for a location (carrying more meaning) in memory, and used to store a 'value'. The indicator associates (i.e., "tells you") what is going to be measured (i.e., what is of significance). The means of verification relates to how that which is significant will be measured. The indicator, which is an entity, has the attribute of a 'numerical value', representing an actual number, proportion, percentage (i.e., rate).

11.6 Societal conceptual indicator types

In application in a societal information system, there are several types of indicators.

11.6.1 Performance indicators

Performance (a.k.a., Results and Output) indicators measure the results of action (efficiency and effectiveness), providing a measure of the efficacy of an activity. In order to ensure optimal performance, indicators are needed in order to enable the decision controller (or decision space) to compare the results of action with the objectives for action.

Performance indicators are the results of the previous decision as evaluated against requirements. Performance indicators measure a/the performance, to understand better how performance is occurring (i.e. how well things are working, to introduce corrective actions, to validate results, to improve accountability, etc.).

> *A performance indicator is a specification (a plan, a decision solution) that allows for comparison between itself, the target, and some execution, the actual result.*

Performance indicators include the following two additional characteristics:

1. Actionable (a measurement of ability).
2. Achievable (a measurement of ability).

In application, a performance measure is an aggregate [measure] that signifies (describes) the human-relevant condition of an ecosystem, or one of the ecosystems critical components/dynamics. Wherein, an indicator may reflect: biological, chemical, or other physical attributes of an ecological condition. Performance indicators are used to monitor the progress toward and objective.

A performance indicator is a 'strategic instrument' (tool capable of being integrated into the unified information space), which allows for some user (or group of users) to evaluate performance against targets (intended/ expected, demanded performance).

A performance indicator must have a target measure. There must be a target measure (or metric), because the organization is being moved [by change] toward an objective target.

In other words, an indicator has a metric that measures the direct results of decisions as to the overall direction of the organization. Technically, a performance indicator is not an 'objective' measure, since the measurement is not independent of the observer. In the contrary, the indicator is defined by its decisions ("author") in accordance to the type of action conducted and the goals pursued.

Herein, a performance measurement system is an information system that allows a user to track the execution and results of an objective (strategy) through the monitoring of performance indicators.

11.6.1.1 Performance indicator formatting

It is always necessary to define clearly each indicator with fundamental parameters.

1. (Label, name) The symbolic identification of the indicator.
2. (Optimal relationship articulated) The objectives (requirements) of the indicator.
3. (Issue articulated) The problem drivers (issues) related to the indicator.

A performance indicator becomes an objectives [chain] combining associated decision variables.

A performance indicator is: the objectives [chain] and decision variables.

1. A reference model, which gives a structure of the HSS system.
2. A structured approach, leading step-by-step from an existing system state to a future one.
3. Various modeling formalisms to describe the components of the structured system (graphic formalisms, entity/relationship formalisms).

The habitat service system (a.k.a., production) is classified by discrete service processes. The global model is composed of the description of the physical, decision, and information systems.

Performance measures include:

1. Output: Tangible and quantifiable results from efforts entirely within the project/activity, not involving interactions with individuals or organizations that are not project/activity members. Examples include planning workshops and conferences, staffing and equipment plans, publications, reports, draft standards or codes, software, algorithms, assimilated data.
2. Outcome: Measurable results of projects/activities. Examples include new expertise, knowledge, or capabilities; adopted codes and standards; and practitioner acceptance.
3. Impact: Substantial, positive changes enabled by, or due to, project/activity outputs and outcomes, including impacts on other agencies, industry, or society. Changes are associated with external entities, not internal to the project/activity. Examples include changes in societal behavior, changes in building codes and standards, etc.

11.6.2 Scientific indicators

A scientific indicator may be defined as an aggregate measure, index of measures, or a model element, that signifies (characterizes) an ecosystem, or one of its components. Scientific indicators are used to monitor the ecological environment.

Note here that scientific indicators do not have targets. Not all indicators have to have targets; they could just be reporting patterns of change.

11.6.3 An environmental indicator may be defined as

An environmental indicator is a variable related to any aspect of the environment, supposed to respond to modification, and representative for a delimited area. It is a variable for which a value in the reference state can be estimated. The set of indicators should cover as homogeneously as possible all aspects of the environmental system, an any addition of a new indicator should result in the addition of information.

An environmental indicator might refer to the density, abundance or distribution of a population, a taxonomic, functional or genetic metric, a behavioural parameter, or any other natural parameter fitting the definition.

11.6.4 An engineering indicator is

An indicator is used for the visual detection of the completion of a particular behavior (and/or reaction). Engineering indicators are used to monitor the progressive development and operation of a system.

Engineering indicators support the effective decision control of habitat systems by providing visibility into the current, as well as, expected project performance and potential future states.

> **INSIGHT:** *The specification of functional requirements involves mathematical concepts (e.g., number, and operation) and their metrics and indicators that quantify and evaluate them.. The specification of non-functional requirements involves calculable concepts (e.g., quality, accessibility, productivity) and their metrics and indicators that quantify and evaluate them.*

11.6.5 From an environmental coordinator perspective, an indicator is

An indicator is a characteristic or an entity that can be measured to estimate (predict) status and trends of the target environmental condition and/or resource, over time. Wherein, the numerical attribute of an indicator is a quantitative datum (value, level, etc.) that reflects (shows) the presence or amount, quantity, of a factor under observation by the system.

The conception of indication has three directability characteristics:

1. **Indicators indicate environmental nodes and relationships**. These type of indicators are often referred to as environmental indicators, or scientific indicators. There is are no directional "value" weights applied to them. They are indicators of extant quantities, or not, objects, relationships, and dynamics. The directability here is the measurement process itself, for which there are three types:
   A. Non-experimental research - Only measure once and no information need to compare over time or group. Non-experimental research can express if some event/behavior took place, describe the details, and concurrent occurrence (is this occurrence associated with another occurrence), but it cannot say that one thing caused another, there is no causality.
   B. Quasi-experimental research = group 1 and group 2, compare.
   C. Experimental research = group 1 (intervention) and group 2 (control), compare
2. **Indicators indicate environmental significance** (i.e., indicate something significant in the environment). Indicators represent data (of significance about the environment) whose meaning is consolidated and expressed at a higher level than the information upon which the data themselves are based. A factor (i.e., indicator) in the environment is carrying capacity. The directability here is the alignment, error and its correctability.
3. **Indicators express a link between the environment and an intended outcome**. These indicators are set by understanding prior data (as baselines, targets, benchmarks), and evaluated against incoming actual data to determine error and inform the control (i.e., correction) decision ... in order to maintain course (or the direction of human and ecological flourishing in the case of community). The indicators holds the "baseline" or "target" information on an issue of concern and presented in a form which informs an algorithmically pre-determined [common, objective] decision space. The directability here is the potential for alignment, error and its correctability.

11.6.5.1 Indicators as an objective expression

In objective expression, indicators are generally quantitative variables. They are expressed in single terms or brief descriptions, and in their container are generally the following associations:

1. Of a quantity.
2. Frequency of event.
3. Result of a scoring (weighting and/or comparing) system.
4. They can also be qualitative indices.

11.6.5.2 The indicator's metric view

An indicator is sub-classified as a metric, or a classified combination of metrics, that provides insight into (accounts for) the process, project, or product itself (i.e., accounts for the status or state of a system). Indicators (a type of inquiry identifier) define a trace from inquiry to that which is required to resolve the inquiry. Indicators are necessary for directional comparison and knowing whether something has occurred. If a metric reflects performance, it is a performance indicator. If a metric reflects risk, it is a risk indicator. Indicators are derived from questions, which are themselves derived from goals and objectives. Indicators are sub-composed of metrics. There are two types of change control indicator; one type of indicator that signals positive progress or quality, and another (risk) that signals delay or damage of progress or quality:

1. **Environmental indicators (scientific and resource indicators)** - indicators that indicate the state, status, or health and/or availability of an environment. General environmental indicators include the measurement of: humans; other living beings; ecological resources/services; knowledge (scientific); and equipment (infrastructure, components). An indicator is a linguistic representation that points to some signalled existence in the real world. An environmental indicator is an attributive, measurable characteristic of the environmental state. Scientific indicator is a single piece of information which acts as a surrogate for an environmental variable to serve a particular use or interest". Environmental variables and Environmental indicators. Each environment variable is analysed separately and an indicator representing this particular environmental aspect is adopted to monitor a phenomenon in time, in space, or to estimate progress toward goals that should be reached. An environmental "indicator" is a scale indicating various degrees of environmental quality with regard to a particular environmental variable. A scientific indicator is a linguistic representation of something dynamically observed in the real world.
2. **Performance/quality indicators** (a.k.a., "good" indicators, quality indicators, a type metric; positive progress; results indicators) - indicates the quality or state of a system; it is the goal of an expected performance in/through time. An performance indicator is a linguistic representation of something specified through requirements as being in the real world. A performance metric is something that can be pointed to in an information system or a physical system that indicates a quantitative use (i.e., is something useful) based on one or more metrics, observations, or both. Performance indicators (and their metrics) represent a desired state or status. Performance indicators indicate that which is desired from a project, process, or product. Performance indicators are factors that a system needs to monitor (and benchmark). To engineers, performance indicators indicate functions and quality indicators indicate conditions required from a system. Generally, performance indicators are quantitative variables, and are defined with a threshold (or standard) value. An objective to be reached or maintained can, at times, be considered an indicator without establishing a threshold as long as requirements are defined precisely. Performance indicators evaluate how successful a service system is at meeting a service directive, objective or requirement. Performance indicators define and measure ("express") progress toward the successful completion of a process or project. Performance indicators define and measure performance (progress) relative to project or process, organizational goals (objectives). Once an organisation has analysed its mission and defined its goals, it needs to measure progress towards those goals. A performance indicator expresses the achievement of a desired level of results in an area relevant to the evaluated entities activity. What are the "success" factors of the project/process? Performance indicators ask if a project/process is on track, or its results were as expected, and if not on track or as expected, where not.
   A. What are past goals (past performance indicators)?
   B. What is current goal (current performance indicator)?
   C. What is future goal (future performance indicator?
3. **Risk indicators** (a.k.a., bad indicators potential negative progress) - indicators of the potential to express negative change progression. Note that 'risk' is a measure of the probability that a negative outcome will occur. Risk indicators indicate an undesired state or status, one that could harm, delay or damage. Risk indicators may provide an early warning of increased risk exposure (i.e., metrics to define and measure risks). By monitoring risk indicators, the problems expressed by them are possible to identify early, whereupon a pro-active (planned) approach of mitigating risks before they escalate and have more serious consequences occurs. A risk indicator (a.k.a., effectiveness indicator) is a sign that an incident may occur, or is occurring.

Scientific performance indicators include:

1. Performance indicators express positive (i.e., evolutions) and negative (i.e., problems) change of progression.
2. Scientific indicators record intentional change.

Performance indicators components:

1. The measure - What is being measured.
2. The target - The expected value.
3. The source - System of input of the data.
4. The frequency - how often to report.

In this case, a metric is essentially a target - a quantitative value for a goal or objective.

Performance indicator sub-types:

1. **Count indicators** - How many, raw count.
2. **Progress indicators** - What percent complete of objective.
3. **Change indicators** - percent increase (possibly, compared to some prior date).

> **NOTE:** *Performance metrics data that indicate a problem area should not be considered "negative"; instead, these data are merely an indicator for [process] improvement, and an opportunity to be better.*

A **metric** is something set up as an example against which others of the same type are compared. A metric (a sub-type of indicator) is a collection of the same type of data used to understand and change optimally over time across a number of unified dimensions or criteria. Specifically, a metric is a quantitative [statistics] measure of the degree of alignment to which a system, component, or process:

1. Possesses a given attribute, or
2. Describes a given event, or
3. Predicts a given trend.

A metric is an aggregation of one or more measures to create a decision context (a.k.a., actionable information context, intelligence context). Actionable information is information that can be used in system control decisioning. Technically, every measure is a 'metric' when associated with contextual information. In this sense, metrics are the numerically counted values (measures) and their meanings (units and indicators). A metric is any contextualized measurement; it may refer to anything in the real world, which can be counted ("measured"). Any real world measure could be a metric. Metrics involve properties of the environment that can be measured directly.

A performance metric is a quantitative measure or derivation from two or more measures, which may not necessarily indicate something useful to particular observers. It is a measure of something that does not necessarily indicate something useful to particular observers.

Metrics for organizations include:

1. **Time** - hours or days elapsed from the time a request is made until evaluation is complete (t_{queue}).
2. **Effort** - person-hours to perform evaluation, (w_{eval}).
3. **Time** - hours or days elapsed from completion of evaluation to assignment of change order to personnel, (t_{eval}).
4. **Effort** - person-hours required to make the change, (w_{change}).
5. **Time** - required hours or days to make the change, t_{change}.
6. **Errors** - uncovered during work to make change, (e_{change}).
7. **Defects** - uncovered after change is released to the customer base, (d_{change}).

A **measure** (a type of indicator) is the directly recorded observable value or performance. A measure is measurement of the value of a specific characteristic of a given entity (collected data). A measure is a quantitative indication of extent, amount, dimension, capacity, or size of some attribute of a product or process. A measure is, How much there is of some thing that "you" can quantify. Measures enter an information space as data **--** a collection of facts and/or statistics for reference or analysis.

11.6.6 Applied societal control indicators

The common societal indicator types include:

1. Resource indicator (RI).
2. Environmental indicator (EI).
3. Material [economic-access] indicator (MI).
4. Human indicator (HI).
5. Social indicator (SI).
   A. Indicators of well-being.
   B. Indicators of social cohesion (note can't measure social cohesion directly).
   C. Indicators of human fulfillment.
   D. Indicators of human capability (capacity).

For example,

1. Air condition is a metric, because it can be measured.
   A. Air condition is a performance indicator for the habitat service system, because the organization is concerned with the impact upon and change of air condition.
   B. Air pollution is a risk indicator for the habitat service system (health, safety, security, and environment).

11.6.7 Common societal indicators

Common societal indicators include, but are not limited to:

1. **Hours used vs. hours estimated vs. hours remaining** through statistical calculation upon the log time data. Wherein, time is logged continuously and/or regularly (i.e., time tracking occurs).
2. **Resource loading (per person)** through statistical calculations and algorithmic expressions determines if the system (person or otherwise) is carrying too many tasks (i.e., too much responsibility). Wherein, time is logged (i.e., time tracking occurs).
3. **Earned value analysis** a method of measuring a project's progress at any given point in time, forecasting its completion date and final cost, and an analysis of variances in the schedule and resource requirements as the project progresses.
   A. **Potentially ineffective projects ("Projects at risk")** - projects with the potential to harm the functioning of an optimized and adaptive, resilient and regenerative [community-type] societal system.
   B. **Effective projects ("Healthy projects")** - projects with no potential to harm the functioning of an optimized and adaptive, resilient and regenerative [community-type] societal system.
   C. **Ineffective projects ("Trouble projects")** - projects possessing the potential to (*future*), or actually harming (*present*), the functioning of an optimized and adaptive, resilient and regenerative [community-type] societal system.
4. **Estimated priority ("Estimated value" and "Estimated profitability)** through estimated, predicted consequences to human and ecological fulfillment caused from a[ny given] state change to the material (or otherwise conscious) environment. Wherein, value is traced through to need. In other words, orientation is traced to reliably direct toward a set direction[al heading], and a test is predictable, regularly.
5. **Average time tasks take** to stay in each stage of the process.

11.6.8 Societal indicator types

There are three indicator types for any society:

1. **Systems-based indicators:** Indicators that relate more to the coordination and the information system; societal systems level. Indicators that relate more to the coordination of the societal system.
2. **Operations-based indicators:** Indicators that are relevant to the functioning of an organization's infrastructure (e.g. machinery, operations); potentially site-specific. Indicators that are relevant to the functioning of the societal system's structure.
3. **Behavior-based indicators:** Indicators that measure the behavior or actions of individuals or groups (in the workplace); people-to-people interactions related to work; useful at site-specific level through society level. Indicators that measure behavior or actions of individuals or groups in InterSystem Team Service.

11.6.9 Living environmental indicators

A total living environment has three primary types of indicators:

1. **Environmental condition indicators** (ecosystem service indicators).
2. **Indicators of societal coordination** (societal/social performance indicators, social cohesion and fulfillment indicators). These indicators are otherwise known as human development indicators. These indicators indicate the fulfillment of human needs, requirements, and capabilities. These indicators refer to the requirements of the unified societal system.
3. **Indicators of operational coordination** (operational performance indicators). These indicators are otherwise known as human service indicators. These indicators indicate the quality of the service [by the operational habitat service system]. These indicators refer to the requirements of the materialized habitat service system. On an activity level, it allows the assessment and control of ongoing processes and environmental impacts.

To living beings, indicators and metrics conceive and resolve decisions. In the real world, there are two primary types of environmental [performance] metrics, each defining a set of correctly orienting metrics for a specific environment:

1. **For ecosystem services**, defining the right metrics involves scientific investigation into the global ecosystem (i.e., the global habitat service system). Services at the planetary scale.
2. **For the human service system**, defining the right metrics involves the engineered construction, and scientific investigation of, the societal habitat service system. Services designed by humans for humans.
3. **For the personal system**, defining the right metrics involves the knowing of ones own capabilities through regular practice. The personal practice of capability as a service to oneself.

11.6.10 Ecosystem service indicators and metrics

Global environmental indicator's indicate the state/ status of the planetary [environmental] ecosystem, given what is known. These indicators include:

1. What is necessary for all planetary life.
2. What is necessary for human life.
3. What is necessary for individual flourishing.

Humanity requires an ecology (ecosystem services to feasibly provide for itself on any major scale. Humans can purify air for themselves on an astronaut navigated spacecraft, but on earth, plants and other systems perform this operational service.

At a high-level, every ecosystem service is an environmental indicator. There are six major environmental indicators to determine the health of ecosystem (i.e., ecosystem sustainability):

1. Biodiversity – number and variety of organisms in an area.
   A. Genetic diversity – code for re-configuring provides resilience within a population.
   B. Species diversity – variety of living beings
   C. Ecosystem biodiversity – looking at planetary ecosystem.
2. Extinction rate – rate at which species disappear.
3. Food production – the amount of food an environment can produce.
4. Temperature and CO2
5. Population size relative to carrying capacity.
6. Resource depletion rate.

11.6.11 Societal information system indicators

Information organizational indicators (i.e., in the real world and in an in an organizational systems context, there are two usages of indicators):

1. **Conceptual (indicates potential meaning, understanding)** - structure a conceptual framework for understanding and working with information and problems therein. Conceptual indicators make use of scientific values (to form the semantic structure of science).
2. **Decisional (indicates potential decision, selection)** - the use of indicators to select decision options, resolve decision spaces. Decisional indicators make use of a *target value* (to take decisions once new data, new information, has arrived and integration is complete).

In the real world, the following types of indicators (and metrics) exist:

11.6.11.1 The scientific type (conceptual, to derive meaning)

A scientific-type information system involves, at least:

1. **Scientific indicators** - A scientific indicator is a single piece of information (a single identifier, with description) that associate the [real world] environment with [an environmental] variable to serve a particular inquiry (use or interest). Simply, a scientific indicator indicates what is being measured with a symbol and accompanying description of what is being indicated in the context of all knowledge (i.e., all science).
   A. For example, a direct scientific indicator is 'water' (H20).
   B. For example, an indirect scientific indicator is 'biodiversity'. All indirect scientific indicators are made up of direct scientific indicators. Indirect scientific indicators are abstract groupings of indicators conveying greater meaning and allow for intentional re-orientation within a useful information space (i.e., within society).
2. **Scientific metrics** - a measure(s) in the context of the whole scientific use interest.
3. **Scientific measures** - a specific, point measure composed of a value and unit.
   A. For example, a direct scientific metric is 2.3Liters of H20 in Pond X. Note that, in general, it is the 2.3 that is referred to as "the metric".
   B. For example, an indirect scientific measurement is the biodiversity of square kilometer X. Biodiversity is made up of multiple indicators, including number of species types, number in each species type, and size of region.

11.6.11.2 The performance type (decisional, to derive selection)

A performance-type information system involves, at least:

1. **Accuracy performance indicators** - indicates how well the system is performing, in a given environment, as compared to ("against") the specified system (with a descriptive requirements specification).
   A. For example, correct classification of data points could be one indicator
2. **Accuracy performance metrics** - measures how well the system is performing, in a given environment, as compared to ("against") the specified system (with a set of requirement's metrics).
   A. For example, check to "see" how many of the data points from a data set were classified

correctly. The name for this type of performance metric is "accuracy". A metric of 20 classified correctly, and 10 incorrectly, which is 2 away from the 22 threshold. The metric indicates, when evaluated by the standard threshold of 22, that performance is below standard.
3. **Decision/Selection performance indicators** - indicates which of a group (set, {}) of options (choices, probabilities) is better (positive, +) or worse (negative, -).
4. **Decision/Selection performance metrics** - input into decisioning to resolve the determination/ selection (to determine, select) one probability path over another.
   A. For example, one classification algorithm 'A' classifies 80% of data points correctly, and another classification algorithm 'B' classifies 90% of data points correctly. An observer with a decision space realizes [through this 'experience'] that algorithm B is performing better than (in comparison to) the other algorithm. Note, that there are nuances (intricacies) here.

11.6.12 Project life-cycle phase indicators

In general, every phase of a project will have its own indicators:

1. **Input level indicators** - survey of resources. For example, survey of availability of a specific type of water pump and horses that drink water to live.
2. **Process level indicators** - operational performance. For example, water pump performance.
3. **Output level indicators** – amount of output. For example, gallons of water pumped 3; number of buckets to carry 10; feet of leading rope prepared 40; bridles on horse 1.
4. **Outcome level indicators** – amount of outcome. For example, liters of water made ready for horse to drink 1; number of horses ready and willing to drink 1.
5. **Impact level indicators** - the environmental affect. For example, # of horses independently accessing water; # of gallons of water consumed by horses in the city 5.

11.6.13 Project and process indicators

Process performance indicators:

• On time delivery, user satisfaction.

Project performance indicators:

• Percent of project complete, milestones against target.

11.6.14 Project[-scale] indicators

Project indicators include, but may not be limited to:

1. **Process indicators** (Process indicators indicate the change process) – indicators that are used to measure project process or activities. For example, in a water project, this could be: the number of chlorine dispensers installed at water points, or the number of households that have received training on chlorination of water.
   A. # of farmers supplied with drought resistant crops.
   B. # of community awareness meetings conducted.
   C. No of wells/dams constructed.
   D. No of farmers enrolled in crop insurance.
   E. No of irrigation systems constructed.
2. **Outcome indicators** (Outcome indicators indicate the short-term change) – indicators that measure project outcomes. Outcomes are medium impacts of a project. For example, in a water project, this could be: the proportion of households using chlorinated drinking water, or the percentage of children suffering from diarrhoea.
   A. Proportion of food secure households.
   B. Percentage of malnourished children under 5.
3. **Impact indicators** (Impact indicators indicate the long-term impact of the change) – indicators that measure the long-term impacts of a project, also known as project impact. For example, in a water project, this could be: the prevalence of under 5 mortality.
   A. Employment rates of the region.
   B. Prevalence of under 5 mortality.

11.6.14.1 Project metrics

Project metrics include, but may not be limited to:

1. Effort/time per task.
2. Errors uncovered per review hour
3. Scheduled vs actual milestone dates.
4. Changes (number) and their characteristics
5. Distribution of effort on engineering task

11.6.15 Project progress indicators

Project indicators can have several uses and be of several types:

1. **Monitoring** (the state of health/progress of the project in a common-parallel decisioning space).
2. **Observing** (discrepancies in the memory-state of the project and the resulting deliverable of planned executions) .

3. **Analysing** (possible solutions 1. to project-level discrepancies, and 2. to system-level discrepancies).
4. **Synchronizing** (activities and tasks with availabilities).
5. **Anticipating** (issues, risks and improvement opportunities).
6. **Decisioning** (optimal solution selection).

Five project indicators (five indicators for the assessment method):

1. Entry criteria (scope of process).
2. Cost *of process.*
3. Duration *of process.*
4. Resource *of process.*
5. Expected criteria (scope of process).

> **TERMINOLOGICAL CLARIFICATION:** *The cost, duration and scope of a project are sometimes called the "project management triangle".*

> **NOTE:** *Additional indicators may be specified depending on the project.*

Project indication has three types of possible indicator values:

1. **Planned value (PV):** the pre-decided value (e.g., budget or planned value of work scheduled).
2. **Actual value (AV):** the actual resulting value (of work completed, for example).
3. **Earned value (EV):** the "earned value" of physical work completed. This is a market-only term; there is no concept of "profit" or "market-State economic growth" in community.

11.6.16 Service indicators

There are two true services:

1. One support[ing] service.
2. For the whole societal system.

Whereas, 'technology' is "true" support, 'life' and 'facility' are "true" 'services'. Of course, technology support is also a type of service. In an operating system (societal), imagine technology support as the combination of firmware and hardware, which functions through physically and logically discoverable processes. In computing, this combination forms a computing platform upon which more complex computing operations can be run. For society, this means that (given what is known) the 'life' and 'facility' systems the two second layer platforms upon which the base, technological is formed. New experimental discoveries occur the Facility System and maintain operational processes protocols as common to all systems.

Each HSS has a set of indicators:

1. Medical indicators – health of individuals.
2. Energy indicators - energy usage of individuals.

From a general point of view, the term 'requirement' could be considered, "a thing that is needed or wanted". Requirements define the services expected from the [habitat service] system (functional requirements), and the [societal organizational decision-inquiry] constraints that the system must follow (i.e., obey; more practically, protocols-algorithms). Constraints may otherwise be known as non-functional, or qualitative (i.e., qualifying, constraining) requirements. Constraints place restrictions on the system been developed, notably in the fields of usability, reliability, mobility, regenerability.

Each time a system must be designed or re-engineered the design/re-engineering decisions are composed and resolved ("taken") on the basis of objectives flowing as user requirements. This is the basis for all outputs, results, performance, process performance, quality, and assessment and evaluation.

> **CLARIFICATION:** *The process named "requirements engineering" is 'the systematic process of eliciting, understanding, analysing and documenting requirements'.*

11.6.17 Societal service performance indicators

There are several levels of societal indicator representing the different layers of society:

1. **Systems-based indicators (project-based metrics, Level 0 indicators)** - Indicators that relate to the planning, coordination, and change control of systems (i.e., systems indicators). Everything is a project.
   A. Systems-based metrics (project-based metrics), may include: Assess the status of an ongoing project; Track potential risks; Uncover problem areas before "critical" flag; Adjust work flow or tasks.
2. **Societal-based indicators (Level 1 indicators)** - measure a societal system model's level of alignment with society standards. Indicators that relate to the design and functioning of the unified, societal information system (i.e., societal systems indicators). Indicators of their presence and functioning (as, how?, and how well?). Information system impact.
   A. Social system indicators.
   B. Decision system indicators.
   C. Lifestyle system indicators.
   D. Material system indicators.
3. **Operations-based indicators (Level 2 indicators)** – measure the habitat systems inputs, activities, outputs ('activities' are sometimes classified here, under 'outputs'), and performance. There

are indicators relevant to the functioning of an organizations infrastructure, the network of integrated and materialized habitat service systems (i.e., city operations indicators). Habitat Service System Operations impact. For example, amount of hazardous waste, total resource operating cost, # of activities to maintain service system.
   A. Life Support System.
   B. Technical Support System.
   C. Facility Support System.
4. **Behavior-based indicators (Level 3 indicators)** – measure the potential impact that the materialized habitat system's presence and activities have on its users (the community), its workers (InterSystem Team members), and the surrounding environment. Indicators that measure the behavior or actions of individuals or groups of actors, humans and/or machine (i.e., the behavior of humans and their services as indicators). Behavior impact.
5. **Ecological Service-based indicators (Level 4 indicators)** - measure the operation of the ecological service system. Ecological impact. These indicators measure how the network of city systems, and their production activities (i.e., the operating services) affect the larger picture of an ecologically sustainable society. For example, % renewable materials used at a lower or equal to renewal rate, community quality of life, worse health status compared to other companies in industry.

11.7 [Decision] Indication interface

A.*k.a., Dashboard, passive system interface, monitoring interface tool, visual data analytics tool, indicator display, analytical indicator visualization*

In general, a 'dashboard' is a visual monitoring and data analytics interface for operating in a specific type of information space.

CLARIFICATION: *Interface refers to a point of interaction between components, and is applicable to the level of both hardware and software (via an input/output system with associated protocols).*

A dashboard is a screen/page (a digital-computational information interface) that indicates, items and/or issues, in some sort of priority. A dashboard is an interface with two possible functions:

1. Viewing the information sub-system
2. Executing analytics on indication and measurement data, including upon indicators, metrics, measured values, and synthesized data itself.

INSIGHT: *A dashboard is a visual interface into decisioning. All decisioning is procedural. A dashboard is a multi-functional display; the ability to execute analytics and synthesize the results of a transparent decision resolution inquiry.*

A dashboard is simply a monitoring and analytics interface into an information space. Dashboards provide an overview of current, past, and/or future, system status, including data about the events collected and generated by the system. A dashboard is a useful, highly customizable monitoring feature that provides actionable data given an objective direction.

NOTE: *Analytics tools other otherwise known as discovery tools, because they synthesize new information from the information given (Read: prior available), and this new information may be said to be, "discovered".*

A dashboard is a visual display of the most important information needed [by a user] to achieve one or more objectives; consolidated and arranged on a single window. All dashboards have a visual layout (pre-designed to meets user requirements).

There are different names for different information space views, and hence, different names for different dashboard configurations:

1. Project dashboards.
2. Evaluation dashboards.
3. Assessment dashboards.
4. Change control dashboards.

A dashboard is a single place for viewing all key indicators (and metrics). A window into the overall assessment (health, progress, etc.) of all projects (or other directional information packages). A dashboard (and its backend) visually tracks all indicators, and provides raw, graphed, and calculated data. The system is [in part] capable of visualizing due to a backend tracking and statistical calculation system to which all project variables and metrics are available. Note that a metric appearing on the Community-user's dashboard is not necessarily a performance indicator.

The dashboard shows indicators, which carry pools of values associated with scientific measures and/or diagnostic measures.

1. Dashboard reporting (monitoring) of this operation.
2. Dashboard analytics (calculation on measures, and on, results). The sub-operation where new data is calculated in the system from prior.

In an integrated system, diagnostics are consistently run on sub-systems to ensure that they are functioning appropriately and to catch errors or potential further problems. Diagnostics are an essential element of the 'maintenance' operational process.

11.8 [Decision] Indicator assessment

A.k.a., Indicator analysis.

Analysis upon indicators may involve evaluation, assessment, or calculation.

11.8.1 Assessment (an analysis of results)

Information monitoring and analytics capabilities and tool.

Assessment and evaluation mean the same thing in an information systems context. However, evaluation is more commonly used in some engineering contexts, and assessment in some scientific-environmental contexts. Regardless of context, the meaning the re-solution [tool], is the same:

1. Requirements are assessed through evaluation of a system's alignment with those requirements (validation and verification).
2. Or, some variation of the same meaning, such as, An object and/or event is evaluated (to produce a new value for the environment) through an evaluation process (by means of a method) that compares states [wherein accurate information has been collected].

In general, the term, 'evaluation' connotes a direction of meaning (as in, engineering). And, in general, the term, 'assessment' connotes a meaning of direction (as in scientific research). Verification and validation data is used to determine the results of an evaluation.

11.8.2 Assessment of the project's progress

A.k.a., Measured achievement, measured success, metric analysis, indicator-metric analysis, solution progress, project progress, etc.

In order to assess the project, it is necessary to define the assessment indicators, including the two most common:

1. Milestones (are more broad than KPI) - a milestone is a strategically marked future event or deliverable (or, condition) that has occurred, has been documented to have occurred, and was measured in relation to the requirement that was expected for strategic longer-term outcomes/conditions.
   A. Work toward the completion of a project must get to "this document point, then get to this next, and so on".
2. Key project indicators (KPI) - are events, deliverables, or conditions that are expected to be occurring now or in the nearest future, and have been designed to occur, are documented and is measured in relation to the requirement for expected current (and near-term) operation.

A project's controlled execution uses the current project indicators to validate the conclusion of an expected project process (situation) at a specified project end time. Hence, in this case, monitoring is the comparison of a measurement-based estimation (e.g., derived from measurement of effort) with the respective project goal (e.g., total effort/resource budget [bounded access] of the project).

11.8.3 Project indicators and assessment

I.e., Project performance and completion indicators.

Indicators are widely used to measure the success (alignment) of any type of project, service, or product. At the project level, performance indicators are defined to assess the progress of the project. Project indicators informs decisioning, rather than being an end in themselves.

While a given task is under way, the execution of indication is applied to monitor changes to each indicator. By comparing them with the objectives (expected value, if present), it will be possible to detect deviations (in scheduling, performance, quality, materials, etc.).

Project indicators can have several functions and be of several types:

1. Monitoring (the state of "health" of the project).
2. Observing (discrepancies).
3. Analyzing (possible solutions).
4. Synchronizing (activities).
5. Anticipating (risks and opportunities).
6. Facilitating (decision-making).
7. Characterizing project progress in a summarized form.

Shared indicators include:

1. Resources (I_r).
2. Entry criteria (I_{en}).
3. Expected criteria (I_{ex}).
4. Duration (I_d).
5. Cost (I_c).

11.8.4 Environmental impact assessment

An environmental impact assessment identifies the various impacts (to all impactable systems) should a change (intentional or not) occur. As part of the societal design process there exists a sub-decision module that conducts continuous environmental impact assessments on information passing through the decision system. An assessment provides significant information on indication and measurement, and may be used to decide, select indicators and metrics.

There are [at least] five impactable systems in human society (i.e., the primary types of environmental impact are):

1. Social.
2. Decision (economic).
3. Lifestyle.
4. Material (local habitats).
5. Ecological (global habitat).

Evaluation through structured information flow:

1. Objectives - identify objectives that establish a need.
2. Goal - define one or more goals required to achieve stated objective.
3. Question - develop one or more questions that when answered, help determine the extent to which the objective or goal is met.
4. Indicator - identifying one or me pieces of information that are required to answer each question.
5. Metric - Identify one or more metrics that will use selected indicators to answer the question.

11.9 [Decision] Indicator evaluation

Evaluation refers to the evaluation of alternative designs [given a set of social inquiry criteria for integration into our active specification]. Evaluations are carried out for each domain (i.e., each service system; such as, life support, energy, tech support). The evaluation is based on indicators that verify requirements.

In order to acquire and calculate change data, a systems-level change [control] process of evaluation must exist. Evaluation provides necessarily useful information on change in time (e.g., increase or decrease, improvement or decline).

Because evaluation is a time affected process, if an outcome (project, engineering, etc.) mentions (requires) an increase, improvement, or decline, then the indicator will need to be compared/measured at least 2 different instances over time (i.e., two temporally separated measurements must be taken), for example:

1. Pre-testing, or a baseline [value] for initial conditions.
   A. Prior measure(s) (pre-tests) from which a baseline (target, metric) has been composed and/or selected.
2. Post-testing, or a post [value] for the new conditions (Read: value for conditions after the change was executed).
   B. Baseline to current [actual, post-test] comparison.

This process of comparing over time is quasi (almost/sort of)-experimental research.

11.10 Real world evaluations

There are two types of evaluation relating to the two measurement dimensions (categories of information) in the real world (two outcomes, one is scientific and one is engineered quality assurance).

11.10.1 Scientific evaluation (a.k.a., direct measurement and analytics)

Scientific evaluation involves the scientific method, and the real word, to collect and analyze data (i.e., to do true 'research'). Scientific evaluation produces and uses scientific indicators and scientific metrics. Scientific metrics (true research, facts) are the results of scientific experimentation.

The best way to establish change is to look for the occurrence of the indicators [of change] in two groups. One group is the 'target' group, to which an intervention is applied, and the second group is the 'control' group to which no intervention (or uncontrolled influence) has occurred. This process of comparison is often called [true] experimental research.

11.10.1.1 Experimental evaluation

Experimental evaluation refers to monitoring and evaluation of outcomes under conditions of direct controls over inputs and processes. Experimental evaluation refers to more scientific usability tests where hypotheses are being made and tested, and statistical results are collected and processed. (Preece, 1993:117) In experimental and quasi-experimental evaluation, the estimated impact of the intervention [in the experiment] is calculated as the difference in mean outcomes between the treatment group (those receiving the intervention) and the control or comparison group (those who don't). This method is also called randomized control trials (RCT).

11.10.2 Quality/progress evaluation (a.k.a., effectiveness evaluation, performance evaluation, program evaluation, process evaluation, environmental evaluation, diagnostic evaluation, and indirect measurement, and "monitoring")

Evaluation toward (i.e., progress, performance, quality, etc.) goals. Effectiveness evaluation involves statistical calculation on collected real world data, true analysis. Quality evaluation produces quality metrics and quality indicators. Quality metrics are the result of statistical calculations. Here, evaluation exists to determine progressive alignment with a direction by the method of 'measurement'. Evaluation (and monitoring) is the process of collecting and analyzing measurement data (measures) to inform decisioning and ensure process results align with input objectives. Quality/progress evaluation answers the questions: What indicates achievement of objectives by a project or process, in time? What does not indicate achievement of objectives by a project or process, in time? How effectively is a project or process achieving objectives (directives and/or orientations) in time? Questions related to a process (i.e., "What is the process for ...?") lead to implementation metrics. Questions related to effectiveness (i.e., "How effective is ...?") lead to effectiveness metrics.

To measure quality, there are [at least] the quality indicators and metrics of:

1. **Correctness** – the degree to which a system (program) operates according to specification.
2. **Maintainability** – the degree to which a system (program) is amenable to change.
3. **Integrity** – the degree to which a program is impervious to outside attack.
   A. **Threat** – is the probability (which can be estimated or derived from empirical evidence) that an attack of a specific type will occur within a given time.
   B. **Security** – is the probability (which can be estimated or derived from empirical evidence) that the attack of a specific type will be repelled.
   C. Integrity can be defined as: sigma (1 - (threat x (1 – security)))
4. **Usability** – the degree to which a system (program) is easy to use.

From the engineering perspective, quality is conformance to requirements (i.e., "this is what is required, and this is what is designed"). Requirements are the foundation from which quality is measured, because requirements become designs, which become actualized. From the user's perspective, quality is conformance to a design specification (i.e., "this is what was designed, and this is what was built").

11.10.2.1 Project evaluation (Project performance)

> **NOTE:** *The measurement of project performance is an assessment of the magnitude of variation from the original scoped baseline (i.e., from the requirements).*

A project evaluation systems necessitates following life-cycle of information elements:

1. **Inputs** - Those elements that are used in the project to implement it. Inputs are what is composed to make the outputs. Time, resources, humans, equipment.
2. **Activities** – What the people and machines do in order to achieve the goal(s) of the project.
3. **Outputs** – The first level of results associated with a project. What has the project achieved in the short term.
4. **Outcome** – The second level of results associated with a project. Usually refers to medium term consequences of a project. Outcomes usually relate to the project goal or aim.
5. **Impact** – the third level of project results, and is the long term consequences of a project.

A more complete description of project evaluation is a follows:

1. **Input evaluation (input indicators and metrics):** At the initial phase of a project, indicators are important for the purpose of defining how the intervention (state change) will be measured. Through the use of indicators, engineers are able to pre-determine how effectiveness (of the engineered system) will be evaluated in a precise and clear manner. Input evaluation involves the evaluation of those elements (as indicators) that are used in the project to implement it: time (availability), resources (availability), humans (availability), equipment (availability).
2. **Process evaluation (a.k.a., formative evaluation, monitoring [project] progress, activity evaluation; process indicators and metrics)** determines the value alignment of a project/program while the project activities are forming (in progress). Process evaluation involves the evaluation of those actions that people and machines do (execute) to complete goals and objectives. Therein, monitoring is a continuous process of observing and assessing progress. Monitoring involves conceiving and perceiving change to the progress of a process, and that progress can be evaluated [at least] quantitatively. Technically, it is the routine collection of data that measures progress toward achieving objectives using record keeping and reporting. Here, evaluation is the process of measuring 'progress' toward a given direction (e.g., goal or objective). Progress is the status of the current state of a system in relation to prior and/or desired states. Here, an 'evaluation' is a measurement of 'progress', which provides information relevant to the resolution of *access*, *scheduling*, and *effectiveness* of a project.

   These are primarily process indicators; they indicate the change process itself.

During project implementation (project coordination), indicators serve the purpose of assess project progress and highlighting areas for possible improvement. In this case, when the indicators are measured against project goals, coordinators are be able to measure progress towards goals and inform the need for corrective measures against potential errors through to catastrophes.

Formative evaluations generally start with a baseline survey, carried out before an actual project is implemented:

A. Ask context questions about relationship and capacity.
B. Ask implementation questions about the quality and quantity of activities.
C. For each question, then develop ‘process indicators’ that are measures of whether planned activities are being carried out, and how they are carried out. Process indicators indicate a measure of whether planned activities are being carried out, and how well they are being carried out.

   The purpose for formative evaluation (monitoring) is:

   1. To keep processes/projects/programs on track.
   2. To assess the extent to which a process is having its desired impact.
   3. To maintain transparency.
   4. To understand and support decisioning.

CLARIFICATION: *Formative evaluation generally refers to evaluation during a project, and summative means at the end of a phase of the project or the end of the project itself. A well-conducted and well-planned project will have several rounds of evaluation, at varying levels of fidelity.*

3. **Output evaluation (a.k.a., summative evaluation, end-term evaluation, ex-post evaluation, outcome evaluation, and impact evaluation; output indicators, outcome indicators and metrics)** assesses the final or overall [value alignment of the] result (i.e., product, outcome, output, impact, and effect). Although the concept of quality can be measured throughout, from the user's perspective, the concept of quality primarily applies here. Poor quality outputs indicate to the user poor quality inputs and/or processes.

   It is intended to be carried out immediately at project or sub-project conclusion. Summative evaluation is carried out to evaluate project outputs and immediate outcomes, with results of the evaluation compared to the results at baseline. This evaluation generally informs all involved on the project of its success and is important for documenting success and lessons learned, and progressing at the supra-project level. At the end of every test through to final (and beyond) release, there is a summative evaluation.

   Summative evaluations are primarily outcome indicators; they indicate the change to the web of life after the initial results on the change have returned, and are collected over a medium to longer duration of time.

   Outcome indicators indicate change over medium and long-term periods of time. Summative evaluation occurs over time, and reveals the depth of the actual change. It is intended to capture the total impacts of the project's effect (output and activities) over time. Although not always conducted, a summative evaluation is usually the final evaluation associated with a project. If the project is a service, then every modification becomes a summative sub-component evaluation of the continuous formative evaluation of the projected service. Impact indicators indicate the long-term impact of a change.

   During summative evaluation, indicators provide the basis for which evaluation will assess the project impact.

   The purpose for summative evaluation (which may be continuous as a service, like monitoring is continuous) is:

   A. To keep processes/projects/programs on track.
   B. To assess the extent to which a process is having its desired impact.
   C. To maintain transparency.
   D. To understand and support decisioning.

11.10.2.2 Qualitative and quantitative evaluation

There are two types of project evaluation: qualitative and quantitative. A balanced approach combines both qualitative and quantitative evaluation.

Qualitative evaluation involves (the subject), for example:

1. Asking users about their expectation of what the

system will do and how it will function
2. Observing users interacting with a system while "thinking aloud" and noting areas that cause user confusion or frustration
3. Probing for suggestions from users and asking users about their level of satisfaction with the system

Quantitative evaluation involves measuring the following (work performance), for example:

1. Task Completion Rates: Percent of users who successfully complete each task
2. Time on Task: Time it takes for users to perform a task from beginning to end
3. Error Rates: Number of errors made during the course of a task

11.10.3 Evaluation as navigation

Evaluation involves the three navigational elements:

1. **Direction (goals and objectives, questions)** - the setting of a direction; what is to be accomplished, improved, expected? This includes long and short term intention(s) as well as broad to specific desired outcome/effect upon a system.
   A. **Articulate the objective (goal).** All measurements have an objective that structures the measured response.
   B. **Articulate a question** to refine the objective/goal to a quantifiable amount.
2. **Indication (indicator)** - factors that are significant to the successful completion of the direction (achievement of the outcome). Indicators measure success[ful accomplishment] of the direction. Indicators are composed of metrics, and metrics are composed of measures.
   A. **Identify indicators.**
   B. **Identify metrics (measures).** Metrics indicate the measurements required to answer each question.
3. **Determination (evaluator, comparator, analytics) -** was it accomplished as expected, and if not, what action is determined to correct alignment with direction. New value compared to baseline (benchmark as a selected historical, trace value).
   A. Calculate directional comparison.

Evaluation may involve calculating differences between temporal points:

1. **Current actualized ("current")**- The current actualized state (may not align with specified).
2. **Current specified ("target")** - The current specified state (may not align with actual).
3. **Next predicted ("predicted")** - The predicted (next probable future) state.

11.10.3.1 Performance measurement

A complete performance measure includes:

1. **NAME:** The use of an exact and expected (intuitive) name to avoid ambiguity.
   A. Name of metric: HSS.
2. **PURPOSE/OBJECTIVE:** The rationale underlying the measure has to be specified, otherwise one can question whether it should be introduced. The relation of the metric with the organizational objectives must be clear. Typical purposes include monitoring of the rate of change, ensuring that all delayed services are eliminated, and ensure that the asset materialization is efficient and effective for everyone's fulfillment.
   A. Reason for measure: Human fulfillment.
3. **RELATES TO:** The organizational (societal) objectives to which the measure relates should be identified, otherwise one can again question whether the measure should be introduced.
   A. Description of what is measured: Common human need fulfillment and ecological services.
4. **TARGET:** An explicit target, which specifies the level of performance "to be achieved" and a time scale for achieving it. A benchmark is another word for a target, and it means that some value is present. An appropriate target for each measure should therefore be recorded. Typical targets include 99 percent, global human access fulfillment, given common resources and knowledge. By what percent per year are we achieving this on a local and/or global scale. Improvement year on year, ? percent closer to global human access fulfillment during the next ? months, and the target is to "achieve" 95 percent global human fulfillment (given what is known) for a population of 500 on 1295 hectares with on-time delivery by the end of next year.
   A. **Specification** of next system state, as planned in execution.
   B. **Threshold calculation** for each *inquiry process*.
   C. **The target value** is the optimal value [range] as within a range, per indication along the lines of a units of measurement.
5. **FORMULA:** The formula—the way performance is measured -- to 'specify' affects how people behave. As a matter of fact, an inappropriately defined formula can encourage undesirable behaviours. The formula must therefore be defined in such a way that it induces good societal-organizational [ordering] practice. The exact calculation of the

metric must be known to everyone. Also, what is/are the units used (units of measurement must be known).
   A. Measured procedure for how the metric is measured.
6. **FREQUENCY OF MEASUREMENT:** The frequency with which performance should be recorded [and reported] depends on the importance of the measure and the volume of data available (in a technical solution space).
   A. Measurement frequency for how often the measurement is taken.
7. **WHO MEASURES?:** The person/system who is to collect and report the data should be identified.
8. **SOURCE OF DATA:** The exact source of the raw data should be specified. A consistent source of data is vital if performance is to be compared over time.
9. **DRIVERS:** As factors influencing the performance of entities in the decision space.
10. **INTERSYSTEM TEAM:** The team accountability for ensuring the specified performance. The actions taken by accountable persons to change the performance.

11.10.3.2 The performance evaluation process

Performance evaluation (related to ISO 1400:2015):

1. **Assessment** via monitoring, measurement, analysis and evaluation - Assess the organizations environmental performance in relation to society objectives.
   A. **Internal "auditing"** – performs conformity assessment to the requirements defined by internal standards. How does a local system conform to the Society Standard with a set of requirements.
2. **Project review** – review and evaluation for improvements, supra-system decision, and next steps.

11.10.4 The evaluation process

The following is a common example of an evaluation process:

1. Entry criteria of the process
Entry criteria (I_{en}) is the minimally acceptable inputs in order to perform the process. The values for this indicator are as follows:

- $V^{pv}{}_{en}$ = the number of the inputs required by this process.
- $V^{av}{}_{en}$ = the number of the inputs finished at this moment.
- $V^{ev}{}_{en}$ = the number of budgeted inputs performed.

For example, the "Human Resource Plan Process" of PMBoK has four inputs: "project management plan", "activity resource requirements", "enterprise environmental factors" and "organizational process assets". If at the calculating moment, four inputs should be finished but only "project management plan" is finished, and the number of the budgeted inputs performed is 2. Then the three values are:

- $V^{pv}{}_{en} = 4$
- $V^{av}{}_{en} = 1$
- $V^{ev}{}_{en} = 2$

2. Cost of the process (Market only)
Cost (I_c) is the money allocated to the process. It will be used to evaluate if the process is over or under budget. The values for this indicator are as follows:

- $V^{pv}{}_{c}$ = the planned value of the cost of this process (planned cost).
- $V^{av}{}_{c}$ = the actual value of the cost of this process (actual cost).
- $V^{ev}{}_{c}$ = the budgeted cost of this process performed.

For example, a process had a cost allocation of 10 money. If the cost that has been spent for the current moment is 5 money, and the budgeted cost of the finished work of this process is 3 money, then the three values are:

- $V^{pv}{}_{c} = 10$
- $V^{av}{}_{c} = 5$
- $V^{ev}{}_{c} = 3$

3. Duration of the process
Duration (I_d) is the money allocated to the process. It will be used to evaluate if the process is behind or ahead of schedule. The values for this indicator are as follows:

- $V^{pv}{}_{d}$ = the planned value of time required of this process.
- $V^{av}{}_{d}$ = the actual value of time spent on this process.
- $V^{ev}{}_{d}$ = the budgeted time of this process performed.

For example, for a process requiring 400 hours of work, the time spent at the current moment is 200 hours and the budgeted time of the finished work is 300 hours. Then the three values for this indicator are:

- $V^{pv}{}_{d} = 400$
- $V^{av}{}_{d} = 200$
- $V^{ev}{}_{d} = 300$

4. Resource of the process
Resource (I_r) is the resources allocated to the process. It will be used to evaluate if the resource usage is over or under capacity. The values for this indicator are as

follows:

- $V^{pv}_{\mathbf{r}}$ = the planned value of resource required of this process.
- $V^{av}_{\mathbf{r}}$ = the actual value of resource spent on this process.
- $V^{ev}_{\mathbf{r}}$ = the budgeted resource of this process performed.

For example, here the resource allocated to this process is 11 bricks; the resource assigned to this process is 9 bricks at the moment, and the budgeted resource of the finished work of this process is 8 bricks. Then the three values for this indicator are:

- $V^{pv}_{\mathbf{r}} = 11$
- $V^{av}_{\mathbf{r}} = 9$
- $V^{ev}_{\mathbf{r}} = 8$

5. Expected criteria of the process
Expected criteria ($I_{\mathbf{ex}}$) is the minimally acceptable outputs in order to perform the next process. The values for this indicator are as follows:

- $V^{pv}_{\mathbf{ex}}$ = the number of the outputs required by this process.
- $V^{av}_{\mathbf{ex}}$ = the number of the outputs finished at this moment.
- $V^{ev}_{\mathbf{ex}}$ = the number of budgeted outputs performed.

For example, the process "Quality Plan Process" of PMBoK has five: "quality management plan", "process improvement plan", "quality metrics", "quality checklists" and "project documents updates" expected outputs. If at this moment, two inputs should be finished but there is only "quality management plan" is finished, the number of the budgeted outputs performed is 1. Then the three values for this indicator are:

- $V^{pv}_{\mathbf{ex}} = 2$
- $V^{av}_{\mathbf{ex}} = 1$
- $V^{ev}_{\mathbf{ex}} = 1$

11.11 [Decision] Evaluator

An evaluator is a statistical tool for comparison to provide analysis of a metrics, performance indicators and risk indicators, to explore trends, data patterns, and interdependencies for informing optimal decisioning, and ultimately the achievement of intended results. Here, process analytics are applied to indicators to produce actionable decisioning information.

> **QUESTION:** *What is the actual progress against goals.*

In general, the first analytic process is the determination of the 'base rate' [of a change]. The **base rate** is a statistical measure of what percentage of a population has a particular characteristic. This statistic is then used as the base (or prior probability) upon which to compare other measurements.

Determination involves analytics (assessment):

1. Assessment and analytic techniques provide the mechanism for measuring and evaluating the defined factors to evaluate progress and impact.
   A. **Performance assessment** - determine current and future performance by identifying performance indicators and measuring them over time.
   B. **Risk assessment** - determine current and future risks by identifying performance indicators and measuring them over time.

The primary three system change control evaluators are input, process, and output:

1. **Input indicators/metrics (Project indicators/ metrics, Project control)** - measures of the project, used to monitor and control the project. Through continuous monitoring and control: the development space may be minimized by making adjustments necessary to optimize and avoid problems; and product (service) quality can be assessed (evaluated) on an ongoing basis, and the technical approach modified to improve quality. Project control determines the targets of 'resources' and 'timing' in a project:
   A. Resources (cost) - alignment with required resources usage.
   B. Timing (schedule) - alignment with required timing.
2. **Process indicators/metrics (Project coordination)** - measures of the development process. Process metrics are collected across all project (forever), and provide indicators that lead to long-term process improvement. Process metrics reference/measure attributes of a process (people, environment, tools, techniques).
   A. Overall development time (% complete).
   B. Type of methodology used.
   C. Work products delivered (productivity, work delivered).
   D. Human effort expended.
   E. Errors uncovered before release.
   F. Calendar time expended (% on-time delivery).
   G. Conformance to schedule.
3. **Output indicators/metrics (Product indicators/ metrics, Service distribution)**
   A. Usage/productivity rate.
   B. Defect rate - Defects delivered to and reported by end-users.

C. Change request rate.
D. Resource usage / schedule variance.
E. Quality - alignment with required quality.
F. Effects - alignment with [required] goals and objective (effect).

Table 25. Decision Approach > Monitoring & Evaluation*:*
Table of project coordination metrics.

Indicator Category	Metrics
Productivity	The number of (lines of code, modules, classes, deliverables, etc.) developed on time unit or per resource.
Quality	The degree of completion of project objectives.
Deliverables	The ratio between the achieved deliverables and the planned deliverables. The number of rewords because of no concordances between the specifications and the results.
Resources	Statistics regarding resource usage. Statistics regarding resource costs. Statistics regarding resources loading and distribution.
Risks	The number of identified risks. The number of raised risks. The number of avoided risks.

Control metrics are classified based on their roles, importance, and functionality:

1. **Roles (evaluation)**
   A. Forecasting (a.k.a., predicting) - predicting project resources (cost) and timing (schedule) outcomes based on the current understanding of project progress and performance.
   B. Diagnostic - signalling progress and performance issues to inform corrective actions.
2. **Importance (prioritization)**
   A. Priority (core) - "must have" metrics that provide the greatest insight into project controls (resources and timing).
   B. Significant - supplement or complement Core metrics as needed.
3. **Functionality (application)**
   A. Data (data collection).
   B. Information (progress measurement).
   C. Knowledge (performance assessment).
   D. Insight (performance forecasting).

12 [Decision] Quality indication

Quality ([high-level] "management") engineering indicators (quality assurance, performance indicators)

Quality is conformance to requirements. Quality is the totality of features and characteristics of a product, or service that influence it's ability to satisfy stated or implied needs. Fully satisfy user ("customer") requirements at the lowest resource usage. In engineering, service quality is now measured with performance-based measures. Quality is indicated by a source of/for feedback in order to re-orient the next state by controlling the adaptation.

High-level quality indicators are the quality and/or performance requirements, which are assessed through evaluation of a system's materializing/-ed alignment with it's [user] requirements (as validation and verification).

Quality is evaluated through feedback types:

1. **Metrics (an objective's criteria)** - provide ways of measuring each stated quality (objective). There may be multiple metric for each quality. At the level of systems engineering, metrics are measurable requirements -- requirements with an objective and/or subjective measure of progress or completion.
2. **Weightings (ranking)** - define the relative importance of different qualities in a particular problem environment.
3. **Strategies** - are methods for sustaining and/or improving the current quality and/or progress.

Project metrics are used, in part, to improve quality:

1. As quality improves, defects are minimized. A defect is a verified lack of conformance to requirements.
2. As defects go down, the amount of rework required during the project is also reduced.
   A. As rework goes down, the overall project input (e.g., time, resources, cost) is reduced.

The three service quality indicators (factors) are:

1. **Service/product operation (system operation):** its operational characteristics (do they align, meet requirements, meet metrics); its operational characteristics.
2. **Service/product revision (system revision):** its ability to undergo change.
3. **Service/product transition (system transition):** its adaptability to new environments.

The system quality functions (factors/indicators) are:

1. **Functionality** - the degree to which the system satisfies needs.

2. **Reliability** - the amount of time the system is available for use.
3. **Usability** - the degree to which the system is easy to use.
4. **Efficiency (optimality)** - the degree to which the system uses system resources optimally.
5. **Maintainability** - the ease with which the system may be repaired and enhanced.
6. **Portability** - the ease with which the system can be transposed from one environment to another.

System quality performance inquiries:

1. What are the results of task or test execution?
2. What are the results of their timings?
3. What are the results of the comparisons and calculation of all data their timing(s).

High-level project-coordinator indicators of quality include [the performance of tasks]:

1. The 'performance' [of a service system], in the context of an organizational outcome, can be measured (and have its quality determined) by [calculating] the number of tasks/projects closed.
2. Is the number of tasks of another, related project, becoming sufficiently overwhelming that that system is flagging an alert (leading indicator)?
3. Within the last # of days, how many tasks were not closed as expected? What is the user/requirements accessibility threshold for the closure of expected tasks (a lagging indicator)?

Quality can be simplified by measuring:

1. Quality is specification driven – does it meet the set requirements
2. Quality is measured at start of life – percent passing customer acceptance.
3. Quality is observable by number of rejects from customers.

> **NOTE:** *The quality characteristics of a service or product (functional object) are known as the 'Determinants of Quality'.*

12.1 Indicator(s) determinants of service quality

In the market, there are different theories of determinants of service quality, generally, satisfiers and dissatisfied with the following definitions:

1. **Satisfaction** refers to the outcome of individual service transactions and the overall service encounter,
2. **Service quality** is the customer's overall impression of the relative inferiority/superiority of the organization and its services.

In the market there is an expectation-perception gap view of service quality (i.e., customer expectation and perception). There is business [management's] perception and business [management's] expectation, and there is the customer's equivalent. Therein exists "the zone of tolerance, a range of service performance that a customer considers satisfactory". The importance of the zone of tolerance is that customers may accept variation within a range of performance, and any increase in performance within this area will only have a marginal effect on customer perceptions. Only when performance moves outside of this range will it have any real effect on perceived service quality. There sets up a desire to conceal real quality on the part of the business-service provider.

The following are how service quality was best understood in the literature circa 1995 (Johnston, 1995):

Parasuraman et al. (1985) provided a list of ten determinants of market-based service quality as a result of their focus group studies with service providers and customers:

1. Access.
2. Communication.
3. Competence.
4. Courtesy.
5. Credibility.
6. Reliability.
7. Responsiveness.
8. Security.
9. Understanding.
10. Tangibles.

Johnston and Silvestro (1990) suggested a refined list of 12:

1. Access.
2. Appearance/aesthetics.
3. Availability.
4. Cleanliness/tidiness.
5. Comfort.
6. Communication.
7. Competence.
8. Courtesy.
9. Friendliness.
10. Reliability.
11. Responsiveness.
12. Security.

Johnston and Silvestro (1990) went on to add the customer's perspective to the 12 service quality characteristics. They identified five "customer" service quality determinants:

1. Attentiveness/helpfulness.
2. Care.
3. Commitment.
4. Functionality.
5. Integrity.

Walker (1990) suggested that the key determinants are:

1. Product reliability.
2. A quality environment.
3. Delivery systems that work together with good personal service (staff attitude, knowledge and skills)

Grönroos (1990) postulated six criteria of perceived good service quality:

1. Professionalism and skills.
2. Attitudes and behaviour.
3. Accessibility and flexibility.
4. Reliability and trustworthiness.
5. Recovery.
6. Reputation and credibility.

Albrecht and Zemke (1985) suggested:

1. Care and concern.
2. Spontaneity.
3. Problem solving.
4. Recovery.

Armistead (1990) split the dimensions into “firm” and “soft”:

1. The firm dimensions are time (including availability, waiting time and responsiveness), fault freeness (including physical items, information and advice) and flexibility (ability to recover from mistakes, to customize the service or add additional services).
2. The soft dimensions are style (attitude of staff, accessibility of staff and ambience), steering (the degree to which customers feel in control of their own destiny) and safety (trust, security and confidentiality).

Essentially, there are several emotional and physical determinants users apply when evaluating (the satisfaction, fulfillment, etc., of) their experience. Generally, these include,

1. Accessibility.
2. Service.
3. Expectations.
4. Communication.
5. Competence.
6. Courtesy.
7. Credibility.
8. Reliability.
9. Responsiveness.
10. Product or service attributes (the tangible characteristics of a product or service, for example, if acquiring a car, its size, colour, shape and engine size).

The names of the determinants of service quality do not distinguish between the effect of the determinants in terms its creation of satisfaction or dissatisfaction in a service user. It is implicitly assumed that they are the two aspects of the same conception. For example, reliability was Berry et al.’s (1985) most important factor, which implies that unreliability will lead to dissatisfaction and that reliability will lead to satisfaction.

In community, there are only contributors, who are themselves the users. There are two socio-economic identities in community: the user and the InterSystem Team. In the market, there are at least three: the employer, the employee, and the customer. The competition for access that such a system sets up is likely to leads to significant diversions from real world understanding and fulfillment. Because, individuals of the same society are competing for access [to some thing] through a set of relationships based upon power over others, and not access cooperation through a perception of common heritage. Guest/customer satisfaction (dissatisfaction) has meaning in the market, but community has only users who contribute, there are no economic guests or customers. Hence, when the real world is more greatly considered, then creation and awareness more based in the real world where humans have needs that are fulfilled from particular organizations of the environment.

In community, there is no business (i.e., no monetary) gap between access to fulfillment (i.e., the “customer”, or user) and the production of fulfillment services and goods (i.e., the “employer and employee”, or InterSystem Team contributor). In the market there are a number of different sub-gaps, including the real world knowledge gap (engineering), then, the policy gap, the delivery gap, the communication gap, the customer gap, etc. In the market, there is a customer gap, as the difference between expectations and perceptions; there are also provider gaps:

1. **Not knowing** what customers expect > ensure what customers expect, via research and analysis (or, user input, as well as, research and analysis).
2. **Not selecting** the right service designs and standards > establish the right service quality standards, via management (or, decision analysis).
3. **Not delivering** to service standards > ensure that service performance meets standards, via employees (or contribution to the InterSystem Team).
- **Not matching** performance to promises > ensure that delivery matches promises (of the enterprise).

But, when all of the information is present because the service is designed to directly fulfill needs, then there is no provider gap.

Note: In engineering, service quality is now measured with performance-based measures.

In the market, there are customer expectations and perceptions. Customer expectations are the beliefs and assumptions of what an organisation's products, services and all-round customer service will be like. Customer perceptions are how consumers feel and regard an organisation's product or service after purchasing their product and using it first-hand. The company has a perception of the consumer's expectations, and the customer has a degree of expected service, and there are provider "gaps" in between. In community, however, service is derived from a open, transparent, and unified model, and hence, expectations become based upon the societal information system itself, and not on any specific business or industry (as in the case of the market).

1. Accessible time.
2. Accessible space.
3. Accessible services.

12.1.1 Validity (quality of information)

The term validity is used (by researchers) to characterize the degree to which information reflects the phenomenon being studied.

12.1.2 Reliability (trustability-testability of information)

Reliability, or the extent to which information is "trustworthy", can in principle be tested. It is "ensured" when indicators are unambiguous or measurements have no systematic errors. To test the data for reliability, several people independently using the same indicator for the same problem should obtain the same result. Sources and methods of acquiring information are decisive (in order to ensure reliable information).

INSIGHT: *You can trust other people to do research and discovery for you, if they are following a transparent method and if their arguments are sound, free of bias and transparent.*

There are two ways to ensure higher validity information:

1. By choosing indicators that provide the most direct measure, and
2. By using several indicators that together comprise a good indication of the phenomenon describe[d by the indication].

NOTE: *Unambiguity (Read: clarity and precision) is a precondition for dependable information, for intelligence in action.*

12.2 False quality indicators (false indication)

There are also false indicators of quality (i.e., indicators that appear to stand on their own as a representation of quality, but require a larger context to be integrated). For example, 'total lifetime' in age (cycles around sun) is not an indicator of 'life quality'. Similarly, 'age at death' is not an indicator of the quality of the life. Here, what it *means*, or *is*, to be alive needs to be defined. As a definition, what does it *mean* and/or *require* to be 'alive'? Life is something that needs to be measured with a matrix that is more comprehensive and nuanced than minutes or years (time) or currency (market). Life is something that needs to be measured with a matrix that is more comprehensive and nuanced than minutes or years or currency.

A false indicator by itself:

- Total lifetime in age is not an indicator of life quality -- age of death is not an indicator of the quality of the life.

12.3 Requirements quality indicators

A.k.a., Requirements traceability ensures reliability.

The quality indicators of a requirements statements include, but are not limited to:

1. **Imperatives** - Command words (e.g., shall, must, is required to, are applicable to, should).
2. **Directives** - Words are often used to make requirements more understandable (e.g., for example, figure, table, note).
3. **Continuances** - Words that introduce more detailed specification (e.g., below, as follows, following, listed, in particular, support, essential, fundamental).
4. **Options** - words that allowing the developer latitude in implementing a requirement. this introduces risks to schedule and resources.
5. **Weak phrases** - Words and phrases that introduce uncertainty into requirements statements (as appropriate, as preferred, as possible, customizable).

12.4 Access derived quality control indicators

The meaning of access can be derived from various societal perspectives. Here, the relevant perspective is that of systems engineering. In an engineered system, 'access' is derived through/from the intentionality of the system's user. At a population level, users will (for fulfillment or not) determine the meaning of 'access', from which the meanings of 'stewardship' and 'quality' are similarly inter-defined (or left excluded):

1. Determination of meaning of 'access' (i.e., determining relationship of individual to societal access).
    A. *Access* to product (in service resources).
    B. *Stewardship* of product (in service resources).
    C. *Quality* of product (in service resources).

12.5 Measuring quality

Quality is a multivariate measurable, which generally includes:

1. **Correctness** - degree to which a system operates according to specification.
    A. For example, verified non-conformance with requirements.
    B. For example, defects per KLOC.
2. **Maintainability** - the degree to which a system is amenable to change and lifespan.
    A. For example, mean-time-to-change (MMTC) - given an incoming change requirement, what is the time to analyze, design, implement, and deploy a change.
3. **Integrity** - the degree to which a system is impervious to outside attack, environmental instability, or failure.
    A. Threat probability and security (likelihood of repelling an attack).
        1. t=likelihood of threat occurring and S=likelihood of repelling the attack
        2. Integrity = sigma [1 - (threat x (1 - security))].
        3. t=0.25, S=> I=0.99
4. **Usability** - the degree to which a system is easy to use.

13 [Decision] Measurement

DEFINITION: *A mathematical model consists of one or more equations, in-equations, and objective functions and it has a role to describe the associated state. The metrics measure the project, service, or product characteristics based on the characteristic's influencing factors.*

'Metric space', also 'measure space', is the conception of distance in the real time (line).

Definition 1.1 A metric space is given by a set X and distance function d : X x X → R, such that

1. *(Positivity) For all x, y make X*
    - 0 ≤ d(x,y)
2. *(Non-degenerated) For all x, y make X*
    - 0 = d(x,y) set equal to x = y
3. *(Symmetry For all x, y make X*
    - d(x,y) = d(x,y)
4. *(Triangle inequality) For all x, y, z make X*
    - d(x,y) ≤ d(x,z) + d(z,y)

At level 4, a triangle can now be (a metric space construction:

1. *Found* for orientation (in real time).
2. *Formed* for construction (in real time).

The construction of a metric space allows for the following ability functions:

1. Measurement (measurability, ==, set equal to).
2. Comparison (comparability, !=, not equal to).
3. Analysis (decomposability, <, <=, less than and/or equal to).
4. Synthesis (composability, >, >=, greater than and/or equal to).
5. Estimation (probability).
6. Verification (verifiability, equatability, boolean, =, is equal to).

13.1 Indicators and metrics fundamentals

The following are the essential operational elements of an intentionally indicated measurement and its application:

1. **Indicator** - Indicators are categories (meaningful concepts associable to a context). There are many different sub-types of indicators.
2. **Metric/measure** - Metrics are the description of the variable that the indicator is expressing in some alignment, and the value itself in numerical form. Metrics that express a goal are sometimes called "targets".
3. **Statistics** - It is upon the values themselves that

statistical mathematics are run (computed).

4. **Parameter** - Parameters are what the values should or can be between in relationship to a single variable indicator; the range of acceptable or available values (important note: the word, "parameter", has other definitions applicable elsewhere, for instance, a parameter may not be defined as a range of values, but instead, a single parametric value itself). Note that in the decision system, user customizability would be considered a contextually available parameter; where, users could customize available variables within a set range of parameters or available values (or set, set the parameter).
5. **Threshold** - Thresholds are the minimums and maximums, which may act as limits and/or triggers (for events). Note that in the decision system, thresholds are used to resolve many supra-economic decision inquires.

13.2 Mathematical metric construction

A mathematical measure is a function that assigns a non-negative real number (or, +infinity) to (certain) subsets of a set X. As defined in [IVAN04] a metric represents a mathematical model (function model) developed around an equation having the following form[ed construction]:

1. The identity function: f(x) = x
   A. The identify function allows graphing.
2. The indicator function: y = f(x)
   A. The indicator function allows estimating probability.

13.3 'Metric' from a mathematical perspective

In mathematics, in part, a metric is a real [time line] function that measures the distance between two coordinated entities. Mathematically, a metric is a measure between two itens in a set. One mathematical definition of a metric is: Let A be a set of objects, let R be the set of real numbers, and let μ be a one-to-one function such that $\mu: A \oslash A \rightarrow R$, where ⊘ denotes the Cartesian product of A with A. Then, μ is a metric for A if and only if:

- $\forall \alpha, \beta \in A: \mu(\alpha, \beta) \geq 0$; (P1)
- $\forall \alpha, \beta \in A: \alpha = \beta \Rightarrow \mu(\alpha, \beta) = 0$; (P2)
- $\forall \alpha, \beta \in A: \mu(\alpha, \beta) = \mu(\beta, \alpha)$; and (P3)
- $\forall \alpha, \beta, \gamma \in A: \mu(\alpha, \gamma) \leq \mu(\alpha, \beta) + \mu(\beta, \gamma)$. (P4)

13.1 Simplified definition of 'metric'

A metric is a quantitative measure of the degree to which a system, component or process possesses a given attribute. The simple definition of metric containing the following two sub-characterizations is insufficient for computer processing:

1. A metric is a standard of measurement in comparison.
2. A metric is a function that describes distances between pairs of points in a space.

13.1.1 [Decision] Metrics classifications

The metric shape of that which can be classified as having shape.

The concept of a 'metric' carries two related meanings:

1. **A scientific/discovery meaning:** A 'metric' is defined to measure distance between two linear systems (real to real and real to abstract, where engineering is abstract to material). This type of metric refers to scientific metrics (scientific measurements in context).
2. **An engineering/evaluative meaning:** A measure (metric) is the objective allocation of a value to an entity, in order to characterise a specific feature. This type of metric refers to project metrics (project-program-process-quality evaluation).

Thus, there are two types of metrics, project metrics and scientific metrics.

1. A **scientific metric** is any scientific measure with context.
   A. Measures degree of alignment with the real world.
2. A **project metric** is a quantitative measure of the degree to which a system (engineered or not), component or process possesses an attribute.
   A. Measures degree of alignment with a direction (intention).

Defining the metrics for projects consists of building models and indicators (of occurrences) that start from values measured objectively with numbers (values).

13.1.1.1 Quantitative and qualitative metrics

Quantitative metrics are considered those that are based on factors that can be measured or counted. Such metrics include, but are not limited to: work productivity, project value, resource usage, costs, etc.

For example, <u>work productivity based on inputs</u> (is computed as):

- $W1 = ({}^{n}\sum_{i=1} O_i) / ({}^{m}\sum_{j=1} I_j)$
- Where,
- O_i = the output i (deliverables, results)

- Ij = the input j (human effort, resources per time unit)
- n = the number of outputs
- m = the number of inputs

Work productivity based on time:

- $Ws = ({}^{n}\sum_{i=1} O_i) / T$
- Where,
- T = time period

For example, a given project portfolio value at a gen moment in time (is computed as):

- $PPV^s (t) = {}^{ks}\sum_{i=1} VP^s{}_i (t)$
- Where,
- PPV^s (t) - project portfolio s value at a given moment, t
- $VP^s{}_i$ - the value of project i from the portfolio s
- ^{ks} is (k^s) - the number of projects in the portfolio s

For example, the degree of resource loading for a portfolio of projects (is computed as):

- $LD = ({}^{s}\sum_{j=1} UR_j) / ({}^{t}\sum_{i=1} RR_i)$
- Where,
- UR_j - the number of resources involved in the project s
- RR_i - total number of required resources for project s

For example, the degree of resoure usage at a gien moment in time (is computed as):

- DU (t) = NR(t) / TR
- Where,
- NR - the number of resources involved in a project
- TR - the total number of resources available

For example, the cost of resources per some other unit, such as energy (is computed as):

- $C = {}^{w}\sum_{i=1} NR_i\, d_i\, p_i$
- Where,
- NR_i - the umber of resources from the category i
- pi - energy per unit for the resource category i
- d_i - units of usage for the resource category i

For example, level of complexity, which assumes a project as a basis of comparison (is computed as):

- $C = ({}^{k}\sum_{i=1} r_i) \log_2 r_i$
- Where, k - the number of tasks in the project
- r - the number of unique resource types involved in the project.

Qualitative "metrics" are formal[ly meaningful] answers. For example, Why did something happen (with linguistic reasoning)? What is the source, cause, or influence of something? These are not expressed as indicators. The first three qualitative metrics are: quality of work, team cohesion, and degree of satisfaction. The two qualitative contexts are social abilities and personal experience. Note that social abilities depend on the communication skills and knowledge, which could/should be quantified.

For example, the degree of satisfaction (can be computed):

- $DS = ({}^{p}\sum_{i=1} DSR_i) / TR$
- Where,
- DSR = the degree of satisfaction for the requirement i.
- TR = total number of requirements.
- p = the number of requirements
- The degree of satisfaction for a user of requirement is a value from 0 (no satisfaction) to 1 (fully satisfied).

13.2 Metrics service-level overview

Every constructed system has the following initial metrics:

1. **Functionality delivered** - provides an indirect measure of the functionality of the system
2. **System size** - measures the overall size of the system defined in terms of information available as part of the analysis model.
3. **Specification quality** - provides an indication of the specificity and completeness of a requirements specification.

13.2.1 Service metrics

The following are a high-level list of metrics for indicating the presence of a service [type].

1. **Architectural metrics** - provide an indication of the quality of the architectural design.
2. **Component-level metrics** - measure the complexity of system components and other characteristics that have a relevance to quality.
3. **Interface design metrics** - primarily focused on usability
4. **Specialized object-oriented design metrics** - measure characteristics of classes and their communication and collaboration characteristics.
5. **Complexity metrics** - measure the logical complexity of a system.
6. **Length metrics** - measure the amount between.

13.3 A metric indicator random variable

A.k.a., The indicator function, characteristic function.

The set of possibilities is the *sample space*, in which each possibility is an *outcome*. A proposition (a set of possibilities) is an *event* in statistical usage. The indicator function (a.k.a., characteristic function) of a proposition is its *indicator function*. Indicator functions are a type of random variables. Any function defined point-wise on the sample space is a random variable. By convention, its range is usually the set of real numbers or subset thereof, such as {0,1}. Generally, random variables are continuous (there is no substantial difference between the discrete and the continuous type with respect to this context. Herein, conditionals denote random variables, and the probability of a conditional is the expectation of its values. Probabilities are defined in terms of the expectation of the assignment function.

Essentially, an indicator function links expectations and the probability of that event/result occurring, which would be represented by an indicated variable.

For random variables X,Y, expectations and conditional expectations are defined as follows:

- **X = f(x)**
- $E[X] = \sum_{x \in range(X)} x \cdot Pr(X=x)$
- $E[X|Y=y] = \sum_{x \in range(X)} x \cdot Pr(X=x|Y=y)$
- where,
- E[X] = the expectation of X

INDICATOR RANDOM VARIABLE - A random variable that has the value 1 ("true") or 0 ("false"), according to whether a specified event occurs, or not.

- For example, X is an indicator random variable for the Event A, where p denotes P(A).
- If: E(X) = p
- Then: Var(X) = p(1-p)
- The derivation: EX = 1 • P(X= 1) + 0 • P(X=0) = P(X=1) = P(A) = p

There is an expressible relationship between the taking of an expectation of an indicator random variable, and the probability of that particular event occurring (as represented by an indicator random variable). To "take an expectation" is to set an event value; whereupon, there may be calculated a probability for the event value occurring. The indicator function of an event takes on (associates) a value of 1 when an event occurs (true), and 0 when an event does not occur (false).

The indicator function is a function that returns the value 1 when something is true:

- $1[A] = \begin{cases} 1, & \text{A is true,} \\ 0, & \text{A is false.} \end{cases}$

As the name implies, an indicator random variable indicates something:

1. Either a value of '1' when the event happens, or if expression is true.
   A. For example, the value of I_A is 1, when the event occurs.
   B. Where, I is the random variable assigned to the occurrence of an event A.
2. Or, a value of '0' when the event does not happen, or if the expression is false.
   A. For example, the value of I_A is 0, when the event does not occur (that is, A^c occurs).

Thus, I_A is a Boolean variable that indicates the occurrence of the event A. This Boolean variable has value 1 with probability P(A) and so its average value is P(A). Over time, I_A will have value 1 on N • P(A) of N [trials of an experiment, for example].

The indicator random variable method involves randomness in two ways:

1. The variable assigned is random. 'I' in I_A for example.
2. The intentional agent cannot be sure whether the next time I_A is check, that the variable I will have value 1 or 0.

The expectation is the same thing as computing the expected value of the variable: the value 1 times the probability that A is true, plus the value 0 times the probability it is not.

In application, indicator random variable is a method to convert between probabilities and expectations.

- For example,
- x{f} *(set f is given the indicator random variable x)*
- x{f} = 1 if f occurs || 0 if f does not occur

A density (distance) function may be expressed for a random variable:

- For a continuous random variable:
  - $E(X) = \int_{-\infty}^{\infty} x f(x)\, dx$
  - The expectation of X is the integral from negative infinity to infinity of x f(x) dx.
  - $f(x) \geq 0$
  - $\int_{-\infty}^{\infty} f(x)dx = 1$
- For a discrete random variable (replace integration, ∫, with summation, ∑)
  - $E(X) = \sum_{x \in S} x\, f(x)$

Whereupon, the expectation of a discrete random variable is defined as the sum over all type of values which that random variable can take multiplied by the probability of that particular value occurring. The indicator function is:

- $E(x) = \sum_x n P(x=n)$
- Where,
- $\sum_x$ - sum of all values which the random variable x can take
- P(x=n) - probability of that value x occurring

When the outcome is a continuous number, then a continuous random variable is expected. Examples of random variable are weight and height.

Probabilities are specified over an interval to derive probability values:

- $P(a < X < b) = \int^b_a f(x)dx$
- Where, the probability of taking on a single value is 0

13.3.1 The resolution of a metric space, boolean

Boolean expressions use relational and logical operators that result in either a 0 (true) or 1 (false). Boolean expressions allow for the existence of an instruction (programs) that decide whether to execute code (a decision). Code is a set of rules, that when an input (of energy) is applied, information is processed and a result is produced.

13.3.2 Arithmetic (counting expression) operators:

The arithmetic operators are:

- **Operator & Name**
- + addition
- - subtraction
- * multiplication
- / division
- % modulo (remainder)

13.3.3 Assignment (expression) expressions (operators):

The assignment operators are:

- **Operator & Name**
- = set equal to
- += set greater than
- -= set less than
- *= set multiplicator
- /= set divider
- %= set modulo (set remainder)

13.3.4 Relational operators

Determine the relative ordering between values. Relational operators may be used to compare expressions that evaluate numeric and character data.

The relational operators are:

- **Operator & Name**
- == equal to
- != not equal to
- < less than
- ≤ less than or equal to
- > greater than
- ≥ greater than or equal to

13.3.4.1 Logical operators

Combine boolean values and evaluate to a boolean result.

The logical operators are:

- **Operator & Name**
- ! logical NOT
- && logical AND
- || logical OR

13.3.5 Operator precedence

The operator precedence types are:

Table 26. Decision Approach > Measurement: *Counting operators and precedence.*

Operator type	Operator	Associates
grouping	(expression)	left to right
unary	++, --, +, -	right to left
cast	(type)	right to left
multiplicative	*, /, %	left to right
additive	+, -	left to right
assignment	=, +=, -=, *=, /=, %=	

13.4 [Decision] Measurement method

Measurement is the method of producing metrics (i.e., actionable, relational data), and it includes of the following components:

1. **The measurement method (the sense-feedback, observation method)** - The method used to measure something; collection of information.
2. **The measurement (actual) value (The counted value)** - The resulting value obtained from measuring, also called the measure.
3. **The expected value (the predicted value)** - The predicted, intended, or otherwise expected count.
4. **The calculation (the mathematical value)** - the resulting calculation or combined set of measures. In order to determine the degree of alignment, statistical processes are used.

One simple example of the measurement method is, Goal, Question, Metric (GQM). The GQM method's fundamental principle is that the carrying out of the measurement must always be oriented (alignable) towards an objective. GQM defines an objective, refines that objective into questions and defines [mathematically precise] measures that are most probably likely to answer those questions, given what is known. Indicators are then generated to collect and process information for useful synthesis by calculating the separation between a result and that which was probably expected as a metric.

13.4.1 Simplified definition of a 'measure'

A measure provides a quantitative indication of the extent, amount, dimensions, capacity or size of some attribute of a product or process. When a single data point has been collected (e.g., the number of errors uncovered in the review of a single module), a measure has been established. Measurement occurs as the result of the collection of one or more data points (e.g., a number of module reviews are investigated to collect measures of the number of errors found during each review).

Measurement is, by definition, empirical. Measurement, as a collection of information, is knowledge that is derived from observation and/or experimentation. A 'measurement' is the act of determining a 'measure' by counting that which is perceived ("having that information of quantity"). A measure provides a quantitative indication of the extent, amount, dimension, capacity, count, or size of some attribute of a product or process. To measure is to inquire into the sensed unified-separation of the environment through something already known (something standard or common). Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world to describe them according to pre-defined rules. Therein, measurement is the act of determining a measure. Measurement provides data into the way systems *change* and *operate*.

> **NOTE:** *Measurement is a system boundary function. Measurement is the act of determining a measure.*

The term 'measure' is used to mean the 'value' measured (identified and recorded) by whatever mechanism is used. All measures are composed of a value (a number) and a unit of measure. The number provides magnitude for the measure (how much), while the unit gives number meaning (what is measured).

From a data context, a measure is a number or value (and unit) that can be summed and/or averaged, or have other statistical calculation applied, such as distances, durations, temperatures, and weight.

Note that the term, 'measurements', is often used alongside 'dimensions', which are the categories that can be used to segment, filter or group, such as the physical dimensions of length and volume, or the societal dimensions of social, decision, lifestyle, and material organization.

Measure[ment] is a common process to all embodied consciousness, and a necessary process for:

1. **Understanding a system (discovering)** - *associations* in space and time.
2. **Changing a system (decisioning)** - *modifications* in space and time.
   A. **Adapting** - processes to remain resilient.
   B. **Optimizing** - processes to improve functioning.

There are two real world domains of measurement related to the experience of separation by consciousness:

1. **Direct measurement (physical collection)** - the real world occurrence is observed and recorded as a 'measure' (and in context, 'metric'). Some [statistical] factors can be directly measured.
   A. For example, defects uncovered during testing.
2. **Indirect measurement (abstraction level)** - the abstract occurrence is implied to have occurred in the real world by one or more direct measures. Some [statistical] factors can only be measured indirectly
   A. For example, usability or maintainability.

The three process domains of measurement are:

1. **Measurement objectives (inputs)** - intention for measurement.
   A. For example, to know temperature difference between two days.
2. **Measurement process (process itself)**
   A. **Measure** ("measurement record") - Measurement occurrence via a method recording new data.
      1. For example, 30 kelvin is a measure.
   B. **Metric** ("measurement record in context") - New data is contextualized to be used in decisioning, providing orientational information.
      1. For example, 30 kelvin on 24 March.
   C. **Indicator** ("signal")
      1. For example, thermometer system change as indicator of temperature.
3. **Measurement result (output)**
   A. **Directional comparisons ("indicator of objective")** - new decisioning data is statistically processed (compared to itself and/or past data) in order to determine direction.
      1. For example, measure (1) at 30 kelvin on 24 march, and measure (2) at 29 kelvin on 23 march, and their statistical comparison results.

Thee full change cycle involves the following phases:

1. Intending (Objectives).
2. Questioning (Objectives).
3. Measuring (Measurement process).
4. Evaluating (Measurement process).
5. Planning (Measurement process).
6. Forming (Measurement process).
7. Monitoring (Measurement process).
8. Effecting (Results).
9. Evaluating (Results).
10. Cycle repeats.

13.5 Measurement optimizes decisioning

The measurement method enables more optimal decisioning by allowing allows for the estimation of probabilities, and thus, informed decisioning (as decisioning that is capable of orienting correctly toward a direction of alignment).

Measurements a re used in decisioning to:

1. **Form a reference baseline** value (benchmark, base rate) for estimated change (*of* quality/progress).
   A. Is reference value set by prior measurement?
   B. If reference value not set by prior measurement, then synthesize reference value.
2. **Determine if change is necessary** (*for* progress).
3. **Inform a necessary change** (*to* progress).

Measurement objectives include those abilities necessary for decisioning:

1. **Conceivability** - To what (where) did something happen?
2. **Observability** - Did something happen?
3. **Comparability** - To what degree did it happen?
4. **Temporality** - How often is it happening?
5. **Stability** - Is the change sustained, or not; to what degree?
6. **Predictability** - Did that which was expected or predicted to happen actually happen?

These objectives are generally expressed as:

1. The number of...
2. The percent of....
3. The ratio of...
4. The incidence of...
5. The proportion of...
6. The probability of...

13.6 Measurement from a scientific (discovery) perspective

Measurement is the [experimental] process in which, to precisely describe the entities or events in real world, numbers or other symbols are assigned to its attributes by using a given scale and clearly defined rules. The result of the measurement is called measure. A Metric is a quantification of a specific characteristic from an entity in the real world, which can be inferred from a set of attributes.

13.7 Measurement from an engineering (technical) perspective

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.

In engineering, measurement contains information about attributes of entities. An entity is an object (such as a person or a room). Entities are described by the characteristics that are important to distinguish one entity from another. An attribute is a feature or property of an entity. Entities can be:

1. The products (deliverables) generates as outputs and outcomes from the service life cycle, as requirements specifications, documents with design, source code, testing, etc.
2. The project/development environment.
3. The user(s).
4. The events corresponding to the phases if the life cycle or to activities and incidents.

The attributes that can be measured depend on the entity or event considered.

Table 27. Decision Approach > Measurement: *Measurable attributes in relationship to generation entities and events.*

Entity or Event	Measurable Attributes
Requirements Specification	Words, phrases, paragraphs, verbs, adjectives
Block Diagram	Modules, coupling between modules, dependencies

14 [Decision] Tabular database

A tabular systems is an extension of relational databases. A tabular system is a system that can visually specify both conditional data (conditional rules) and unconditional data (unconditional knowledge, data patterns). A tabular system consists of a table with columns labelled by attributes. Any row of such system specifies characteristics of some object defined in the attribute space; it can also define a rule, provided that some attributes refer to preconditions and at least one is a decision attribute.
Tabular systems can encode both facts and rules:

1. Facts provide knowledge that is unconditionally true (given what is known).
2. Rules specify conditional knowledge.

Any tabular system specifies characteristics (knowledge) of certain objects. For some objects this knowledge can be valid, while for other it may be not true.
Tabular rule-based systems may be used to define attributive decision tables or control algorithms.
In computation, there are at least two types of computational systems:

1. Tabular computational - a system that supports tables as a data structure, but not the set of algebraic operators.
2. Physical computational - a system that supports algebraic (geometric) operators as a data structure.

NOTE: *Algebraic relation operators include, but are not limited to: greater than, less then, etc.*

14.7.1 Tabular system usages

The following are usages of a tabular system:

1. A tabular system may be used to perform material yield (e.g., water yield) calculation.
2. A tabular system may be used for recording the characteristics of material yields (e.g., the characteristics of each seedling fruit produced). In other words, material object characteristics can be tabulated (e.g., for a fruit, the following could be recorded: the external characteristics of size, color, skin; the internal characteristics of color of flesh, firmness, texture, grain, juice, degree of acidity or lack of it; and the environmental characteristics of quality, season, and the desirability of the fruit.

14.7.2 Database characteristics

The following are characteristics of a tabular database:

1. 'Minimally relational' is a system that supports tables, access, project, and join operators, but no other relational operators.
2. 'Relationally complete' is a system that supports tables and all of the operators of the relational algebra, and can thus be spatially visualized (as an object in relation to other objects).
3. 'Fully relational' is a system that support all aspects of the model when executing a SELECT command [for a 'solution'], and JOIN command using the SUB-LINK command. More simply, a fully relational system is a system that may be fully realized as a material solution from a selection of material solutions. The SUB-LINKING of relationships occurs between different material solution configurations (in the reference frame, context of, their expected results). Then, designs that match potentials can be JOINED, from which a single is SELECTED ... for EXECUTION by CONTRIBUTION.
4. In classical relational database (RDB) systems all the attribute values must be atomic ones.
   A. Atomic values (a.k.a., single values) are values where single cell contains single value. For example, a violation of a single value per cell would be an RDB with one column in the table named 'Energy', and beneath it there is a cell with two values, 10 and 5, instead of just the one value 10. Additionally, in an atomic value RBD, each cell (record) needs to be unique, and there should not be any repeating groups. Repeating group means a table contains 2 or more values of columns that are closely related. For example, the existence of repeating groups would be an RDB with multiple columns that contain only energy data and have the names 'Energy 1', 'Energy 2', etc. These columns that contain only energy data are repeating groups. The three design coherency requirements for an atomic value RDB are:
      1. User shall eliminate repeating group in individual tables.
      2. User shall create separate tables for each set of related data.
      3. User shall define the primary key for related data.
5. An object is:
   A. Value objects:
      1. Any atomic object:
         i. $o \in C$
         ii. For example: travis, John, 28, 389
      2. Any interval object:
         i. I = [a,b], where a and b are atomic objects belonging to the same ordered set being a subset of C (such as integers and floats)
         ii. For example: ∈[2,5], [17,123], [a,b]
      3. Any sequence object:

i. Q = [o1,o2,...on] (*1,2,n*=subscripts), where o*1*,o*2*,...o*n* are objects.
ii. For example: [1,2,3,4,5,6,7], [2,4,6,8], [1,2,3,5,7], [English, French, Russian]

4. Any set object:
   i. S = {o*1*,o*2*,...o*n*}, where o*1*,o*2*,...o*n* are objects.
   ii. For example: {potato, carrot, tomato}, {john, mary, sue}, {5,1,3,7}

B. Structural (tuple) object:

1. Any tuple object:
   i. O(*a1* : o*1*, *a2* : o*2* :,...,*an* : o*n*), where *a1*,*a2*,... ,*an* ∈ A are distinct attribute names and 0*1*,0*2*,...0*n* are objects
   ii. For example: O*1*(first:travis, second:john, age:28), O*2*(town:London, street:Oxford, number:25), O*3*(languages: [English, French, Russian]), O*4*(cars: {honda, audi, bmw}), and a more complex object of the form O*5*(first:travis, second:john, age:23, O2(town: London, street: Oxford, number:25), children:[james, mary, jane], languages: [English, French, Russian], cars: {honda, audi,bmw}).

14.7.3 What is data

An atomic data item is some piece of information represented in certain accepted language, and:

1. As precise as possible (within the selected language).
2. Meaningful (having some interpretation).
3. Positive (no negation is used).
4. Unconditional.

14.7.4 What is knowledge

Knowledge emerges from data when information is systematically collected, organized, and interpreted, transforming raw facts into meaningful insights and understanding. It is best defined as the theoretical and practical understanding of the computational ability of common-kind. By using different systems approaches and methodologies, data can be collected in quantitative and qualitative form for the purpose of explaining, interpreting, and reflecting on the various aspects of a [societal] system. The sharing of knowledge has the potential to optimize technical interest in the prediction and control of natural and social systems (causal explanation); a practical interest in communication and creation of shared understanding among all individuals in a social systems (practical understanding); and a desire for self-integration to protect them from constraints imposed by power structures (reflection).

An atomic knowledge item is any data item and any more general elementary item of the accepted language, which:

1. May contain variables/sets/intervals/structures (according to the selected language).
2. Meaningful (having some interpretation).
3. Positive or negative.
4. Perhaps conditional.

Data and knowledge can be differentiated by their intended interpretation: a data item (such as attribute value, record, table) is considered to be data if the main intended use of it is to provide static, detailed and precise image of some fragment of real world while a knowledge item (such as fact, simple conjunctive formula, DNF formula, and especially rules) is intended to provide more general knowledge defining universal or local properties of the world. From practical point of view, one can consider data to be the part of knowledge expressed with the finest granularity and unconditional.

If the specification contains variables (e.g. universally quantified, or defining some scope ones) or it is true only under certain conditions (e.g. takes the form of rules, allows for deduction or any other form of inference), then it should be normally considered to be knowledge. However, in the uniform, simplified model proposed in this paper explicit distinction is in fact not necessary. A RDB table would be normally considered as data, but it may be considered as most detailed knowledge as well. On the other hand, tabular system of data templates can be considered as extensional specification of data.

Scholarly references (cited in document)

- Armistead, C.G. (1990). *Service operations strategy: framework for matching the service operations task and the service delivery system*. International Journal of Service Industry Management, Vol. 1 No. 2, 6-17.
- Blackburn, M., & Bone, M. (2016). *Transforming System Engineering through Model-Centric Engineering*. Technical Report SERC-2016-TR-109. Stevens Institute of Technology. https://web.sercuarc.org/documents/technical_reports/1526315380-SERC-2016-TR-109-Transforming_SE_Thru_MBSE_Interim_RT-157_v6.1.pdf
- Johnston, R. (1995). *The determinants of service quality: satisfiers and dissatisfiers*. International Journal of Service Industry Management, Vol 6, No. 5, 53-71. https://pdfs.semanticscholar.org/5923/4dfc81e571232626e15cf2feecf72cc5c2a6.pdf
- Johnston, R., Silvestro, R. (1990). *The determinants of service quality – a customer-based approach*. in The Proceedings of the Decision Science Institute Conference, San Diego, CA, November.
- Johnston, R., Silvestro, R., Fitzgerald, L. and Voss, C. (1990). *Developing the determinants of service quality*. The Proceedings of the 1st International Research Seminar in Service Management, La Londes les Maures, June.
- Newcomer, J.T. (2012). *A New Approach to*

Quantification of Margins and Uncertainties for Physical Simulation Data. Sandia Report: SAND2012-7912. Sandia National Laboratories. https://prod-ng.sandia.gov/techlib-noauth/access-control.cgi/2012/127912.pdf
- Parasuraman, A., Zeithaml, V.A. and Berry, L.L. (1985). *A conceptual model of service quality and implications for future research*. Journal of Marketing, Vol. 49, Fall, pp.41-50.
- Rizzo, D., & Blackburn, M.R. (2015). *Use of Bayesian networks for qualification planning: a predictive analysis framework for a technically complex systems engineering problem*. Procedia Computer Science, 61, pp133-140. https://doi.org/10.1016/j.procs.2015.09.173 | https://www.researchgate.net/publication/283197628_Use_of_Bayesian_Networks_for_Qualification_Planning_A_Predictive_Analysis_Framework_for_a_Technically_Complex_Systems_Engineering_Problem

Scholarly references (non-cited)

- Perny, Patrice & Pirlot, Marc & Tsoukiàs, Alexis. (2013). *Algorithmic Decision Theory Third International Conference, ADT 2013,* Bruxelles, Belgium, November 13-15, 2013, Proceedings. https://doi.org/10.1007/978-3-642-41575-3
- Allaire, B. (2007). *A system of indicators for culture and communications in Quebec. Part One: Conception and concerted development of the Indicators*. http://www.stat.gouv.qc.ca/statistiques/culture/systeme-indicateurs1_an.pdf
- Certain, G., et al. (2011). *The Nature Index: A general framework for synthesizing knowledge on the State of biodiversity*. https://doi.org/10.1371/journal.pone.0018930 | https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0018930
- Cowlagi, R. V., & Saleh, J. H. (2015). *Coordinability and consistency: Application of systems theory to accident causation and prevention*. Journal of Loss Prevention in the Process Industries, 33, 200–212. https://doi.org/10.1016/j.jlp.2014.12.004
- Elfouly, M., et al. (2015). *General Indicator Modeling for Decision Support based on 3D city and landscape models using Model Driven Engineering*. Peer Reviewed Proceedings of Digital Landscape Architecture, Buhmann/Ervin/Pietsch (Eds.), Wichmann Verlag, 2015.
- Heini, O. (2007). *Performance Measurements Designing a Generic Measure and Performance Indicator Model*. Geneva, 2007. https://pdfs.semanticscholar.org/a54a/3605dea5d5cd37c162bc9d9436f047d74fa1.pdf
- Olsina, L., et al. (2004). *Ontology for software metrics and indicators. Journal of Web Engineering*. Vol. 2, No.4 (2004) 262-281. https://pdfs.semanticscholar.org/047e/4cce849f91abdfa1c5c699521f572fe65cf5.pdf
- Vincente, K.J. (2008). *The Human Factor*. Expanding Frontiers of Engineering Journal. 32(4). https://www.nae.edu/19582/Bridge/ExpandingFrontiersofEngineering7308/TheHumanFactor.aspx

Book references (cited in document)

- Grönroos, C. (1990). *Service Management and Marketing.* Lexington Books. Lexington, MA.
- Preece, J., Benyon, D., et al. (1993). *A guide to usability: Human factors in computing*. Harlow, England: Addison-Wesley.
- Russel, S.J., Norvig, P. (2015). *Artificial Intelligence: A Modern Approach*. 3rd ed. Pearson Education India. https://www.pearsonhighered.com/assets/samplechapter/0/1/3/6/0136042597.pdf | https://www.slideshare.net/MohammedRomi/ai-03-solvingproblemsbysearching?next_slideshow=1

Book references (non-cited)

- Salles, M. (2015). *Decision-Making and the Information System, Volume 3*. Wiley & Sons, Inc. https://doi.org/10.1002/9781119102663
- *Systems engineering leading indicators guide*. (2010). Version 2.0. January 20, 2010. INCOSE-TP-2005-001-03. http://www.psmsc.com/downloads/other/seli-guide-rev2-01292010-industry.pdf

Online references (cited in document)

- Acheson, N. (2016). *What is the difference between an algorithm and a protocol, and why does it matter?* Linked In. https://www.linkedin.com/pulse/what-difference-between-algorithm-protocol-why-does-matter-acheson

Online references (non-cited)

- *Reliability & Validity.* Warwick. Accessed December, 2009. http://homepages.warwick.ac.uk/~psrex/Lecture%20W7%20Reliab.ppt
- *Summary of the 2018 Department of Defense Artificial Intelligence Strategy: Harnessing AI to Advance our Security and Prosperity*. (2018). US Department of Defense. https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF
- *zGuide 6: Systems engineering competency framework*. (2010). INCOSE UK. Issue 1. https://incoseonline.org.uk/Program_Files/Publications/zGuides_6.aspx?CatID=Publications

Approach: Standardizing

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: standardization, the standardization approach, standards, unified information production,

Abstract

This article explores the pivotal role of standardization in the transition of society towards a community-based operational model. In any given environment, the expression of function is fundamental, with these functions serving as manifestations of underlying capabilities. It posits that these capabilities can be effectively standardized through the implementation of consistent procedures and technologies. A standard, defined as a commonly agreed-upon knowledge set and/or method of operation, plays a critical role in simplifying life and enhancing the cooperation, interoperability, effectiveness, and efficiency of any repeated interaction. The core purpose of developing and adhering to standards includes: ensuring appropriate performance levels, meeting safety requirements, achieving consistency and repeatability in products, systems, or processes.

The article emphasizes that standards are established beyond the organizational level, incorporating the collective intelligence and efficacy of community-wide working groups. It highlights the process through which societal standards can be developed, suggesting the use of working groups drawn from the community's population to collaboratively outline operational procedures and standards. These societal specification standards not only dictate the operational framework for a community, but also elaborate on the rationale behind these procedures, ensuring a shared understanding across the community.

Graphical Abstract

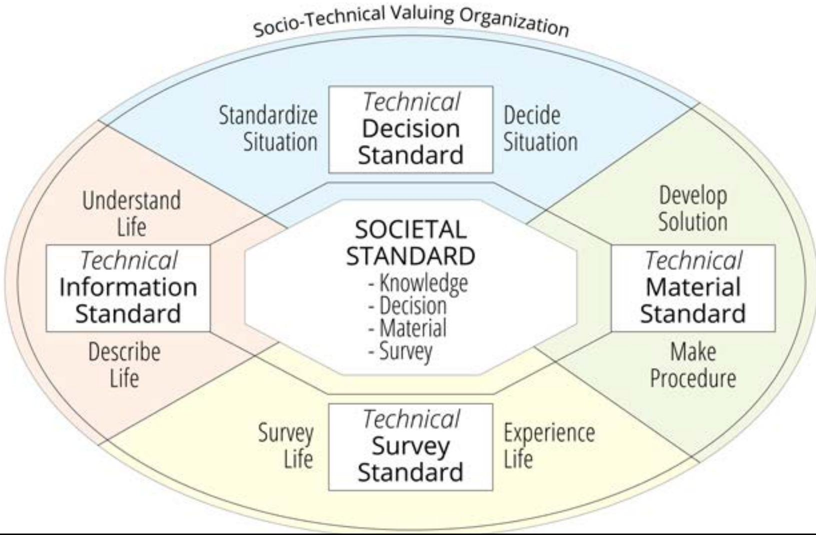


Figure 7. *Standards are a repository of useful knowledge and ensure technical systems are built and operated correctly. Every technologically advanced society uses standards documentation to sustain expected standards of function, quality, and safety.*

1 What is a standard?

A.k.a., Socio-technical standard, integrated knowledge set useful for understanding, creating, and operating anything socio-technical.

A standard is a commonly agreed way of doing something. A standard is any set of technical and/or social specifications that either provide or are intended to provide a common design and/or operation for a product, process, service, or system. Standards not only make life simpler, but are essential in increasing cooperation, interoperability, effectiveness and efficiency of any repeated interaction. The purpose of developing and adhering to standards is to ensure minimum performance, meet safety requirements, make sure that the product/system/process is consistent and repeatable, and provide for interfacing with other standard- compliant equipment (ensure compatibility). Standards are primarily to ensure interoperability and, in matters relating to safety of the product, to ensure that the producer has not overlooked important safety-related design requirements. Fundamentally, standards are functional (useful) documents. Standards specify the characteristics, reasoning, and/or performance requirements of countless aspects of the early 21st century socio-technical world. Standards are an important part of society, serving as rules to measure or evaluate capacity, quantity, content, extent, value and quality. Standards are structured documents that iterate and contribute to a structured body of content that is clear, coherent, precise, unified, non-contradictory, and unambiguous. Working group experts draft and develop the content within technical committees (working groups). Working groups comment and decide on the content. Decisioning is done through protocols and working group consensus voting. Project working group coordinators prepare and publish the final product. Standards are usually developed through discovery and information integration (analysis) by working sub-groups. Generally, as new information is discovered that is relevant to a standard, the standard will integrate the newly given information, and evolve/adapt.

A standard is a process or system that has been agreed upon (formally or informally). Standards emerge when cooperation is present, and when safety is required. Therein exists the necessity for a repeatable process for comparing within a single category some known quantity and/or value with some to be known quantity and/or value. Standards define that which is required to effectively and cooperatively perform some function (or, "do some thing"). Take note, however, that there are multiple kinds of standards, and their meanings may vary slightly; they include, but are in no way limited to: measurement standards, standards of practice, protocol standards, law standards, policy standards, etc.

Standards vary in nomenclature across industries, resulting in numerous distinct terms used to refer to them within specific sectors. Standards can be published in the form of documents that contain:

1. Technical standards (a.k.a., technical specifications).
   A. A technical standard is an established norm or requirement for a repeatable technical task.
2. Social standards (a.k.a., social specifications, behavior and procedural standards).
   A. A social standard is an established norm or requirement for a repeatable behavior, or, not engaging in a behavior.
3. Political standards (a.k.a., policy, legislation, constitutions, legal standards).
4. Rules (a.k.a., code standards).
5. Requirements.
6. Agreements.
7. Guidelines.
8. Procedures and practices (a.k.a., protocol standards).
9. Models.
10. Definitions and explanations.
11. Articles.
12. Etc.

Standards may be called different names in different disciplines and under different applications:

1. Standards that are used for information storage are called '**formats**' (e.g., *information formats*).
2. Standards that are used for transmitting information are called '**protocols**' (e.g., *decision protocols*).
3. Standards that are used for material transformation by humans are called '**procedures**' (e.g., *material procedures*).
4. Standards that are used for material transformation by machines are called '**instructions**' or '**commands**' (e.g., *material commands*).

Technical standards are often documented in a so-called "standard specification" (or, "specification standard") that describes ways to consistently organise information so that it can be understood and used by multiple independent applications and users.

Using a standardized method of expressing information and a standardized way of delivering it cuts out the need to adapt your systems for every organization you intend to do business with. If everyone is using the same standards, communicating data becomes easier and cheaper, ultimately meaning there is more revenue to be distributed across the whole digital supply chain.

In the language of innovation, standards help to harmonize technical specifications of products and services making global materials cycling more efficient, while breaking down barriers to cooperation. Conformity is what the InterSystem Team does. The InterSystem Team conforms the environment to the set societal standard while following other set societal standards.

Standards are used to control processes and outcomes, and establish levels of excellence. The benefits of standardization to the individual, society, and the ecology are many. It is difficult to imagine a world without industry (and industrial) standards. Without standards early 21st century society would not function. Human interaction depends on standards. Human speech is ruled by rules and standards. And, in early 21st century society, human behavior is significantly governed by the standards of manners and laws. We can't live as a society without an agreed upon set of expectations to make our interactions and systems predictable, rational, safe, and stable. The ubiquity of standards indicates that nothing less than 'quality' should be settled for, particularly at the societal level.

Open societal standards are the backbone of a community-type society, ensuring the safety and quality of products and services, facilitating transparency, understanding and improving the environment. Conformity to standards reassures everyone that products, systems and organizations are safe, reliable and good for the community.

Standards and specifications are documents that describe and/or recommend a set of rules and conditions for how materials and products should be manufactured, defined, measured, or tested. Standards are used to establish minimum levels of performance and quality and optimal conditions and procedures for the purpose of ensuring compatibility of products and services from different sources. Specifications tend to have more limited applications than standards and generally establish requirements for materials, products, or services. Standards and specifications may be issued by voluntary technical or trade associations, professional societies, national standards bodies, government agencies, or by international organizations. Standards may be developed by organizations operating at national, regional or international levels, at the market level, the State level, or some combination thereof.

In the context of data, a standard is a technical communications file-document that applies collectively to codes, specifications, recommended practices, classifications, test methods, and guides. Standards represent the integration of multiple sets of data by multiple parties (humans and machines) into an optimal 'standard' data set about a socio-technical topic. Standards are composed in accordance with an established social procedure in order to ensure clear and coherent communications. A standard is like a blueprint; it provides guidance to someone when he or she actually build or operate something. A standard can refer to a level of quality or attainment, or an item or a specification against which all others may be measured. A technical standard is a set of commonly agreed decisions, rules and behaviors, in regard to technical systems; and a social standard is a set of commonly agreed to decisions, rules and behaviors, in regard to individual humans and the organizations in which they participate. A standard establishes common ground that provides means for cooperative development and shared operation.

> **NOTE:** *A practice is a repeatable approach to doing something.*

Standards serves several purposes:

1. Standards describe and explicate a design in a way that makes it duplicable.
2. Standards facilitate communication and shared understanding through a shared set of definitions and visualizations.
3. Standardization improves consistency in understanding and in results.
4. Standards are a set of final/last integrations on a subject matter.
5. By using standards, the end user can be sure a minimum due diligence has been exercised (quality control and assurance).
6. In the case of dispute one can use "following a standard" as a defense.

A standard is a formalization, a record, a log, official information, standardized information, selected information, and an understandable model. In concern to formality, standards are:

1. **Formal**, in that they are officially established, documented, and recognized as authoritative guidelines or benchmarks within a particular field/context.
2. **Semi-formal**, meaning they are not entirely mathematically formal. While math embodies complete formality in data representation, standards may incorporate formal mathematical elements along with other forms of data representation.
3. **Semi-structured,** deviating from complete tabular structure. Tabular data is highly structured, unlike the semi-structured nature of standards, which can encompass formal mathematics, graphs, completed tables, and other forms of structured and unstructured data representations.

There are three modeling language types:

1. **Informal** (e.g., human natural language).
2. **Semi-formal** (e.g., graphical languages such as flow charts).
3. **Formal** (e.g., mathematics).

In the context of a formal document, "formal" is an attribute of information identifying that it is stored in memory (has a documented state), such that the information can be recalled and acted upon at a later time. A formally documented solution that has influenced results in the real world, can have its results assessed (evaluated) at a later date, or even, in-between documented usages.

> **NOTE:** *A formalized structure gives people a method and location from which to work together.*

A formally documented visualization can be expressed in any of the following, and any combination thereof:

1. **Written (text)** - Linguistic directions with an accompanying set of understandings to ensure comprehension.
2. **Diagrams (graphic)** - A diagram is commonly understood as a means to convey information through symbol and figure, and as such, it is used to synthetically represent concept and form.
   A. **Drawings** - diagrams with spatial information.
3. **Simulations and Computer Aided Designs (computation)** - computing spatially dynamic information over processing cycle-time.

Simply, a standard answers the question, commonly, given what is known: How do "I/you/we" know how to do (build or operate) something (read: something material)? A standard is something that should, given the integration of all that is known by "me/we", be followed when doing something. If standards aren't used then design iteration and project intercommunication becomes exceptionally challenging and is more likely to lead to conflict. In a sense, a standard is a protocol, and protocols are how individuals (i.e., we) communicate. When protocols aren't define communication is poor.

Among a material network of integrated habitat service systems, global and local Intersystem teams need to be able to reliably depend upon each other; that other individual Intersystem team members and other local habitat service systems are doing what is expected, following standards and doing the right thing.

> **CLARIFICATION:** *A standard of work is the standard (quality and/or function) at which something is made, built, or operated. Standards ensure quality of service, clear communication, and operations transparency.*

Standards exist for anything that can be materialized and operated, such as software computer hardware, telecommunications, health care, automobiles, aerospace, and many areas of manufacturing. Standards are also employed when we have to ensure that things made by different people will either work together or work in the same way. There are standards that describe the "blueprints" for the plugs and jacks, but the standards themselves are not the actual plugs or jacks. We separate the ideas of "a standard which may be implemented" and "something that is an implementation of a standard."

For clarification, there are different types and sub-types of "standards":

1. In science and engineering, and operations, there are technical and procedural standards.
2. In the State there policies, political standards [delimiting when the violence of the State occurs].
3. In the market, there are contracts, social standards [delimiting when engagement of the State occurs].
4. In a community system, there are decision standards (sets of inquires) that become computable thresholds at an understandable, algorithmic level [delimiting issue prioritization].

Standards often appear complex. To some degree, this is unavoidable. To be useful, standards are details. Standards may specify characteristics or performance levels of products, processes, services, or systems. Humans need standards to generate and operate society together. In other words, humans need standards for information construction, coordination, and materialization if they are to work together at a population scale. Standards are required to meet global human needs and human advancement.

As systems are being designed, new standards issues or need for clarifications may arise. An iterative/adaptive process should be used to incorporate any updates, changes, or clarifications into the standards document and supporting materials.

Humanity has long needed a unified societal standards to realize the intended benefits of standardization and complete effectiveness of community at the global scale. Shared goals and principles embodied in the a societal standard provides motivation and direction. Societal standards establish a basis for collective action so that members of the community can contribute and participate together efficiently. All humans have some fundamental set of mutual goals. Humanity's mutual goals oblige everyone to work pro-actively with one another to further shared technical, social, and individual interests. These goals they commit some of humanity to participating regularly in the critical activities of technical workgroups (for standardization of the habitat support system, life support system, etc.).

A socio-technical standard is an established procedure or requirement for a repeatable technical task. It is usually a formal document that establishes uniform engineering or technical criteria, methods, processes, and practices.

Standards are necessary prerequisites and complement of products, processes, and services. Wherein, standards can:

1. Ensure safe materialization and operation, everywhere.
2. Promote technical efficiency.
3. Foster cooperation and integration.
4. Lower barriers to access.
5. Diffuse new technologies.

6. Protect human health and the environment.
7. Transparently meet human needs/demands.

NOTE: *Unlike in community, in the market-State, standards can be used to disadvantage others.*

In the market-State, standards are hierarchically organized in the following ways:

1. National standards - are federal State created by authorities that dictate territorial-level regulations.
2. Regional/sub-regional standards - are created by local authorities that dictate local regulations.
3. International (e.g., ISO, IEC, ITU) standards - are developed by inter-national standards setting organizations (SSOs; or standards development organizations, SDOs).
4. Associations - are guidelines setting bodies that produce guideline documents and research studies to inform future standards.
5. Companies - have business standards, also known as their standard operating procedures.

In general, going up the hierarchy the standards are more restrictive and less generic, and going down the hierarchy, the standards are less restrictive and less generic.

1.1 Why apply standardization?

Standardization enables, beneficially,

1. The accumulation and integration of knowledge [into more unified/integrated forms].
2. The optimization of states and processes [via the repeatability of the standard process, which allows for the optimization].
3. The collaboration of self-directed entities by the standards processes [producing a value set and behavior conducive to sharing and cooperation].

Thus, resulting in:

1. Increased quality.
2. Increased speed.
3. Reduced effort.
4. Increased safety and control.

1.1.1 Coherence

A.k.a., Alignment.

Coherence (alignment) is important for standards (and therein, humanity and the ecology) in the following way:

1. Internally coherent.
   A. Standards should not conflict.
      1. How do you ensure that standards do not conflict with one another?
         i. Final approval, after working group approval, of the coordinator is necessary, because the role of the coordinator is that of someone who must have shown and maintains an understanding of the whole system (and not just an expert at the sub-group/-team subject).
      2. How do you coordinate superseded and withdrawn decisions, solutions, standards and articles?
         i. Coordinators meet regularly and monitor each other's feeds (where appropriate).
      3. How do you ensure that two groups do not develop different decisions, solutions, standards and articles for the same subject?
         i. Coordinators meet regularly and monitor each other's feeds (where appropriate).
   B. One system, one standard.
      1. Axioms divide systems into separate (but, unified) standards.
2. Externally coherent.
   A. Avoid conflict between standards and operations. Users and habitat service teams apply the standards.
   B. Avoid conflict between users and contributors (i.e., meet real-world human need fulfillment).
   C. Agreement by contributors and users is/feels intuitive.

1.2 Information adaptation

In the early 21st century, standards are developed in a distributed way with many equivalent-level standards present for any given need. Hence, there is a need for developers (of standards) to seek out and embrace opportunities to harmonize standards and adopt the best ideas from the range of standard setters, contributing standards into a commons for the formation of a unified standard for community operations within a societal commons. Collaboration must take precedence over competition, and perpetuation in the commons must take precedence over privatization (commodification/ commercialization).

During the period of uncertainty that will exist until the most appropriate standard(s) become widely adopted, to allow for consistent interpretation by readers (educators and adopters) it is essential that reporting of data (observed alignment) include sufficient detail regarding the underlying methods used. Such transparency will enable the ability to apply an additive and iterative approach to adapting the executed societal specification to a higher potential for human fulfillment, regardless of which standards become more globally accepted. It will also be important to ensure a clear transition from interim standards (a.k.a., transition standards) and the final accepted standards, including methods to determine and present comparative figures for interim results that reflect the final measurement method chosen (i.e., so that appropriate comparison can be made between results of different periods). It is important not to harm needs (i.e., interrupted needed human support services) during transition between standards.

1.3 The specification standard

> *A.k.a., The design standard, the construction and operation standard.*

A specification often refers to a set of documented requirements to be satisfied by a material, design, product, or service. A specification is often a type of technical standard. Specifications are a type of technical standard that may be developed by any of various kinds of organizations.

There are different types of technical or engineering specifications (specs), and the term is used differently in different technical contexts. They often refer to particular documents, and/or particular information within them. The word specification is broadly defined as "to state explicitly or in detail" or "to be specific". Specifications are a specific communication about a system.

Common specifications for systems include, but are not limited to:

1. Requirements specification.
2. Functional specification.
3. Design/product specification.
4. Construction, assembly, disassembly specification.
5. In-service, maintained as, operations specification.
6. Usage specification [usage parameters].

1.4 Standards developing organization (SDO)

Early 21st century understandings have evolved over the last 100 years to meet the needs of international industry and society in general. These standards systems primarily operates on a sector-by-sector basis. If humanity is to survive and flourish, then global standards developers must share and cooperate at the level of what is actually possible, with each other, in standards and conformity assessment activities.

An SDO is an organization that is an accredited representative of:

1. International Organization for Standardization (ISO), or
2. International Electrotechnical Commission (IEC) [iec.ch], or
3. has been accredited by these organizations.

The American National Standards Institute (ANSI) [ansi.org] is the sole U.S. representative to the ISO/IEC, and in turn, ANSI accredits more than 270 public and private standards developers that adhere to ANSI criteria for developing voluntary consensus standards. In contrast, Standards Setting Organizations include not only formal SDOs, but trade organizations, consortia, alliances, and others. Note that organizations like IETF, OASIS, and the W3C are considered SSOs, and their patent policies are independent of governing SDOs.

1.5 Standard setting organization (SSO)

A standards developing organization (SDO) generally refers to an industry or sector based standards organization that publishes and develops industry specific standards. Other names for this type of organization include, but are not limited to: standards setting organizations (SSO) or consortia. Many standards are developed by the standards body itself, or developed by a corporation and accepted by a standards body (and the standards body may, or may not, be a corporation, itself).

There are many SSOs, national, regional as well as industry-based. A formal SSO refers to one that is recognized directly or indirectly by a government entity. Very often, there will exist a formal SSO in a country that the government recognizes as the national standards body and which has the authority to designate a specification as the national standard for the country. Thus, for example, in India, the Bureau of Indian Standards (BIS) is the national standards body; in the USA, the American National Standards Institute (ANSI) is the official body; while in the United Kingdom, it is the British Standards Institute (BSI).While any organization can come up with its own specification and call it its standard, to be an internationally acceptable standard, it has to be either set or adopted/adapted by an SSO that is recognized as an international standard-setting body. The three organizations having the highest international recognition are the International Organization for Standardization (ISO), International Electro-technical Commission (IEC) and the International Telecommunication Union (ITU). ISO is an international standard-setting body made up mainly of representation from national standards bodies. IEC is a standards organization that deals mainly in setting standards for electrical, electronic and related technologies. A body that is an accredited representative to ISO or IEC is called a Standard Development Organization (SDO); most national standards bodies are SDOs. ISO produces standards in many domains, including IT. Many of its standards are also developed jointly with IEC, in particular, the ISO/IEC Joint Technical Committee 1 (JTC 1) is active in setting standards for the IT domain.

> **NOTE:** *The State often acts as a standard setting organization as well as the coercive enforcer of standards. Some laws are legal contracts for*

enforcing cultural standards (socio-standards) and others are legal contracts for enforcing technological standards (technical standards).

1.5.1 What is a "proprietary" standard?

A.k.a., De facto standards.

A de facto standard is a specification that became popular because everyone just happened to use it, possibly because it was implemented in a product that had significant market acceptance. The details of this specification may or may not be available publicly without some sort of special legal arrangement.

The basic problem with a de facto standard is that it is controlled by a single commercial entity, who can, and often does, change it whenever internally decided. At that point, everyone else who is trying to interoperate with the information. The owning vendor gets a time-to-market advantage, possibly increasing its market share, again.

Traditionally, it was not in the interest of the owner of a de facto standard to make the details too widely available because they didn't want to make it easier for anyone else to move into their market space. They would say, "Why would I voluntarily let other people build products compatible with my data? They might steal away my customers!". In answering these questions, it is essential to think in terms of transparency, community, democracy, costs, freedoms and permissions, and restrictions.

In the market-State, proprietary standards require financial payments (i.e., have a fee, require trade, need money). Conversely, in community, standards relevant to the design and operation of society are not proprietary (i.e., do not require a fee or trade). Proprietary standards use the State enforcement mechanism to restrict the ways and opportunities with which people can interact with their societal information system (i.e., with their "government").

For example, proprietary software is usually made available in a form that will run on your computer, but you are not given the original material from which it was created. You cannot freely incorporate proprietary software in your own products, though you may be able to obtain some sort of fee-based license to let you do this. The basic idea here is that proprietary software contains intellectual property that was created by the software provider and that is not shared because it offers competitive advantage. Licensing proprietary software to users for a fee is a long standing business model in the software industry. Licensing is not the only way revenue can be created, and it is often supplemented with subscription, maintenance, and support charges.

Problems arises when a standard is owned by one market player that uses the position of advantage (over others) to control the further development of the standard, or tries to manipulate it through licensing policies in order to exclude or include some specific groups of actors. In this case, the standardisation is used for contrary purposes than promoting co-design, co-operation, and co-usage.

The full co-operation in the community is, therefore, provided by standards that are open; because, open standards are freely available without any restrictions, they allow standardised information and technology to be used in products and services without ownership. As a consequence, the access to information and technology is accessible globally to everyone.

1.5.2 What is an "open" standard?

Open standards are usually defined as standards that are available for all to read and implement without any royalty or fee (i.e., they are free [of trade] to use, share, and modify. An open standard is free from legal or technical clauses that limit its utilization by anyone. Simply, an open standard refers to a format or protocol that is:

1. Subject to full public assessment and use without constraints in a manner equally available to all parties.
2. Without any components or extensions that have dependencies on formats or protocols that do not meet the definition of an open standard themselves.
3. Free from legal or technical clauses that limit its utilisation by any party or in any business model.
4. Managed and further developed independently of any single vendor in a process open to the equal participation of competitors and third parties.
5. Available in multiple complete implementations by competing vendors, or as a complete implementation equally available to all parties.

Open standardization ensures that technology is accessible for everyone, irrespective of business-model, size, or exclusive rights portfolio.

1.5.3 What is a "voluntary" standard?

Voluntary standards are standards established by any organization, and that are available for use by any other person or organization, private or government. The term includes what are commonly referred to as "industry standards," as well as, "consensus standards." In the market-State, it may become mandatory for the regulatory-enforcement authority system.

Different licensing practices have been developed in order to overcome the issue of patents essential to standard implementation. For example 'royalty-free' (RF) licensing and 'fair, reasonable, and non-discriminatory' (FRAND) licensing. Take note here that FRAND terms are incompatible with Free Software.

1.5.4 Patents in standards

Sometimes, the standard specification includes technical solutions that are needed in order to implement the standard. In the market-State, these technical solutions can be protected by patents. Whoever wishing to adopt and implement the standard in a relevant jurisdiction has to, therefore, acquire the appropriate licence from the patent-holder.

1.6 Standardization in the market

In the market, there are competing standards development organizations. Some of the competing standards organizations are called "de jure" organizations, because they have particular credentials in State jurisdictional (national or international) settings. Some governments have laws that make it very difficult to use standards that do not come from de jure organizations. ANSI, ITU, and ISO are examples of de jure organizations while groups like the W3C, OASIS, and the OMG are usually just referred to as consortia. Sometimes a standard produced by a consortium will be submitted and accepted ("blessed") by a de jure organization to make it more palatable for government procurements. Of course, de jure organizations, like all standards groups, must be very careful what they publish ("bless") because they have reputations for quality and relevance that they hope to maintain.

Consortiums have a formal governance structure wherein a consortium governs the standard. Typically, a consortium comprises key members and contributors, either from commercial or non-profit organizations, or being individuals. Consortium members are elected or appointed to a binding by law.

In the market, certification marks and logos "prove" they have been certified to certain safety standards. In the market, compliance with standards is often a jurisdictional issue. The brand/logo is to "prove" to the customer that the supplier has produced an item that conforms to the standardization. In the market, jurisdictional law enforces compliance to standards; thus, the necessity for the mixture of technical and legal documentation under market-State conditions.

1.7 What is the difference between a specification and a standard?

A specification is the result of (i.e., strictly bound to) the requirements. A standard is something that is consistent until new information is learned (i.e., what is probably optimal, contextual). A specification is a communicated [or communicable] design.

1.8 What are technical interoperability standards?

Technology interoperability standards are specifications that define the boundaries between two objects that have been put through a recognized [societal] decision process. In community, the decision process is supported, transparent, and open; in the market-State, the decision process may be a formal de jure process supported by national standards organizations (e.g. ISO, BSI), an industry or trade organization with broad interest (e.g. IEEE, ECMA), or a consortia with a narrower focus (e.g. W3C, OASIS). The standards process is not about finding the best technical solution, and codifying it, but rather to find the best value-encoded ("consensus driven") solution with which all the participants can live well and optimally. Whereas market implementations of interoperability standardization can be highly challenging, community application of interoperability tends toward system integration as the interoperable standard of priority. Market implementations are expected to benefit customers, by enabling choice in a marketplace. Alternatively, a global habitat-service standard specification enables the effective and efficient functional (and quality) design of a global access service system for all of humanity.

> **INSIGHT:** *A system which is optimally interoperable is open to unifying (and not, trading).*

In the market, instead of one unified and cooperatively developed standard, there are [often] multiple separate competing standards (and hence, product designs), which generates the market-based need for an interoperability standard. In Community, interoperability is the norm, because the information system is openly unified by a population of cooperating human contributors. Collecting data in community is simple because interoperability is designed in-to the system's design, and it is not an afterthought (or externality) of the result of market-State organizations developing socio-technical systems on their own, or in secret. Collecting data from lots of different sources, expressed in lots of different ways, is a result of proprietary (market-based) standards, and it is a waste of human energy and resources, because interoperability is being considered "after the fact" (i.e., after the standard has been developed, or a product has been produced). In the market, companies often implement a particular function in ways that do not build on current open standards. They might do this because no standard exists to meet their needs, because they decide to implement the same function without relying on standards for business reasons, or because they are unaware a standard exists. When there is competitive advantage in a global socio-technical system, then there is the need for additional layers of unnecessary, abstract, potential hurtful relationships (Read: licenses to engage State (i.e., the coercion and violence of the State) against a competing entity in the market-State; "jurisdiction").

When every body can view the standard, then everyone can follow the standard, then intercommunication (and sub-system interconnection/-ability) have the potential of being optimized in the next iteration.

In the market, by using interoperability standards, software and hardware systems made by different market-State organization can nevertheless communicate in a high level way that does not depend on the underlying implementation details. This means we don't all have to buy our computer hardware from the same vendor and we don't all have to use the same operating system and applications. In this sense, interoperability is the open source value applied between vendors, but it is not integration. It is the result of not having integration to begin with.

In the market, standards enable interoperability, compatibility, and consistency across markets.

1.8.1 System [service] interoperability

QUESTION: *What, fully described and explained, visualized, does 'service' mean?*

Interoperability is the ability of systems to provide services to and accept services from other systems and to use the services so exchanged to enable them to operate effectively together." A more precise definition of interoperability would require at least two steps: (1) identifying the vocabulary and syntax of service interfaces, and (2) defining interoperability mathematically. In this paper, I address the first requirement. Preliminary results of an ongoing debate suggest that the theory of institutions (Goguen and Burstall 1992 ; Goguen 2004 (draft)), building on category theory, supplies the necessary formal foundations for the second requirement. The notion of interoperability needs to be understood broadly enough, encompassing the interoperation between human beings and systems. But it should also remain precise enough, allowing for a common syntactic basis.

1.9 What is standardization?

Standardization refers to the process of establishing a common, shared model of the criteria, terms, principles, practices, materials, items, processes, equipment, parts, sub-assemblies, and assemblies appropriate to achieve the greatest practicable uniformity of products and practices, to ensure the minimum feasible variety of such items and practices, and to effect optimum interchangeability or interoperability of equipment, parts, and components. The standardization processes naturally create compatibility, similarity, measurement and symbol standards. Standardization can help to maximize compatibility, interoperability, safety, repeatability, or quality.

NOTE: *Socio-technical organizations have the potential to become more efficient through standardization.*

The four levels of standardization (in the context of interoperability between sub-systems in a unified system) are:

1. **Compatibility** - the sustainability of products, processes, services for use together under specific conditions to fulfill relevant requirements without causing unacceptable interactions.
2. **Interchangeability** - the ability of one product, process, or service to be used in place of another to fulfill the same requirements.
3. **Commonality** - the state achieved when the same knowledge, procedures, or equipment are used. Standardization of measurement and symbol standards.
4. **Reference** - the state of having the ability to trace information back to an evidence base.

A short history of standardization might be:

1. Philosophic and tribal standardization
2. National [market-State] standardization
3. International [market-State] standardization
4. Planetary [community] standardization

1.9.1 What is an asset identity code?

An asset identity code is a collection of mandatory standards, which has been codified by an information control system, and thus, has become part of the informational decisioning framework represented by that materializing system.

1.9.2 What is laboratory accreditation?

Laboratory accreditation is the formal determination and recognition that a laboratory has the capability to carry out specification tests in accordance with prescribed procedures.

1.9.3 What is harmonization?

Harmonization is the process whereby two or more habitat service systems (or, nations or standards bodies) reliably replicate and explicate ("agree on") the content and application of a standard.

1.9.4 What is meant by design decision standardization?

Design decisions are controlled to ensure design standardization as the adherence to specifications, tasks, standards tests, or other requirements. For example, a high-level design decision standardization is that of the requirement and specification for an network of integrated city systems. The majority of the planetary population live in a network of integrated city systems. The integrated city systems are a standardized, repeated, and sub-service bounded populated geoinformatic environment.

1.9.5 What is meant by validation (conformance) assessment?

Validation (conformance) is the state of having satisfied the requirements of some specific standard(s) and/or specification(s). Validation (conformity) assessment is the procedure by which an operation, product, process, service, or system becomes recognized in the decision system as accepted solution to the user's issue(s). Validation (or conformance) is used with respect to voluntary standards and open specifications, whereas compliance is used with respect to mandatory standards and regulations.

1.9.6 What is a service ("certified") product?

A service ("certified") product is a product that has been inspected, evaluated, tested, or otherwise determined to be in conformance or compliance with applicable or specified provisions of referenced standards, codes, or other requirements and certified by an authority which is recognized or has the legal power to grant such certification. Certified products imply a guarantee or warranty of product conformance and that the product is under the test and surveillance procedures of a specified certification system.

Information service standards for service fall into three categories (Read: the domains of software interoperability):

1. Data formats - A data format is how information is represented and structured.
2. Protocols - A protocol wraps up the data format with additional data necessary for transmission, so it can be moved reliably from one computer to another.
3. Interfaces - The interface is the exact specification of how you tell a service to do something, whether it is a query or an action to be performed.

All together, these three things describe how you talk to a service and how they talk to each other.

1.9.7 How do I locate standards?

Most standards developing organizations have search tools to locate and order standards that they develop. The SES web site provides links to most of these organizations. There are also several databases and websites that provide searches across standards developers at the national, regional, and international levels, including but not limited to:

- NSSN: A National Resource for Global Standards of the American National Standards Institute at [nssn.org] contains over 250,000 references to standards from over 600 standards developers worldwide.
- Standards Store of the Standards Council of Canada at [standardsstore.ca] contains over a very large listing of standards from hundreds of standards developers worldwide.
- Stanford Libraries Standards Reference Guide. [library.stanford.edu]
- The web site [standards.gov] maintained by the National Institute of Standards and Technology provides useful links to many databases worldwide that can help locate standards.
- The U.S. Department of Defense Acquisition Streamlining and Standardization Information System (ASSIST) database at [assistdocs.com] helps locate military and federal specifications and standards that can be downloaded free of charge.

1.9.8 Who are the globally known standards setting bodies?

In early 21st century society, the significant, globally recognized standards organizations are numerous; versus community, where there is one unified and optimized standard. Some of the most well-known are relevant standards settings bodies to global technological knowledge in general, include but are not limited to:

> **CLARIFICATION:** *Some organizations produce multiple standards and have no unified standard. Other organizations produce one unified standard, within which there may or may not be multiple sub-standards.*

1. Auravana.
   - *One unified standard (many sub-standards).*
2. Electronic Industries Alliance (EIA).
   - *Multiple standards.*
3. International Organization for Electrical Engineering Standardization (IEEE).
   - *Multiple standards.*
4. Council on System Engineering (INCOSE).
   - *Multiple standards.*
5. Institute of Electrical and Electronics Engineers (IEEE) .
   - *Multiple standards.*
6. International Electrotechnical Commission (IEC).
   - *Multiple standards.*
7. International Standards Organization (ISO).
   - *Multiple standards.*
8. Project Management Institute (PMI).
   - *One standard.*
9. National Institute for Standards and Technology (NIST).
   - *Multiple standards.*
10. American National Standards Institute (ANSI).
    - *Multiple standards.*

In the early 21st century, there are an incredibly large number of professional and useful socio-technical standards-issuing organizations throughout the world; the following list only identifies some of the more well recognized standards organizations at a global level. Note that not only do non-profits and governments produce standards, but corporations often develop their own standards also. Some corporations contribute to/ participate in the working deliverables of major national and international standards bodies also (there may be financial and other relationships in these cases)

A list of common standards setting organization may include:

- American National Standards Institute (ANSI). [ansi.org]
- American Society of Civil Engineers (ASCE). [asce.org]
- British Standards Institution (BSI). [bsigroup.com]
- German Institute for Standardization (DIN). [din.de]
- International Code Council (ICC). [iccsafe.org]
- International Council on Systems Engineering (INCOSE). UK Chapter. [incoseonline.org.uk]
- International Electrochemical Commission (IEC). [iec.ch]
- International Standards Organization (ISO). [iso.org]
- Institute of Electrical and Electronics Engineers (IEEE). [ieee.org]
- National Aeronautics and Space Agency (NASA). [standards.nasa.gov]
- National Institute of Standards and Technology (NIST). [nist.gov]
- National Academies of Sciences Engineering Medicine. [nap.edu]
- MITRE Corporation. [mitre.org]
- The Open Group. [opengroup.org]

1.10 What is a unified standard?

A unified standard is a single standard developed through the cooperation of a whole population who have an interest in participating in the coordinated development and/or use of the standard. Existence as a standard requires that all views and objections be considered, that reliability, objectivity, and certainty are available to everyone, and that an effort be made toward the resolution of all potential issues into a more optimal organization of useful information for the whole population. Unified implies more than the concept of a simple majority, opinion agreement, or consensus, but not necessarily unanimity. At the societal level, a standard requires reason and evidence for reliability, because it represents the optimal, and hence, safely reliable, way of doing anything in society, even regenerating society itself (Read: the societal information system specification).

Because society is an information system, it can be designed in a way that works better for everyone. There is a choice between openness in which information is shared by all, or we can have a closed model in which information is exclusively owned and controlled by competing interests. And, that choice gives very different worlds. If we choose open, then we have a world of access and fairness and fulfillment, and on the other side, 5he closed side we end up with digital dictatorships, in a world where the few dictate and dominate, whether that is online or shaping and controlling designs in ways that they choose and threatening or just excluding competitors and those seen as untrustworthy to their competitive advantage. In an open world. Viewing the societal system as fundamentally informational is key.

Open standards are open to the contribution of all (voluntary), open to usage by all (habitat), and evolve over time to more greatly fulfill all individual human beings.

There are two possible (at least, diametrically) constructions of an information-based society:

1. **Information symmetry** - open source, global cooperation (Read: the community).
2. **Information asymmetry** - closed source, competition, the artificial boundaries of the market (Read: the "market" and the "State").

There are two construction transparency phases:

1. **The transparency of the result:** The release of an operational, real-world, moneyless, access-based, open-source [code], integrated city-society.
   A. Is the result, global access (an open-source society); regardless of scale?
2. **The transparency of the development:** The open source development of the city and larger society.
   A. Does transparency of development matter (e.g., closed source), if the result is likely global access (e.g., open source)?
   B. **Open development** (e.g., open source projects, and global access licenses, transparency events and decisions).
   C. **Closed development** (e.g., NDA agreements, employment contracts, secrecy events and decisions).

1.11 Who uses standards?

Standards are useful, and sometimes essential, for anyone constructing anything at any level of socio-technical design, where information has previously been integrated into a standardized knowledge set of how to know and do something well and with intelligence. Where there is materialization (and hence, visualization), there is the potential usage of a standard to benefit the whole using population.

Every socio-technical society is built on written down standards. What is written down, documentation and visualization, is of primary importance for understanding, for coordination, and for adaptation. What is written down is useful information for all users, and essential information for workers (to varying degrees, with coordinators expected to be the most familiar with the unification of the standards). Standards are the description, explanation, and standardized way of working together in an understandable, ordered, and formalized manner, in an efficient manner. Without written socio-technical standards nothing of socio-technical value can be produced at scale. Without an understanding of socio-technical standards (their existence and content) it is not possible to coordinate a socio-technical society in the 21st century and past. Coordinators require a high-level understanding of the standards to effectively complete their project role; because they must both (1) know where to communicate, and (2) know in what alignment (with the existing standards) is the communication. In this second way, they are integration facilitators, and to integrate without contradiction, error, or bias, there must be a high-level understanding of the system's unification. In the early 21st century, this requires reading and visualization, particularly on the part of coordinators of teams/groups. In a team structure, this means coordinators must have Here, it is important to recognize that this is an engineering hierarchy, which is capable of safely bringing into existence new social creations; conversely, an authoritarian hierarchy brings coercion and competition into socio-technical creation. Coordinating leaders are accountable for a knowing the content of the societal standards.

> **NOTE:** *In the future, an AI may be trained on these standards and assume the functions of human coordination using software.*

1.12 Why are standards used at the societal level?

At the societal level, standards are used for many reasons, one of the most important being construction of the information and global habitat service systems. A societal systems specification is a standard that varies based upon the data and the intention of the population with access to the data. In society, there are standards for information infrastructural interfacing, for pesticides, for food processing and storage, etc. Safety standards provide an additional layer of safety in order to fully control and monitor water and air quality.

> **INSIGHT:** *By working together to develop planetary human societal standards, organizations from different industries are able to implement standards that benefit humankind, everywhere across the planet.*

1.13 What is a societal standard?

A societal standard is a standard that uniformly generates a socio-technical, societal, materialization. And, the intention for the standards creation is to generate optimally, given the integration of all that is known. The standards is the first knowledge set (wherein, data precedes knowledge, and structure precedes data, pattern precedes structure, intention precedes pattern). In the market-State, failure to comply with a mandatory standard usually engages enforcement, which carries out sanctions, competitors (civil) or State (criminal) penalties, or loss of money and ability to continue to profit. In the market-State, standards exist in this context, and they are developed by organizations embedded into this context. Here, standards may be used as a competitive advantage: if all other factors are equal, the market entity that can prove compliance to the applicable standard will have advantage over another that does not meet the requirements. Standards can be used by companies to avoid sanctions and penalties. Monitored compliance to standard adds trust to market competitor relationships. Because of the complexity of the market-State, the labeling and numbering of everything, including standards documents is highly confused. In community, the most well-known standard is the unified, global societal systems [standards] specifications. These documents specify past, present, potential future, and executed future standardized ways of constructing together in a common real-world environment. A societal standard provides a harmonized, stable and globally recognized framework for fulfillment of human individuals through the use of common resources and technologies. A standard that encompass multiple possible habitat service system configurations, customized to the intentions of their inhabitants.

> **QUESTION:** *Is the unified [societal] information system that holds all project information openly visible to everyone, and available for any to better (given, societal InterSystem protocol access)?*

1.13.1 What are human access standards

> **A.K.A.,** *Human well-being standards.*

A human access standard (a.k.a., human societal standard) identifies, given what is currently known, the lowest common denominator and highest common denominator of a standard of living among the population. The population exists within a network of locally integrated habitat service systems. A human access/societal standard identifies the presence of a universal, irreducible and essential set of material and informational conditions (really, conditional life-cycles) for achieving basic human well-being, along with indicators and quantitative thresholds, which can be operationalized for society based on local preferences. Humans have a set of material and informational, experiential requirements that are essential for human flourishing. A human access/societal standard identifies the set of material and informational conditions that everyone has the fundamental (basic, absolute, required) opportunity to access. These requirements are essential pre-conditions to meet basic needs, or provide central capabilities. A global human access standard specifies the extent to which, and how, such identified for everyone specified in documentation, and where preference processes would have to take over to reach the level of specificity required for their full operationalization. It is possible to coordinate for a universal set of material services, objects, and conditions that individuals, habitats, and the biosphere require, at a minimum, and maximum, for enabling flourishing for all.

1.13.1.1 Human access standard indicators

Indicators of an understandable level of human access, that is mutually desirable, includes, but may not be limited to:

1. Physical well-being:
   A. Nutrition (food, cold storage).
   B. Shelter (sufficient, safe, comfortable, hygiene).
   C. Living conditions (sufficient, safe, comfortable, hygiene).
   D. Clothing (sufficient, safe, comfortable, hygiene).
   E. Medical care (accessible and adequate).
   F. Air and atmosphere quality (accessible and adequate).
2. Social well-being:
   A. Education.
   B. Communication.
   C. Information and computation access.
   D. Mobility (access to transport, if required).
   E. Autonomy (personal space, freedom to contribute and participate).

1.14 What are a societal-level projects documentation requirements?

The project's societal-level documentation suite consists of (note that some of these are overlapping views):

1. Socio-technical documentation set - A set of socio-technical references for building and operating a socio-technical system.
2. The overview documentation set - Provides the reader with a top-level overview of the project and its proposal, a guide to the technology, a roadmap to the technology documentation set,
3. The online interface set - An overview of the project's web site. This document is aimed at the entire global audience.
4. Training materials documentation set - A set of instructional material, as well as a set of review/test questions with answers, that can be used to ensure understanding of societal concepts and systems. This documentation set is aimed at trainers/ trainees.
5. Marketing materials documentation set - A set of materials that provide a high-level overview of the project and its products, as well as a brief synopsis of the society-related work of the societal contributors and partners. These are used at conferences, demonstrations, and briefings as handout material. This documentation set is aimed at senior managers, project managers/system engineers, and operators/users.

Wherein, there are sub-domains:

1. The Synthetic Environment Domain - Provides background information on the creation and use of synthetic environment databases required to understand the problem that society (as community-type) solves. This document addresses the "why do we need Community" question. Additionally, the terms/technology that the reader needs to know to fully understand the synthetic environment domain problem are introduced and defined. This document is aimed at senior InterSystem Team Coordinators, operators/users, and trainers/trainees.
2. Technical Reference Set - Provides technical guidance to members of the data provider and data consumer communities. Provides explicit "how-to" information for the development of new Auravana products, as well as the use of existing products. Due to its size, this document is divided into many stand-alone "volumes". Volume 1 provides a detailed description of the contents of each individual volume contained in Part 4. As necessary, each volume of the reference set provides technical information covering all hardware platforms supporting the product. This reference set is aimed at developers/contractors, operators/users, and trainers/trainees.
3. Tools and Utilities User's Guide Set - Contains multiple stand-alone volumes that provide "how-to" information for the use of each Auravana software tool and utility. As necessary, provides specific instructions for each hardware platform supporting a tool. This document is aimed at developers and contractors, and trainers and trainees.
4. Procedures and Processes Manual - Provides a series of procedures and processes used to manage the project. It addresses configuration management, the FTP site, and the development process for core software, among others. This document is aimed at project coordinators, system engineers, and developers/contractors.

1.15 How do 'standards collaborations' differ from 'open source collaborations'?

Society is an open source, standards project, and therein, there are standards that are developed and maintained as open source sub-projects. In a market place, standards collaboration and open source projects are seen, generally, as different socio-economic tools in with different goals, outcomes, and processes. As Stephen Walli explains:

1. Standards take longer to develop and change. Whereas open source projects can develop quickly, standards encourage multiple implementations and tend to enter a market with some maturity and competition. Standards and specifications don't change quickly, so they are developed with the expectation that they'll need to last for longer periods of time. For example, moving from HTML1.0 to HTML5 standard took about 18 years, and we've had TCP since 1981 with few changes.
2. Standards are consensus-based compromises. Open source projects are driven by contribution and meritocracy.
3. Standards define useful predictable boundaries. Well-run open source projects are the building blocks of rich, varied ecosystems.

1.16 In terms of standards, what does this project propose?

This project proposes the world's first globally workable, unified societal systems standard. An open source project-based organization that forms a bridge between the potentials (e.g., Community and market-State sectors) by publishing the first societal-level information systems standard, and doing so, openly under a trade-free license.

The mission of the project is to create a unified, global societal information standard, and to promote the development of societal standardization and related activities in the world with a view to facilitating the global access fulfillment of all individuals to common heritage services. The mission should lead to the development of highly cooperative spheres of intellectual, scientific, technological, and social activity, which materializes (given that which is known) into a network of highly-automated, free-access, integrated city systems.

The project will realize (and materialize) a unified and global standard information-decision-materialization protocol of societal development and operations, which is disseminated as a published, globally accessible (transparent), unified (integrated) societal standard. To realize this goal, the project supports collaboration, development, and adoption of this standard across the globe.

Other names for the type of standard (and standards organization) this project proposes, are:

1. Planetary societal standard.
2. Planetary societal specification.
3. Human societal standard.
4. Societal specification standard.
5. Human life standard.
6. Universal community standard.
7. Planetary societal standard.

This is a global standards setting project (SDO; body) composed to realize, continuously a community-type societal standard for a planetary-scale human population. The development of a unified societal standard that "works" for all individuals among humanity. A standard is a medium of integrated alignment, a communications structure and protocol between people.

A community-type societal system is fundamentally based on the existence of openly developed standards. Open standards are a foundation of a community-type society. Open standards let people and organizations set up new services and make them available across the rest of the human network without permission. A good example of this is the World Wide Web, which was developed—without permission from anyone. The next example will be a societal-level information system. These standards are key to allowing information, services, devices, and applications to work together across the global network of habitat [city] service systems.

> **NOTE:** *A globally cooperative societal system must to the greatest extent possible have a de-personalized and de-commercialized societal standard.*

1.17 [Standard] Linguistics

Like most spoken languages, English is full of words that have multiple definitions and which evoke subtle nuances of meaning. The presence of multiple definitions and subtle nuance can lead to confusion and unhelpful disagreement when it comes to specifying and interpreting systems and their meaning.

A good tactic for reducing ill-definition and misinterpretation is to standardize the language used to express meaning (concepts). Appropriately standardized language optimizes communication by reducing the likelihood of confusion. Strictly defining terms, and adhering strictly to definitions, will not only reduce conflict and confusion in interpreting communication, but through its universal practice, all of society will "save" time and reduce the likelihood of conflict in developing systems that serve human fulfillment. In other words, linguistic standardization allows for efficient and effective communication and development between individuals.

In the context of this project, it is optimal to include a section dedicated to linguistic clarification accessible toward the beginning of the plan. This section defines exactly how certain terms will be used within the project itself, and how they should be interpreted (i.e., "read" or input).

Herein, precise language makes the meaning of the directive clear to the user (of the standard). There are different linguistic standards used globally for directive statements.

ISO, for example, uses:

1. **Shall** - requirement. When specifying a requirement, use the word shall.
2. **Should** - recommendation.
3. **May** - permission.
4. **Possibility** or **Capability** - can / can not.
5. **Must** - an external constraint (e.g., jurisdictional/ legal).

The following are some common and globally definable directive statements:

> *Note: The use of 'Shall' is used frequently throughout this document as a specification sets out clear absolute requirements. Recommendations, permissibility and possibilities are expressed as 'Should', 'Can' and 'May'.*

1. **Shall** - is used to express requirements of this standard.
2. **Must** - is used to express a previously decided decision or denote likely negative consequences if not followed.
3. **Should** - is used to express recommendations.
4. **May** - is used in the text to express permissibility (e.g., as an alternative to the primary) recommendation of the clause).
5. **Can** - is used to express possibility (e.g., a consequence of an action or an event).

Other organizations use different terms and definitions:

1. **Must/will** - requirement.
2. **Might** - recommendation, best practice, guideline.
3. **Could** or **Able** - can / can not.

1.18 [Standard] Semiotics

Semiotics refers to the axiomatic structuring of all language by consciousness, given the ability to influence a real-world, physical environment. In communication among a social population, semiotics facilitates linguistic standardization by acting as a refer for the creation of commonly meaningful structures, through:

1. The study of the communication of existence.
2. The study of how to most accurately represent a potential, and a real, world existence.
3. The production of models for understanding.
4. The production of models for additional capacity realization.

The current semiotic model is sub-composed of the following three inter-related conceptions, which enable communication and safe realization [of society] among a population:

1. **Semantics:** meanings, propositions, validity, truth, signification, denotations. Semantic means unambiguous. This is the semantic web. At the semantic level, the words, the technical and non-technical terms, and the things referred to in the conversations must be understood by the two people. The sentences and the contents of the conversation must make sense to both of them.
   A. Meaning - a 'sign' (as a unit of semantic, meaning) is normally considered as a relationship between a 'sign' as a unit of language and what that unit of language refers to a 'sign' denoted denotatum (real-world shaped surface). All real-world meanings have a reference in the shapeable real-world. Under this definition of meaning, there has to be a 'reality' assumed, a datum, so that signs can be mapped onto objects in the 'reality'. Meaning is a logic function mapping words to reality in some way useful to consciousness.
   B. The social system analogue - There exist individuated units of consciousness with the ability to sense an environment and open resolvable decisions spaces that have material consequences to the individually social environment. In other words, there exists a social population of individuals with the ability to sense an environment and integrate information through an open resolvable decisions space with material consequence to the individual and social environment.
2. **Syntactics:** formal structure, language, logic, data, programs, software, files, categories, functions, etc. Communication must follow the same grammatical-procedural rules to be shared. Syntactics is the aspect of semiotics concerned with structure. At one level it concerns the structure of sentences, claims, or procedures in or through a language. At another level it concerns the models as the instantiation of entities in relationship, patterns, algorithms, etc.
   A. The decision system analogue - There exist a calculated computational space where decisions may be resolved and designs may be compositionally solved as solutions. There is a

logical procedure [for referencing resources].

3. **Empirics:** pattern, variety, noise, entropy, channel capacity, redundancy, efficiency, codes, and the technical infrastructure to fulfill needs.
   A. Physical world - Humans have needs within a socio-technical environment. This environment is observable. The observation of a conception is to sense something which is technically understandable as appearing in the common, socially experience[-able] environment. Here, conceptions can be unified and when sensations are common sensed and communicated, then technical service system have the potential to arise into materiality to fulfill human needs as intentionally communicated to one another within the unified societal system.
   B. The material system analogue - There exists a real-world material-physical environment that is shared by our individually embodied consciousnesses.

1.19 [Standard] Unifying language

The Unified Modeling Language (UML) is an axiomatical-purpose, developmental, modeling language in the field of engineering ("creation" and "operation") that is intended to provide a standard way to visualize the design of a 'system'. UML is simply a diagrammatic, visual notation based on the system method.

> **NOTE:** *Modeling is the unifying language. Modeling is visualizing, and visualizing together requires technical modeling alignment on the part of all communicating entities.*

1.19.1 [Standard] Unified modeling language (UML)

> *A.k.a., Systems modeling language (SysML), unified requirements modeling language, (URML), and unified operations modeling language (UOML).*

Unified modeling language (UML) is the semiotic representation of conceptual information in visual form as purposeful communication between consciously processing entities (e.g., humans). UML is a communications standard, a set of rules for visualizing relationships between objects that exist, or may exist, in the real world. Information expressed through the rules of UML appears as an integrated set of diagrams forming a unified visualization, as a model, for the "network" of objects and relationships. UML could be considered the first element of a systems-based communications (i.e., visual) protocol between processing entities for arriving at a common understanding. Concept models are the most simplistic form of visually modeling objects and relationships. Concept "network" models are more complex descriptions, models, of objects and relationships.

> **CLARIFICATION:** *Note here the conceptual difference between a 'description' and an 'explanation'. The description is the visualization itself, which is perceived by the senses of the conscious processing entity. The explanation is the reason processing itself, which is processed by the cognition of the conscious processing entity. From explanation, more than one conscious processing entity can construct and share a common visualization. In communications, there can be description and not explanation. To have explanation and not description would be to not have a unified visualization language between consciously communicating entities.*

In order to create and operate any system in the real world there are correct alignment relationships that must be expressed (enacted). Conscious entities with the intention to operate together, to cooperate (co-operate), a common visualization rule processing structure is required. UML fulfills the requirement for that common visual-rule processing structure. UML was developed (discovered, naturally expressed) to allow system engineers (developers and operators) to visualize together, to co-operate, which is necessary in order to specify a possible design [for both entities], and construct that possible design [for both entities].

A specific visualization of a real world system (existent or not) is shared through a UML-based 'design-operation package', which is otherwise commonly known as a visual system specification document (an information set, or in digital storage, a 'file'). That 'design-operation package' file is shared between engineers co-operating (either as developers and/or operators). The 'design-operation(s) package' is the set of visual information (diagrams) for understanding (self), selecting (together), constructing (together), and operating (together) a real world system.

UML is a coherent and complete system visualization language applied cognitively (i.e., used to process information) that can account for the individual and the social. However, as a tool (i.e., a method, technique, process, etc.) its application by consciously processing entities may not always necessarily be so [at the societal level].

In the process of creation and operation, the UML represents a set of rules 'engineers' (the consciously processing entities expressing action) may use successfully to model large and complex, real world, systems. The UML is a requirement for developing system-ware (i.e., hardware and software, real-world interfaceable systems).

When expressed through a digital information system, the UML appears as graphical notations applying some set of semiotically coherent rules. To the graphical notation, there may, or may not be textual notation. All constructable and constructed 'design operation packages' are developed and operated through 'projects'. 'Projects are a sequence of operation's objects (action) and relationship's links (communication) that exist concurrently (together in 'time', sensory experience) between conscious processing entities.

Using the UML, project participants (team) communicate, explore potential designs, select a single design, create that design, verify that design, operate that design and test-study-learn from that design.

1.20 [Standard] Applied language

> **NOTE:** *The linguistic standardization of the two information sets necessary for intentionally re-creating a different sensible-experienceable-observable, real-world, physicalized environment. Here, coordination involves the consciously-unified sharing of information useful for a "peak-state" (Read: optimal state) of [required, given conditions] fulfillment.*

The Project and Engineering information sets are unified at the societal level, there is only a single, unfed information set, which can be viewed from two perspectives, that of the coordination (control and communication of resources; projects-tasks) and that of engineering (en-/ab-lization or en-/dis-ablization given a solid, materially-density constrained, environment). The common physicalized environment that consciousness en-habiting human form experiences changes through this process; where, individual can take the change, and groups of individuals can come together to cooperate to take the change. The Intersystem Team consists of Engineers who follow openly sourced rules, procedures, in their following of each new instruction. The instructions originate from the resolution of unique decision spaces in the given (common) information system to be executed by the InterSystem Team.

In the market-State, all humans are have some probability of being in competition with each other for the fulfillment of their human requirements (where, some people therein, cooperate). In other words, people are pitted against one another with some organizations of people pitted against one another having more control over the next instantiated iteration of the given material-physicalized environment (the State of regulation).

1.20.1 The systems language

> **APHORISM:** *It becomes very difficult to make progress when the lexicon (vocabulary) is not agreed upon.*

Modeling and designing complex, societal service systems requires a language capable of explaining services and describing their components by users who are also the service's creators. The language must produce a shared understandability to deal with the individuality of users and contributors. The language must integrate the autonomy of individuals and component parts, so that the creation is adaptive. That language must be able to represent a real common world in some degree of falsifiable alignment (levels of conceptual alignment) to deal with complexity (networks), context (situational issue), and nuance (common human need and individual histories).

In the information technology discipline (IT) there is a service-oriented architecture (SOA) standard that allows for the effective and efficient design and operation of human [service] systems. A service-oriented infrastructure is the integration of a wide divergence of components into a specific unified system to fulfill a purpose (Read: a service the application of socio-technical information for a purpose). A service-oriented structure provides users (who may also be contributors) a common interface and set of protocols for them to communicate, through a common process (sometimes called a 'service bus'). With the recognition that there exist the potential to design a service, exists the potential to design a societal organizing structure oriented around human need as the organizing form of service fulfillment.

To approach language systematically, definitions have to be criticized before explanation are evaluated (i.e., before someone expects another to adopt their theory). If definitions cannot be critically examined, then reasoning is irrational. If explanations cannot be critically examined, then [human initiated] constructions from those explanations are unlikely to produce optimal [human] environments.

1.20.1.1 Systems language applied to complex societal organization as simplified use-case scenario

A user - is going to 'drink' a 'cup' of 'coffee' under an 'umbrella' from the 'sun' and in a 'pleasant' - environment". In order to do, to produce, a consumable coffee in a nice location, the user and producer need to bring together many bits of information and shapes of material resource (from coffee beans to machines, and human effort). Some common platform must be designed for all these "things" to interoperate and deliver the final service, optimally.

In common practice, the service-oriented structural systems method associates sub-elements (parts) as delivering a service, which may be a:

1. Function (output as service process itself).
2. Object (output as shaped material).
3. Condition (output as state of processing shaped materials).

To the user (higher system need) there is the experience of a service, which does or does not meet the need [by the user for the services purposeful existence].

In the market there is something called "service autonomy" where market services run by business and States operate as black boxes with subjective interfaces. Note that the market concept of "service autonomy" plays no role in a unified human service system, and it's application is reflective of a dis-unified societal configuration where user and developer entities are competing against each, and, one another. When users are developers then services aren't "discovered by consumers through a market", but are instead, 'designed' and 'developed' by a 'community' of 'contributing users' who are discovering more about themselves and their world while living.

In community, services may or may not still be "broadcast" as being available; the decisioning is different in the market-State.

In market-based systems service terminology there is the concept of "loose coupling" to the whole system, which means that employees and employers, can be changed out easily, including by consumers, all of whom maintain their independence.

In community, "loose coupling" could mean individual contribution and freedom to access all the opportunities that all availabilities provide, because anyone is contributing. From a contributors perspective, a service is an internal adherence to a communications agreement as defined by one or more service description documents, and practiced as a protocol by teams of humans and hard-/soft-ware systems.

A unified service structure (unified service architecture) allows for the unified provisioning and de-provisioning of resources to sub-systems to optimize the overall service system. Here, unified means that all resources are accounted for.

> **NOTE:** *In the market all services are designed to eventually generate money (income), otherwise the service would not survive in the market (without philanthropist money support). In the market, services are not axiomatically independent of the market. In community, services are not axiomatically independent of human need, because the service providers are the users.*

Systems language is a language that visibly applies at all known levels of socio-technical scale, from the micro to the macro. A standard (generic) language that can be used to describe all the components in the system and their service. Interfaces translate between their local functionalities into the global language, which are given descriptors (Metadata tags) to describe the components functionality, availability, access protocol, conditions, and various other parameters to its coupling and service provision. It is possible to create a societal 'service bus' (a habitat service system) to integrate functional societal components into a complete [habitat] service system, and an interface for the end-user to interact with the services they need.

In a sense, service-oriented design (service-oriented architecture) is the selected structure for doing systems integration within complex engineered systems. It provides a formal language with the ability for abstracting to different levels [of abstraction] as required by the entity using it in any particular application.

At an individual level, having systems thinking allows for the autonomy of thought. A method for accurately modeling and aligning with the natural world is required if real-world individual fulfillment is the goal. If the real world can't be conceived of without serious error then all manner of environmental influence will be having all manner of negative network effects in the human system of autonomously fulfilled individuals. No one individual human is feeding of moving for another human individual unless there is some dis-ability present.

Insight: Just as someone can stop eating and moving in a healthy manner, so too can they not think in a healthy manner. Thinking can be out of alignment with the nature of their mental fulfillment, just as diet and movement can be out of alignment with the nature of their physical fulfillment, and to complexity the situation and make it 'real', each dimension of experience influences the other (because thought is being expressed through matter). The only language currently know of to express this complexity is systems language, which has carries the ability to self-correct (adapt alignment) and scale correctly (model coherence). Without the ability to model coherently, self-correction will likely be out of alignment with stated intentions, and without the ability to accept and integrate sensation a coherent model will likely not be developed.

A unifying system of language, systems language, is required:

1. A system has a given environment, by an interface.
2. A system has coherence, among its internal parts.
3. A system can self-correct, if it is living.
4. A system can be optimized, when it is unified.
5. A system can be designed (planned) and operated (executed) by life.
6. Life has requirements.
7. Life that uses 'systems' language can evaluate its service designs as 'systems' for purposefully completing life requirements.
8. The completion of life requirements may be optimal or sub-optimal.

1.20.2 Knowledge

Knowledge is the significant independent variable that will decide whether or not society moves forward into a community-type of society. The involvement of the global population ("masses") is necessary, but not sufficient. The masses have to know what and how to create a societal-level community, and in order to know that information, knowledge is required.

> **STATEMENT:** *If you gently read this document, you will receive unique insights that will assist your human minds development. But, this comes at the expense of being able to read dozens of pages at a time.*

1.20.3 Optimization

A unified societal system may be optimized when all core structures are accounted for:

1. An event-driven structure that represents temporal systems. (event-driven also means task, activity, etc.)
2. A positional-driven structure that represents spatial systems. (positional-driven also means material, physical, shape, etc.)
3. An intentional-driven structure that represents conceptual systems. (conceptual-driven also means semantic, meaning, purpose, etc.)

Here, a given system may be optimized by analyzing from, and synthesizing with, a unified structure. A unifying societal systems structure includes a unified, real-world model, which is structurally sub-composed of an event-oriented structure, a positional oriented structure, and a conceptual-oriented structure. It is through these data structures (information structures) that a transparent set of societal sub-system specifications are built (project, social, material, etc.). These categorical data structures may be applied as information constructors (by users) to combine data (previously existing and newly collected) into patterns (packages) of usability information for other societal sub-system. These data structures structure data in the social system that outputs into the decision system, when decisions are executed there are affects (some predictable, some not) in a material world that have consequence to consciousness, which inputs data in a variety of forms as feedback and design.

1.20.4 Simplified societal design for humanity

QUESTION: *What would society look like if it were arranged to complete human need?*

It is possible to analyze the composition of a societal design that works for all of humanity:

1. The whole unified societal human system
2. Has a whole unified information system
3. Expressing a whole unified habitat service system
4. Contributed to by whole unified individuals
5. For the human need fulfillment of all individuals.

It is possible to synthesize the composition of a societal design that works for all of humanity:

1. Whole unified individuals have a requirement for human need fulfillment.
2. Human need fulfillment may be contributed to by individuals who know what is needed.
3. Humans have a need to control (socio-technical state) a portion of their total habitat to develop and use complex socio-technical service systems.
4. To control (to decide the solution to) complex systems, information is required.
5. To control complex systems in alignment with a given intentional direction, then a sufficient amount of information is required (to ensure the solution is 99% predictable may be the highest level of information completeness).
6. Decisioning can have complete (sufficiency) or incomplete information in its database to determinedly resolve the execution of a decision in a complex control system.
7. Information in the determination of a decision can be unified (given access to all that is potentially shareable) and/or sufficient, or it can be incomplete.
8. To coordinate a complex societal system a unified information system must be as completely accessible as possible for human need fulfillment.
9. When all structures are modeled coherently, then individuals among society can more objectively account for why society is the way it is, and how society could be differently configured tomorrow to adjust for greater human individual need fulfillment.

NOTATION: *Is an individual's 'mood' the feeling someone has when following natural genetic programs, and the feeling comes from having those needs 'feel' fulfilled or not, 'suffer'. Whereas 'mood' is instinctual, 'emotion' is the conscious or sub-conscious drive, and the 'feeling' is the conscious feeling from the complex systems mixture.*

2 [Standard] Working group

A.k.a, Workgroup, working-group, work group, working party, task groups, or technical advisory group, the project integration working process, intersystem team working groups, working group conferences, solutions inquiry team.

The execution of solution design and integration is likely to involve working groups and workgroup conferences. Workgroup conferences are integration points for the team. In concern to the societal systems model and information system [article set], the result of societal engineering working group conferences are updates, sometimes, to the societal system. The concept model for the societal information system is resolved currently through these workgroup conferences, whose results are accepted or not and integrated via a larger management (or InterSystem Team). Organization of people and machines.. This/these individuals should be the most knowledgeable about that subject area since their names are listed as those who last developed the content. Life circumstances may complicate the issue of accountability. Former content developers are logged and removed. Generally, new iterations to the information sets come from workgroup conferences regularly/cyclically pre-scheduled, some of which may lead to changes, and others not. The results are accepted by the affiliates as the results of a transparent decision.

Working groups are self-directed organizations of skilled and motivated individuals who are working on the articles of standardization of one or more aspects of a community type society. Working groups are composed of those who are sufficiently motivated to contribute and sufficiently informed to understand (or some mixture thereof). Working groups are composed of informed and capable individuals (the term "experts" connotes wrongly here that only those who have put in 5-10,000hrs can make contributions, and is thus replaced with, 'informed' and 'capable').

The coordinator structures information and material flows between the developers ("experts"), and schedules conferences where appropriate.

Market-State organizations generally form working groups by time, technology, or territory. The weakness of this is that boundaries interfere with the desirable sharing of knowledge and experience, and so, learning suffers and work becomes less optimal (efficient and effective). Self-directed and self-regulated groups do not require supervisors to manage the boundaries of the group (e.g., ensure the group has adequate resources and coordinate activities with other groups) and foresee coming changes.

The responsibility for work on standards begins in a working group. Standard[ized] operating procedures facilitate the effort of working group participants and the deliverable by establishing the necessary framework for a workable organization. These [standard] operating procedures outline the orderly process of work by the working group.

A common working group procedure is, for example:

1. Working group personal and sub-group work.
2. Working group meeting/conference for discovery presentation and integration [draft integration].
3. Public comment period: October 9, 2019 through December 9, 2019
4. Working group meeting/conference for integration primarily [final integration].
5. InterSystem Teams implement and/or apply new societal standard; teams conform information and material environment to the standard.
6. All working groups are live streamed. All working documentation is public except for personal notes. All comments are transparent, and generally, accountable.

Working group standards information flow involves:

1. Pre-conceptualization.
2. Conceptualization.
3. Discussion.
4. Writing, modeling, simulating.
5. Implementation.

2.1 Technical working groups

Technical working groups discover, integrate, and develop socio-technical systems. By forming a technical working group (TWG), high-level practitioners working in on the same article of the systems composition can coordinate activities and align resources to better work toward common objectives in their sector or area of focus. Collaborative development leads to more efficient use of resources.

Working group deliverables, the community specification standard is the main deliverable that the Project publishes. However, there are other sub-societal deliverables that technical working groups may publish, including but not limited to:

1. Technical reports (TR) - cannot contain requirements.
2. Research reports (RR).
3. Publicly available specifications and standards (PAS) - can contain requirements. Free and open source specifications by other organizations.
4. Technical specifications and standards (TSS) - draft and sub-societal specifications.

2.2 Working group conferences

A.k.a., Technical working group sharing and integration events to produce standards.

Technical working groups (TWGs) come together at a working group conference to learn and decide. At a working group conference, articles that compose the societal standard are developed. When appropriate, groups split off into smaller sub-groups to work on different sub-sections (sub-problems or solutions) of a total article. In general, working groups develop standards under standardized ("approved") scope.

Working group core members focus on models, clauses, drawings, simulations, coding and coordination. If deliverables are developed and approved, then the names of those who attended and approve the deliverable output are assigned to the new article of their contribution.

In a working group conference. A decision in the form of "consensus" is the resolution of serious objections sufficiently for the coordinator to effectively move forward with the effort of the working group. At decisions points there must be sufficient information to resolve the decision such that there are no serious objections sufficient for the coordinator to prevent the forward movement of a working group or prevent conflict.

In a working group, from the submitted modifications, the members decide to accept or not the. If the group thinks the modification will benefit, he will choose the best code from all of the submittals and incorporate it into the updates.

Coordinators may sign off that there are no significant remaining objections. "Consensus" is general agreement (90% and above), characterized by the absence of sustained opposition to substantial issues by an important part of the concerned interests and by a process that involves seeking to take into account views of all humans concerned and to reconcile/integrate any conflicting arguments. If voting is required, then a 90% threshold is required to move the project forward. Sustained opposition means sustained opposition on the part of another working group member of the same working group (and not another working group of member of the public).

Working group conferences can be exhaustive exercises, and so the work must be checked post conference by at least the contextual coordinator. When the next publication is ready, new content and names will be published. If the next publication won't be ready for some time, then workgroup results can be published temporarily as addendums, waiting for the next iteration of the complete publication.

Content scheduled for presentation at the workgroup conference, should in general, be sufficiently complete and open that it can be worked and reworked into the next iteration of the system by sub-teams of the whole population of workgroup attendees the workgroup works with the prior and new (should be easy to work with) information to produce a better ultimate design and/or understanding. Workgroups should be of an appropriate size to complete work effectively. The term working group or work group conference can be confusing at first. What happens is people do work before the conference, this is their personal work which they may or may not have made public to everyone. They then get their work sufficiently reading so that it can be worked into other work by a team of workers. They then attend the conference with their sufficiently completed work. Teams first learn about the new work. Then teams integrate, as possible, the new work into the old, all the while working on achieving greater understanding. This whole process may last a few days, or weeks.

This whole process generally occurs with most of the attendees together in the same physical space so that communication and work is real-time. Remember, this is a process of integration, most of the discovery was done earlier by those who presented their work at the working group conference.

Some conferences have nothing produced in terms of changes to the actualized or described system, and instead only personal learning and greater understanding occurs for conference attendees.

Workgroup conferences are populated by their specific Intersystem team members, and by significant contributors who are presenting their discoveries or their significantly complete (to be workable) work. Coordinators generally, though not always, try to unbiased themselves from the events of the workgroup, acting as a peer reviewer of the output and not participating work working group re-working teams precisely.

There is no formal rule against this though. It is just a potential flag of bias for when the open source commons public has the opportunity to view the new workgroup content and its peer accountability reviews. There could be bias here, which everyone should be aware of. Yes, workgroups are where work occurs, but most of the work should be done ahead of time. The result of a conference may just be an article of work for researchers or workers outside the conference. A working group conference can produce many outputs, some types of which will lead to changes to the core kernel.

2.2.1 Standards publication

Every societal standard is a continuous living document (and visualization); it is updated continuously over time. Published versions of the societal standard for a community-type society represent snapshots in time. Frequently updated standards will likely be published more frequently by the organization (i.e., every quarter, as opposed to once a year or two). The most up-to-date version of a standard may be published in digital and/or physical print formats.

The Auravana Project has the following protocols in concern to publishing:

1. If there are no updates to a specific standard, then there is no need to publish again.
2. If there are more frequent updates, then the standard should be published more frequently.

3. Draft articles in a standard are publishable with newly updated standards if they maintain a base level of integration without significant contradiction. These draft articles will carry the following statement in all capital letters:

   IMPORTANT: *THIS IS A DRAFT DOCUMENT ONLY AND HAS NOT BEEN FULLY DEVELOPED AND INTEGRATED INTO THE STANDARD. THIS DRAFT DOCUMENT MAY NOT BE RELIED UPON.*

2.3 What is an open-source societal standards setting working group (workgroup)?

There is a specification/model of society that incorporates a series of articles that together represent the societal standard(s) system. The specification as a composition of articles is the decided upon standard for information-spatial processing in society. These adaptive standard-articles change how the societal system itself is understood and also lead to changes in the informational-materialization of society. In this later sense, the standard articles represent the specification for the society as explained and to be actualized upon. Any given socio-technical society is made up of standards. A society, uniquely, can compose these standards into the form of a unified specification for the next optimal iteration of the society itself. Workgroups can be composed to discover and decide the societal standards, which are described and explained in text and visualized spatially. The societal specification articles could be viewed as articles of specification for a community-type society. Each article represents a standard[ized] as understandable and intended element within that society.

The articles that compose this document and the whole societal system specification (social, decision, lifestyle, and material) include all operative (at a Habitat InterSystem Team level) standards in society. The currently decided articles are the current standard for society. Each article represents the composition and reasoning for a sub-construction of the whole societal system. Standards, sub-composed of articles, adopted by working groups, forms the specification for the design of a societal-level operating system.

The standards societal system specification sub-composed of articles must be adopted:

1. Adoption of [articles of] societal standard for city-network and nations.
   A. Working group 1 (e.g., ISO37101, System Management) - This standard sets requirements, guidance and supporting techniques for sustainable societal development among all sub-communities. It is designed to help all kinds of sub-communities coordinate their sustainability, smartness and resilience to improve the contribution of communities to sustainable human development and self-performance progression.
2. Adoption of [articles of] technical standards for cities, operations and usage, and interoperability.
   A. Working group 2 (e.g., ISO 37120; ISO TC 268 WG 2, City Indicators) - This standard sets requirements, guidance and supporting techniques for sustainable technological development among all sub-communities.

2.4 Community-type society workgroup sub-composition

In a community-type society, there are:

1. The intersystem spatial teams (people doing socio-technical, material things to sustain the population as life, technical, and exploratory).
2. The intersystem information teams (people taking decisions and integration determinations as individuals, team contributions, algorithms, and [accepting and developing a] information system).

A working group (a.k.a., working group, work group, working party, task groups, workgroups, or technical advisory groups) standards setting body (higher level), community of practice (lower level). is a group of knowledgeable individuals working together to achieve specified goals. Working groups are domain-specific and focus on discussion or activity around a specific subject area. A working group can be disciplinary or interdisciplinary.

The lifespan of a working group can last for years or only a few months. Work groups that extend over years have the tendency to develop a quasi-permanent existence when the assigned task is accomplished;[citation needed] hence the need to disband (or phase out) the working group when it has achieved its goal(s). It is imperative for the participants to appreciate and understand that the working group is intended to be a forum for cooperation and participation; the working group exists for those who want to contribute work, only related to the groups work.

Characteristics of a work group:

1. A work group may be ad hoc or exist continuously.
2. A work group may be team-oriented, team-centric, or non-team affiliated (note: team here refers to InterSystem Team).
3. A work group may be a formal standard setting body, conference, event, or some other point of integration
4. A work group may produce a formal specification iteration.

5. Generally, a work group conference is the point of common integration and production for a working group.

Examples of common goals for working groups include:

1. Creation of an informational document.
2. Creation of a standard.
3. Resolution of problems related to a system or network.
4. Continuous improvement.
5. Research.

Real-world working groups may be:

1. Social - workgroup teams.
   A. Social service teams carrying out informational processes.
   B. Social information work groups.
2. Decision - combination, and computation.
   A. Decision support service teams carrying out decisional processes.
3. Material - habitat service teams.
   A. Habitat service teams carrying out operational processes.
   B. Habitat information work groups.
   C. The habitat life-planning operational process team work group.
   D. The habitat technical-operating operational process team work group.
   E. The habitat exploratory-discretionary operational process team work group.
   F. A habitat service system has a set of operational process teams (planning, operations, discretionary). Each habitat service system has a work group. All operational systems have actively accountable teams.
   G. In the case of the decision system algorithm, the kernel, the decision system work group conference iterates, and habitat operation process team oversees the systems operation.

There are effectively three levels of designation for the societal system from a work group view:

1. **Exploration work group.**
   A. The societal information system workgroup, and many sub-workgroups. Development of the total information system itself.
2. **Kernel integration work group.**
   A. The societal decision system algorithm workgroup. The procedural algorithm itself.
3. **Habitat service team operations work group**
   A. The habitat operation process teams. Teams that follow procedures have a continued interest in those procedures.

What are the open standards requirement for society?

- Which sets forth a number of criteria to ensure that specifications can be implemented under open source licenses. The OSR will be used by the working group as a set of guiding principles and best practices.

The Open Source and Standards Working Group will:

1. Explore current SSO understanding of OSI approved licenses, and more generally, open source software, development, and projects;
2. Educate SSO in current principles and practices widely excepted by open source communities of practice;
3. Support authentic engagement across open source communities (i.e. implementers, contributors, projects, foundations) to ensure alignment with best practices in open source licensing, development and distribution, and;
4. Produce reference resources (educational materials, professional development activities, expert opinions, consulting services, etc.) to address gaps in understanding, support current practices, and increase the recognition of OSI approved licensing and the OSI License Review Process.
5. Encourage SSOs to request and maintain formal peer relationships with OSI. The Working Group will act as the formal Correspondent.

> **NOTE:** *Working group proceedings may be hosted on Github, a collaboration platform especially well-suited for open source projects.*

2.5 Workgroup decision criteria

A.k.a., Decision criteria, workgroup criteria.

Criteria is the plural form of the word criterion, which means a standard, rule, or test (ideally with reasoning) on which a decision (determination, selection, evaluation, etc.) can be based. In application, criteria are used for the *evaluation* of probables and *selection* of a singular [solution]. A criteria for the selection of a solution will lead to the ranking of potential solutions. The application of the criteria to some information set lead to the ranking of solutions; wherein, solutions are inquired into, and are ranked, according to the criteria.

> **NOTE:** *Workgroups and algorithmic decision processes resolve decisions (in part) through criteria.*

A threshold may exist beyond which a solution is acceptable and/or is not acceptable [to the complete, 99%, resolution of the inquiry cycle]. A criteria may be used to determine this threshold [at which a particular solution, from all the many probable solutions, is selected to be executed upon].

> **NOTE:** *Thresholds require a resulting value against which to compare. The resulting value is sometimes known as a "score", and in such an analogy, the threshold would be the "goal".*

More technically, a criteria for a newly incoming set of information lead to the ranking of its processed outputs. From an information systems perspective, a criteria is an information search and resolution program. The criteria is pre-selected. New information comes into content with the criteria. If the new information is absorbed, then the system that established the criteria can run calculations on the results to discover-learn more about the information environment.

When a criteria for a design [project] is decided/determined, it is then used to evaluate the success or failure of the design (as an inquiry, a solution to meet an inquiry, and/or, a project). Criteria is something that may express (or, result in) an evaluation. For example, a set of criteria for buying a new television may be location shape, visual quality, sound quality, battery life, cabling, or brand name (market only).

Analyze solutions problems to evaluate them against a set of criteria that match a completely (decision system acceptable) set of pre-determined criteria for selection of one solution [to materialize and feed-back into ourselves].

Among the population of an organism, the most essential criteria for survival and thriving is that of moving toward the satisfaction of life needs.

Among a global population, it is essential to transparently reason (i.e., justify) 'why' every action has been taken (i.e., to explain with some evidence).

The inquiry resolution protocol (i.e., markers, examiners, etc.) will constantly look at (inquire into) the product/system or environment that that is being produced as a solution for the evidence of its intended physical- or informational-oriented objective. In the case of a team, inquiries will look into the application of skills, application of your research and application of the results of experimentation, testing, and integration. The validity of the work is evidenced by the application in the system of the designing [specification]. The work is valid because the experience of the 'what' works as expected. The 'how' requires materials and technical knowledge.

2.5.1 One of the more simplest workgroup decision criteria

Each team or workgroup member may provide a final score based on an equal weighing in each of the following four criteria as well as a set of short comments (risks and biases must be noted for each criteria):

1. Clarity of vision - quality of visualization or writing in understandableness (comprehensibility).
   A. Is there a visualization?
   B. Can it be understood?
   C. Can it be integrated?
2. Past performance - given what has occurred, what is most likely to occur?
   A. Is there a predictably less beneficial likelihood of current trends continue?
   B. What changes can be made to make alternative potentials most likely?

3 Technical "peer" review

A.k.a., Peer review, technical review, socio-technical review, engineering review, merit review, work review, challenge review, critical review.

In order to ensure accurate, up-to-date, and safe system design, development, and operation, socio-technical articles go through a review process. Review (peer-review, technical-review, etc.) is the name given to any judgment of technical merit by others working in or close to the field in question. Reviews should consist of a critical analysis, and should include careful reasoning, citation, and conclusions.

A [peer] review is a documented, critical review performed by technically competent persons ("peers"). A technically competent person is a person having technical knowledge and/or skills in the subject matter (discipline) to be reviewed (or a subset of the subject matter to be reviewed) to a degree at least equivalent to that needed for the original work. Reviewers can be part of a working group, part of the a larger team, or part of the public. In the strict case of "peer-review, the reviewers are generally expected to be independent of the work being reviewed. The peer's independence from the work being reviewed means that the peer, a) was not involved as a participant, supervisor, technical reviewer, or advisor in the work being reviewed, and b) to the extent practical, has sufficient freedom from funding considerations to assure the work is impartially reviewed.

Technical design reviews occur during the system engineering lifecycle. These reviews can be supported by peer reviews, which are deeper technical reviews by technical experts in the subject matter to be reviewed.

Note here that in some cases, there are different terms of the sub-types of review. In general, a review is an in-depth critique of some combination of knowledge, synthesis, assumptions, calculations, extrapolations, alternate interpretations, methodology, acceptance criteria, results, and of conclusions and content written, drawn, or modeled. An engineering peer review can be a resource for a product team to find potential defects, design weaknesses, or implementation flaws as early as possible in the development process. A peer review tends to fall along the lines of improving quality, aiding decision-making, ascertaining that objectives and/or requirements are being met, and/or providing validation. Such reviews, conducted by a team of peers, bring the system team a broad experience base and lessons-learned from previous operations, without which design oversight can be missed. Sometimes, reviews (generally, "peer reviews") confirm the adequacy of the work. Sometimes the term "technical review" refers to a review to verify compliance to predetermined standards or requirements. Other times the term "technical review" means a confirm the adequacy of work. Different disciplines may use similar terminology to refer to different types of review. Additionally, most organizations restrict the term "peer review" to only those technical reviews conducted by independent, external [technical/peer] experts. Reviews conducted publicly can enhance credibility by increasing confidence in the process. Also, the identification of peers can help to identify interdependencies between disciplines.

Critical review analyses are generally structured by means of concern levels, where issues that the reviewer observes are assigned a level of concern. Concern levels include:

1. Yes, high concern.
2. Yes, medium concern.
3. Yes, minor concern.
4. No, not a concern.

Alternatively, the categories (with their associated impact level) could be:

1. High/critical - critical failure.
2. Moderate/major - major failure.
3. Low/minor - failure that does affect overall system stability or functioning (in the short term).
4. Negligible - unaffected.

The impact level refers to the consequences that could occur if the issue isn't solved.

Note, these issue levels could relate to the engineering [design through to construction] life-cycle, or they could refer to the scientific [research] life-cycle.

The technical "peer" review process involves the following steps:

1. Step 1: Author writes and submits article to working group and coordinator.
2. Step 2: The editor/coordinator sends the content to technical (expert) personnel to review and evaluate quality of content, research, writing, and conclusions.
3. Reviewers return content to editor/coordinator with suggested changes, as well as a recommendation to publish or not publish the content.
4. Editor/coordinator reviews suggestions and returns the content to the author for revision (if necessary).
5. Author revises and resubmits the article to the editor.
6. Step 6: The article is published in the journal/standard.

Technical/peer review is a system based on knowledge, improvement, and appropriate coordination. Effective and rigorous technical peer reviews provide quality to a product realization process. More specifically, peer review is a process in which an article is evaluated by a group of technically competent people in the same field (or related field) to make sure it meets the necessary standards and quality for acceptance and publication. Therein, technical peer reviews involve a well-defined review process for finding and fixing defects, conducted by a team of peers with assigned roles. Technical peer reviews are carried out by peers representing areas of life cycle affected by material being reviewed (usually limited to 6 or fewer people). The reviewer(s) must give an honest and impartial evaluation of the article.

Review processes can be carried out by:

1. Documentation coordinator.
2. Subject matter expert (technical expert).
3. Technical support representative.

> *Note: For the current standards, a peer/technical review is quasi-optional, and each working group can design and select a review process according to its needs. The review is quasi-optional because of the present size of the working group team and the fact that the coordinating editor, who is technically competent already, must conduct a review of the content.*

The goal(s) of peer review processes may be to:

1. Verify whether the work satisfies the specifications.
2. Identify any deviations from the standards or knowledge base.
3. Ensure standard engineering/scientific rigor has been completed.
4. Ascertain that the methodology and/or science underlying the conclusions and/or technology is well-understood (and does not have contradictions, flaws, errors, or biases).
5. Provide suggestions for improvements.

Responsibilities of the reviewer:

1. Provide a prompt, thorough, and impartial review of the content.
2. Ensure the content is without errors.
3. Give constructive feedback with reasonable suggestions and professional tone.
4. Avoid suggesting the addition of irrelevant or unnecessary references.
5. Alert the coordinator to any suspected moral/ethical issues.

Responsibilities of the author:

1. Accurately (and without bias) report research findings and/or knowledge content.
2. Ensure the content is without errors, and correct errors if present.
3. Describe methods and materials with enough detail that the work can be reproduced.
4. Cite only articles/content that are directly relevant to the submitted article.
5. Ensure the content meets all publication requirements standards.
6. Revise the content as per reviewers' suggestions (or give a reason why not).

Responsibilities of the coordinator:

1. Select, invite, and coordinate reviewers.
2. Ensure an accountable, useful, unbiased, and speedy peer review process.
3. Ensure the content is without errors.
4. Synthesize disparate peer review reports and arrive at a final decision.
5. Any questions from authors or reviewers about the peer review process should be directed to the coordinator.

In concern to item #4 in the list directly above, content may receive one of three possible decisions:

1. Acceptance: The content will be integrated and published without edits. "You" may be asked to upload final files or to sign a copyright form.
2. Acceptance with revision: The content will be accepted after suggested/required edits by the reviewers are finished. "You" will be asked to provide a revised version.
3. Rejection: The content will not be integrated and/or published.

Table 28. *The difference between useful publications: peer-reviewed and non-peer-reviewed publications.*

	Peer-reviewed	Not-peer-reviewed
	Scientific/scholarly journal	Trade journal/ magazine
User	Scientists and experts (researchers and professionals)	Scientists and experts (members of a specific discipline, business, or organization)
Author(s)	Technically qualified personnel	Technically qualified personnel
Editing	Peer-review process; 2-3 technically qualified personnel	Editor/coordinator of publication
Review	Technical peer reviews are a well-defined review process for finding and fixing defects, conducted by a team of peers with assigned roles	Well-defined review process for finding and fixing errors, conducted by coordinating editor
Content	Research projects, methodology, theory, results, etc.	Trends, products, techniques and organizational news, etc.

Sources	Sources cited in bibliographies and/ or footnotes	Sources mentioned occasionally with bibliographies
Purpose	To share facts.	To share the latest information and news

Book references (non-cited)

- *How to write standards*. (2016). ISO Standards. ISBN 978-92-67-10686-1 https://www.iso.org/files/live/sites/isoorg/files/archive/pdf/en/how-to-write-standards.pdf

Online references (non-cited)

- Hornung, A. Krivosheev, G. et al. (2006). *Standards Wars.* University of Washington. CSEP 590A: History of Computing; Autumn 2006. https://courses.cs.washington.edu/courses/csep590a/06au/projects/standards-wars.
- Lucas, G., Hatcher, L. Introduction to standards and specifications for design in mechanics or Strength of materials. Penn State College of Engineering. Accessed: December, 2019. https://sites.esm.psu.edu/courses/emch13d/design/design-tech/standards/specs_8.pdf
- *Quality Resources*. (2019). American Society for Quality. https://asq.org/quality-resources/standards-101
- Rakesh, Vanya. (2016). *Adoption of Standards in Smart Cities - Way Forward for India*. The Center For Internet & Society. March, 19. https://cis-india.org/internet-governance/blog/adoption-of-standards-in-smart-cities-way-forward-for-india
- *Standards development*. (2019). International Electrotechnical Commission. https://www.iec.ch/standardsdev/
- *Standards: What are they and why are they important*. (2005). NPES Standards Bluebook. http://www.npes.org/pdf/Standards-WhatAreThey.pdf
- *The Standards Development Process*. (2019). The Open Group. https://www.opengroup.org/standardsprocess/standards-dev.html

Approach: Opening

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: openness, the open approach, the open source approach, the participative approach, the voluntary approach, the collaborative approach, the socially cooperative approach, the sharing approach, the free approach, the cooperation approach. openization, contribution, open-source, open standards, open licensing, copy-left, patent-left, open-patenting,

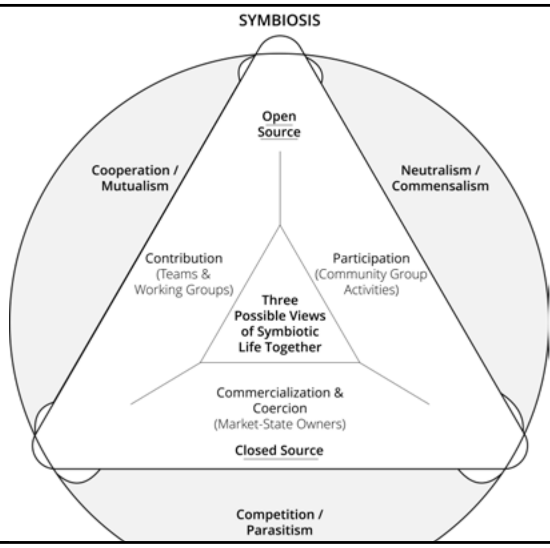
Abstract

This article delves into the transformative potential of open and left licensing frameworks in facilitating the transition of society towards a community-based operational model. At the heart of this transition lies the concept of contribution, characterized by work driven by intrinsic motivation rather than the pursuit of trade or tangible extrinsic rewards. Such contributions thrive within open information environments, spaces that are inherently "free," "open source," and "left." In these environments, the essence of open source is manifest; it democratizes access to the source code, enabling collective participation and innovation across community projects. This paradigm shifts away from conventional market-driven approaches, emphasizing a royalty-free license to use that embodies the true spirit of "free" - operating without the constraints of trade or currency.

The article explores how, within community projects, contributors engage with open-source systems and standards, laying the groundwork for a collaborative ethos that transcends traditional proprietary restrictions. Open-source licenses serve as the backbone of this collaborative potential, ensuring that all participants have equal access to modify, improve, and distribute the collective intellectual output. Meanwhile, left licenses play a crucial role in safeguarding these contributions within the commons, ensuring that redistributions adhere to the principles of community benefit and shared value, rather than being co-opted for private gain.

Graphical Abstract

Figure 8. *It is possible for humanity to work together to meet one another needs by means of open source design and freely contributed effort. Simplistically, there are three possible approaches to completing work at population scale: together fulfillment, apart fulfillment, or neutral fulfillment.*



1 Introduction

Simplistically, in order for individuals in society to have trust and certainty in their society, there must be transparency and contribution to society's formalization, construction, and operation:

1. **Openness and transparency** (facilitates *trust with high certainty*): Everyone has access to the information and materials necessary for doing the best work. When these information and materials are accessible, humanity can build upon each other's ideas and discoveries. Humanity can make more effective decisions and understand how decisions affect one another. Humanity can resolve solutions more effectively. And, human can more efficiently use resources.
   A. What is being transitioned to is a coordinated information commons.
2. **Contribution** (facilitates *trust with high certainty*): When humanity is free [to participate], anyone can enhance another's work in beneficial ways. When we can modify what others have shared, humanity "unlocks" new possibilities [for individual fulfillment and freedom]. By initiating new projects together, humanity can solve problems that no one can solve alone. When humanity implements open standards, every individual is enabled to contribute to the present and the future.
   A. What is being transitioned to is a coordinated [physical commons] contribution service system.

1.1 Openness policy

A.k.a., Opening strategy, objectives and procedures, open access approach, transparency approach, trust approach, open-source approach.

When any project or organization (including, government or corporation) wants to increase transparency, one approach is to institute an openness policy. Open source is the global categorization of a policy of openness. Open source increases commons access and creates socio-technical standards and technologies without a profit or coercion bias. Fundamentally, community designs systems that are as open as possible, as restricted as necessary, and always safe (and secure). Habitats in community are open source creations and operations, by requirement. Open source should pertain to all sources of project content:

1. Open-source data repositories should be created/used to find a common language for data sharing.
2. Open-source licenses should be created/used to find a common language for entering content into the commons. Here, persistence licensing is valued in its ability to maintain content in the commons.
   A. Software should be open-source licensed (i.e., open-source software, open software designs).
   B. Hardware should be open-source licensed (i.e., open-source hardware, open-hardware designs).
   C. Information should be open-source licensed (i.e., open-source copyright, open access journals, open standards).

2 Market-State property ownership

In the market-State, there is property ownership; there is property and there is the legal [entity] person that controls the property (Read: the owner of the property). Therein, property must be defended [by those who own the property and by the State] against those who seek to use, but do not legally own, the property. All property in the market-State necessitates an organization that exists to enforce the "rights" of the legal owner of the property. In a market-State (i.e., environment with property-market and State-enforcement) where there is private intellectual property, then there is the coerced and State-enforced systematic underutilization of information. In community, we have the systematic full-utilization of information. Property is harmful to everyone and it holds humanity back from its adaptation to an optimized state of fulfillment.

> **INSIGHT:** *Property is a "right" against the rest of the world.*

The flow of propertization generally follows the following steps:

1. **Identification** of types of things that can be owned or otherwise registered as property.
2. **Claim** of property ownership by someone or some organization.
   A. Claim of property ownership submitted to property licensing authority (i.e., State).
   B. Acceptance of property ownership by State; property licensing by the State.
3. **Inspection** of property [for compliance] by property licensing authority.
4. **Conflict** about who is authorized to own the property.
5. **Punishment** and censorship of those who are not authorized by the State to use the property.

2.1 Property ownership (property law-enforcement)

> *A.k.a., Ownership, property rights, property law, property license.*

When property is the law, then is ownership enforcement. Property ownership is created and accounted for at the State level in the form of property law-code. A property ownership law is any law that includes the idea of private and/or State-separate-Public (i.e., no commons) property. The idea behind all property law is the belief in property itself as a that is reified to be transferable [between legal persons] beyond the information and/or physical object itself. The property itself is the item (physical and/or informational) owned by some legal [property owning] person.

The owner of "property" (is a *concept*, *not object*) is the owner of the exclusive "rights" (is a *concept*, *not object*) granted by the owner who creates a:

1. **Civil contract** (contract initiated by [private] property owner).
   A. **Court of civil matters law-suit processing**, who resolves contract conflict in terms of prior trade and access agreements.
2. **State contract** (contract initiated by territorial [State property] owners):
   A. **Physical property law** (integrated into the State through "lawful" constitutions, statutory and regulatory codes, and defended by all (employees, employers, and consumers).
   B. **Intellectual property law** (integrated into the State through "lawful" constitutions, statutory and regulatory codes, and defended by all (employees, employers, and consumers).
      1. The owner of a *copyright* is the owner of the exclusive "rights" granted by "copyrighting".
      2. The owner of a *patent* is the owner of the exclusive "rights" granted by "patenting".
      3. The owner of a *patent* is the owner of the exclusive "rights" granted by "trademarking".

In the market-State, everything is someone's property, or someone's potential property:

1. **Private property** - property that is owned by one or more legal property owners.
2. **State property (a.k.a., "public" property)** - State "public" property.
   A. All land in the territory of the State.
      1. Some land in the territory of the State is rented out to legal property owners as private property real-estate.
   B. Some objects in the territory of the State (i.e., State buildings and government materials).
3. **Not yet anyone's property** - so, anyone's potential property.

Market-State property rights (as those rights defensible with State power) categories include, but are not necessarily limited to:

1. Information property rights:
   A. Copyright property rights.
   B. Patent property rights.
   C. Trademark property rights.
2. Object property rights:
   A. Land property rights.
   B. Land-fixed property rights.
   C. Movable object property rights.
3. State property rights

A. Job authority rights.
B. Public access rights.

There are two general categories of private property in the market-State:

1. **Intellectual property (a.k.a., information property, intangible property law):**
   A. Copyright.
   B. Patents.
   C. Trademarks.
2. **Physical property (a.k.a., personal property, tangible property law):**
   A. Real property (a.k.a., immovable property).
      1. Land.
      2. Objects fixed to land.
   B. Chattel property (i.e., objects not fixed to land).

A more complete description of the two types of property is as follows:

1. **Information property (a.k.a., intellectual property)**:
   A. **The "right" of privatization:**
      1. **Copyright property** - A copyright gives a legal person the ability to stop other legal persons from freely using one's productive information content.
      2. **Trademark property** - a right to prevent others from using one's organizational identity mark(s).
      3. **Patent property** - A patent gives a legal person the ability to stop other legal persons from freely using one's productive/useful invention. Patent rights refer to the control of the following aspects of production (by property owners):
         i. Manufacture.
         ii. Usage.
         iii. Sale (commerce).
   B. **The "left" of commonization/openization:**
      1. **Copy-left property.**
      2. **Patent-left property.**
2. **Physical property (a.k.a., land and objects):**
   A. **Real property (immovable property) -** real-estate landed property, usually only refers to land and land-fixed property. Real property is, the land and other objects. The building is "attached" to the land (by being built on it) and therefore, for legal purposes, becomes part of the land. Structures fixed to the land (and structures fixed in those structures fixed to the land) become real landed property. The house is legally inseparable from the land (unless it can be moved). Real property is also territorial property because it is property fixed on the land within some State's territory. Real property is created and licensed by means of (i.e., proof of ownership comes in the form of): titles, deeds, and other contracts.
   B. **Chattel property (movable property, personal property) -** usually only refers to tangible movable personal property. Cars and pens are choices of possession, or chattels. "Chattel" is the technical name for property someone can touch, pickup and hold, which is not land, and is not something affixed to the land, such as a houses. Chattel property is created and licensed by means of (i.e., proof of ownership comes in the form of): titles, purchase receipts, and other contracts.

There are several axiomatic forms of [private] property:

1. **Informational property** (a.k.a., intangible property rights, incorporeal property rights).
   A. **Information other than money as property:** Information about function, identity, documentation, and culture.
      1. *Private information property.*
   B. **Money as property:** Information about a universal trade/token accounting.
      1. *Private bank account property.*
2. **Physical property** (a.k.a., real property rights; tangible property rights).
   A. **Objects other than means of production as property**: Objects that consumers use.
      1. *Personal private property.*
   B. **Means of production as property:** Objects used to produce the objects consumers use (a.k.a., machine/resource capital).
      1. *Business property.*
3. **Human property** (a.k.a., labor property rights).
   A. **Humans as labor for other humans:** Humans used to produce the objects consumers use (a.k.a., human capital).
      1. *Employee property.*

There is informational property, just like there is physical property. The "copyright" is like the "deed/title" to a house. It says "you" own the house. If "you" want, "you" can rent the house out to someone, and that rental agreement is the 'license'. Copyright is different than a "deed" - it concerns information only - in that it is possible to license the material in different ways, to different people, all at the same time. Like a "deed", though, "you" can sell ownership of the property to someone else, and that would be called a "copyright transfer".

In society, there are two market-based population dimensions in concern to copy protection (i.e., copyrights and patents) and licensing (i.e., licensee and licensor):

1. The producer-user dimension:
   A. Producer - the legal property rights of owner (i.e., licensor).
   B. User - the legal people/organizations that want to use the property (i.e., licensees).
2. The economic production sector dimension:
   A. Copyrights and patents - Works of artistic, engineering, and exploratory expression.

2.1.1 Property rights

A.k.a., Proprietary rights.

Under market-State conditions, there are "rights" granted to property owners [by other property owners and by the State]. Property rights refer to "control of" some thing directly, and/or, control of another's freedoms. Where there is property [ownership], there is the right to engage a larger force to defend one's right to exclude another from using one's property. In a community-type society there is no tradeable property. However, in societies where there is property, and profit is derivable from property, then individuals are incentivized to withhold and restrict potential advancements and other useful products for their own selfish gain. Intellectual property, copyright, plagiarism, and even many cases of cheating, are reflective of an environment where there is little global cooperation and little facilitation of intrinsic motivation. In the market, property rights are privileges given from authority for some prior work.

A user/owner can have access to and/or control over the following potential freedoms:

1. See (is it secret?).
2. Perform, run, or use.
3. Copy.
4. Modify.
5. Distribute (including commercial).
6. Display.

Once someone has the "rights" of control over a property, they may then grant the "rights" of control over the property to another as a trade (i.e., they may contractually "license" some or all "rights" to another legal entity). In general, the State grants "rights" to the producer and to the consumer, such that if there is a legal "rights" violation, then legal "rights" actions can be taken. In this way, the State has the ultimate "right", because it creates and presides judgement when there is conflict.

"Rights" are not objects; they are a social construction. Copyrights and patents are choices in action - to engage in legal-State action against someone who takes more freedom than permitted by the "rightful" [proprietary] property owner. As choices of action, property rights are equivalent to a debt - the copier then owes a debt to the producer for taking more freedom than allowed; whereupon the property owner can engage legal property proceedings (using State resources for conflict resolution of the property-based social conflict) in hopes of acquiring financial damage and/or an injunction on continued copying.

The idea of property "rights" (in a market) provides users (among the population) two sets of consumer product "rights:

1. **The "rights" to purchase "rights of title"** to the product (and use per various State-overseen agreements; e.g., patents, copyrights, etc.).
   A. Producers can sell the product for a one-time (or multi-payment fee) and forget it; it becomes the users (with State-overseen "rights").
2. **The "rights" to purchase rented access to some of the "rights of title"** to the product (and use per various agreements; e.g., terms of service, patents, copyrights, etc.).
   A. Producers get continued fee compensation for a users continued access to, and/or the producer servicing of, the product.

2.1.1.1 Property rights durations

A.k.a., Entering the public domain.

In the early 21st century, most copyright and patent laws set a fixed number of years dedicated to the "right" of private ownership of information. Given current copyright laws, all copyrights and patents (intellectual works/labor) will eventually enter the common public domain in some fixed number of years after their creation. In this way, the "rights" are said to be turned over to the public (Read: public domain) for the public good of innovation and growth. Works in the public domain (P0, CC0) are not protected by copyright acts (i.e., copyright laws). Anyone can freely use them (in a copyright sense) without obtaining permission or paying a copyright fee. Anyone can even edit, adapt and republish these government/public works without permission. Anyone can adapt them and release them under a restrictive (non-commons) copyright license.

The time period for copyrighted material to enter the public domain fluctuates by country and by the politicians and corporations in power at any given time. In the United States, for example, copyright extensions have been increased approximately eleven times over the past forty years. Specifically, the Sonny Bono Copyright Term Extension Act of 1998 (SBCTEA) extended copyright protection for present works and all future works in the United States by twenty years. However, in the United States, material published prior to 1923 is entirely in the public domain. Please be aware that there is no single go-to reference guide for identification of public domain works. If such a list were generated, it would constantly be changing because copyright standards, rules, and laws are in a constant state of bureaucratic flux.

Today copyright laws are partially standardized through international and regional agreements such as the Berne Convention and the WIPO Copyright Treaty. Although there are consistencies among nations' copyright laws, each jurisdiction has separate and distinct laws and regulations covering copyright. National copyright laws on licensing, transfer and assignment of copyright still vary greatly between countries and copyrighted works are licensed on a territorial basis. Some jurisdictions also recognize moral rights of creators, such as the right to be credited for the work. Depending upon jurisdiction, copyright law still applies when copyrighted material are used in educational settings. The fair use of copyrighted material in educational settings is known as educational fair use doctrine and permits limited utilization of copyrighted material without the owner's permission for use in education and research settings. In order for Fair Use to apply, an educational institution and their learners must be using the material for non-commercial, educational-only purposes.

Note that there is one notable exception to "rights" of ownership not entering the public domain after a fixed number of years. The Italian State has a Cultural Heritage and Landscape code, which grants public institutions in the country who own works of cultural heritage the ability to request concession fees for, or outright bar, commercial reproductions of important artworks, regardless of their copyright status.

When intellectual property expires, or there is no patent/copyright at all, then that opens up the production so that anyone else can copy, sell, and adapt and re-privatize an invention or documentation/record. When a patent expires, the information enters the "public domain", where there are no "rights" [granting private property ownership].

Of additional note, to acquire public domain content, it is typically necessary to acquire that content from a valid public domain repository. Note that in the 21st century, there are businesses online that take public domain content, and then, create higher resolution versions of it, and index the new versions on their servers, whereupon they claim copyright over the "improved" formerly public domain content now on their servers. This technique is typically legal, but has also been named "copyfraud". These businesses do not claim that they own the copyright of the public domain versions, but they do claim copyright over their versions.

The property rights in a market-State economy involve:

1. **Personal (individual & family) property rights (family property rights and law)** - owned by a legal person or family. The power of "invitation".
2. **Civil (organizational) property rights (contract property rights & law)** - owned by a partnership of legal persons.
   A. **Market organization (i.e., business, partnership of legal individuals and families)** - individuals and families as legal property owners of a partnered (Read: partnership) organization.
      1. What do competing entities in the market own, and what do the do with what they own survive and win [profit] with trades?
   B. **State organization (i.e., governmental)** - owned by some partnership within the territory, such as: owned by a political party, owned by a democratic participating public and political parties, owned by faith, owned and coordinated by members of community, etc.
      1. What is the State authorized to do?

The property rights to land (and, buildings thereon) can be either (i.e., the terms of the duration of the "rights" license is either, or):

1. **Freehold** - "you" permanently (qualified by continued State-tax payments) own the property and the land the property sits upon.
2. **Leasehold** - "you" temporarily (qualified by continued State-tax payments) own the property and the land the property sits upon. Once the term [time period] of the lease is complete, its ownership "rights" will return to the original owner, who is either:
   A. Another private property owner.
   B. The State.

2.1.2 Property [rights] licensing

A.k.a., Permitting, permission, invitation, allowing, authority giving property rights, property licensing, proprietary rights licensing, property rights agreement, property rights contracts, property rights boundary restrictions.

A property can have its "rights" licensed to another legal property owner. A license is a permit or allowance of use of some item of property by the property owner. Licensing is the legal term used to describe the terms under which people are allowed to use any property (e.g., from copyright to trademark, and from the driving of an automobile to the installation of an air conditioning unit). Licenses permit or deny the creation, construction, or use of something. Another word for "license" is "permit". Effectively, a license is a permit to do and/or have something. Licenses are gatekeepers to access. Licensing is essentially an agreement between two legal persons. Licensing agreements allow people to control property. A licensing agreement creates a business partnership. It identifies how the partners enter, what all the partners agree to, and possibly, how the agreement can be dissolved. Effectively, property licenses govern how others – besides the originator – can use, modify, or distribute information, software, and resources (or resource compositions) .

For users, proprietary systems (Read: permissively closed; e.g., territory, hardware, software, etc.) generally require purchase of a license to use (by payment of a one-time fee or recurring fees), and the source code (design) is typically hidden from them (i.e., hidden from the users). Proprietary software, for example, is also called closed-source software or commercial software. The copyright limits use, distribution, and modification, imposed by the copyright holder's publisher, vendor, or developer. Proprietary systems remains the property of its owner/creator and is used by end users under predefined conditions (e.g., terms, contracts, etc.) usually defined in a license (for use). Alternatively, open-source socio-technologies are available free of cost to all in community; whereas, proprietary software is generally not free of cost.

Licensing is a market-State verb that means to give or grant permission on behalf of a market and/or State entity. The noun license (American English) or licence (British and many other places) refers to that permission as well as to the document recording that permission. A license may be granted by a party ("licensor") to another party ("licensee") as an element of an agreement (market-State) between those parties. A shorthand definition of a license is an authorization (by the licensor) to use the licensed material (by the licensee). A license may stipulate what territory the rights pertain to , and the length of time the license is valid.

A shorthand definition of "license" is:

1. A promise by the licensor (seller) and licensee (buyer) not to engage conflict resolution services if the property is exchanged as permitted and conditioned.
   A. An indication that conflict resolution will be engaged if the property is used *(by the buyer)*, or service is not provided *(by the seller)*, in some way without permission.
2. An indication of what the seller of the license will and will not do.
3. An indication of what the buyer of the license will and will not do.
4. A debt of payment (if the license has a fee).

In concern to the State, a license may be issued by authorities, to allow an activity that would otherwise be forbidden. It may require paying a fee and/or proving a capability. The requirement may also serve to keep the authorities informed on a type of activity, and to give them the opportunity to set conditions and limitations.

In concern to the market, a licensor may grant a license under intellectual property laws to authorize a use (such as copying software or using a (patented) invention) to a licensee, sparing the licensee from a claim of infringement brought by the licensor. A license under intellectual property commonly has several components beyond the grant itself, including a term, territory, renewal provisions, and other limitations deemed vital to the licensor.

A license is a legal fiction; it is a concept with only meaning, and does not exist in the real world, like a pen-object or house-object. In the market-State, licensing is simply another word for propertization. A licence is a grant of a permission of control over property. Herein, information and objects become property controllable through permissions, which if violated have consequences. In the market-State, technologies (physical and process), as well as land, and significant amounts of information, can become property licensed out to other legal persons.

Let's say someone owns some land. That owner can give permission to somebody to use the land in a legal way and/or subdivide it into a whole share of "lots" for people to habitat, dwell, and work within. The permission given by the owner is known as a license; and a license is written as a contract. The contract need not always be signed and agreed to in order to need to be followed. Websites' Terms of Service contract agreements are one example. Simply accessing a website necessitates compliance (as legal and reasonable) with the terms in the contract.

Licensing can be given in any of the following forms, some are better records than others:

1. Granted verbally (a.k.a., accepted verbally).
2. Granted in writing (a.k.a., accepted in writing).
3. Implied by law.
4. Supported by contractual consideration.
5. Exclusive, non-exclusively or sole.

The most common property license agreements are for:

1. Trademarks (information that conveys identity).
2. Copyrights (information that conveys documentation or culture).
3. Patents (information that conveys function).
4. Trade secrets (information that conveys competitive advantage).
5. Technologies (physical machines).
6. Real-estates (physical land and fixed buildings).
7. State license agreements (supposed to convey public safety).

In the market-State , there are really two dimensions to the concept of a "license":

1. **Market-based licenses (a.k.a., property licenses, civil licenses, service licenses, user licenses, business agreement, terms of service, etc.)** - those licenses given by other market owning property entities (i.e., legal property owners). A property owner can allow other legal persons to use the property in a state permissive/allowed way.

A. If this license is breached, then conflict resolution comes into effect. The resolution may stay local to the business, it may involve a commercial conflict resolution business, or it may involve the State.

2. **State-based licenses (a.k.a., State licenses)** - those licenses given by the State to create, construct, or use anything in a permissible/allowed way. This is effectively the technical process of quality assurance and certified/record of knowledge and skills verification.
    A. Herein, conflict resolution always involves the State.

NOTE: *When any organizations issue a licenses for a limited/temporary period of time (e.g., 1 year, 5 year, etc.), these types of licenses are typically called, "limited term licenses".*

In the market-State there are two types of licenses:

1. There are licenses for property (informational and objects) held by legal persons.
    A. Similarly, in community, there is an [authorized] access profile, which involves decisions about actual habitat service access and the available options for access. The program also shows decisions about past, current, and future potential access, and allows for user participation. In community, there individuals have an access profile with different [access] availabilities during the 3 self-sustained phases of their lives.
2. There are licenses for socially consequential actions held by legal persons (e.g., a license for driving, license for cultivating, license for policing, license for engineering, license for construction, etc.).
    B. Similarly, in community, there is certification for usage/operation of potentially dangerous systems by individuals accessing personal and common usage items (e.g., a car, saw blade, etc.) and accessing InterSystem [Contribution] Team items (e.g., medical center operations, electrical system operations, coordination operations, etc.).

The State licenses (Read: gives licenses to) people and systems, typically to ensure the safety of the public under conditions of competition and incentives of profit. Under the State people and objects may be licensed (i.e., can acquire a license to use, and/or be constructed and operated). Legal persons can acquire licenses, for example, a drivers license. And, objects can acquire licenses to operate, such as a motor vehicles registration, or hazardous chemical facility. The public can acquire a drivers license by certification (i.e., study and examination), showing/proving the ability to operate a motor vehicle safely. State employees often require licensing, which involves the taking of and passing a "public" exam. State licenses may be:

1. Permanent (as in, permanently associated with an object's or person's identity).
    A. Exist for one's lifetime (unless taken away because of an incident).
2. Temporary (as in, temporary permission to do something).
3. Need periodic inspection/review (as in, periodically inspected for quality and quantity).
4. Need periodic re-testing (as in, continued practice and certification requires another test to ensure sufficiently effective operational standards).
5. Need periodic education (as in, there may be new data available which ought to be learned, and/or there ought to be time dedicated to formally facilitated learning practices, for self-improvement).

NOTE: *Only the [copyright] owner (or their agent representation) can initiate a license agreement.*

2.1.2.1 Using licensing to reduce competition

I.e., The cost of oversight reduces access to production.

In the market-State licensing can be used to reduce competition. Licensing in the market is often a way to keep out competition due to the financial cost of the State licenses itself. Sometimes businesses lobby and advocate for regulation and expensive licenses, because they are very likely to be the only ones, or only one of a few businesses that have the finances to afford the expensive licenses. In this sense, there is licensing for two dimensions of societal operation:

1. **Safety** - regulation to ensure safe production and operation of societal systems.
2. **Competition reduction (objective-effect; may also be a safety reason)** - regulation to reduce access to production, by adding costs (financial and otherwise).
3. **Competition escalation** - alternatively, a State could increase competition by specifically setting a low cost for licensing.

In this context, State licenses (with or without a cost to the producer/user) may be required for:

1. Design and development of a product.
2. Manufacturing of a product.
3. Sale of a product.
4. Maintenance service of a product.
5. Quality assurance service of a product.

In general, all of these licenses cost money, and sometimes companies don't have the money or don't want to spend the money to pay for the license. In general, State oversight equates to expensive licensing; because, the oversight costs [a lot of] money.

It must also be noted that releasing content (partially or fully) open source is an often used tactic in competition in order to get a wider audience-adoption of the software on public platforms. The early 21st century Android operating system by Alphabet-Google is one example of this. In order to compete with Apple, Alphabet-Google made the Android operating system they produce open source, so it would be most accepted by all other smart-device producers.

2.1.2.2 Licensing types of property

A license simply means, "permission to use or not use, and punitive consequences for violation". A patent license is simply permission to use a new technology. A copyright license is simply permission to use some content/data. A "licensee" generally pays the [patent] owner for this permission, who tells the buyer of the technology/content what they can and cannot do with the technology/content.

The breakdown of licenses in the market-State follows as:

1. **Intellectual (a.k.a., informational) property** - belief in property; property exists.
2. **Real-property licenses** - belief in license, State/ public permission exists.
3. **Licenses with enforcement** - belief in forced defense of property; enforcement.
4. **Businesses partnership** - belief in trade of private physical property, and licensing of materials for profit.
5. **Patent and copyright** - belief in trade of private informational property, and licensing of ideas for profit.
6. **Hardware, machine licenses** - belief in licensing economic production (i.e., habitat service system) objects for profit.
7. **Software, computation licenses** - belief in licensing economic production (i.e., habitat service system) objects for profit.
8. **Citizen license** - belief in license for traveling through different State territories (e.g., "passport" license).
9. **Trademark license** - identity actualization requirement.
10. **Personnel qualification licensing -** ability actualization requirement.
11. **Production partnership** - agreement to work co-operatively and co-ordinated together for production of their own services (and products). Not to be confused with a business partnership.
12. **Privacy license** - Does the project collect data on users? Is the data related to some special category, e.g., health, etc.). Jurisdictions may have compliance regulations for that information. For example, the European Union State GDPR compliance regulations.

In the 21st century, there is pre-transition (closed) and direct transition (open) licensing:

1. **Closed-source** (proprietary, permissioned license restrictive license):
   A. Hardware - patents (all rights reserved).
   B. Software - copyright/patent (all rights reserved).
   C. Documentation - copyright (all rights reserved).
   D. Expressed content - copyright (all rights reserved).
2. **Open-source** (community, commons license "lefted" with reciprocity):
   A. Hardware - open patenting (patent-left).
   B. Software - open patenting (patent-left) / open copyrighting (copy-left).
   C. Documentation - open copyrighting (copy-left).
   D. Expressed content - open copyrighting (copy-left).

In community, individuals have a community access profile.

1. **Community access profile** - profile of access throughout the network to habitat services. Including a habitat residency profile within which a set of habitat rules are agreed to.
2. **Identity license** - identity actualization requirement.
3. **Personnel qualification licensing -** ability actualization requirement.
4. **Production contribution** - agreement to work co-operatively and co-ordinated together for production of their own services (and products) in a coordinated and contribution-based manner.

2.1.2.3 Possible licensing types

Licensing is the process of giving another permission to have and/or use something, wherein the "owner" of the defensible "rights" of the "property" gives (permits) some "rights" to another sub-hierarchical property owner. Technically, all property licenses are permissive, because they, by permission, give "rights" to another [legal property] owner that the [legal property] owner higher in the hierarchy has available to give. Hence, there are the following types of copyright licenses:

1. **Copyright and proprietary licenses (a.k.a, permissively closed licenses; exclusive license, "right" licenses, copy-right licenses, proprietary**

licenses, closed system licenses, closed-source licenses, etc): These are rights granted by law to the creator of "original" works of authorship (e.g., text, music, imagery, software, etc.) that allow control over the use of their "creations". Proprietary or "closed" licenses restrict the use, distribution, and modification of the copyrighted work unless specific permission is granted by the copyright holder. These licenses are not "permissive" in the open-source context; rather, they restrict how the work can be used by others. These licenses impose authority based restrictions on how work can be used. This license is closed and/or private; exclusive to those given individual permissions by the owner in the form of a sub-license [to use, make, sell]. Permission is required to use, make and sell the object or information.

A. Effectively, there is an owner with complete rights (given the contract and jurisdiction) to use State resources to prevent copying, modifying, and redistributing.
B. If there are permissions beyond those necessary for safety, then there is likely an unnecessary boundary of restriction being placed upon the creation and sustainment of community.

2. **Copyleft (a subset of open licenses, not permissively open):** Copyleft licenses ensure that works and their derivatives remain open but require derivatives to be shared under the same or compatible open terms. However, they require that any derivative works are distributed under the same or compatible license terms, ensuring the work and its derivatives remain free and open. In other words, copyleft licenses still impose at least one demand - copyleft licenses require users to distribute derivative works under a license that offers the same, or sufficiently similar, rights as the original work. This requirement does not make them "permissively open" in the usual sense, as they impose conditions to maintain the openness of the work (i.e., they maintain the work in the commons). The term "copyleft" reflects the requirement to keep the work and its derivatives open, contrasting with the term "permissive," which usually refers to open-source licenses that do not have this restriction. Left licensing means that for the duration of the copyright, once something is placed in the commons, the "creator" cannot take it back and privatize it for their own private gain. Contributions made to the left commons cannot be withdrawn. This approach ensures that the deliverable remains freely available for both commercial and non-commercial use under the terms of the chosen copy-left license, so long as the license is maintained. No permission is required to use, make and sell the object or information. However, all users and redistributors are typically required (under the conditions of a copy-left license) to give appropriate credit, provide a link to the license (where applicable), and indicate if changes were made to the original work. Furthermore, if the deliverable is adapted, modified, or built upon, the resulting work must also be shared under the same or a compatible copy-left license, thus ensuring the freedom to use, modify, and share the work is preserved for all subsequent creations. Example of copy-left licensing in terms of creative works is the Creative Commmons By-ShareAlike license (CC-BY-SA), and in terms of software code, there are the GNU General Public License (GPL), Lesser General Public License (LGPL), and Mozilla Public License (MPL).
   A. Effectively, there is an owner with no right to use State resources to prevent copying, modifying, and redistributing.
3. **Permissive open licenses:** Permissive open licenses allow greater freedom with minimal requirements, usually just attribution. No permission is required to use, make and sell the object or information. These licenses, such as the MIT license or the Apache license, allow copyrighted works to be used, modified, and redistributed with minimal restrictions, often only requiring attribution to the original author. They do not require derivative works to be distributed under the same license, making them more "permissive" than copyleft licenses. The Creative Commmons By only license (CC-BY) is an example of a permissive open license. Something distributed with a permissive license can be taken by another person, adapted, and then, privatized.
   A. Effectively, there is an owner with limited right to use State resources to prevent copying, modifying, redistributing, and adding a copy-left or copy-right license.
4. **Public domain (P0) and Creative Commons Zero (CC0):** A public waiver of, or public permissive, license. Works in the public domain are not protected by copyright and can be freely used by anyone for any purpose. Not only does this involve a waiver of copyright, but it also naturally involves a waiver of the commons. The CC0 license is a legal tool that allows creators to voluntarily waive all their copyright and related rights, effectively placing their work in the public domain (commons), and then, taken out and privatized at any time. It represents the most permissive approach, as it removes copyright restrictions altogether. These

works are open to all to do with as they choose given certain societal safety restrictions. A public permissive P0/CC0 license "grants" users the freedom to do anything they want (within societal safety limits, of course). This "license" is not really a license, it is more of an explicit disclaimer of intention to enforce copyright. CC0 (a.k.a., CC Zero) is a public dedication tool, which allows creators to give up their copyright and put their works into the worldwide public domain, which they would enter into anyway given some legally set number of years. Public domain and CC0 are a waiver of market-State (copyright) property rights. Public domain could be considered a type of commons under market State conditions. However, if something is public domain, it can be adapted, and then, its adaptation can have a restrictive copyright placed on it. So, in this sense, something that was in the commons under market-State conditions can be taken and made part of something that is privatized, and State resources cannot be used to prevent that. The content has all its rights [legally State enforceable claims] waived.

A. Effectively, there is no owner and no on has the right to use State resources to prevent copying, modifying, redistributing, and adding a copy-left or copy-right license.

CLARIFICATION: *There is also a separate category of open-source licensing that is called, "permissive licensing". In this context, permissive means the permission to privatize derivations of an open-source original.*

2.1.2.4 License permissiveness

The most permissive license is a free license - the most free (Read: freest) license is the most permissive. Free licenses can be classified in two groups:

1. Free, as in, **not having a financial cost** (no necessary fees, human labor trade, or purchases). Permissions allow free of financial cost access.
2. Free, as in, **freedoms for the users**:
   A. **Closed** (no social freedom) - not open to copy and not open to redistribute without the given permission of the owner.
   B. **Openness freedom** (social freedoms) - open to copy and redistribute without permission. Open-source license that allows the work to be freely used, modified, and distributed.
      1. **Permissive open licenses** (freedom to remove from the commons) - users can privatize future version. Future versions can be privatized by users (ownership freedoms). Permissions allow future exclusion. Permissions may disallow commerce. Permissions may disallow derivations.
      2. **Not permissively open licenses** (no freedom to remove from the commons; a.k.a., left-licenses, "left" freedom, copyleft, share-alike, commons persistent, recursive open-source licenses) - open to copy and redistribution without permission, and future versions cannot be privatized. Permissions requires free future access.

Permissive licenses, such as the MIT License and the Apache License, only refer to the use, redistribution and modification of software. They do not require that redistribution of the software keeps itself free (in the commons; persists in the commons), and it does not require that open-source code be redistributed, which allows free software to become proprietary. Permissive licenses carry on the property market, because they allow future propertization (i.e., to make property) of information. This is a problem because, even if the original code/design remains free, if its subsequent developments become proprietary, in the long run, this can derail its use without purchasing the subsequent modifications. Copy-left only (a.k.a., commons share-alike) licenses apply language that require, in case of redistribution, conditions that ensure the software will keep its original freedoms, acting to prevent that a later version will become closed. Of course, all open projects need motivated contribution.

2.1.2.5 Competitive licensing and cooperative licensing

This system of competitive licensing (Read: Patent and Copyright State Departments) was designed to prevent socio-technical inventions/discoveries and individual media creations from being used or copied without potential for individual profit, and thereby, incentivize and facilitating scarcity-driven, profit maximization, and power-over-others types of behavior. Cooperative licensing moves property into the commons, where it persists in the commons. Cooperative licensing is a transition type of licensing that is likely to transition people and resources into a community configuration of society.

In community, there are no competitive contracts or competitive licensing; there is no extraction from the commons for personal gain and advantage over others. In community, there is no private property, no trade, and hence, no forms of intellectual property, including patents and copyrights (none are written or believed to exist). In concern to the equivalent of trademark in community, individual and group identities are cryptographically secure.

2.1.2.6 Political economic orientations under conditions of private property

A.k.a., The (copy)"right" and (copy)"left". The closed-source (privatization) and the open-source (commons).

The political economic orientations under market-State conditions inclusive of private property presents a two-sided political economic movement:

1. Metaphorical "right" [licensing] movement:
   A. The right is movement toward private property (privatization), and thereafter, permission by an owner.
      1. This is an orientation toward private control through legal State enforced private property.
2. Metaphorical "left" [licensing] movement:
   A. The left is movement of the permissions into an "open" and "common" contribution-usage environment.
      1. This is an orientation toward community service control through a legal State enforced commons.

2.1.2.7 Information property violation

In general, the State-legal justice system's determination of guilt for copyright infringement results in either:

1. Damages paid (forced money transfer to property owner) and/or
2. Injunction (stop using the property).

2.1.2.8 Notice [of license] requirements

Licenses each have their own specific notice requirements; in terms of how the property owner must associate a specific licenses (and its contractual terms) with a specific item. Notices of a license typically include symbolic language (e.g., ©) and/or abbreviated text, as in CC BY-SA. Sometimes notices are required to include the title of the license and other relevant associated data. Technically, if "you" want your content in the public domain, just put it out there without a license.

> **NOTE:** *Sharing represents a truly difficult situation for someone who survives off of the protection of information as their exploitable product.*

3 Information property [law]

> *A.k.a., Intellectual property (IP), intangible property law, privatized information property protection, intellectual regulation, intellectual property regulation, information regulation, information property regulation, information privatization protection laws, private information property creation laws, profitable intellectual property creation laws, anti-community law, etc.*

Intellectual property (IP; a.k.a., information property) privatizes the results of intellectual activities (i.e., privatizes information). In the early 21st century, the expression of intelligence can be privatized (property-creation-ability) and profited from. Intellectual property law exists to create property out of information in order to profit. Intellectual property law exists to perpetuate trade and market competition by using State resources to punish cooperation. Intellectual property law refers to the terms of a copying agreement.

3.1 Intellectual property

What is possible in the information age is in direct conflict with what is permissible [in early 21st century society]. No social order, no matter how entrenched and how ruthlessly imposed, can resist transformation when new ways of producing and sharing emerge. With closed and "secure" content (i.e., "protected content") the "author" of the content is the sole creator and owner of said content. With open content the "participative creator" is in a state of collaboration with those who have come before as well as the community of users of the content. The community of users and the "participative creator" are all creators, and to an extent, accessors (or "owners") of the content. An open system involving openly participative content is a closer approximation to the existence of every living systems (in nature), and closed content goes a long way toward limiting the evolution of a community and causing unnecessary inefficiency (and suffering) in the world. Closed content does not account for the fact that the "participative creator" of the content would have been unable to create the content in the first place were it not for their prior learning, informed by the earlier work of many socially participative others. Fundamentally, information does not have the same [spatial] scarcity potential as materiality, unless imposed by force of violence by other humans. Therein, in order to optimize societal services for all individuals, material resources coordination should not be imposed by force of violence. In reality there is impermanence in everything; to keep a permanence when openness is essential is folly.

If "we" are restricted from sharing, then "we" are effectively restricted from navigating this real world together in common. And with this realization in mind, we ought to ask ourselves, who benefits and profits from the barriers are barbs raised to reduce cooperation?

Possibly the few, though also, possibly no one.

The pervasive culture of turning everything and anything into a commodified piece of property, a "commodity", that can be artificially restricted, bought and sold, is squeezing the space for and awareness of community (and of common access). Exploitation for private gain has systematically diminished the commons. This is happening not only in the case of tangible life support services and natural spaces, but also with more intangible things such as ideas and information, now increasingly referred to as "intellectual property". If a market entity can own an idea, and have that ownership forcibly defended, then the entity can stop progress on that idea and its synthesis into new ideas and new designs. Therein, there is potential profit for market entities in all forms of property. Further, there is profit [for the few] in conditioning the perception that space and time are "ownable".

When there is a claim of ownership, then there is the legally enforceable right to possess. Property is "rightfully" defended. When ownership is rightfully defended, then the idea of cooperative access is not understandable.

Intellectual property (IP) represents the "rightfully" enforceable enclosure and control of that which was designed and discovered, of what is essentially, just information. Intellectual property is claimed to include: patents; copyright; industrial secrets; and trademarks. Intellectual property is the exploitation of an idea for profit or social recognition. The moral claim for intellectual property is that an inventor has an exclusive, enforceable "right" to his/her useful, novel application of an idea, while an author or composer has such a "right" to his/her original work or expression. Those who believe in IP generally insist that what is owned is not an idea, per se; but, it's hard to make sense of that assertion since an application or expression of an idea is itself an idea. Hence, in the real world, IP is about the ownership of ideas, which are equivalent to thoughts, which are also, just information.

> **INSIGHT:** *The provenance of an idea is completely irrelevant to the evaluation of its rationality and/or truth. And in fact, in social discourse, often times the provenance of an idea can bias critical examination.*

The term "intellectual property" itself is a marketing term; it is ownership jargon created by powerful States and market entities; it has no actual meaning when critically examined. Intellectual property is the doublethink encoding of the idea that the intangible thought of a design, itself, can be property. For all practical purposes, the term "intellectual property" is identical to the statement "thought is property" (or, more precisely, "some thoughts are property"). Yet, there is no scarcity in thought -- thought moves through our minds and we can replicate it with our minds. Logically, how can there be exclusive control over, and access to (Read: "property" to), something intellectual? There can't, it's mental (herein, "mental" is a double entendre).

"Intellectual property" isn't even "property" in the traditional market philosophy sense of the concept. For example, two or more people cannot use the same pair of socks at the same time and in the same respect, but they can use the same idea—or if not the same idea, ideas with the same content. Ideas can be multiplied infinitely and almost costlessly; they can be used non-rivalrously. When someone articulates an idea in front of other people, each now has his/her own "copy." Yet, the "original" articulator retains the idea. In fact, one could go so far as to say that there was even productive effort on the part of the attendee who had to do the work of listening and integrating the idea, which was simply projected into an environment.

At the level of an information system, [digital] information isn't depleted or consumed by usage. When shared openly, [digital] content is added to over time. [Digital] information is non-zero, it provides benefit to more people without taking anything away from the composer.

Patents create legal monopolies. Historically, patents originated as royal grants of privilege. Copyright originated in the power to censor. One of the reasons someone might take a patent is to collect royalties. Therein, something renewable by its very nature (i.e., information), is now made non-renewable, externally controlled by its "owner" with the right to protect its property backed up by political property enforcers (e.g., the police & military).

From the perspective of "property rights", and in practical terms, when one acquires a copyright or a patent, what one really acquires is the power to ask the government to stop other people from doing harmless things with their own property. Hence, intellectual property is inconsistent with the "right to property".

Intellectual property does not stimulate innovation and it does not reward innovators. It is important to remember here that it is the owner of the idea (being an abstract piece of property) itself, who is rewarded, not the individual(s) who designed or discovered it - sometimes they are the same, and sometimes they are not. Intellectual property can be bought and sold, and often times companies write 'intellectual property clauses' into employment contracts, which give the company rights to works created inside, and sometimes outside, the scope of employment.

Intellectual property is a form of State-market protectionism. Intellectual property prevents others from using ideas that could benefit everyone. Fundamentally, information is a financial asset to the business that owns it. Some States even go so far as to have "culture reproduction laws". These States have code entitled something akin to, "Cultural Heritage and Landscape Code", which grants the State (a.k.a., "public institutions") in the country the ability to request concession fees for—or outright bar—commercial reproductions of important artworks, regardless of their copyright status.

In the market, "downstream" discoveries, namely marketable products, often depend on "upstream" discoveries in basic research. Yet the former can be more easily patented than the latter because they are more tangible, and so companies that operate downstream may be able to benefit financially from discoveries made by not-for-profit institutions upstream. And, much of the upstream research is government subsidized through threat (i.e., "tax").

Also of note, to some degree the patent system skews money and research toward things that are patentable, while limiting research on things that are not patentable and profitable, but would still help humanity. In other words, the system skews research priorities. The patent system also causes entities to hide or otherwise sit on research for competitive advantage. Abundance enabling technologies are not welcome when scarcity is useful for control and for profit. Software, for example, is reproducible at almost no direct financial cost, and hence, without property rights software would essentially not have a price. Price is only secured through government monopoly of force and the derivation of rights therefrom.

Copyright is the type of intellectual property that protects fixed, expressive works (novels, movies, songs, sculptures, paintings, programs, maps, charts - "creative works" as a fixed and original expression of authorship). Copyright is a restriction on freedom given from those of privilege. Sometimes individuals have to use other peoples words to make their points or visualizations because those most precisely meet objectives; copyright takes from an individual (and humanity as a whole) the ability evolve and adapt to the extent copyright to restricts sharing. Copyright is a government granted privilege by those themselves who have privilege to be authorities. Copyright restrictions exist to defensibly protect some peoples competitive economic positions in the world (i.e., copyright protection); they are a restriction on others backed up by violence. When a copyright is enforced, freedom of behavior is artificially restricted by the value of violence; because, the enforcement is enforced (i.e., backed up by violence). Copyright is a strong negator of the value of freedom [of expression], because it gives some human or organization the legally enforceable ability to control everyone else's actions by inhibiting the publishing or re-publishing of information...without the prior consent of the "copyright" owner. Copyright, intellectual property, etc. are all forms of relationship based on power-over-others; wherein, one individual or some group has the power to restrict the sharing of others, and even punish them if they do share.

Imagine a situation where some group or individual has a patent on a technology (e.g, energy technology) or copyright on a design that could move all humanity forward, and they choose not to use it, because they have other (energy) investments that they want to continue to make money on. Conversely, all of humanity becomes the beneficiary when useful information is shared. When the requirement for a monopoly on force is removed from the conception of government, then there may arise a structure to facilitate the coordinated sharing of information and controlled operating of services for the highest potential benefit of everyone in society.

The world's first copyright law, the Statute of Anne, was enacted in England in 1710. Exercising its power under the newly adopted Constitution to secure the rights of authors and inventors, Congress passed an act almost identical to the Statute of Anne as the first United States copyright law in 1790. In the United States, copyright emerged in 1790s, and one year later the authorities ratified the United States Constitution. The same people who ratified the constitution also passed the first copyright act. That act had 1308 words. In 2014, the act has approximately 79602 words. If all the statutes administered by the United States copyright office were examined then there are over 130000 words, not including all the various circulars and regulations the copyright office issues.

In early 21st century society, "creative works" are registered when financially and otherwise possible [with an authority] or through a very special mark (e.g., "©", which is supposed to signify something meaningful). What does registration or marking imply? In the intellectual property system, holding a patent or a copyright means you hold an enforceable, violence-based, and artificial monopoly over that property for some duration of time. From the encoding of IP into a socioeconomic system, consumers get [at least] all of the following:

1. They have to pay monopoly prices.
2. There is reduction of choice.
3. The sharing of information is artificially limited.
4. Progress on an idea or technology is blocked.

Owners get all of these benefits because other market entities cannot use and improve upon the creative idea [without payment and/or permission].

The neologism "creators rights" are a slap in the face against those who create for the joy and appreciation of the creative experience itself. "Creators rights" are not creators "rights" at all, but the "rights" of business and State entities toward maintaining social power and economic profit.

Intelligence seems to involve the intellectual accumulation of something, and almost every concept we come across is an intellectual accumulation. We realize that we have all "stood on the shoulders of giants" to accomplish what we have accomplished. Knowledge that has come before us has permitted us the opportunity to understand that which we now experience and create that which we create. We are all benefactors of others efforts, of others learning. For example, the knowledge that led to the 'car' was developed over/by thousands of generations of discoveries, not by a single individual. The argument that anything which can be manipulated becomes property is an unfortunate and inaccurate one. It's like saying, "We made this car, but because I painted it, it now belongs to me". An industrial manufacturer is simply adding paint to the work of others and claiming ownership. In truth, no one "invented" anything, all creations are a series of discoveries and arrangements/ modifications.

When a robot constructs and pieces together a car, then there is no longer human labor involved. If someone wants to argue that the output of an automated system is property, by extension, then it must be asked, "What about everyone else who contributed to it along the way, why are they not part owners as well?" This leads someone to the conclusion that we are all owners or not-owners at all. Anything in-between is a contradiction.

The very idea of intellectual property is a detrimental social construct. And yet, intellectual property is not just a particularly destructive and unjustified form of property, all "rights" to property are destructive and unjustified as is argued elsewhere in this document. (Tremblay, 2010)

Fundamentally, the encoding of the idea of intellectual property into a socio-economic system is highly likely to orient society in a direction opposed to human fulfillment.

In reality, ideas live in an intellectual commons where no one needs the permission of another to access and apply such information. The idea that ideas and replicable creations are property, and that using and building upon on others' ideas always requires permission, is insane.

In the real world, "intellectual property" is fraud and coercion. "Intellectual piracy" is not stealing (Read: not theft); why do you think they call it 'file sharing'. Sharing is real life. Sharing is synergy. Sharing is a means of life coordination. If sharing is piracy in a pejorative sense, then doublespeak is present and such a society is a criminal-society at its foundation. Therein, "anti-piracy" is likely to mean, in fact, the enacting of harm and obstruction against those who share and cooperate for mutual benefit. And, where there is violence done to reduce and obstruct sharing, then there is likely to be found equivalent exploitation and enterprising profit. In such a society, censorship and inefficiency become profitable. In truth, sharing is an issue of freedom and the use of violence to protect profit or reputation is the domain of tyranny (of law and government). Yet, there is the acknowledgement that sharing in the market is difficult when you have to live and survive by market rules (and principles).

When circumstances rob you [of that which is common], sometimes you have to rob it back. In a monetary system there will always be people looking to monetize information. There is no secret to health and fulfillment. Claiming information is "yours" is an ego-reward, while growth and competition lead to "ecological egocide". In community we free our minds from attachment identifications with our creations so that our creations may benefit everyone. When the sharing of knowledge and entertainment becomes a wrongful act and "defendants" are claiming great and "irreparable" injury, then something is wrong with fulfillment in that society.

If intellectual property does not exist, then what is an 'idea'?

- An 'idea' is a new combination of old elements. An 'idea' is the capacity to bring old elements into new combinations, which depends largely on the ability to see relationships. In the market-State, it also depends on education and financial purchasing power, as well as motivation.

Producing new ideas is a process of combining items already known and understood, in new ways.

> **INSIGHT:** *In reality, copyright is nothing less than tyranny, plagiarism is nothing at all, and intellectual property advances the special interest groups of that property. Today, ideas are exchanged at rates that no one, not even some just 30 years ago, could have imagined. We are fundamentally entering a more 'thought responsive' environment. Protectionist paradigms of thought, such as intellectual property, copyright, and even plagiarism and cheating are anachronistic to this new way of living.*

3.1.1 Protectionism and repression

> **INSIGHT:** *When there is secrecy, there is possible conspiracy, behind which there is possible malice. Secrecy enables corruption.*

In 2014, the message appearing on file-sharing websites censored by French telecommunications read: "You cannot access this website because it infringes on others rights". This message is doublethink - there is an entity out there (a server on the Internet) and "you" are not permitted to freely communicate with it because it is infringing on others rights. Fear has unfortunate ramifications. A fearful society sets limits on sharing, cooperation, and total progress. Censorship in access to information is hindering human progress at the very least, and is totalitarian at the worst. By artificially inhibiting access to available information and claiming said access infringes on others rights, then the question must be asked: Is there not a greater infringement of rights going on?

At a fundamental level, we as individuals learn [in part] by copying (or mirroring) one another, by adapting and modifying the works of others. Sharing is compatible with human fulfillment, copyright and patents are not. There is no way to enforce copyright in the non-commercial sense without abolishing a great many "human rights" -- the only way to dictate what two consenting people choose to transmit to one another is to remove their ability to communicate privately and freely. Hence, copyright considerations as a violation of another's "rights" are either voluntary on an individual basis, or completely unacceptable for a society oriented toward human fulfillment. There is no middle ground to be found and there never was. The entire concept of "copyright", in a world which includes the Internet, is a concept of financial-monopolization and protectionism that revolves around the idea that a market entity can prevent individuals from sharing with one another. Because, in the end, that's what ctrl+c to ctrl+v does.

The encoding of the idea of 'copyright' cannot serve society. Its encoding principally serves State-commercial industry -- if it is applied, it must be applied society wide, and when it is applied society wide, it inhibits society-wide change. It inhibits systematic social change and transparent economic change -- useful information is artificially restricted in its distribution and re-modification; it becomes locked up with elements of culture, privatization, fear, and threat.

3.1.2 Property and the commons

Ideas, information, and understanding are emergent and are an accumulation of that which has come before (i.e., all knowledge and understanding are serially developed). Almost everything we come across is an intellectual accumulation. Ownership of knowledge is thus illogical because we have all "stood on the shoulders of giants" to accomplish what we have accomplished. Knowledge that has come before us has permitted us the opportunity to create that which we now create. We are all benefactors of others efforts regardless of our beliefs.

Social and economic rules that restrict the sharing and evolution of knowledge limit humanities evolvement and individuals' betterment. If individuals are not permitted by an authority to make use of the knowledge they have acquired, due to external exclusive ownership involving threat of force, aggression or coercion, then that is akin to exclusive external ownership over elements of a being's internal cognition as well as their self-directed freedom in the material world. All claims to external ownership over another are an illusion.

Intellectual property could properly been seen as "property over mind" and "property over ideas", which leads quickly to an economic system that re-conditions (or "takes over") the minds of individuals. The word "intellectual" in the term "intellectual property" is really just there for obfuscating the encoded purpose of the concept. In other words, property over mental ideas, abstractions and thoughts (or "intellectual property"), is in fact a form (or possibly, the form) of mind control. The belief in "intellectual property" is, conceptually and metaphorically speaking, an economic system re-encoding itself into the minds of individuals in its society in order to perpetuate its own principles [at a more refined level]. A socio-economic system that encodes "property" in general, and "intellectual property" in particular, will find the economic system taking precedence in the social lives of individuals wherein the society as a whole is not oriented by social concern and well-being, but by abstract economic principles and power-oriented leaders.

The idea that someone might have useful information in his/her head (i.e., information that may be of benefit to oneself and others), and cannot use that information because it is defensibly owned by an "entitled" entity, is tragically absurd. Quite possibly, the growth and persistence of "intellectual property" comes from years of social and cultural indoctrination [of a belief system].

Quite predictably, "intellectual property" furthers the vanity, prestige and protectionism that are involved in coming up with (or "arriving at") an idea. Conversely, in community, we are pleased when our ideas are past around and shared, for it means that others have possibly benefited as a result.

> **INSIGHT:** *Non-disclosure agreements are all about the maintenance of a competitive advantage under a state of competition.*

3.2 Closed[-source] intellectual property agreements over real-world systems

> *A.k.a., Privatization, proprietary systems, proprietary licenses, commercial licenses, private property licenses, closed source, "right" source", etc.*

Proprietary systems (information and software) generally requires purchase of a license to use by payment of a one-time fee or recurring fees, and the source code is typically hidden from users. Proprietary systems (software) are also called closed-source systems (commercial software). Another term for a proprietary systems license is a commercial license. The copyright limits use, distribution, and modification, imposed by the copyright holder's publisher, vendor, or developer. Proprietary software remains the property of its owner/creator and is used by end users under predefined conditions usually defined in a license (Techopedia 2016). OSS source code is available free of cost to all, whereas proprietary software is not (Crooke 2016).

3.2.1 Closed [proprietary] licenses

Closed-source means some degree of:

1. Other humans are forbidden to adopt, adapt, or distribute, because of "your" personal decision about "your" personal property.
   A. Therein, through "licensing" there is property made out of work, wherein, it is possible to permit (or not, and/or charge a fee to) other humans to adopt, adapt, and redistribute it.
      1. Financial licensing in the market is a way to profit off of the labor of other humans using common heritage resources.
2. In the labor market, it is common for businesses and State organizations to have employees sign an IP agreement that gives the owners of the business or State enterprise some control of the employees current and future projects, whether at the business/State or with another organization.

3.3 Closed-source and open-source

In the market-State, Open/closed source refers to a legal [licensing of property] status. Open and closed source are particular ways of implementing and distributing something, enabled by jurisdictional legal language that gives a range of permissions for what people may do. Open source encourages the development of the commons, collaboration, and interoperability (among different market-State entities). Open source ensure no one has a monopoly over knowledge and others community freedoms. This also means that no one is prevented from attaining knowledge.

> **INSIGHT:** *When there is only open source and free sharing, there is no need to police every computer user for "pirating", life and the State are more efficient.*

Whereas open source means transparency and cooperation, closed source means competition, secrecy, and trade. When the condition of secrecy, of competition, is present, then individuals and organizations will withhold useful (or potentially useful) data for their own benefit. If you are the only person who has the data in a state of competition, you are highly likely to keep it secret.

"Open source" always means:

1. **Transparency of access** - Open-source means contributions are publicly observable.
2. **Contribution** - Open source is the application of the idea that it is possible to contribute to the whole (self and social) as a value, simultaneously.
3. **Free of cost** - Open source means "you" are giving away information and technology for free.
4. **Ephemeralization** - Open source fits with allowing the free social evolution and development of ideas.
5. **Economic efficiency** - Open source fits with efficiency in providing shared access to optimal fulfillment, by not creating artificial limitations on access, use, and development.

"Closed source" always means:

1. **Secrecy of access** - The source is private and owned by someone.
2. **Unnecessary duplication (economic inefficiency)** - Closed source efforts necessary entail duplication, because not everybody can be aware and involved in the closed source effort.
3. **Distrust** - Closed source efforts necessarily entail distrust, because not everybody can be aware and involved in the closed source effort.

In concern to software, for example,

1. When someone sells closed-source software, they only give the buyer the pre-compiled binary - the actual 1's and 0's that the computer actually understands. In general, executable (or, ready-to-run programs) are identified as binary files and given a filename extension such as .bin or .exe. Programmers often talk about an executable program as a binary or will refer to their compiled application files as binaries. Technically you could try to reverse engineer it but it is a challenge and not precise.
2. When someone releases open-source software they usually give you the pre-compiled binaries AND a copy of the source code (ie, the code that the programmers actually typed out in some form of programming language).

> **NOTE:** *Digital technologies have enabled open source, and many of the tools and infrastructure behind the Internet are open source.*

In a university context, open source is useful:

1. For both public sector and private organisations, open source makes good use of money, promotes freedom of choice and helps customers avoid becoming 'locked in'.
2. Open source makes it easy and most efficient and effective:
   A. To use and reuse software solutions, pooling efforts to create services that are interoperable.
   B. To add features where desired to open source software, the benefits of which are freely shared with anyone and for any purpose, so that everyone can benefit.
   C. There is a closer connection (no profit) between the users and creators.
   D. To increases efficiency.

3.1 Open[-source] intellectual property agreement over real-world systems

> *A.k.a., Open source, left-source, open innovation, open science, open society, open data, open standards, de-privatization, open-ness, commons access, "left" source, etc.*

An open source license means to give permission so that anyone should be able to view and modify the source code/design of a piece of software or hardware technology. Releasing content under an open source license, will limit the ability of the worker/organization-of-workers to assert their patent/copyright "rights". Open source licenses effectively mean that there are no longer proprietary "rights", only human fulfillment "rights" to common heritage resources.

Open-source is a value-action agreement set (Read: license) that may be applied to [the means of] production in the context of information/data, to the following effect:

1. On the demand/user side there is:
   A. No price/cost or restrictions on accessing the information (except for safety).
   B. No price/cost or restrictions on using the information (except for safety).
   C. No price/cost or restrictions on copying the information (except for safety).
   D. No price/cost or restrictions on adaptation and re-distribution of the information (except for safety).
2. On the production side there is:
   A. Must provide the source code/design file.
   B. Must maintain the same openness level of license of the information.

The idea of a "commons" relates at a fundamental level to what can be done with the two [consciousness applicable] states of the world (i.e., concepts/information and objects/matter/materialization:

1. **What can be done with concepts (information):**
   A. The *identifi-ability (a.k.a., identitification)* of the information.
      1. The *read-ability (a.k.a., observation)* of information. Note that to read something, it must first be identifiable.
   B. The *use-ability (a.k.a., application)* of information.
   C. The *copy-ability* of information.
      1. The *distribut-ability (a.k.a., distribution, re-distribution)* of information. Note that to [re-] distribute something, it must first be copied.
      2. The *alter-ability (modification)* of information. Note that to alter something, it must first be identified.
2. **What can be done with matter (materialization):**
   A. The *interface-ability (a.k.a., connection)* of matter.
      1. The *sense-ability (sensation)* of matter)
         i. The *touch-ability* of matter. Touch-ability is object-to-object contact.
         ii. The *EM*-ability of matter. EM-ability is object-to-energy-to-object contact by means of one object torquing the electro-magnetic ropes connected to another object.
            1. The *sight-ability* of matter. Sight-ability is object-to-energy-to-object contact by means of one object torquing of the electro-magnetic ropes connected to another the eye of an organism.
   B. The *use-ability (a.k.a., application/functioning)* of matter.
   C. The *re-configur-ability (a.k.a., re-configuration, re-working)* of matter.
   D. The *transfer-ability (transportation)* of matter.

In concern to 'readibility' in particular, there are typically three categories, in both the market and the State, which are:

1. Human readable *(real-world)* - has meaning to a human.
   A. Can a human understand it?
2. Legal readable *(a social construction)* - can be performed by the State (conflict resolution process).
   A. Can a conflict resolution authority understand it?
3. Machine readable *(real-world)* - can be performed by a digital machine.
   A. Can a machine understand it?

There are multiple domains of potential openness in society, including but not limited to:

1. **Open source (open-source):** is content, documentation, software, algorithms and calculations, and hardware that is publicly accessible so that anyone can see, modify, and distribute it as they see fit.
2. **Open data (open-data)** is:
   A. Publicly available data.
   B. Structured data that is machine-readable, freely shared, used and built on without permissive restrictions. Open data is, therefore, about freely available informational products.
3. **Open standards (open-standards):** are documented information that is findable, freely accessible, redistributable, adaptable, and reusable.
4. **Open science (open-source science):** is a commitment to the open sharing of software, data, and knowledge (algorithms, papers, documents, ancillary information) as early as possible in the scientific process. The principles of open-source science are to make publicly funded scientific research transparent, inclusive, accessible, and reproducible. The idea of open-science involves free availability of research information to encourage all-source contributions (as in, contributions from all available sources).

Open source requires the source to be distributed. The source [code] is kept visible and open to all. Open source does not reserve special "rights" to property owners; it only reserves the "right" to possible attribution. Open source is global cooperation. Items licensed under open source licenses grow the community commons and increase interoperability and integrability. Open source means the ability to use works in the commons together, usually in the form of adaptation, without legal property and personal restriction barriers. Open source means free for everyone; open source is a part of transparency. Open source software exposes all source code, features and functions for free.

> **CLARIFICATION:** *Open and transparent information environments are generally referred to as 'free', 'open source', and/or 'commons'.*

Open source is self-explanatory in its title -- it means that everyone gets to share [openly] in the source [code] - everyone has the opportunity to participate in a[n open source] system's innovation. In the market, open source means a royalty-free license to use. And, free means without trade or currency (without the market). Open source encompasses two related concepts regarding the way systems are developed and "licensed". They are codified in the "free xyz" (e.g., free software, public domain) and the "Open Source" definitions. "Free and Open Source" refers to systems that have been made available under a free market-State "license" with the rights to run the system for any purpose, to study how the system works, to adapt it, and to redistribute copies, including modifications. Open source is where anyone can see, re-use, and redistribute all or part of the source code of some thing's construction or operation. Fundamentally, an open source orientation allows for safe operation of a population wide control system.

If a design is released with any restriction other than attribution, then it cannot be classified as open source (actually free):

1. If it is non-commercial, then it is not open source.
   A. Some rights are reserved to the property owner(s), specifically, commerce.
2. If it does not allow for derivations, then it is not open source.
   A. Some rights are reserved to the property owner(s), specifically, derivations
3. If it cannot be re-distributed, then it is not open source.
   B. Some rights are reserved to the property owner(s), specifically distribution.

> **CLARIFICATION:** *If there are any special rights of access and usage by a property owner only, then the work is not open source.*

A design is open source if anyone is free to share it (commercialize it as a trade, or not), and adapt it. To facilitate transition, the design must be something akin to Share-Alike (SA) license; wherein, any change to the design must carry the same openness license requirement, same as the original. In an open source context, anyone who works with the design may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Open source works when a group of people all embrace a set of shared human goals (e.g., the fulfillment of human need) and establish an organization/platform for sharing based co-operation (which is, in part, based on trust). Open source allows for self-direction and self-organization, because it isn't restrictive of others access to and usage of the information to better themselves and the organization of which they are a part. Here, trust enables cooperation. The sharing of goals creates a reason for people to participate in the open-source project. Community design is a common interest, and therefore, its design must be open source and transparent to all.

Note that anything in the public domain (CC0, P0) can be modified and then copyrighted restrictively. The same goes for the CC-BY license. To maintain an open copyright, a Share-Alike conformant license must be used. CC BY-SA is intended for general sharing of documentation and "cultural works". CC BY-SA allows for the sale and redistribution of the "work" by others as long as the work is licensed in the same way. Alternatively, the GNU Free Documentation License (GFDL or GNU FDL v1.3) was made with the main purpose of being used for software documentation specifically.

The idea of an open-source community transition license (a.k.a., "left" license) is a license that allows others to do all of the following:

1. Use, modify, distribute and reproduce the content (e.g., source code) provided the content (e.g., source code) be kept open and available to all (a.k.a., the public).
   A. To reproduce means to create (i.e., produce and make).
   B. To use means to operationalize.
   C. To modify means to change (Read: adapt/mix).
   D. To distribute means to transfer a copy of the work from one legal person to another.
2. All derivatives must be transferred with the same free (community migration) license that came with the original work.
   A. Left licensing prevents the appearance of personal permissive restrictions being placed over a derived or modified work. Note here that citation ("attribution") is a global coordination process and is not considerable as a personal restriction.

The open-source licensing of information comes with several qualified freedoms (all together to be open-source, copyleft):

1. Freedom 0 - observe and study.
2. Freedom 1 - Use (a.k.a., adopt).

   A. For study.
   B. For personal use.
   C. For commercial use.
3. Freedom 2 - Share (a.k.a., redistribute, copy, publish, display, communicate, etc.).
4. Freedom 3 - Copy (a.k.a., duplicate).
5. Freedom 4 - Modify (a.k.a., change, adapt, mix, etc.).
6. Freedom 5 - Distribute modifications (a.k.a., redistribute).
7. Control 0 - Cannot have personal restrictions attached and cannot convert to private, closed license (i.e., share-alike, SA).

> **NOTE:** *Information, hardware, and resource access freedoms can be extended to societal as a whole. For instance, there are freedoms that every individual in community (as a type of society with software, hardware, and user) should have, and that no user should have restricted.*

The producer-freedoms side of an open-source, commons-oriented license involves the freedom to:

1. The freedom to share [“your“] changes.
   A. Note this freedom in terms of the GNU GPLv3 license is written as, The freedom-ability to share the changes [“you”] made.
2. The freedom to not have [“your”] productive work on the system, privatized out of the commons, ever.
   A. Note this freedom in terms of the Creative Commons license category, Share-Alike (SA).
3. The freedom to not have common heritage physical resources, where not essentially necessary for economic re-production, sold out of the commons.
   A. Note this freedom in terms of licensing, this clause is often written as, the freedom-ability to not have resources [“you”] configured and traded out of the community.

When a society offers all of these freedoms, it is sometimes called, a free society. When a software program offers all of these freedoms, it is sometimes called, free software.

Hence, after a full community transition license has been applied, the property owner no longer has the ability to choose a license type for the adaptations of their work other than a community one, which is a protocol for community standards.

The open-source licensing of objects comes with several qualified freedoms:

1. Freedom 0 - Contribution of a resource input or human effort to development and operate society.
2. Freedom 1 - Access as a team member - Habitat operations via habitat service teams.
3. Freedom 2 - Common access to community services and objects - common access to habitat services.
4. Freedom 3 - Personal access to community services and objects - personal access to habitat services.
5. Control 0 - Must reasonably operate an integrated decision system and reasonably follow its determined results.

3.1.1 Free and open source license

> *A.k.a., Freedom license, tradeless license, moneyless license, priceless license, etc.*

Each open source license states what users are:

1. Permitted do with the system components.
2. Their user obligations.
3. What users cannot do as per the terms and conditions.

> **NOTE:** *There are over 200 open source licenses available with a complex of permissive variations.*

Free and open source systems (hardware, software, etc.) is the common name for systems/software that is available under a licence that grants (gives permission to; is permissive to) the recipient the “right” to:

1. Use it,
2. Modify it, and
3. Distribute it (either the original or the modified version),
4. Free of charge or royalty, and
   Free of necessity to be Inclusive of the source [attribution].

> ** A key point here is that the source-code for the software remain open and accessible as the software is copied and adapted over time.*

As part of the principles of open-source, all “open-source” licenses permit commercial use. To restrict commercial use is to restrict the potential spread of important knowledge and technology required for human fulfillment.

> **NOTE:** *Free and Open Source Software (FOSS) – As the acronym suggests, FOSS refers to free and open source software that is provided to the user to copy, exchange, share, and use. Free/ Libre Open Source software (FLOSS) – FLOSS is similar to FOSS but allows more freedom to edit/ modify and distribute the software in original or modified version without any restrictions. FLOSS emphasizes the value of freedom, that is, with few or no restrictions, and it encourages the modification and redistribution of the source code.*

That something is open-source only means that the source code is accessible to everybody; it does not mean that the software is free, though usually it is. One of open source's biggest advantages is that access is usually free of cost, although some features and technical support may have a financial cost. Also, because the code is available to anyone who wants it, public collaboration can fix bugs, add features, and improve performance within a relatively short amount of time, if there is the motivation to do so. If there is a financial royalty to be paid because of any form of use or copy, then it is not a free access license.

Free access may means (no cost):

1. Access (use, copy, redistribute) without 'trade', without 'price', without royalty (royalty-free), open source.

'Free access' may mean (open source):

1. The code (project file) is open and available for study, modification, and redistribution.

'Free access' may, or may not, mean:

1. Without 'contract' (e.g., public domain, open source access license,).
2. With 'contract' (e.g., end-user license agreement, "Creative Commons" licensing, and the like).

> **NOTE:** *A license can be written as royalty-free, but still demand attribution.*

In the early 21st century market-State, royalty-free use/service licenses generally state that a royalty (price) fee does not need to be paid to use the content regularly. However, the license may state that the consumer pays a one-time fee in exchange (membership) for the right to use a photograph (or some other work protected by copyright, patent, or trademark) according to agreed upon terms, with no ongoing license fees due for further use. In this case, the term "royalty free" has come to mean membership cost, and not free of all cost, including that of membership.

In terms of cultural works, credit (attribution, citation) should be something "you" a user of societal resources, chose to do if "you" want:

1. To be nice to someone. Here, there is no one forcing any other to attribute/cite. There is no monopoly of force that can be used to stop another modifying and redistributing without citation. Here, there is flexibility of citation.
2. Are part of the InterSystem Team and are expected to follow citation protocols. Here, there is protocol, and potential consequences for violation of protocols. Here, there are protocols for citation. The team is accountable for logging and recording events, actions and decisions, and with concern to information, there is appropriate referencing between information sets (documents), appropriate identification of working group members, and appropriate identification of agreements and approvals.

3.1.2 Open-source licenses and patent provisions

> *A.k.a., Open patenting, open licensing, open copyrighting, etc.*

Some open-source licenses (e.g., GPLv3, Apache 2, etc.) include stated patent license provisions, which grant recipients a license to any patents covering the software product. Other open source licenses (e.g., BSD, MIT, GPLv2) do not include such patent provisions.

> **INSIGHT:** *Proprietary software developers have the competitive advantage of money-over-other; free system developers need to cooperate-for-fulfillment of each other.*

3.1.3 Open-source license persistence

> *A.k.a., Non-persistent open-source, non-copyleft licenses, etc.*

Generally speaking, open[-source] licenses can be either permissive or be copyleft licenses -- open source licenses can be divided into two main categories based on whether or not future copies of them must remain in the commons (i.e., persist), or if they can be privatized and traded after being placed into the commons (as in, CC-BY, CC0, P0, etc.). With copy-left licenses, the property is licensed so that derivations and copies can be shared equally into perpetuity (or, whenever a copyright expires), after which they become commonly open for integration into private property.

> **NOTE:** *Copyleft licenses are said to follow the "reciprocal effect", in that those who have benefited from the content are expected to reciprocate by sharing the content alike (i.e., in the same way, via the same copyleft license) by which they received the content.*

Hence, the two forms of "open/commons-based" licenses, based upon the licenses persistence (until copyright expiry) are:

1. **RECIPROCAL LICENSING (COPY-LEFT LICENSING) - persistent commons licenses** *(a.k.a., copyleft licenses, persistent open licenses)* require the sharing of derivations only with same license, share-alike (SA, reciprocity). These licenses "guarantee" that the licenses will remain open.
   A. These include, but may not be limited to: CC-BY-SA.

2. **PERMISSIVE LICENSING - non-persistent commons licenses** *(a.k.a, non-persistent open/ common licenses*) do not require the sharing of derivations to only occur under the same license, share-alike (no requirement for reciprocity). These licenses do not "guarantee" that the licenses will remain open.
   A. These include, but may not be limited to: CC-BY, CC0, P0.

   NOTE: *This division is based on the requirements and restrictions a license places on users. With few exceptions, all licenses eventually stop persisting and the content enters the public domain.*

The only difference between copyleft and permissive is that copyleft licenses are persistent (share-alike) since the same license applies to all the works that are a result of it. On the other hand, permissive licenses are not persistent. Also, copyleft carries a 'viral effect,' which means that when a derivative work is made by using two different licenses, out of which one is copyleft, the derivative work is governed automatically under the terms of the copyleft license. The same doesn't hold for permissive licenses.

A permissive open-source license is a non-copyleft open source license that guarantees the freedom to use, modify, and redistribute, while also permitting proprietary derivative works. Permissive open source licenses are often referred to as "Anything Goes Licenses", which place minimal restrictions on how others can privatize open-source components. That means that this type of license allows varying degrees of freedom to use, modify, and redistribute open-source content, permitting its use in proprietary derivative works. In some cases of permissive licenses, the license is essentially equivalent to the public, requiring nearly nothing in return in regards to obligations moving forward.

In concern to the transition of property over to community, there are commons reciprocity licenses (i.e., copy-left commons, open-source licenses): :

1. Hardware - open patenting (patent-left).
2. Software - open patenting calculations (patent-left) / open copyrighting code (copy-left).
3. Documentation - open copyrighting (copy-left).
4. Expressed content (film, audio) - open copyrighting (copy-left).

Conversely, closed-source (proprietary, permissioned license restrictive license):

1. Hardware - patents (all rights reserved).
2. Software - copyright/patent (all rights reserved).
3. Documentation - copyright (all rights reserved).
4. Expressed content - copyright (all rights reserved).

3.1.4 Common commons licenses

Creative commons is one of the most well-known commons (copyrighting) organizations with a licensing structure (and organization) designed for producing a legal contract that makes it permissible to share documentation and cultural works.

Commons licenses make claims about the openness of licensing. A fully open license grants a number of rights, as statements that, "recipients can":

1. Use the work or run the software (for a licensed purpose, any purpose, commons purpose, etc.).
2. Obtain the source code in order to study it.
3. Share and/or sell source code.
4. Share and/or sell work or software.
5. Modify, adapt, and improve sources to specific needs, making and sharing, and selling, derivatives.
6. Share the information alike (copyleft), or not.

The Creative Commons license is used for opening information creation:

1. Creative Commons [market-State identified] license - Free, easy-to-use copyright licenses provide a simple, standardized way to give the public permission to share and use your creative work — on conditions of your choice. CC licenses let you easily change your copyright terms from the default of "all rights reserved" to "some rights reserved." Creative Commons licenses cannot be revoked once issued. In concern to creative commons, it is in the re-sharing that restrictions are set.
   A. The Auravana Project has selected the Attribution Share-Alike (CC BY-SA) license. The Attribution CC BY-SA license lets others distribute, remix, tweak, and build upon "your" work, even commercially, as long as "they" credit "you" (*"a nice to have"*) for the "original" creation, and carry the license forward in their adaptations and copies.

Licenses similar to Creative Commons, which are also used for design and information creation of an open source nature, are:

1. GNU General Public License - modify, distribute, and charge people (with a few rules). GPL is a copyleft license. This means that any software that is written based on any GPL component must be released as open source. All code in a single program must be either be subject to GPL or not subject to GPL. To be clear, if a developer were to combine GPL code with proprietary code and redistribute that combination, it would violate

the GPL. Whenever someone conveys software covered by GPLv3 that they've written or modified, they must provide every recipient with any patent licenses necessary to exercise the rights that the GPL gives them. In addition to that, if any licensee tries to use a patent suit to stop another user from exercising those rights, their license will be terminated. What this means for users and developers is that they'll be able to work with GPLv3-covered software without worrying that a desperate contributor will try to sue them for patent infringement later. With these changes, GPLv3 affords its users more defenses against patent aggression than any other free software license.
2. BSD License - modify, distribute, and charge people with no restrictions on charging people.
3. MIT License - lets you do whatever you want, you just have to include license with software being given away.
4. Apache License - Able to use on copyrights and patents and doesn't expire. The Apache License allows you to freely use, modify, and distribute any Apache licensed product. However, while doing so, you're required to follow the terms of the Apache License.
5. The Microsoft Public License - a free and open source software license released by Microsoft, which wrote it for its projects that were released as open source.

The following are license selection relevant questions to facilitate determination of the optimal license for a project:

1. Will the project be used as a dependency by other projects?
    A. It may be best to use the most popular license in your relevant community. For example, MIT is the most popular license for npm libraries.
2. Will the project appeal to large market organizations?
    A. A large business, for example, will likely want an express patent license from all contributors. In this case, Apache 2.0 has you (and them) covered.
3. Will the project appeal to contributors who do not care if their contributions are to be used in closed source systems.
    A. For example, it may require a permissive license so that the company can use your project in the company's closed source product. In this case, Apache 2.0 has you (and them) covered.
4. Will the project appeal to contributors who do not want their contributions to be used in closed source systems?
    A. GPLv3 or (if they also do not wish to contribute to closed source services) AGPLv3 will go over well. This project may have specific licensing requirements for its projects.

Table 29. Contribution Approach > Market-State Licensing: *Licensing requirement comparison table. Superscript references: (1) Application needs to be licensed under GPL if redistributed with the GPL asset. (2) Library code modifications need to be licensed under the same license as the originating asset. (3) Usually requires a commercial license from the copyright holder. (4) Although much more permissive than an OSI license, some BSD based licenses, such as Apache V2, still have some copyleft materials..*

Capabilities (without	GPL	Dual-GPL	LGPL/MPL	Apache/ BSD
1) Download	Yes	Yes	Yes	Yes
2) Evaluate	Yes	Yes	Yes	Yes
3) Deploy	Yes	Yes	Yes	Yes
4) Redistribute	No[1]	Yes[3]	Yes	Yes
5) Modify	No[2]	No[2]	No[2]	Yes[4]

3.1.5 The best license for simulation-related content

The safest option when looking for images, music, models (objects), and code for social media marketing and simulation development are CC0 public domain (P0) licensed content. This licensing avoids the attribution/citation requirement of all (or, most) commons licensing. When developing a complex distributed simulation, BIM and other object metadata information is essential, but the identity of someone or some partnership that made some texture, model, or code, is potentially helpful, but not essential information. In fact, more often than not, it is irrelevant information that complexifies an already complex development process. Simulations require so much effort and resource that collecting irrelevant references can make development more challenging and time intensive than necessary. Of course, it is possible to include CC BY content (including CC BY-SA, but not CC BY-NC), if that content's source is mentioned in an attached document.

3.2 Information property agreements types

A.k.a., Intellectual property agreement types.

Intellectual property agreements restrict the sharing of information and progress among a population. Intellectual property (IP) is a legal tool wielded in competitive battle for profit through market advantage. Intellectual property builds protection around business-owned content and business-owned technology. Patents, copyrights, trademarks are aspects of the same thing -

information property "rights" held and traded by legal persons in an enforceable contract system managed by the State-authority.

There are four main intellectual property (IP) codes that allow for regulation of systems and service:

1. **Patent law (physical exclusivity rights)** regulates the market on behalf of producers of functional systems that are "novel" and "non-obvious", after the producer applies for protection from the a Patent & Trademark Office. Patents are monopoly protection for systems that have real-world functionality. Patents are packets of information about production put into public vision (likely, eventually, public domain), and they exist in large part, so the State knows what it is protecting, and who it is to enforce the "property rights" of. Patents enters into the public domain at the expiration of the term of the patent. As part of the terms of granting the patent to the inventor, States publish patents into public vision and eventual public domain. Except for secret patents. National patent secretes are patents without the transparency, without the patented specification showing up on the register. The fact that a patent's description is in the public vision does not give others permission to manufacture or use the invention during the life of the patent without permission from the inventor. A patent gives the owner the right to exclude others from potentially making, using, and selling the claimed invention (patent). A patent is a legal method for the owner of an invention to control how others use the invention. All patents are explicit licensing" in that they must be submitted in writing to the State, and approved. Patent law offers private property holders the right to State protection to forbid usage and copying without permission/trade by the owner; this protection extends to the "original works of authorship" which is "fixed in written/visual form/ medium representing a tangible physical creation" by the "State-private partnership" as a "functional technological creation". Note here that all things with functionality are "born" free -- patents are not received automatically when something is created. Creator interaction with the State must take place and a set criteria for approval must be met. No one can patent anything that is already in the public domain, or was published before.
    A. The requirement that functional inventions be "novel" and "non-obvious" are high legal bars that few inventions meet. Additionally, patents are very expensive to obtain and the process is quite complicated, usually requiring help from specialized lawyers. You must take affirmative steps to obtain patent protection for your hardware.
    B. An open patent is a patent that is freely shared with others under a copyleft-like license.
2. **Copyright law (informational exclusivity rights)** regulates the market on behalf of producers of "original works of authorship" which are "fixed in a tangible medium", such as books (literary), music, paintings, photographs, artistic things, engravings, models, and architecture. Copyright law "protects" against the dissemination of a work on information networks without permission or remuneration. Copyrights enters into the public domain at the expiration of the term of the copyright. A copyright is a legal method for the owner of some expressed content to control how others use the content. Copyrights may be implicit assumed (Read: implicit licensing) when a content creator doesn't explicitly identify a copyright. Copyrights may also be identified with licensing and associated symbols that explicitly (explicit licensing) identify permissions. Copyright law offers private property holders the right to State protection to forbid usage and copying without permission/trade by the owner; this protection extends to the "original works of authorship" which is "fixed in a sensible and communicable form/medium" (e.g., music, software, etc.), and "representing a privately tradeable expression of creation" (market, trade) by the "State-private partnership" (law; legal-State). Often, small alterations to public domain content can be re-licensed in a proprietary ("end-user" license) out of the public/commons domain and into restrictive permission licensing.
    A. While certain hardware elements might be creative, the creativity is often constrained by functionality, which prevents most physical aspects of most hardware from being protected by copyright. For example, the way in which parts of a 3D printer's extruder work together is governed by functional concerns. That means that it cannot be protected by copyright law.
    B. An open copyright is a copyright that is freely shared with others under a copyleft license. It is relevant to note that copyright law means that any of this-type of expressible content is created, by default, closed (proprietary); it is "born" closed.
3. **Trademark law (identity exclusivity rights)** regulates the market on behalf of producers with "source identifiers", which may include any brand names, product names, logos, or even the design and packaging of your product. Double check that a

market-State project's name does not conflict with any existing trademarks. If an organization uses its own trademarks in a project, check that it does not cause any conflicts.

A. While trademark law may protect the names, logos, and other elements that signal who the producer of the product is, in most cases trademark law do not protect the physical object itself.

4. **Trade secret law (business exclusivity rights)** - The primary objective of trade secret law is to protect a business' decision to maintain a valuable business secret from which it derives a competitive advantage in the market. Generally, a "trade secret" refers to information that derives competitive economic value (competitive advantage) from being kept secret, and is subject to reasonable efforts by the business to maintain its secrecy. In contrast to patents and trademarks, trade secrets do not require registration, never expire, and never become public (unless independently discovered by others or misappropriated). If, however, a competitor reverse engineers a business' trade secret, or independently develops the process, the original business with the trade secret has no recourse to prevent their use of the (formerly) secret information. Consider whether there is anything in the project that the business does not want to make available to the general public.
   A. Note that in some states trade secret laws include "negative know-how". Negative know-how refers to information discovered during development about what does not work. Therein, an employee who resigns from a business and joins a different business can be liable for not repeating the mistakes and failures of his or her former employer in the development of the same thing.
   B. Common trade secrets include, but are not limited to:
      1. Ingredient mixes (a.k.a., proprietary blends).
      2. Manufacturing methods.

CLARIFICATION: *Where functionality starts, copyright ends in patents. Copyright-based licenses don't apply to hardware as hardware is mostly functionalities. Within the functional domain (patenting), there is still copyright-based licenses for the documentation + design files = the source > of the hardware*

Copyrights and patents intermix in concern to software. Copyrighting software code protects the actual code itself, but would not stop someone else from creating their own code that implemented the same calculation method used in the copyright protected code. The calculation method would need to be patented (separately from the claim of copyright) to protect the property owner of the method of calculation.

To the extent that hardware may be regulated by one or more of these intellectual property (IT) codes, properly applying an open source license to the project ensures that downstream users can use the product within the bounds of the license. These regimes will not protect every element of hardware. Purely functional elements of hardware are not generally protectable by copyright. Other types of protection such as trademark and patent usually require creators to take active steps in order to obtain. As a result, the hardware for many functional open source hardware products will not be protected by any kind of right at all. Protection will begin to attach to hardware as decorative and aesthetic elements are added. While this protection will not extend to the functionality of the hardware, in some cases this protection will effectively control reproduction of the entire physical product.

INSIGHT: *Copyrights and patents are commodities that can be bought and sold like any other property – a house, a car, a pen.*

In the early 21st century, the following types of information may be privatized (Read: made into property):

1. **Hardware designs** - are patented (function and look).
2. **Hardware quasi-functional designs** - copyright, such as the hull of a boat.
3. **Software designs** - are copyrighted.
4. **Calculations** - are a patentable processes.
5. **Socio-technical processes** - are patented.
6. **Software language code** - is copyrighted.
7. **Images, text, audio,** and **video** - are copyrighted.
8. **Architectural drawings** - patentable in some jurisdictions and copyrightable in all.
9. **Logo and brand name/image identifiers** - are trademarked.

In the market-State, legally enforceable copy restrictions by State action include the following as possibilities:

1. **Hardware license, hardware usage and copy restrictions:**
   A. **Patent for function** - can license function, cannot copy/use without licensing the function.
   B. **Patent for design look** - can license design, cannot copy/use without licensing the design.
2. **Software license, software usage and copy restrictions:**
   A. **Patent for calculation** - can license algorithm, cannot copy/use without licensing the algorithm.
   B. **Copyright for [source] code** (software, text,

images, and animation) - can license [source] files, cannot copy/use without licensing the source files (i.e., files the product was created with; means of production of final product).

3. **Copyright license, communication (messaging) usage and copy restrictions:**
   A. **Copyright for expressed content (Read: messages)** - can license all sensed forms, including: text, image, audio, or animation.
4. **Identity license, identity usage and copy restrictions**:
   A. **Identity (trademark) usage and copy restrictions:**
      1. Trademark for preventing and resolving identity conflicts over one or more organizational-specific identifiers.

3.3 Patent law [for profit through property]

> **INSIGHT:** *Notice how, in the market, human lives, opportunities, and access are always discussed in terms of cost/benefit, rather than individual freedom and fulfillment?*

A patent is an issued license by a State authority granting right to restrict others usage. A patent, if issued by a State, allows the owner to prohibit others from selling and/or using a patented invention. A patent "protects" the property owner and their ownership "rights" over new inventions, processes, and compositions of matter (e.g., medicines, machines). Patents must be applied for at the State-level, by means of the submission of a proposal for review, which is either accepted or denied [by the State]. Importantly, ideas cannot be patented (because they are too general) - the invention must be embodied in a process, machine, or object. A patent is a government granted monopoly to use some system. Patents exist to make a profit and to control technology usage/transfer for private advantage. Patents must be applied for. Patents holders are the only party allowed to bring the product to market, and may license its use to others to collect royalty fees. A patent provides the "inventors" of new items with a source of ownership-profit by preventing others from making, using, or selling the invention for a specified amount of time. As a government granted monopoly, patenting grants the right to use the force of government to exclude others within the government's jurisdiction from:

1. Making (just making, or making for sale?).
2. Using (just using, or offering for use?).
3. Offering for sale.
4. Selling.
5. Importing.

There are effectively three types of patent licenses the government can issue (utility, design, and organism):

1. **UTILITY PATENT** - specific design functionality. Utility patents" are granted for "original" functions. Utility patents are issued for a useful, novel, and non-obvious method, device, composition, or system. Utility patents may be granted to anyone who "invents" or "discovers" any new and useful process, machine, article of manufacture, or compositions of matters, or any new useful improvement thereof. A utility patent provides protection for the way an object works and is used. This includes patents for new and non-obvious compositions, formulations, methods of use, and manufacturing techniques. Effectively, a utility patent provides profit for the way an object works and is used. Most filed patents are utility patents.
   A. **A utility patent represents the creation/ abstraction of property:** *How something functions becomes property; utility becomes property.*
   B. **Calculation patent** - For software, the patented invention is often a method of calculating something. A calculation is a utility item.
      1. **A calculation patent represents the creation/abstraction of property:** *A method of calculation becomes property; calculation becomes property.*
   C. **Composition patent -** For chemicals, the patented invention is a specific composition of ingredients/chemicals.
      1. **A composition patent represents the creation/abstraction of property:** *A mixture of chemicals becomes property; chemical mixes become property.*
   D. **Synthesis patent** - For materials (including, minerals, genes, organisms, and biological processes), the patented invention is a synthesized material.
      1. **A synthesis patent represents the creation/ abstraction of property:** *A new material becomes property; materials become property.*
   E. **Secret patent (national invention secrecy orders)** - specific design functionality that is too dangerous to make public. Technologies related to sensitive military applications are often made a State secret under secrecy orders. The patent application and all details become a [national security] secret, guarded under sensitive secrecy orders. This could be considered a temporary State-national public defense and safety issue. States have State secrets for: (1) competitive advantage; and (2) public safety. State-level public-safety technology secrets may be used licensed to State-defense certified organizations and personnel (i.e., licensed to those with a

"security clearance").

2. **DESIGN PATENT** - specific look and feel. Design patents are granted for "original" designs or articles of manufacture. Design patents may be issued to anyone who "invents" a new, original, and ornamental design for an article of manufacture. Effectively, a design patent provides profit for the way an object looks.
   A. **A design patent represents the creation/ abstraction of property:** *The look and feel of something becomes property.*
3. **PLANT PATENT (A.K.A., ORGANISM PATENT)** - this type of patent may be granted to anyone who "invents" or "discovers" and asexually reproduces any distinct and new variety of plant or other organism. Effectively, an organism patent provides profit for creation of a life (i.e., an organism).
   A. **An organism patent represents the creation/ abstraction of property:** *The code for an organism becomes property.*

NOTE: *A* **Provisional patent** *is given when a patent application has been submitted, but acceptance not yet complete.*

3.3.1 State patent license

Whereas copyright is an automatic right, a patent right must be applied for. A State patent license grants the property owner the ability to claim a legal suit of "patent infringement", which occurs when another person uses one's innovation without consent in a country where the innovation has been registered as a patent. Patent infringement generally has larger potential financial loss consequence than copyright infringement (because the State analyzed and issued the patent license).

A State patent [license] is requested by filing a written application at the relevant patent office. Every patent application must contain one or more "claims", or detailed definitions of precisely what is being patented. Simply, patents can cover technologies, materials, methods, and aesthetics.

3.3.2 Civil patent licenses

A.k.a., Patent-right licenses, right patent licensing.

A patent license is an agreement that lets someone else commercially make, use, and sell your invention for a specified period.

3.3.3 Open-source civil patent license

Open source patent licensing is the practice of licensing patents for royalty-free use. Collaborative innovation (is the same as "market innovation", and it) flourishes within communities of independent people that choose voluntary to cooperate.

3.3.4 Patent-left [open-source] licenses

A.k.a., Left patent licensing, left-patent licensing, patent left licensing.

Patent-left is the practice of licensing patents for royalty-free use, on the condition that adopters license related improvements they develop under the same terms. Re-distribution is allowed, but only under equivalent licensor terms. The concept of "left patenting" is not standard terminology in intellectual property law. However, it is possible to write an open-source left patent license, where a patent is made available for free use under certain conditions, similar to the principles of copy-left in software, such as writing in a condition requiring improvements to be shared similarly.

3.4 Copyright law [for profit through property]

In the early 21st century market-State, any media displayable work (e.g., text, image, video) that anyone creates is automatically protected by copyright (as permitted by the State). No explicit copyright notice is required; the copyright comes into existence because the work came into existence. If "you" want others to re-use and build upon "your" work, "you" - as the copyright holder - have to give others an explicit license (permission). A copyright, regardless of registration with the State, allows the owner to prohibit others from using the content (except for educational and other exempt purposes). Copyright exists to make a profit and control textual and image usage and transfer for personal advantage. Copyright is an automatic right that comes through the invention of some [piece of] property.

CLARIFICATION: *Copyright effectively means that others are forbidden to adopt, adapt, and redistribute that which is copyrighted without the permission of the copyright owner.*

Copyright gives a legal person who created text or image content the ownership ability to decide what others can do with the text and images after they observe it (with specific exemptions, such as education and critique). Copyright is designed, typically, to prevent text, images, and audio from being copied by creating a legal method of punishing those who do. Copyright "protection" extends to a description, explanation, or visualization of an idea or system, assuming that the requirements of copyright law are met. Copyright is designed to create and protect literary or pictorial expressions created by an "author". However copyright law does not give the copyright owner the exclusive rights to the idea, method, or system involved (if those are sought, then a "patent" is necessary). Copyrights are separate from patents, which is made clear by Section 102 of the United States Copyright Act (title 17 of the U.S. Code) clearly expresses this principle: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, ..."

> **CLARIFICATION:** *Suppose, for example, that an author writes a book explaining a new system for food processing. The copyright in the book, which comes into effect at the moment the work is fixed in a tangible form, prevents others from copying or distributing the text, illustrations, audio, and videos describing the author's system without the author's permission. But, it will not give the author any "right" to prevent others from adapting the system itself for commercial or other purposes, or from using any procedures, processes, or methods described in the book. If they "author" wants to prevent others from using the food processes without paying a "licensee" fee, then the author needs a patent.*

Copyright is the exclusive legal right to use, copy, and distribute a documented or creative work. A copyright is a legal [State authority] process used by someone to create and protect their property and to control distribution of their product. Copyright is the exclusive and assignable legal right, given to the originator for a fixed number of years, to print, publish, perform, film, or record literary, artistic, or musical material. One can't use it without permission of the owner. A copyright infers that any further copying or sharing of someone's initial work may only be done with their permission. Copyright is the right given to the creator/s of books, audio cassettes, video cassettes, journal articles, music scores, DVDs, computer software, etc., to determine when and how their work will be used for a specific length of time.

A copyright "protects" the property owner and their ownership rights over "original created" works. Anyone can copyright by just creating. A copyright is effectively issued when anything new is created. In publication of text and images, copyright may be attached as metadata to inform others of one's ownership and identify what permissions and restrictions the owner grants them. In the United States, along with publishing works, it is possible to proactively register with the US Copyright Office. Once a copyright is registered, it is active for the rest of the registers life, plus an additional 70 years (for the inheritors).

Types of State information privatization protection laws (i.e., laws of competition and scarcity induction) include:

1. **Content copyright**- is copyright protection for the property owner of any original content, which extends to all of the "copyrightable" expressions embodied in the content.
   A. **Software copyright** - is copyright protection for the property owner of a computer [software] program, which extends to all of the "copyrightable" expressions embodied in the program. The copyright law does not protect the functional aspects of a computer program, such as the program's algorithms, formatting, functions, logic, or system design. For these to be proprietary, a patent must be applied for.

3.4.1 State copyright licenses

Whereas copyright is an automatic right, a patent right must be applied for. Copyrights can often be registered with relevant State authorities as future, efficient proof of ownership. However, copyrights do not typically need to be filed for the property to have been created.

3.4.2 Civil copyright licenses

A.k.a., Copy-right licenses.

A [civil] copyright license is an agreement that lets someone else use, and/or sell "your invention of property" for a specified period of time agreed under contract.

3.4.3 Copy-left [commons-reciprocal copying] licenses

A.k.a., Open-reciprocal copying, copy left, left copyright, copyright left licensing, left licensing, persistent open-source licenses, persistent commons licenses, open-source licenses, etc.

All copy-left licenses are also classifiable as open-source licenses. Copyright is a law that restricts the right to use, modify, and share creative works without the permission of the copyright holder. Copyleft gives users' freedoms, copyright takes away users' freedoms. Information can become the intellectual property of its creator. Copyleft prevents proprietary derivatives. When an author releases a program under a copyleft license, they make a claim on the copyright of the work and issue a statement that other people have the right to use, modify, and share the work as long as the reciprocity of the obligation to keep the work in the commons (Read: open-source) is maintained. If a future user adapts the program, they are using a component with a "left" kind of open license, then they too must make their code open for use by others as well. Hence, copyleft[ed] software is still actually copyrighted (with a terms of contract shown), but instead of using those rights to restrict users, like proprietary/closed systems do, the rights are used to ensure that every user has freedom. During transition into community itself, the goal is, "use and contribute back". The goal is not, "use and privatize" or (use and create proprietary derivatives).

> **NOTE:** *Copy-left licenses allow/permit others to adopt (copy), adapt and remix (modify), and redistribute, under the same license.*

Copyleft is the legal technique of granting certain freedoms over copies of copyrighted works with the requirement that the same rights be preserved in derivative works. Copyleft is a strategy of utilizing copyright law and licensing to pursue the policy goal of fostering and encouraging the equal and inalienable "right" against State intervention to copy, share, modify and improve creative works of authorship. Copyleft (as a general term) describes any method that utilizes the copyright system (in whole or in part) to achieve the aforementioned goal.

> **CLARIFICATION:** *Copyleft effectively means that others have been given permission by the copyright owner to adopt, adapt, and redistribute, as long as the original copyleft license is maintained.*

Obviously, because of market-State conditions, copyleft/open-source licenses exist within the legal structure of property rights. Copyleft licenses may be used to restore freedom to users, and this facilitate a transition to a community type society. Technically, however, copyleft doesn't just allow freedom; it requires freedom. Share-alike licenses ensure that copyleft freedoms remain even in derivative works.

In community, the population wants to guarantee that it is able to incorporate modifications (derivations, adaptations), updates, improvements, and additions by other people, always. A "left" license guarantees reciprocation with some clause like the Creative Common organization Share-alike (SA) qualification. Here, SA stands for share-alike; meaning, maintain the same commons level of access going forward. The specific license clause is called the "share-alike" clause.

> **CLARIFICATION:** *Copyleft (commons perpetuating) and public domain (commons marketable) are different. Copyleft is not the same thing as public domain. Public domain means that nobody owns "rights" to a particular work and anybody is free to do whatever they want with it. Hence, public domain content can be adapted/modified, and then, distributed/ sold under a less permissive and more restrictive license. For example, MIT-licensed source code can be modified and then released under a stricter license.*

A core concept related to community transition licensing is that of copyleft (Read: copy freely and share-alike). Copyleft licenses state that users have the right to freely use, copy, modify, and distribute works however they want, with one crucial clause: all derivative works must offer the same freedoms to users (a.k.a., the share-alike clause). Any derivation made by anyone must give distributed in a way that permits anyone else to modify and distribute the work, and so forth (Read: reciprocity).

Hence, copyleft licenses are defined by two main aspects:

1. The freedom for users to modify and distribute derivative works.
2. The "share-alike" clause that maintains freedom in derivative works. Share-alike license clauses offer protection for designs/information entering society from being made proprietary, and/or their adaptations being made proprietary.

Copyleft is an arrangement whereby a work may be used, modified, and distributed on condition that anything derived from it is bound by the same condition. Copyleft is used to describe a copyright that requires anyone distributing a copy or derived copy to allow redistribution of their code. Copyleft is an approach that grants rights to others to use, but in order to do copyleft the worker (author) has to have and retain copyright. Copyleft provides a method for software are or documentation to be modified and distributed back to the community.

Copyleft software licenses require that if someone distributes the code or any work based on the code, they must provide recipients the source code under the same copyleft license terms that it was received under—with the same requirement applying to all downstream recipients of the code or any work based on the code.

The requirement that the license apply to or infest other software based on the code, is why these licenses are also sometimes referred to as viral in that the creator will potentially be required to: (1) distribute the source code for any proprietary code; (2) not charge a license fee or impose your own license terms for your proprietary code; and (3) grant to the public a license to the entire software patent portfolio. These requirements typically get triggered by distributing open source code.

The requirement that the license apply to or infest other software based on the code, is why these licenses are also sometimes referred to as viral. Proponents of copyleft licenses don't favor that moniker, by the way. As you could imagine, just keep going with this metaphor, this infection can have major implications for companies. For example, depending on how you integrate open source copyleft code with your proprietary code, the license may require you to (1) distribute the source code for your proprietary code; (2) not charge a license fee or impose your own license terms for your proprietary code; and (3) grant to the public a license to the entire software patent portfolio. These requirements typically get triggered by distributing open source code.

Copy-left is an open source licence wherein the license becomes attached to every replication/adaptation that incorporates the property. Effectively, copyleft exists to transfer information into a growing, permanent collaborative [open-source] commons. Copyleft is a process of moving information, through licensure, permanently into the commons (for a fixed number of years, whereupon given jurisdictional information property laws, it may become privatizable. Remember, copyrights (and patents) expire after a fixed legally set amount of time. So, even if something is share-alike licensed into the commons, it is still considered copyright, and hence, after some time, the copyleft license with the share-alike property disappears and is replaced by the "public domain" where all rights are waived and/or no longer exist (because they have expired). Share-Alike type of clauses in copy-left and open-source licenses prevent adaptations from being privatized.

> **NOTE:** *Open-source copy-left licenses always give permission for both non-commercial and commercial use, as long as the free license remains in place. There are commons-oriented licenses that are non-commercial, and these would thus, not classify as open-source or copy-left.*

Most work in the early 21st century is licensed to take away the users' freedom to share and change the work. There are, however, also licenses worded/intended to guarantee users' freedom to share (redistribute) and make changes (adapt). And also, to ensure work can remain "free of charge" (i.e., free of trade) to its users as it is evolved and optimized. Herein, to not have to trade or pay a price for something is also a type of freedom.

3.5 Trademark law [for private property]

A trademark protects the name of an organization for use by others. Trademarks include: short slogans, or logos. Trademarks protect market and political party identities (a.k.a., brands). A trademark is a word, phrase, design, or symbol that identifies and distinguishes one business' products and/or services from another. Unlike patents, trademarks don't expire and don't have to be registered (because of common law right). However, registering gives some rights advantages since it's a public statement of claim of ownership to a particular mark (symbol of identity).

3.5.1 Trademark licenses

A trademark license is an agreement that lets someone else commercially make, use, and sell a representation of your identity, for perpetuity. Unlike patents and copyrights, trademarks do not expire after a set period of time. Trademarks will persist so long as the owner continues to use the trademark.

4 Open source systems

> **INSIGHT:** *A population can use collaboration (Read: open source) to speed up the arrival of a solution.*

Open source systems are systems whose [source] code is published and made available to the public, enabling anyone to copy, modify, and redistribute the source code without paying royalties or fees. This definition includes two elements:

1. Actual disclosure of the [source] code from the system;
2. The intellectual property rights license, which includes copyright license and, where applicable, patent licenses that can be used, modified and distributed without the payment of software license.

In concern to ware (software/hardware), open source refers to systems [source] code is freely available to users for reference, debugging, modification, and/or extension. Here, open means that the source code must be distributed with every copy of an executable [application] and every recipient must be allowed to modify and distribute the source code freely to subsequent users.

Open standards are, typically, specifications (formal descriptions). Open signifies that the standards process is open to participation and that the completed standards are available to everyone. In concern to societal production, an open societal system is a system to which contribution is possible.

> **NOTE:** *Open source creates a community of designing and contributing users.*

Note that working documents and drafts may or may not be kept private to the individual contributors or issuing organization sub-groups, until released in some more finalized form. Open standards organizations may have membership fees, but any person or company may participate as a member at a meaningful level. Open standards organizations give copies of their standards away for free and the right to implement a standard is typically also free. At a fundamental level, open source means to use without regard to permission and other artificial restrictions on social and technical progression (effectively, self and societal progression). Open source is the turning over of [the concept of] property to the [the concept of] commons.

Open source, functional hierarchies are based on contributed competence involving the presence of knowledge, and the ability to formulate problems and solve them. Some hierarchies are predicated on power and authority, and mostly the pathological ones.

Open source has two principal trust benefits:

1. Transparency = Trust.
   A. Social trust through transparency.
2. Many potential viewers = Trust.
   A. Social trust through networked contribution - there are more observers and contributors to the system (e.g., more people looking at the code such that bugs are discovered more quickly and can be fixed more quickly; hence, the objective/code is achieved/improved.
3. Sharing = Trust.
   A. Open source (sharing) means avoiding having to rebuild fundamental components from scratch.

Because the source code is publicly available, individuals can concentrate on developing the elements unique to their current task, instead of spending their effort on rethinking and re-writing code that has already been developed by others. Code re-use reduces development time and provides predictable results.

Open source systems are considered less likely to fork when there is an accepted and transparent organizational structure, contribution is open, and there is long-term contribution potential; transparency eliminates the economic motivations for fragmentation.

Take Linux for example: Ninety-nine percent of Linux distributed code is the same. The small amount of fragmentation between different Linux distributions is good because it allows them to cater to different segments. The small amount of fragmentation between different regional and local habitat service systems is good because it allows them to cater to different preferences. Users benefit by choosing a Linux distribution (or community-type society distribution) that best meets their needs.

In the corporate model, individuals or small groups of individuals develop systems in isolation, without releasing a version before it is deemed ready. In contrast, the open source (and working group) model relies on a network of volunteer contributors, with differing styles and agendas, who research, develop and debug the system in parallel and serial. Open source allows anyone who is curious or suspicious or critical to take a look for themselves; there is transparency and they can do their own due-diligence.

> **APHORISM:** *Copying is the most sincere form of flattery.*

4.1 Source type and safety

> **NOTE:** *The more accurate information we know, the more capable and likely we are to explain higher mutual life fulfilling intentions.*

When a closed source operation (a business) write code, "we" simply do not know what is in it. A community of users and developers must be able see the source code, for their own safety. Open source means that the functioning is entirely transparent to any user, who may also be a contributor to the systems continued development. Open source is foundationed on the logic that the highest freedom (or, best security) comes from allowing anyone to inspect its code and suggest (or enact, depending on context) improvements.

> **STATEMENT: "***We" have to keep our work open and transparent if "we" are going to thrive. Humanity is likely to discover, resolve, and integrate more rapidly and safely when its societal system is globally cooperative (i.e., open for all to access given what is socially known and based on a societal-level state/condition of optimum fulfillment).*

4.2 Open society

> *A.k.a., Open societal engineering.*

An open society requires thinking in networks. By being willing to be transparent others can discover what "you" are doing, and through that discovery, they can connect their own work to activities that "you" are involved in, thus evolving the whole optimally. In community, participation is global, and hence, open source is the ideal approach, for it allows for efficient global cooperation.

Fundamentally, any societal system based upon a stored program (e.g., a software system that coordinates supply and demand, instead of the price mechanism) must be able to have that program changed when bugs and vulnerabilities are found. Therefore, humanity requires that program to be fixable and updatable. But, that same need for the "ware" to be soft (changeable) inherently opens the "ware" to abuse. Hence, the system must be open so that everyone can see what is occurring. And above the "ware" itself, there is the necessity for a social structure that satisfactorily guides changes to the program, ensuring the social population navigates similarly and anyone is unlikely to abuse the program.

4.2.1 The public [market-State domain]

A community-type socio-economic system necessitates globally cooperative standards for production and access. Conversely, authoritarian-based socio-economic systems specifically restrict access to information about the system in order to advantage some, while disadvantaging others. It is fundamentally dangerous to human fulfillment to actualize a non-public domain socio-economic system into the lives of a population. A socio-economic system that is not in the public domain is not particularly useful to the public (Read: those with unequal access); though it is useful to those with privileged private access. How could a system designed for organizing and coordinating the "public" population not be released into the "public domain"? Or, why (as in, for whose benefit) would the socio-economic information system not be released into the public domain? Who would its concealment/privatization benefit, and who would it harm? Why would a system designed (claimed to be designed) for the public, not be released into the public [domain]? Authoritarian-based socio-economic systems limit and distort corrective feedback from the inherent nature of their structure.

If humanity is constantly sharing what is designed and built, and publishing/realizing examples, while remembering what it is like to learn, then humanity will be able to constantly have an influx of useful solutions to the global fulfillment of all. Open processes, sharing, and a focus on learning, makes community accessible to everyone, which results in so much interesting and useful work. Herein, is strength in sharing, and together humanity can build something useful and fulfilling for everyone.

> **INSIGHT:** *An open society is a society wherein the information system for its conception and production is public.*

4.2.2 Open licenses in the market

Open source is a way of interacting with the market. Openly licensing allows others to replicate, reuse, adapt, improve, adopt, bring to scale, write about, talk about, remix, translate, upstream, fork, digitize, redistribute and build upon what we have done.

If an open license is implement in a commercialized product, it doesn't mean that the product has to be given away for free (the product with the open standard can still be commercialized), unless that is one of the conditional restrictions of the license.

> **IMPORTANT:** *In community, everything added is added community information system and habitat service system [platform] without patents or copyrights.*

4.2.3 Open source engineering

In general, open source engineering means that the engineering file contains the source content that was used in the creation. The content includes:

1. For software, the source code.
2. For all other content, the original design files.
3. Which, may be used by anyone to understand, to study, to apply, and to adapt the design, using:
4. The original source code.
5. The original design files.

4.2.4 Social cooperation

> **NOTE:** *The concept of 'contribution', as an approach is described in its decision context in the Decision System Specification under the sub-title, 'Participation'. Also, the concept of 'open source', as a value objective, is described in its social context in the Social System Specification.*

A social system is a grouping of units of individuation (units of consciousness with "free will") forming a cooperative network. In this sense, individuals are units of awareness that communicate and interact with each other. A social system is an interactive system. There are two fundamental ways in which an individual can interact with another: cooperatively (i.e., togetherness) or fearfully (i.e., competitiveness). Here, cooperation reflects/is caring, because the other is important and significant to "me", because "we" are all in this [environment] together, "we" are all interacting, its all a big interaction, and the cooperative way is more efficient and effective for all individuals. The opposite way is fear.

If "me" has fear -- if each one is fearful, then each individuation thinks only about themselves and acts only in consideration of themselves. Fear is all about "me"; it is not about "we", or "we" and "me".

> **APHORISM:** *In a world where there is only "me", then there is likely to be fear of "we", and in a world where there is only "we", then there is likely to be fear of "me". When "me" and "we" integrate, fear is likely to disintegrate.*

As stated in the project's purpose, a primary goal of a social system is stability, which occurs through the facilitation of cooperation by means of intelligently shared organization and the sufficient completion of human need fulfillment.

> **APHORISM:** *If you love something, set it free.*

4.3 Open standards

The term "open" is usually means royalty-free (RF) technologies, "free" means no trade (no money), while the term "standard" usually means a technology or socio-technical system formalized by information integration. The definitions of the term "open standard" used by academics, the European Union and some of its member governments or parliaments preclude open standards requiring fees for use. In the market-State, obviously, many definitions of the term "standard" and open standards may permit patent holders to impose "reasonable and non-discriminatory" royalty fees and other licensing terms on implementers and/or users of the standard. In concern to openness and standards, standards are considered to be open when they are developed and made available through a community contribution service structure.

> **IMPORTANT:** *Open standards are the foundation for cooperation in socio-technical society. Open standards are a necessary pre-requisite to ensure individual freedom.*

Open standards are publicly available and developed via processes that are transparent and open to broad participation. In concern to participation, an activity is open when it is open to all persons [who are affected by the activity]. There shall be no artificial limitations (e.g., money, birth place) as a barrier to participation. In contrast, proprietary standards are privately owned by one or more entities that control their distribution and access. Open standards let people and organizations set up new services and make them available across the rest of the Internet without permission by a private owner.

> **INSIGHT:** *Open Standards are the foundation of cooperation in modern society.*

Open standards are a socio-technical foundation of community, allowing anyone to learn and contribute to a services design and operation without requiring permission from anyone else. Open standards enable community existence, facilitate its adaptation, and provide a platform that supports social and economic opportunity for billions of users.

Calling a standard "open" makes a clear distinction against so-called "closed", "de facto" or "proprietary" standards. Open standards must be subject to full public assessment and use without constraints in a manner equally available to all parties.

Open standards are important for allowing software made by different people to work together (Read: interoperable). An open standard for interoperability will be either free of patents or they will have been irrevocably declared free of royalty. Patents and copyright pose a privatization threat to open standards (and their implementation). Any person who owns a patent containing claims that are essential to the implementation of a standard can prevent anyone from making, using, or selling products that implement that patent in the market-State jurisdiction(s) in which the patent is acknowledge by the State. It is commonly understood that patents and copyrights prevent sale of another's work in given jurisdictions. A patent does not protect privatized technology from being infringed upon by a competitor. It merely affords the property owner with legal recourse in the event that someone does. A patent gives the patent rights property owner a seat at the [enforcement] table, both offensively and defensively.

> **APHORISM:** *In competition, if "you" have an idea that may help a lot of people, but it gives "you" an advantage in the market on "your" competition, then "you" keep it to "yourself" - intellectual property and concealment are advantageous in an environment of competition, over open source and sharing.*

Consider the implications for the owner of intellectual property (e.g., the owner of a patent or copyright) who wants to have that property integrated into an industry standard. Or, consider the interests of a developer of an industry standard service who learns that another person's intellectual property blocks the implementation of the standard. Is private intellectual property compatible with planetary standards, with global cooperation an open source world? "Industry standards" are not always what they seem to be; some companies or standardizing organizations attempt to control standards through copyrights on specifications and patents in specifications.

Open standards should/ought to be available to everyone on royalty-free terms, or the standards should not be called open. That is one way a clear definition can help distinguish among standards. The term "open standard" is sometimes coupled with "open source" with the idea that a standard is not truly open if it does not have a complete free/open source reference implementation available. Open standards which specify formats are sometimes referred to as open formats. Many specifications that are sometimes referred to as standards are proprietary and only available under restrictive contract terms (if they can be obtained at all) from the organization that owns the copyright on the specification. As such these specifications are not fully Open. Where truly open standards do not have fees associated with their implementation, certification of compliance by the standards organization (generally an organization in the market, may involve a fee). The purpose of an open standard in the market-State is not the same as the purpose of an open standard in community. In the market, the open standard increases the market for a technology by enabling potential consumers or suppliers of that technology to invest in it without having to pay monopoly rent or fear litigation on trade secret, copyright, patent, or trademark causes of action. In the market, no standard can be described as "open" expect to the extent that it achieves these goals. In the market-State, an open standard has certain market-State "rights" associated with it. In the market-State, the definition of an open standards have many different levels of openness.

> **NOTE:** *Many specifications that are sometimes referred to as standards are proprietary and only available under restrictive contract terms (if they can be obtained at all) from the organization that owns the copyright on the specification. As such these specifications are not considered to be fully open. Sometimes the term "Freeware" or "Open" is applied to software which is available free of cost or even as source code but all the same with proprietary distribution terms. This is not Open Source and not Free Software. No matter what, the system must be shipped with an Open Source or Free license to qualify as such (in the market).*

The freedom to use, explore, modify and give away information freely leads to a completely different motivation for creating the software in the first place. The motivation shifts away from primarily making money to solving a problem. The resulting software is typically more focused to solve a single problem at it's best and more open to integrate with other solutions. For users the investment into Free and Open Source design is more lasting because there is no single entity that can take away the right to continue to use the software which is what proprietary vendors can do.

Due to the naturally distributed nature of open source, the flaws of systems are more rapidly and effectively spotted and worked out then with closed source standards.

An open source system (hardware and software) is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source systems is a term for tangible artifacts — machines, devices, or other physical things — whose design has been released to the public in such a way that anyone can make, modify, distribute, and use those things. Hardware is different from software in that physical resources must always be committed for the creation of physical goods.

Table 30. Contribution Approach > Open Standards:
Conditional categorises of open access with their descriptions.

Condition	Description
Availability	Open standards are available for all to read and implement.
Maximize end-user choice	Open standards create a fair, competitive market for implementations of the standard. They do not lock the customer into a particular vendor or group. No vendor lock-in.
No royalty	Open standards are free for all to implement, with no royalty or fee. Certification of compliance by the standards organization may involve a fee.
No discrimination	Open standards and the organizations that administer them do not favor one implementer over another for any reason other than the technical standards compliance of a vendor's implementation. Certification organizations must provide a path for low or zero cost implementations to be validated, but may also provide enhanced certification services.
Extension or subset	Implementations of open standards may be extended, or offered in subset form. However, certification organizations may decline to certify subset implementations, and may place requirements upon extensions.
Predatory practices	Open standards may employ license terms that protect against subversion of the standard by embrace and extend tactics. The licenses attached to the standard may require the publication of reference information for extensions, and a license for all others to create, distribute and sell software that is compatible with the extensions. An open standard may not otherwise prohibit extensions.

4.1 Standards openness index

In order to more greatly discern the openness nature of a standard, the following questions may be proposed:

1. How is the standard created?
2. How is the standard maintained after Version 1.0?
3. What is the cost of getting a copy of the standard?
4. Are there restrictions or permissions on how the standard can be implemented?
5. What is required to demonstrate compliance (i.e., the actual application) of the standard.

In concern to the openness of a standard:

1. The more transparent the standards process is, the more open the standard is.
2. The more the community can be involved and then actually is involved, the more open the standard is.
3. The more democratic the standards process is, where the community can make significant changes even before Version 1.0, the more open the standard is.
4. The lower the standards-related cost to software developers who want to use the standard, the more open it is.
5. The lower the standards-related cost to the eventual consumer of software that happen to use the standard, the more open it is.
6. When the licensing of the standard is more generous in the freedoms and permissions it provides, the more open the standard is.
7. When the licensing of the standard is more onerous in the restrictions it imposes, the less open the standard is.

From these and perhaps other criteria, the development of a standards openness index is possible.

There are varying degrees of possible openness in concern to data:

1. **License-free** (*trade free*):Data are not subject to any form of ownership, copyright, patent, intellectual property or industrial secret. Reasonable restrictions of privacy, safety and access may be allowed.
2. **Non-proprietary**: Data are available in a format on which no entity has exclusive control.
3. **Non-discriminatory**: Data are available for all, without the need of registration to access them.
4. **Machine readable**: Data are reasonably structured to enable automated processing.
5. **Accessible**: Data are available to the largest possible scope of users and for the largest possible scope of purposes.
6. **Up-to-date**: Data are made available as fast as possible preserving accuracy and value.
7. **Primary:** Data are collected in its source, with the highest possible level of granularity, not in

aggregate or modified forms.
8. **Complete**: All data are made available. All data are data that are not submitted to valid privacy, safety, or engineering limitations.

4.1.1 Basic requirements of an open standard

The societal system design specifications are standards, which contain technical and organizational information in documents about the society, as past, present (current InterSystem Team Operations, and future (iteration).

An open standard must be:

1. **Enabling** of future access (i.e., access to habitat services by future humans; procreation).
2. **Contributable** to, by any interested and informed individual (discovery and design openness).
3. **Available** to the planetary population ("public") and developed (or approved) and maintained via a cooperative and contributive process.
4. **Free of trade,** royalty, or fee (i.e., free for all to access, copy, redistribute, modify, use, re-use implement, and accountably comply with).
5. **Free of [State] agreements**, including any requirement for a license, legal agreement, non-disclosure agreement (NDA), grant agreement, click-through, or any other form of exchange, trade, or paperwork (i.e., free for all to access, copy, redistribute, modify, use, re-use, implement, and accountably comply with).
6. **Updatable/adaptable** as required to provide additional clarifications or to include additional information in those areas in which specifications are still evolving.

An open [source] specification (or, standard) has four categories:

1. **Availability**
   A. The specification must be redistributable free of charge.
   B. The specification must be redistributable free of agreements, money, and trade.
2. **Usage rights** (a "license" in the market-State)
   A. Essential patents must be made irrevocably available royalty free.
   B. Essential patents must be licensable free of agreements, money, and trade.
3. **Process**
   A. Further development must be open for anyone to participate in.
   B. Further development must be open for anyone to view.

4.1.2 Basic criteria of an open standard

To comply with the Open Standards Requirement, an "open standard" must satisfy the following criteria. If an "open standard" does not meet these criteria, it will be discriminating against open source developers.

A simplified set of open source criteria are:

1. No intentional secrets: The standard MUST NOT withhold any detail necessary for interoperable implementation. As flaws are inevitable, the standard MUST define a process for fixing flaws identified during implementation and interoperability testing and to incorporate said changes into a revised version or superseding version of the standard to be released under terms that do not violate the OSR.
2. Availability: The standard MUST be freely and publicly available (e.g., from a stable web site) under royalty-free terms at reasonable and non-discriminatory cost.
3. Patents (*a market-State based concept*): All patents essential to implementation of the standard MUST:
   A. Be licensed under royalty-free terms for unrestricted use.
   B. Be covered by a promise of non-assertion when practiced by open source software.
4. No agreements (*i.e., no market-State based agreements*): There MUST NOT be any requirement for execution of a license agreement, NDA, grant, click-through, or any other form of paperwork to deploy conforming implementations of the standard.
5. No Open Standards Requirement (OSR) -incompatible dependencies: Implementation of the standard MUST NOT require any other technology that fails to meet the criteria of this requirement.

High-level organizing components of an open standard include, but are not limited to:

1. Open Source (and free) – providing source code, with permission to use, modify and redistribute it.
2. Reciprocal (or copyleft) – distributed derivatives must remain shared under the EUPL, CC-BY-SA, etc.;
3. Compatible – when merged with another work covered by GPL-3.0, LGPL, etc., and the combined work can be distributed under these licences;
4. Interoperable – interfaces, APIs, libraries, and data structures may be freely copied in order to link with other components.
5. Multilingual – can be translated freely.
6. Complete – covers the use of patents, various

'works' and distribution methods including 'services'.
7. Compliant – with State laws.

4.2 *Open access*

In general, open access (OA) refers to releasing content freely to the public at no cost and with limited restrictions with regards re-use, modification, and re-distribution. The general meaning of open access (OA) is to share research and standards publications freely (without trade or restriction) so anyone can benefit from reading, research, and use toward societal development. For there to be open access, a societal organization must allow others to re-use that research and to apply that research toward societal development. In concern to online content, open access (OA) makes content permanently available online to view without restriction. When research and application is held behind restrictive walls of access, then mutual societal development is likely to be significantly impaired. Making real-world information sets (e.g., research, standards, and protocols) open access (a.k.a., open source, free, etc.), is a requirement for the operation of a community. The best form of open access for facilitating the creation and operation of community is to license content via a creative commons copyleft persistent license:

- **CC-BY-SA (Creative Commons Attribution Share-alike License)**

The benefits of open access to researchers and societal organization includes:

1. Improved reach of research; improved application of research.
2. Improved data collection, facilitating data collection on evidence for impact.
3. Improved reputation for researchers through increased citations.
4. Improved quality of research through open, transparent and reproducible research practices.
5. Improved production, distribution, and material cycling of access to highest quality services.

5 Organizational definitions of open source and open standards

There are many organizations with slightly different definitions for open source, including but not limited to:

1. Open-source hardware:
   A. Open Source Hardware Association.
2. Open-source software:
   A. Open Source Initiative Definition)annotated version 1.9).
   B. Free Software Foundation.
3. Open-source standards:
   A. Open Standards. Open Source (OASIS).
   B. ITU-T.
   C. Governmental definitions.
   D. Open Geospatial Consortium.
4. Open-source (commons) documentation:
   A. Creative Commons.

5.1 *Open-source qualifying organizations and qualifications*

> **NOTE:** *For the vast majority of open source projects, an open source license implicitly serves as both the inbound (from contributors) and outbound (to other contributors and users) license; "inbound=outbound".*

The following organizations qualify open-source:

5.1.1 OpenChain Open Source Specification Standard (The Linux Foundation)

OpenChain ISO/IEC 5230:2020 is the International Standard for open source license compliance. OpenChain exists tos build trust between organizations in the supply chain.

The four principles the standard is built on are (*OpenChain Project*, 2020):

1. Build trust around the open source supply chain.
2. Remember that less is more:
   A. Define the key requirements of a quality compliance program
   B. Do this by solving real pain points in the supply chain
3. Keep our specifications limited to what and why (avoid the how and when)
   A. Embrace different implementations to solve challenges
   B. Avoid mandating specific process content
4. Be open to all to participate and contribute

5.1.2 Open Source Hardware Association

The distribution terms of Open Source Hardware must comply with the following criteria:

1. **Documentation** - The hardware must be released with documentation including design files, and must allow modification and distribution of the design files. Where documentation is not furnished with the physical product, there must be a well-publicized means of obtaining this documentation for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The documentation must include design files in the preferred format for making changes, for example the native file format of a CAD program. Deliberately obfuscated design files are not allowed. Intermediate forms analogous to compiled computer code — such as printer-ready copper artwork from a CAD program — are not allowed as substitutes. The license may require that the design files are provided in fully-documented, open format(s).
2. **Scope** - The documentation for the hardware must clearly specify what portion of the design, if not all, is being released under the license.
3. **Necessary software** - If the licensed design requires software, embedded or otherwise, to operate properly and fulfill its essential functions, then the license may require that one of the following conditions are met:
   A. The interfaces are sufficiently documented such that it could reasonably be considered straightforward to write open source software that allows the device to operate properly and fulfill its essential functions. For example, this may include the use of detailed signal timing diagrams or pseudocode to clearly illustrate the interface in operation.
   B. The necessary software is released under an OSI-approved open source license.
4. **Derived works** - The license shall allow modifications and derived works, and shall allow them to be distributed under the same terms as the license of the original work. The license shall allow for the manufacture, sale, distribution, and use of products created from the design files, the design files themselves, and derivatives thereof.
5. **Free redistribution** - The license shall not restrict any party from selling or giving away the project documentation. The license shall not require a royalty or other fee for such sale. The license shall not require any royalty or fee related to the sale of derived works.
6. **Attribution** - The license may require derived documents, and copyright notices associated with devices, to provide attribution to the licensors when distributing design files, manufactured products, and/or derivatives thereof. The license may require that this information be accessible to the end-user using the device normally, but shall not specify a specific format of display. The license may require derived works to carry a different name or version number from the original design.
7. **No discrimination against persons or groups** - The license must not discriminate against any person or group of persons.
8. **No discrimination against fields of endeavour** - The license must not restrict anyone from making use of the work (including manufactured hardware) in a specific field of endeavour. For example, it must not restrict the hardware from being used in a business, or from being used in nuclear research.
9. **Distribution of license** - The rights granted by the license must apply to all to whom the work is redistributed without the need for execution of an additional license by those parties.
10. **License must not be specific to a product** - The rights granted by the license must not depend on the licensed work being part of a particular product. If a portion is extracted from a work and used or distributed within the terms of the license, all parties to whom that work is redistributed should have the same rights as those that are granted for the original work.
11. **License must not restrict other hardware or software** - The license must not place restrictions on other items that are aggregated with the licensed work but not derivative of it. For example, the license must not insist that all other hardware sold with the licensed item be open source, nor that only open source software be used external to the device.
12. **License must be technology-neutral** - No provision of the license may be predicated on any individual technology, specific part or component, material, or style of interface or use thereof.

Unlike software, which is generally protected by copyright, hardware may have market-State protection by a number of different rights - or no rights at all. That makes licensing hardware a bit more complicated than licensing software.

5.1.3 Open Source Initiative Definition (annotated version 1.9)

According to the Open Source Initiative's definition, open-source does not only imply access to source code, but also compliance with the following criteria for the terms of distribution. (*Open Source Initiative*, 2007)

The open source initiative definition of open source annotated version 1.9 from the Open Source Initiative [opensource.org] defines open source as (*Open Source Initiative*, 2007):

1. **Free redistribution** – license must not require royalty or any fee.
   A. Rationale: By constraining the license to require free redistribution, we eliminate the temptation for licensors to throw away many long-term gains to make short-term gains. If we didn't do this, there would be lots of pressure for cooperators to defect.
2. **Full access to source code** - program must allow free distribution of the source code.
   A. Rationale: We require access to un-obfuscated source code because you can't evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy.
3. **Full access to derived works** - license must allow modification and distribution.
   A. Rationale: The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.
4. **Integrity of the author's source code** - license must explicitly permit distribution of software built from modified source code.
   A. Rationale: Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations.
5. **No discrimination against persons or groups** - license must not discriminate against any person or group of persons.
   A. Rationale: In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.
6. **No discrimination against fields of endeavour** - license must not restrict anyone from making use of the program in a specific field of endeavor.
   A. Rationale: The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.
7. **Distribution of license** - rights attached to the program must allow distribution.
   A. Rationale: This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.
8. **License must not be specific to a product** - rights attached to the program must not depend on the program being part of a software distribution.
   A. Rationale: This clause forecloses yet another class of license traps.
9. **License must not restrict other software** - license must not place restrictions on other software that is distributed along with the licensed software.
   A. Rationale: Distributors of open-source software have the right to make their own choices about their own software.
10. **License must be technology-neutral** - no provision of the license may be predicated on any individual technology or style of interface.
    A. Rationale: This provision is aimed specifically at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee.

5.1.3.1 Open Source Initiative (OSI) criteria for open source licenses

The Open Source Initiative (OSI), the organization responsible for reviewing and approving open-source licenses provides a list of five criteria an open standard must satisfy. "If an 'open standard' does not meet these criteria, it will be discriminating against open source developers," the site says:

1. **No intentional Secrets:** The standard must not withhold any detail necessary for interoperable implementation. As flaws are inevitable, the standard must define a process for fixing flaws identified during implementation and interoperability testing and to incorporate said changes into a revised version or superseding version of the standard to be released under terms that do not violate the OSR.
2. **Availability:** The standard must be freely and publicly available (e.g., from a stable web site) under royalty-free terms at reasonable and non-discriminatory cost.
3. **Patents:** All patents essential to implementation of the standard must:
   A. Be licensed under royalty-free terms for unrestricted use, or
   B. Be covered by a promise of non-assertion when practiced by open source software.
4. **No agreements:** There must not be any requirement for execution of a license agreement, NDA, grant, click-through, or any other form of paperwork to deploy conforming implementations of the standard.

5. **No OSR-Incompatible Dependencies:** Implementation of the standard must not require any other technology that fails to meet the criteria of this Requirement.

5.1.3.2 Open Source Definition principles

The principles that apply to the Open Source Definition of an open standard are:

1. Licensees are free to use open source software for any purpose whatsoever.
2. Licensees are free to make copies of open source software and to distribute them without payment of royalties to a licensor.
3. Licensees are free to create derivative works of open source software and to distribute them without payment of royalties to a licensor.
4. Licensees are free to access and use the source code of open source software.
5. Licensees are free to combine open source and other software.
6. Anything else should not be called an open standard.

5.1.3.3 The Open Source Definition criteria

Bruce Perens, creator of The Open Source Definition, outlined six criteria an open standard must satisfy:

1. **Availability:** Open standards are available for all to read and implement.
2. **Maximize End-User Choice:** Open Standards create a fair, competitive market for implementations of the standard. They do not lock the customer into a particular vendor or group.
3. **No Royalty:** Open standards are free for all to implement, with no royalty or fee. Certification of compliance by the standards organization may involve a fee.
4. **No Discrimination:** Open standards and the organizations that administer them do not favor one implementer over another for any reason other than the technical standards compliance of a vendor's implementation. Certification organizations must provide a path for low and zero-cost implementations to be validated, but may also provide enhanced certification services.
5. **Extension or Subset:** Implementations of open standards may be extended, or offered in subset form. However, certification organizations may decline to certify subset implementations, and may place requirements upon extensions (see Predatory Practices).
6. **Predatory Practices:** Open standards may employ license terms that protect against subversion of the standard by embrace-and-extend tactics. The licenses attached to the standard may require the publication of reference information for extensions, and a license for all others to create, distribute, and sell software that is compatible with the extensions. An Open standard may not otherwise prohibit extensions.

5.1.4 Free Software Foundation (FSF)

It is sometimes helpful to understand that Open Source is a matter of liberty (as freedom), and not necessarily price (as freedom). To this end the Free Software Foundation says that one should think of "free" as in "free speech", not necessarily as in "free-market service" (i.e., not in the sense of having another person give something away at their own expense). In concern to software, free software means that the users of a program have the four essential freedoms (as conditions present in the environment):

1. **The freedom to run** the program, for any purpose.
2. **The freedom to study** how the program works, and change it to make it do what you wish. Access to the source code (Open Source) is a precondition for this.
3. **The freedom to redistribute** original copies so you can help your neighbour.
4. **The freedom to distribute modified copies** of the original version.

These freedoms are the prerequisites to open source software development, and they are studied and promoted by the Free Software Foundation.

5.1.4.1 Free Software Foundation Europe (FSFE)

The Free Software Foundation Europe (FSFE) collaborated with other individuals and organizations in the tech industry, politics, and community to outline a different five-point definition. According to the FSFE, an open standard refers to a format or protocol that is:

1. Subject to full public assessment and use without constraints in a manner equally available to all parties;
2. Without any components or extensions that have dependencies on formats or protocols that do not meet the definition of an Open Standard themselves;
3. Free from legal or technical clauses that limit its utilisation by any party or in any business model;
4. Managed and further developed independently of any single vendor in a process open to the equal participation of competitors and third parties;
5. Available in multiple complete implementations by competing vendors, or as a complete implementation equally available to all parties.

5.2 Open standards qualifying organizations and qualifications

The following organizations qualify open standards.

5.2.1 ITU-T open standard definition

The ITU-T has a long history of open standards development. However, recently some different external sources have attempted to define the term "Open Standard" in a variety of different ways. In order to avoid confusion, the ITU-T uses for its purpose the term "Open Standards" per the following definition:

1. "Open Standards" are standards made available to the general public and are developed (or approved) and maintained via a collaborative and consensus driven process. "Open Standards" facilitate interoperability and data exchange among different products or services and are intended for widespread adoption.
2. Other elements of "Open Standards" include, but are not limited to:
   A. **Collaborative process** – voluntary and market driven development (or approval) following a transparent consensus driven process that is reasonably open to all interested parties.
   B. **Reasonably balanced** – ensures that the process is not dominated by any one interest group.
   C. **Due process** - includes consideration of and response to comments by interested parties.
   D. **Intellectual property rights (IPRs)** – IPRs essential to implement the standard to be licensed to all applicants on a worldwide, non-discriminatory basis, either (1) for free and under other reasonable terms and conditions or (2) on reasonable terms and conditions (which may include monetary compensation). Negotiations are left to the parties concerned and are performed outside the SDO.
   E. **Quality and level of detail** – sufficient to permit the development of a variety of competing implementations of interoperable products or services. Standardized interfaces are not hidden, or controlled other than by the SDO promulgating the standard.
   F. **Publicly available** – easily available for implementation and use, at a reasonable price. Publication of the text of a standard by others is permitted only with the prior approval of the SDO.
   G. **On-going support** – maintained and supported over a long period of time.

5.2.2 Governmental definitions of an open standard

Different organizations define the concept of an "open standard" differently. The following are different organizations' definitions of "open standard".

5.2.2.1 Pan-European eGovernment Programme for Interoperability (EIF 1.0)

The Pan-European eGovernment Programme (IDABC) in DG DIGIT issued their European Interoperability Framework (EIF 1.0) with a strict minimum definition of open standards and mandated their use in pan-European eGovernment services. There, the open standards should be:

1. Adopted and maintained via an open process in which all interested parties can participate;
2. Published and available freely or at a nominal charge;
3. Made irrevocably available on a royalty free basis, even if intellectual property issues apply to patents covering all or parts of the standard;
4. Free of constraints on the re-use of the standard.

5.2.2.2 Danish

1. An open standard is accessible to everyone free of charge (i.e. there is no discrimination between users, and no payment or other considerations are required as a condition of use of the standard).
2. An open standard of necessity remains accessible and free of charge (i.e. owners renounce their options, if indeed such exist, to limit access to the standard at a later date, for example, by committing themselves to openness during the remainder of a possible patent's life).
3. An open standard is accessible free of charge and documented in all its details (i.e. all aspects of the standard are transparent and documented, and both access to and use of the documentation is free).

5.2.2.3 French

By open standard is understood any communication, interconnection or interchange protocol, and any interoperable data format whose specifications are public and without any restriction in their access or implementation.

5.2.2.4 Indian

- 4.1 Mandatory Characteristics An Identified Standard will qualify as an "Open Standard", if it meets the following criteria:
  - 4.1.1 Specification document of the Identified Standard shall be available with or without a

nominal fee.

- 4.1.2 The Patent claims necessary to implement the Identified Standard shall be made available on a Royalty-Free basis for the lifetime of the Standard.
- 4.1.3 Identified Standard shall be adopted and maintained by a not-for-profit organization, wherein all stakeholders can opt to participate in a transparent, collaborative and consensual manner.
- 4.1.4 Identified Standard shall be recursively open as far as possible.
- 4.1.5 Identified Standard shall have technology-neutral specification.
- 4.1.6 Identified Standard shall be capable of localization support, where applicable, for all Indian official Languages for all applicable domains.

5.2.2.5 United Kingdom

1. **Collaboration** - the standard is maintained through a collaborative decision-making process that is consensus based and independent of any individual supplier. Involvement in the development and maintenance of the standard is accessible to all interested parties.
2. **Transparency** - the decision-making process is transparent, and a publicly accessible review by subject matter experts is part of the process.
3. **Due process** - the standard is adopted by a specification or standardisation organisation, or a forum or consortium with a feedback and ratification process to ensure quality.
4. **Fair access** - the standard is published, thoroughly documented and publicly available at zero or low cost. Zero cost is preferred but this should be considered on a case by case basis as part of the selection process. Cost should not be prohibitive or likely to cause a barrier to a level playing field.
5. **Market support** - other than in the context of creating innovative solutions, the standard is mature, supported by the market and demonstrates platform, application and vendor independence.
6. **Rights** - rights essential to implementation of the standard, and for interfacing with other implementations which have adopted that same standard, are licensed on a royalty free basis that is compatible with both open source and proprietary licensed solutions. These rights should be irrevocable unless there is a breach of licence conditions.

5.2.2.6 European Union Public License

The European Union has a public licensing guidelines with stated licensing, published by the European Commission under the Directorate-General for Informatics. (*European Union Public*, 2021)

5.2.3 Open Geospatial Consortium (OGC)

The Open Geospatial Consortium defines Open Standards as standards that are:

1. **Freely and publicly available** – They are available free of charge and unencumbered by patents and other intellectual property.
2. **Non discriminatory** – They are available to anyone, any organization, any time, anywhere with no restrictions.
3. **No license fees** - There are no charges at any time for their use.
4. **Vendor neutral** - They are vendor neutral in terms of their content and implementation concept and do not favor any vendor over another.
5. **Data neutral** – The standards are independent of any data storage model or format.
6. Defined, documented, and approved by a formal, **member driven consensus process**. The consensus group remains in charge of changes and no single entity controls the standard.

5.2.4 Open Standards. Open Source (OASIS)

OASIS, technical committees (TCs) develop the standards, and then for the standard be adopted by the consortium as an open standard, it must:

1. Be created by domain experts (not SDO staff).
2. Be developed under and internationally respected, open process (i.e., be open for public review and debate).
3. Be easy to access and adopt.
4. Have allowed anyone affected by the standard to contribute to the development of it.
5. Not have hidden patents to scare implementers.
6. Have the ability to implement the standard baked in (i.e., OASIS standards must be verified by multiple Statements of Use).
7. Be safe for governments to endorse.

5.3 Open copyright qualifying organizations and qualifications

The following organization qualifies as having open copyrights.

5.3.1 Creative commons (CC) copyright licensing

Commons-type licenses can have the following potential allowances:

1. Share (redistribute) = yes (CC) / no.
   A. If yes, then CC identifier.
   B. If no, then not a commons license.
2. Attribution (identify source, citation of source) = yes (BY) / no.
   A. If yes, then BY identifier.
   B. If no, then no BY identifier.
3. Derivation (modify/mix) = yes / no (ND).
   A. If yes, then no ND identifier
   B. If no, then ND identifier.
4. Commercial (sell) = yes / no (NC).
   A. If yes, then no identifier.
   B. If no, then no NC identifier.
5. Share-Alike (every share/derivation keeps the same license) = yes (SA) / no.
   A. If yes, then SA identifier.
   B. If no, then no identifier.

Note, the CC acronym for what is being transitioned to has two distinguishable meanings:

1. **Coordinated Commons (CC)** - a concept describing how a societal system coordinates global resources for global human need fulfillment, in common.
   A. Attribution - citation sufficient for contextual source identification, and based upon a metadata standard identity collection template. If the commons is informational, then it is linked to a change within the contents of a social database, wherein social and personal profiles can make identifiable changes.
2. The **Creative Commons licensing organization** also abbreviated **(CC)**, who produces the well-known Creative Commons legally enforceable (by the State), copyright license set.
   A. Attribution - Give appropriate credit.

In the market-State, the "Creative Commons" label seems correct, because all legal persons are potential creators of licensable content, which they may contribute to the commons. Herein, a creative commons (CC) licensing organization exists to standardize commons-related State-enforceable licenses. In community, however, the commons is intentionally coordinated (coordinated commons; CC0), and there is no such thing as a property or property license rights. In a coordinated information system, logging of event identities is a synonym of "citation/accrediting)". Some content relates to societal operations, and hence is transparent, and some content relates to societal safety, and is hence personal to individual identities. So, discoveries, engineering, operations, etc., is all information that is in the coordinated commons and shared with all. In community, there is no commerce and no power-over-other type authoritarian relationships, so there is no commercial-related license, and there is no no-derivation related license. There is no privatization, and hence no need for CC0 or CC-BY in economic and/or societal content. Non-personal content is CC-BY-SA, to the extent that BY is logged as an event in the database to a given user identity. Personal information may be controlled on the network by individuals to the extent of the organizations they are participating in and their safety.

Note here that in community, not all information sets need to carry citation. For example, creative images, artwork, graphics, texture images, etc., may or may not have source metadata accompanying their use, and no one can use State force to prevent copying and sharing, expect on safety protocol (Read: legal) grounds. For the consumer/user, the source may or may not have any relevance to the use. Formal information, particularly information associated with standards and societal operations, necessarily carries citations. Items that have no need to carry citation ought to be marked specially, because it doesn't matter what "you" do with them or how "you" cite, because the content isn't relevant to the construction and operation of society. Furthermore, it is sometimes very difficult, and maybe even impossible to determine who an author was. At other times the content (whether considered design or not) may have many authors. There are many prior authors to the newest adaptation of any given art or engineering design. There are some open-source/copy-left licenses that do not require citation. This is particularly the case in software, where software does not require mentions to the program's authorship. In an open society, the default for information openness is set to "open", except where personal safety is the concern, and then it defaults to personal (i.e., Is that something you wanted shared with the world about yourself online?). A coordinated commons must account for the coordination of a common human population's access to a commons of resources and human contributions.

> **NOTE:** *Creative Commons copyright licenses do not contain specific terms about the distribution of source code, which is often important to ensuring the free reuse and modifiability of software.*

Under the Creative Commons licensing organizations license set, the following license categories are available:

1. **CC-BY (Creative Commons Attribution License):** Allows others to copy and redistribute the material in any medium or format and remix, transform and build upon the material for any purpose, even commercially. The source must be cited. CC-BY - must carry citation (a.k.a., attribution, reference,

source, etc.). In order to maintain transparency, accountability, and overall efficiency, it is important maintain attribution/citation. This license is reasonably similar to a public domain license, with the exception that citation/sourcing is required.

A. This is a non-persistent open-source copyleft license. Under the CC-BY licence, anyone who adapts the work can redistribute a modified version under the terms of their choice.

2. **CC-BY-SA (Creative Commons Attribution Share-Alike License):** Allows others to copy and redistribute the material in any medium or format and remix, transform and build upon the material for any purpose, even commercially, provided it is distributed under the same license as the original. The source must be cited. Because of the -BY-, attribution is given to the creator. CC-BY-SA optimizes the potential for more useful adaptations, then anyone in community can create optimizations and reciprocation. Conversely, with public domain, reciprocation is not necessarily the case (i.e., someone can adapt and then more permissively license the adaptation).

CC-...-SA - A Share-Alike licenses that does not allow a change of the license after copying or adapting; hence, the license does not allow someone in the future to restrict its sale or distribution, and without either, it prevents the adaptations from being licensed more privately. This is a reciprocal license. CC BY-SA, however, binds the adapter to the terms of the original licence. These licenses ensure that the commons is maintained going forward. These types of licenses are also called "copy-left", because a legal person can only copy them if that person maintains the copyleft "copy-right" - it has to stay "left" (i.e., in the commons where it is copyable). Whereas, in copy-right the "right" means privatization, personal property rights.

A. This is a persistent open-source copyleft license. Under the CC-BY-SA licence, anyone who adapts the work can redistribute a modified version under only the same terms as the received version.

B. This is essentially a license for transition to community that allows content to be sold. That content may or may not have been produced using community resources. What community resources is it ok to sell into the larger market to maintain the community habitat service system?

C. **NOTE:** *There ought to also be a CC-SA license for when attribution is unnecessary, and the "owner" wants to keep the content as free as possible going forward. Sometimes attribution (citation) is necessary for scientific and scholarly integrity. Sometimes attribution is wholly unnecessary, such as with generic 3D modeling.*

D. **IMPORTANT:** CC-BY is one-way compatible with CC-BY-SA. You may adapt a BY work and apply CC-BY-SA to your contributions, but you may not adapt a CC-BY-SA work and apply BY to your contributions. Two-way compatibility means that you may adapt work under one license (X) and apply a second license (Y), and vice versa.

3. **CC-BY-ND (Creative Commons No-Derivatives License):** Allows others to copy and redistribute the material in any medium or format. However, if you remix, transform or build upon the material these modifications cannot be distributed.

A. This is not an open-source copy-left license. Under the CC-BY-ND licence, adaptations cannot be distributed.

4. **CC-BY-NC (Creative Commons Non-Commercial License):** Allows others to copy and redistribute the material in any medium or format. However, the material may not be used for commercial purposes.

A. This is not an open-source copyleft license. Under the CC-BY-NC licence, neither copies, nor adaptations can be sold/distributed.

B. **CC-...-SA-NC** - This is essentially a license for transition to community that does not allow content to be sold. That content may or may not have been produced using community resources. What community resources is it ok to sell into the larger market to maintain the community habitat service system?

C. **CC-...-NC** - Non-commercial licenses that does not allow the work to be used or added to any sellable/sold product. Content licenced via CC BY-NC cannot be mixed with CC BY or CC BY-SA. When a State releases content under a CC BY-NC license, it must then make the content available at no cost to any citizen upon request. Citizens and economic entities can print the copy and give away the copies, but they cannot sell the copies, and they cannot mix the copies with content that is being sold. This license limits potentially valid and useful access to information, making it's potential availability more limited during transition, but it does protect information resources inside of community that it does not want sold/traded in the market. Similarly, with physical objects during transition, some are produced to be sold into the market, and some to be used locally. The NC part of this license forbids any commercial use; so it forbids sale. It effectively limits, unnecessarily, distribution. It would make integration of others' commons licenses difficult. Technically, you couldn't then sell a book licensed like that on Amazon.com, even if for no profit, because it is a non-commercial license. When States license NC work, they print it with tax money, and give it away. It is true that objects in community ought not be sold outside, unless specified in prior decisioning as a medium of interfacing with the market, but there is no need to place such a restrictive license on information being used for construction and transition to a community-type configuration. The SA part is sufficient and allows for best distribution of information during transition; it is a true community transition license that allows more people to participate.

D. Print versions of the societal specification standard are currently sold to the public. Even when a State makes all access to the digital and print version of the standards available for free, to all, the license for the standard may not change from CC-BY-SA to CC-BY-SA-NC.

E. Content licenced via CC BY-NC cannot be mixed with CC BY or CC BY-SA. When a State releases content under a CC BY-NC license, it must then make the content available at no cost to any citizen upon request. Citizens and economic entities can print the copy and give away the copies, but they cannot sell the copies, and they cannot mix the copies with content that is being sold.

F. Print versions of the societal specification standard are currently sold to the public. Even when a State makes all access to the digital and print version of the standards available for free, to all, the license for the standard may not change from CC-BY-SA to CC-BY-SA-NC.

5. **CC-BY-NC-SA (Creative Commons Non-Commercial Share-alike):** Allows others to copy and redistribute the material in any medium or format, remix, transform and build upon the material for any non-commercial purpose, but the material may not be used for any commercial purpose. If the material is remixed, transformed or built upon, it must be distributed under the same license as the original. This is a highly restrictive license and is sometimes used by States. The State uses tax money to produce and print/publish the works (licensed under this license), and then, the State gives the material away.
   A. This is not an open-source copyleft license. Under the CC-BY-NC-SA licence, neither copies, nor adaptations can be sold/distributed, and if it is adapted and distributed, the license remains the same.
6. **CC-BY-NC-ND (Creative Commons Non-Commercial No-Derivatives License, a.k.a., CC-BY-NC-ND-SA):** Allows others to copy and redistribute the material in any medium or format. However, the material may not be used for commercial purposes and if remixed, transformed or built upon the modifications cannot be distributed.
   A. This is not an open-source license, because it does not allow for commerce.
7. **CC0 (Creative Commons Zero, P0)** - license (contract) stating forfeiture of all rights and an entrance of the content into the public domain. CC0 releasing owners wish to permanently relinquish their rights to a work for the purpose of contributing to a commons of creative, cultural and scientific works ("Commons") that the public can reliably and without fear of later claims of infringement build upon, modify, incorporate in other works, reuse and redistribute as freely as possible in any form whatsoever and for any purposes, including without limitation commercial purposes. There is no need to even cite public domain content.
   A. This is a license that puts the content in the public domain (PD).

6 Auravana license agreements

A.k.a., Contributor agreement, project Terms of Service (ToS), service license agreement, property license agreement, contribution license agreement (CLA), copyright assignment agreement (CAA), contributors work-role agreements, contribution service agreement, contributor open-access agreement.

Licensing is the base of a legal-State operational framework of content creation, distribution, and use across various domains, including software development, media, intellectual property, and real property. It encompasses a wide array of agreements that grant permissions or rights from the owner to another party (Read: rights assignment), often delineating how the licensed work can be used, shared, or modified. These licenses can range from highly restrictive, controlling the extent of use and distribution, to permissively and publicly open, allowing for broad use and privatization of the work. In the context of collaborative environments and open-source projects, licensing becomes especially critical, ensuring that contributions can be legally integrated and distributed while protecting the "rights" of all parties involved. The intricacies of licensing agreements, including Contributor License Agreements (CLAs), which form the backbone of this ecosystem, safeguarding the collaborative orientation by clarifying terms of contribution, usage, and distribution of collective work. A Contributor Licence Agreement (CLA) is strongly recommended when accepting contributions to an open development compilation project

The conditions of license agreements and Contributor License Agreements (CLAs) are closely connected with an organization's Terms of Service (ToS), typically stated on their website. These ToS not only govern the general use of the organization's services and platforms, but also delineate the legal framework within which users and contributors engage with the organization's resources and work. By specifying the rights, responsibilities, and restrictions associated with the service, the ToS complements CLAs by providing a broader legal context that ensures all interactions and contributions align with the organization's operational, legal, and value standards. This comprehensive approach ensures that while CLAs address the specifics of contributions to projects, particularly in open-source environments, the ToS establishes the overarching legal agreement that users consent to, thereby securing a cohesive legal and operational ecosystem that supports the organization's objectives and protects its interests as well as those of its users and contributors.

6.1 Contributors license agreement

A CLA establishes the legal groundwork for an open-source project by defining the terms and conditions under which contributions — be they code, artwork, documentation, or translations — are made to a project. The purpose of a contributor license agreement (CLA) is to define the terms and conditions under which a contribution will occur. A CLA is a legal agreement between the project maintainers and contributors to establish the terms under which contributions are made to the project. A CLA ensures that contributors grant the project the necessary rights to use, modify, and distribute their contributions, and it may specify the license under which contributions are made available. CLAs are particularly important for open-source projects to manage legal risks and ensure that contributions can be distributed under the project's chosen open-source license without encountering legal barriers. Serving as a formal agreement between project maintainers and contributors, CLAs are instrumental in clarifying the rights conveyed with contributions, ensuring that the project has the legal ability to use, modify, and distribute these contributions under its chosen license. This is paramount in managing legal risks and affirming that the project's outputs can freely circulate without legal impediments. For open-source initiatives, the assurance provided by CLAs that contributions integrate seamlessly and permanently into the project underpins the project's integrity and its ongoing distribution under the open-source license. Moreover, the terms and conditions outlined in these agreements, often accessible on the project's platform, fortify the project's legal standing in the modern digital ecosystem, guaranteeing that contributions remain a part of the project in perpetuity.

A contributors license agreement ensures that a project's outputs has the necessary ownership or grants of rights over all contributions to allow them to be distributed under the chosen license. When a contribution is made to an open [source] project, there is an implicit assumption (and sometimes explicit consent) that the contribution (code, translation, artwork, etc.) may be incorporated into the project and distributed under the license the project is using. Often, open source projects will state their Terms and Conditions, an may even link a free or open source license. These terms, which are accessible via the project's platform, are generally all that is required to protect an open source platform in the modern 21st century market-State. Most importantly, their simple presence ensures that contributions cannot be withdrawn by the contributor.

> **NOTE:** *Normally, when a contributor submits a contribution to a project, that contribution is going to be licensed under the terms of that project.*

Regardless of the chosen license, what matters most in the market-State is that the contributions coordinator ensures that when accepting project contributors/contributions, a contribution service coordinator must have received, accepted, and recorded/stored a signed Contributor License Agreement, which is acquired from each contributor. Or, at least, a specific agreement to the Project's Terms of Service. At the minimum there must be agreement of the Project's Terms of Service. This will allow the Project and others to safely redistribute their contributions, and possibly, to change to another open-source license later. A primary responsibility of the contributions coordinator lies in ensuring the seamless acceptance and recording of project contributors and their respective contributions. Central to this role is the acquisition, acceptance, and archival storage of a signed Contributor License Agreement (CLA) and work descriptions, from each individual.

Generally, the purpose of the CLA is: "The purpose of this agreement is to clearly define the terms under which intellectual property has been contributed to the Projet and thereby allow us to defend the project should there be a legal dispute regarding the contribution at some future time." Note that an additional contributor agreement can create additional, unnecessary, administrative work for project maintainers. How much work an agreement adds depends on the project and implementation. A simple agreement might require that contributors confirm, with a click, that they have the rights necessary to contribute under a project's open source license. A more complicated agreement might require a signature from the contributor and from a Project coordinator. A downside of Contributor Agreements is that they pose a small overhead and barrier to contribution.

Contributor Agreements may provide additional confidence that there likely will not be any legal issues in the future regarding the individual contributions that make up the project, such as disputes over origin, ownership, and loss. Also, a project might need to change licenses over its lifetime and want contributors to agree in advance to such changes.

Contributor agreements for a societal project may cover:

1. **Copyright:** Contributors grant a broad set of permissions and they are sometimes asked to assign their copyright to the project. The Contributor Agreement also ensures that contributors are entitled to contribute their changes to the project.
2. **Trademarks:** Contributors ensure that marks (if there are any) are owned by the project rather than by individual contributors. This avoids possible disputes in the future if contributors leave a project.
3. **Patents:** Contributors grant a patent license to the project in order to ensure that a contributor cannot attack the project in the future by asserting its patents against it.
4. **Market-State rights:** Contributors are asked not to assert any market-State rights (where they exist) in order to stop derivative works.
5. **Contributions by minors:** Some Contributor Agreements define how contributions by minors are handled.

The Terms of Service (ToS) of a project/website are similar to a contributors license agreement (CLA). Whereas the CLA focuses on contribution, the ToS outlines rules and guidelines for using a website or service. They are both legal documents that play a role in decisioning, and address intellectual property rights and risk management.

6.2 Auravana Project Individual Contributors License Agreement and Terms of Service

There are three agreement sets (abbreviated in the bullets below):

1. Terms of Service for the Auravana Project:
   - **[ToS Project].**
2. Contributors License Agreement for the Auravana Project:
   - **[CLA Project].**
3. Contributors License Agreement for workshops held by the Auravana Project:
   - **[CLA Workshop].**

[ToS PROJECT]
These Terms of Service ("Terms") govern your participation in and use of the services and resources provided by the Auravana Project ("Project"). By engaging with the Project in any way and via any medium, you agree to these Terms.

PROJECT AURAVANA, also known as AURAVANA or AURAVANA PROJECT, and hereafter known as the PROJECT (auravana.org).

These Terms and Conditions are most up-to-date and current on the Auravana Project's website: https://auravana.org/about/terms-and-conditions

1. **Definitions:**
   "You" and "Your" refer collectively to both "Users" (of the Project's website and any and all Project work and resources) and "Contributors" (to the Project and any and all Project work and resources), encompassing anyone engaging with the Project, whether by using the Website, accessing the Services, or contributing Content.

[CLA PROJECT]
This Contributors License Agreement is made publicly by the undersigned ("Contributor") and Project Auravana ("Project"), hereinafter as "Party" or collectively as "Parties", for the purpose of participation in Project Auravana.

In order to clarify the intellectual property license granted with Contributions from any person or entity, The Auravana Project must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms.

WHEREAS, both Parties wish to clarify the intellectual property license involved in the Workshop;

WHEREAS, both Parties wish to contribute their intellectual property in order to assist the development of the Workshop;

"You" accept and agree to the following terms and conditions for "Your" Contributions (present and future) submitted to the Auravana Project.

1. **Definitions:**
   "You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with the Project. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "Contribution" shall mean any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Project for inclusion in, or documentation of, any of the outputs of the Project (the "Work"). For the purposes of this definition, submitted" means any form of electronic, verbal, or written communication sent to the Project or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Project for the purpose of discussing and improving the Work.

[CLA WORKSHOP]
This Contributors License Agreement ("Agreement") is made publicly by the undersigned ("Contributor") and [Workshop Organizer Name] ("Organizer"), hereinafter as "Party" or collectively as "Parties", for the purpose of participation in the [Workshop Name] ("Workshop") on [Dates].

WHEREAS, both Parties wish to clarify the intellectual property license involved in the Workshop;

WHEREAS, both Parties wish to contribute their intellectual property in order to assist the development of the Workshop;

By participating in [Workshop Name] ('Workshop'), in any capacity, including but not limited to attending in person, engaging remotely, or contributing in any manner related to the Workshop activities, whether before, during, or after the actual event, you ('Contributor' or 'You') hereby accept and agree to the following terms and conditions with respect to your submission of ideas, code, documentation, designs, and any other materials or contributions ('Contributions') made in connection with the Workshop.

By participating in [Workshop Name] ('Workshop'), in any capacity, including but not limited to attending in person, engaging remotely, or contributing in any manner related to the Workshop activities, whether before, during, or after the actual event, you ('Contributor' or 'You') hereby accept and agree to the following terms and conditions with respect to your submission of ideas, code, documentation, designs, and any other materials or contributions ('Contributions') made in connection with the Workshop.

1. **Definitions:**
   "You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with the Workshop. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "Contribution" shall mean any original work of authorship, including any modifications or additions to an existing work, that is submitted by You to the Workshop for inclusion in, or documentation of, any of the outputs of the Workshop (the "Work"). For the purposes of this definition, submitted" means any form of electronic, verbal, or written communication sent to the Workshop or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Workshop for the purpose of discussing and improving the Work.

[ToS PROJECT] & [CLA PROJECT] & [CLA WORKSHOP]

2. **Contributions and licensing:**
You reserve all right, title, and interest in and to Your Contributions.

The Project/Workshop plans to share the final work product ("Deliverable") which includes everyone's Contribution, with a copy-left license. This approach ensures that the Deliverable remains freely available for both commercial and non-commercial use under the terms of a copy-left license. All users and redistributors are required to give appropriate credit, provide a link to the license (where applicable), and indicate if changes were made to the original work. Furthermore, if the Deliverable is adapted, modified, or built upon, the resulting work must also be shared under the same or a compatible copy-left license, thus ensuring the freedom to use, modify, and share the work is preserved for all subsequent creations.

The Project/Workshop employs copy-left licenses and open patent licensing to ensure contributions remain open and freely accessible. These copy-left licenses ensure that contributions can be freely used, modified, and shared, provided that any derivatives are also shared under the same terms. Outputs of the Workshop are managed under the following copy-left licenses to ensure our shared work remains freely accessible and reusable:

- **Creative Content:** Licensed under the Creative Commons Attribution-ShareAlike (CC BY-SA) license.
- **Software:** Licensed under the GNU General Public License (GPL), Lesser General Public License (LGPL), or Mozilla Public License (MPL), as appropriate.
- **Patentable Inventions:** Licensed under open patent licenses, requiring derivatives to be shared under the same terms.

//TOS PROJECT**//**
Under this license anyone is also free to freely distribute (for no profit), as well as trade and sell (for profit), The Auravana Project's intellectual property; however, if "you" choose to sell any of The Auravana Project's intellectual property in whole or in part you must prominently display:

- Referentially credit Project Auravana as the source of the intellectual property.
- Include a link to where a customer may retrieve the intellectual property for free from Project Auravana.

//TOS PROJECT**//**

3. **Grant of copyright license:**
By contributing to the Project/Workshop in any form, including but not limited to submissions such as questions, comments, suggestions, and contributions of any nature ("Submissions" and "Contributions"), you acknowledge that they are for use in a compilation and agree to grant the Workshop a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable license to use, modify, reproduce, prepare derivative works from, publicly display, publicly perform, sublicense, and distribute your Contributions and any derivative works. This grant includes the right for the Project/Workshop to sublicense and create derivative works under its chosen copy-left license, thereby ensuring that your Contributions, as well as any adaptations or modifications thereof, remain freely accessible and distributable under the same or a compatible copy-left license terms. By making your Contributions, you implicitly agree to allow the Project/Workshop to use them in accordance with its selected copy-left licensing model, promoting an environment of collaboration and open sharing that upholds the principles of copy-left licensing to benefit all participants and the wider public.

4. **Grant of patent license:**
By contributing to the Project/Workshop, you hereby grant to the Project/Workshop, and to any recipient of the Work distributed by the Project/Workshop a perpetual, worldwide, transferable, non-exclusive, no-charge, royalty-free, irrevocable, and sublicensable patent license. This license encompasses the rights to make, have made, use, sell, offer to sell, import, and otherwise transfer your Contribution, in whole or in part, alone or included in any Work. The license applies to any patent claims owned or licensed by you that are necessarily infringed by your Contribution or by its combination with any Work distributed by the Project/Workshop. Furthermore, this patent license extends to those claims licensable by you that are necessarily infringed by your Contribution(s) alone or in combination with the Work to which such Contribution(s) have been submitted. Should any entity initiate patent litigation against you or any other entity, including filing a cross-claim or counterclaim in a lawsuit, alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement

for that Contribution or Work will automatically terminate as of the date such litigation is filed. This provision ensures that the Project/Workshop and its users are protected from patent claims while promoting a collaborative environment that respects the principles of copy-left and open patent licensing.

[CLA PROJECT]

5. **Modifications and Derivative Works:**
   At the Projects sole, absolute and unfettered discretion, it may make any changes in, deletions from, and/or additions to Contributions. Contributions may be modified or combined to create derivative works.

[CLA WORKSHOP]

6. **Modifications and Derivative Works:**
   At the Workshops sole, absolute and unfettered discretion, it may make any changes in, deletions from, and/or additions to Contributions. Contributions may be modified or combined to create derivative works.

[CLA PROJECT] [CLA WORKSHOP]

7. **Your Representations and Warranties:**
   You represent that you are legally entitled to enter this Agreement and to grant the above license. By submitting a Contribution, you represent and warrant that: (a) each Contribution you submit is an original work and you can legally grant the rights set out in this Agreement; (b) the Contribution does not, and any exercise of the rights granted by you will not, infringe any third party's intellectual property or other right; and (c) you are not aware of any claims, suits, or actions pertaining to the Contribution. You will notify us immediately if you become aware or have reason to believe that any of your representations and warranties is or becomes inaccurate. Contributors bear full responsibility for ensuring the compatibility of their contributions with Projects/Workshops licenses.

[CLA PROJECT] [CLA WORKSHOP]

8. **No Warranty or Support Obligations:**
   You are not expected to provide continuous support or warranty for your contributions, except to the extent you desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, you provide your contributions on an "as is" basis, without warranties or conditions of any kind, either express or implied, including, without limitation, any warranties or conditions of title, non infringement, merchantability, or fitness for a particular purpose.

[ToS PROJECT]

9. **Modification of Terms:**
   The Project reserves the right to modify these Terms. Changes take effect immediately upon posting to the Project's website. Your continued use of the Project services and resources following the posting of changes to these Terms and Conditions constitutes your acceptance of those changes.

[CLA PROJECT]

10. **Modification of Terms:**
    The Project reserves the right to modify these Terms. Changes take effect immediately upon posting, with Contributors being notified through available communication channels. Your continued contributions to the project after the effective date of the modifications will constitute your acceptance of the modified terms. You are responsible for regularly reviewing the terms of this CLA to ensure compliance with any modifications. Failure to review and acknowledge modifications does not exempt you from the obligations outlined in the modified terms.

[CLA PROJECT]

11. **Attribution:**
    You consent to The Project publicly using your name and the name of any organization you represent in connection with promoting and publicizing the Project's work. This allows us to acknowledge your contributions and association with the project. Contributing individuals will be acknowledged for their contributions to the Project in a manner standard to attribution within a standards setting working group. The Organizer ensures the inclusion of Contributors in published deliverables as members of their respective working groups, without bearing liability for any disagreements related to such attribution.

[CLA WORKSHOP]

12. **Attribution:**
    You consent to the Workshop publicly using your name and the name of any organization you represent in connection with promoting and publicizing the Workshop's work. This allows us to acknowledge your contributions and association with the Workshop. Contributing individuals will be acknowledged for their contributions to the Workshop in a manner standard to attribution within a standards setting working group. The Organizer ensures the inclusion of Contributors in published deliverables as members of their

respective working groups, without bearing liability for any disagreements related to such attribution.

[ToS PROJECT]

13. **Identity Limitations:**
    No trademark rights held by the Project are waived, abandoned, surrendered, licensed or otherwise affected by this document. "You" agree:
    A. Not to use the Workshop's (Project's) name, logo, and motto as they appear on this site and all open source creations and properties, without the prior written consent and agreement of Project Auravana's current Global Projects Coordinator. "You" agree to show the Global Projects Coordinator how the logo will be used, and will not use it to harm the Auravana Project or associate the Project with information that is false about The Auravana Project.
    B. Not to use photographs, biographies, names, and likenesses of anyone involved with Project Auravana, as they appear on this site, for any purpose without the prior written consent and permission of the person.

[ToS Project]

14. **Indemnity:**
    Users, contributors, and participants agree to indemnify, defend, and hold harmless the Project, its organizers, all coordinators, contributors, participants, and their successors from and against any claims, liabilities, damages, losses, costs, expenses (including but not limited to attorney's fees), arising out of or in any way connected with their contributions to the Project, including but not limited to claims of copyright infringement, patent infringement, and misuse of proprietary information.

[CLA PROJECT]

15. **Indemnity:**
    Contributors agree to indemnify, defend, and hold harmless the Project, its organizers, all coordinators, contributors, participants, and their successors from and against any claims, liabilities, damages, losses, costs, expenses (including but not limited to attorney's fees), arising out of or in any way connected with their contributions to the Workshop, including but not limited to claims of copyright infringement, patent infringement, and misuse of proprietary information.

[CLA WORKSHOP]

16. **Indemnity:**
    Contributors agree to indemnify, defend, and hold harmless the Workshop, its organizers, all coordinators, contributors, participants, and their successors from and against any claims, liabilities, damages, losses, costs, expenses (including but not limited to attorney's fees), arising out of or in any way connected with their contributions to the Workshop, including but not limited to claims of copyright infringement, patent infringement, and misuse of proprietary information.

[ToS PROJECT]

17. **Limitation of Liability:**
    The Auravana Project does not represent or warrant that the information accessible via these sites and Project work and resources is accurate, complete or current. These sites could contain typographical errors, miscatalogued information, or technical inaccuracies. The Project makes no representation or warranty regarding the accuracy, completeness, truth, quality, suitability or reliability of such information. The Project reserves the right to add to, change or delete its content or any part thereof without notice. By your use of these materials, you agree that you do not and will not rely on it being appropriately catalogued or complete for any purpose, including but not limited to patentability of any invention, invalidity of any claim of any patent or application, or authorship and inventorship of any article, patent or application. You further agree that you or your agents will not take any action against the Project due to reliance on any information which you believe to be non current, inaccurate or incomplete. The Project is not responsible for any errors, omissions, or inaccuracies contained in any information provided by the Project or Contributors.

    All content provided by the Project is provided as is. The Project and its content providers make no warranties, express, implied or otherwise, regarding its accuracy, completeness or performance. Users, contributors, and participants agree to release the Project, all coordinators, contributors, participants, and their successors from claims, demands, losses, damages, rights, and actions of any kind, including personal injuries, death, and property damage, related to Project activities or deliverables. "You" agree to be entirely responsible for your use of any/all content of whatever nature derived from this site and all the Project's creations and agree to hold us free of any/all liability arising from "your" use or misuse.

[CLA PROJECT]

18. **Limitation of Liability:**
Contributors agree to release the Project, all coordinators, contributors, participants, and their successors from claims, demands, losses, damages, rights, and actions of any kind, including personal injuries, death, and property damage, related to Workshop activities or deliverables. Contributors acknowledge that they are entirely responsible for their use of any/all content derived from the Workshop and agree to hold Workshop coordinators and organizers free of any/all liability arising from such use or misuse.

[CLA WORKSHOP]

19. **Limitation of Liability:**
Contributors agree to release the Workshop, all coordinators, contributors, participants, and their successors from claims, demands, losses, damages, rights, and actions of any kind, including personal injuries, death, and property damage, related to Workshop activities or deliverables. Contributors acknowledge that they are entirely responsible for their use of any/all content derived from the Workshop and agree to hold Workshop coordinators and organizers free of any/all liability arising from such use or misuse.

[CLA PROJECT]

20. **Termination:**
Contributions made to Project Auravana and licensed under this Agreement cannot be withdrawn by the Contributor once submitted. The terms of this Agreement shall apply to the Contributions for the duration specified by the applicable copyleft licenses.

[CLA WORKSHOP]

21. **Termination:**
Contributions made to the Workshop and licensed under this Agreement cannot be withdrawn by the Contributor once submitted. The terms of this Agreement shall apply to the Contributions for the duration specified by the applicable copyleft licenses.

[ToS PROJECT] [CLA PROJECT] [CLA WORKSHOP]

22. **Entire Agreement:**
These Terms constitute the complete agreement between the Parties, superseding all prior communications, agreements, and understandings, written or oral.

[ToS PROJECT] [CLA PROJECT] [CLA WORKSHOP]

23. **Governing Law and Dispute Resolution:**
This Agreement and the rights of the parties hereunder shall be governed by and construed in accordance with the laws of the State of *New York*, exclusive of conflict or choice of law rules.

The parties shall attempt in good faith to resolve any dispute arising out of or relating to this Agreement promptly by negotiation between executives who have authority to settle the controversy and who are at a higher level of management than the persons with direct responsibility for administration of this Agreement.

If the matter is not resolved by negotiation pursuant to paragraphs above, then the matter will proceed to arbitration as set forth below.

Any dispute, controversy or claim arising out of or relating to this contract, including the formation, interpretation, breach or termination thereof, including whether the claims asserted are arbitrable, will be referred to and finally determined by arbitration in accordance with the JAMS International Arbitration Rules. The Tribunal will consist of three arbitrators. The place of arbitration will be New York City, NY. The language to be used in the arbitral proceedings will be English. Judgment upon the award rendered by the arbitrator(s) may be entered in any court having jurisdiction thereof.

You further agree to waive all injunctive relief and to limit any monetary damages sought to a maximum of US $1,000 (one thousand US dollars) should a dispute arise from a breach of these Terms.

[CLA PROJECT]

24. **Acknowledgement:**
The Contributor acknowledges that they have read this Agreement, understand it, and agree to its terms and conditions.

[CLA WORKSHOP]

25. **Acknowledgement:**
The Contributor acknowledges that they have read this Agreement, understand it, and agree to its terms and conditions.

[CLA PROJECT] & [CLA Workshop]
Signatures:
This is a legal contract containing Personally Identifiable Information. Please refer to https://auravana.org/about/terms-and-conditions for the policy governing how this information is used and shared.

By signing below, the Contributor agrees to the terms of this Agreement.

Name: ______________________________
Signature: ___________________________
Date: __________________
Email: __________________
Postal Street Address: __________________
Telephone Number: __________________

[ToS PROJECT]

26. **Accessibility of Website:**
   The Project's aim is to ensure accessibility to the website at all times, however we make no representation of that nature and reserve the right to terminate the website at any time and without notice. You accept that service interruption may occur in order to allow for website improvements, scheduled maintenance, or may also be due to outside factors beyond the Project's control, to which we shall not be liable.

[ToS PROJECT]

27. **Cookie Policy for European State GDPR Compliance:**
   This site uses cookies to make your browsing experience more convenient, provide social media features, and analyse traffic. Cookies store useful information on your computer to help us serve fonts and acquire user analytics. Cookies are essential to providing full functionality to the site. By accessing this site, you consent to the use of cookies. By using our websites, you are consenting to our use of cookies. If you do not agree to our use of cookies in this way, you should set your browser settings accordingly, disable the cookies that we use or not use our websites at all. If you disable the cookies we use, this may impact your user experience while on the website (e.g., the display of fonts).

[ToS PROJECT]

28. **Privacy Policy for European State GDPR Compliance:**
   All information submitted to, or through, the Project has no expectation of privacy. The Project only requires the minimum amount of personal information that is necessary to fulfill the purpose of your interaction with the Project.

The legal basis for processing personal data:

Personal data category — Legal basis

- Email address and name (Contact webpage) — Legitimate interest
- Personal identifying information to generally include name, postal address, and other personal details (Online registration and contracts) — Legitimate interest
- User submissions and User content (site contact forms and Contact webpage) — Legitimate interest
- Cookie analytics (website functioning, social media, traffic analytics) — Legitimate interest
- Others

In addition to the uses described above, the project uses your information in the following ways:

- To conduct analytics and improvement.
- To train artificial intelligence systems.
- To respond to your inquiries related to support, opportunities, and other requests.
- To legally acknowledge and process signed agreements.
- To provide the products and services you request.
- To improve the design of society by including it in a public, common, open source design specification for the next iteration of society.
- To share with the public.
- To add to and modify the system being designed and produced by this open source project.

How the Project collects your information (different methods are used):

- Direct interactions (e.g., email, contact page, forum) and indirect interactions (e.g., surveys), by communicating on, through, or to the project, or to one of its contributors.
- Automated technologies or computer interactions. As you interact with the Project and the Project's website, it may automatically collect Technical Data about your equipment, browsing actions and patterns. The project collects this personal information by using cookies, [server logs] and other similar technologies. [We may also receive Technical Data about you if you visit other websites employing these cookies.] Cookies (You can set your browser to refuse all or some browser cookies, or to alert you when websites set or access cookies. If you disable or refuse cookies, please note that some parts of this website may become inaccessible or not function properly.)

How the project stores your information (different methods are used):

- Your information is stored on email servers and personal computers. "You" accept that any information you make available to the Project or its contributors may be made available to the public at

such information.
- In all interactions with The Project, there is no expectation of privacy.

How "you" can access and control the information collected:

- Please be aware that once your personal information is submitted to The Project, direct access and the ability to alter or delete your data may be limited. This limitation is due to the intrinsic nature of The Projects specific operation, such as data integration, technical constraints, etc. While we strive to maintain the accuracy and integrity of the information in our systems, the specific technical and operational structure of The Project means that submitted data becomes integrated into our data systems in a manner that may not be readily retrievable or modifiable on an individual basis.

The Project may change the privacy policy and practices over time. The Project will not notify "you" of such changes unless legally required. When re-accessing the website, if it is important to you, please re-check the terms then.

29. **Contact Form Compliance for European State GDPR Compliance:**
The contact webpage, including contact form therein, and newsletter subscription form on every page, is used to collect personal information (name and email) so that project coordinators and other team members can contact "you".

Scholarly references (cited in document)

- *European Union Public License: Guidelines*. (2021). European Commission: Directorate of informatics. https://op.europa.eu/en/publication-detail/-/publication/c15c9e93-27e1-11ec-bd8e-01aa75ed71a1

Scholarly references (non-cited)

- Almeida, F., Oliveira, J., Cruz, J. *Open standards and open source: Enabling interoperability*. International Journal of Software Engineering & Applications (IJSEA). Vol. 2, No.1. January 2011. https://airccse.org/journal/ijsea/papers/0111ijsea01.pdf
- *ANNEX 1: GNSO Working Group Guidelines*. (2007). ICAN | GNSO. April, 11. https://gnso.icann.org/sites/default/files/filefield_23493/annex-1-gnso-wg-guidelines-07apr11-en.pdf
- Caplan, P. *Patents and Open Standards*. White paper. Originally published in Information Standards Quarterly, vol. 14, no. 4, October 2003. National Information Standards Organization. https://www.niso.org/sites/default/files/2017-08/Patents_Caplan.pdf
- Cowlagi, R.V., Saleh, J.H. (2015). *Coordinability and consistency: Application of systems theory to accident causation and prevention*. Journal of Loss Prevention in the Process Industries, 33, 200-212.
- *Designing the public domain*. Harvard Law Review. Vol. 122:1489. http://harvardlawreview.org/wp-content/uploads/pdfs/GEdAssistantwebsite,_newVolume_1225_March_PDFsnotesdesigning_the_publicdomain.pdf
- *IEEE P1722 Working Group Operating Procedures*. (2009). IEEE-SA. March. http://grouper.ieee.org/groups/1722/contributions/p1722WG_PandP.doc
- Schmidt, C.M. (2017). *Best practices for technical standard creation: Guidelines for the design, socialization, formalization, and adoption of new technical standards*. MITRE. Project No. 25AOH830-JM. https://www.mitre.org/sites/default/files/publications/17-1332-best-practices-for-technical-standard-creation.pdf
- Shaikh, M. and Cornford, T. (2011). *Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe*. London School of Economics Report. UK Cabinet Office, London, UK. https://eprints.lse.ac.uk/39826/1/Total_cost_of_ownership_of_open_source_software_(LSERO).pdf

Book references (non-cited)

- Hoe, N.S. (2006). *Free/Open Source Software: Open Standards. FOSS Open Standards/Standard-Setting and Open Standards*. Elsevier. International Open Source Network. https://en.m.wikibooks.org/wiki/FOSS_Open_Standards/Standard-Setting_and_Open_Standards
- Hyde, L. (2011). *Common As Air: Revolution, Art, and Ownership*. Fsg Adult.
- Kinsella, N.S. (2008). Against Intellectual Property. Ludwig Von Mises Institute. Auburn, Alabama. https://mises.org/library/against-intellectual-property-0

Online references (cited in document)

- *OpenChain Project*. The Linux Foundation. Accessed: March 8, 2020. https://www.openchainproject.org/get-started
- *Open Source Initiative. (*2007). Open Source Definitions. Open Source Initiative. https://opensource.org/docs/osd
- Tremblay, F. (2010). *The case for socialist intellectual ownership*. The Prime Directive. https://francoistremblay.wordpress.com/2010/12/13/the-case-for-socialist-intellectual-ownership

Online references (non-cited)

- *Copying is Not Theft - Official Version*. (2010). Question Copyright Channel. https://youtu.be/IeTybKL1pM4
- *Copy-me: A Webseries About Copying*. (2014). Copy-Me Channel https://youtu.be/62-UT84-fXM
- *Copyright and Creative Commons: Publishing with open licenses*. (2018). Library Publishing Coalition. https://librarypublishing.org/creative-commons-webinar/
- *Creative Commons Licensing For Authors*. Self-Publishing Review. Accessed: 11 January 2023. https://www.selfpublishingreview.com/2014/10/creative-commons-licensing-for-authors
- *Popular open licenses by Open Source Initiative*. Accessed: 11 January 2023. https://opensource.org/licenses
- *Comprehensively ordered license list.* GNU Operating System. Accessed: 11 January 2023. https://www.gnu.org/licenses/license-list.en.html
- *Debian Free Software Guildelines (DFSG).* Debian. Accessed: 11 January 2023. https://www.debian.org/social_contract#guidelines
- Falkvinge, R. (2013). *The Copyright Monopoly Was Always Intended To Prevent Freedom Of Expression*. Torrent Freak. https://torrentfreak.com/the-copyright-monopoly-was-always-intended-to-prevent-freedom-of-expression-130414
- Falkvinge, R. (2014). *Public Libraries Show Why Sharing Culture Should Never Have Been Banned in the First Place*. Torrent Freak. https://torrentfreak.com/public-libraries-show-why-sharing-culture-should-never-have-been-banned-in-the-first-place-140112
- *Licensing Policies, Principles, and Resources*. Project Open Data. Accessed: 11 January 2023. https://project-open-data.cio.gov/licensing-resources/
- *MDPI Open Access Information and Policy.* (2019). MDPI. https://www.mdpi.com/openaccess
- *Mixed source licensing*. (2016). Project Open. [project-open.com]
- Nimus, A. (2006). *Copyright, copyleft and the creative anti-commons*. https://we.riseup.net/assets/118534/Anna+Nimus+Copyright_pdf.pdf
- *One Community Open Source*. One Community Global. Accessed: May 26, 2019. https://www.onecommunityglobal.org/open-source
- *Open Source*. (2019). One Community Global. https://www.onecommunityglobal.org/open-source
- *Open Source and Open Standards*. (2012). OSGeo. https://wiki.osgeo.org/wiki/Open_Source_and_Open_Standards
- *Open Source and Standards Working Group*. (2018). Open Source Initiative. https://wiki.opensource.org/bin/Working+Groups+%26+Incubator+Projects/OSandStandardsWG/
- *Open Source License Obligation Checklist*. by Carsten Emde and Till Jaeger. Version 5. September, 3, 2017. https://www.osadl.org/uploads/media/Open-Source-License-Obligations-Checklists-v5.pdf
- *Open Source License Checklist*. (2019). Open Source Automation Development Lab (OSADL) eG. https://www.osadl.org/Open-Source-License-Checklists.oss-compliance-lists.0.html
- *Choose an open source license*. GitHub, Inc. Accessed: 11 January 2023. https://choosealicense.com
- *Open Source Guides*. May 26, 2019. https://opensource.guide
- *Open Source Initiative*. Licenses & Standards. Accessed: 11 January 2023. https://www.opengroup.org/standardsprocess/standards-dev.html
- *OSI Approved Licenses.* Open Source Initiative. Accessed: 20 April 2023. https://opensource.org/licenses/
- Richman, S. (2012). *Patent Nonsense*. The American Conservative. https://theamericanconservative.com/patent-nonsense
- Rosen, L. *Defining Open Standards*. Rosen Law. Accessed: 11 January 2023. http://www.rosenlaw.com/DefiningOpenStandards.pdf
- Rosen, L. *The Role of Standards in Open Source.* Rosen Law. Accessed: 11 January 2023. https://rosenlaw.com/the-role-of-standards-in-open-source/
- Stallman. R. *Why Open Source misses the point of Free Software*. GNU Operating System. Accessed: 11 January 2023. https://www.gnu.org/philosophy/open-source-misses-the-point.en.html
- Steele, R.D., (2016). *Open Source Everything Engineering (OSEE)*. Earth Intelligence Network. https://robertdavidsteele.com/wp-content/uploads/2015/12/Open-Source-Everything-Engineering-OSEE-A-Nordic-Manifesto.pdf
- Swartz, Aaron. (2008). *Guerilla Open Access Manifesto*. https://archive.org/stream/GuerillaOpenAccessManifesto/Goamjuly2008_djvu.txt
- *The Center For Internet & Society*. Openness. Accessed: 11 January 2023. https://cis-india.org/openness
- *The Open Source Way. (2017).* Red Hat, Inc. https://www.theopensourceway.org/
- *Top Guidelines for Open Science.* Center for Open Science. Accessed: 11 January 2023. https://cos.io/top/
- Tremblay, F. (2015). *TFAQ by Libsocs for "An"caps*. The Prime Directive. https://francoistremblay.wordpress.com/faqs/faq-by-libsocs-for-ancaps/

Approach: Timing

Travis A. Grant,

Affiliation contacts: *trvsgrant@gmail.com*

Version Accepted: 1 April 2024

Acceptance Event: *Project coordinator acceptance*

Last Working Integration Point: *Project coordinator integration*

Keywords: civil calendar event timing, calendaring, project calendar scheduling, project timing,

Abstract

This article explores the critical role of civil calendar event timing, scheduling, and date and time formatting within the context of project planning for societal operations. The article proposes a comprehensive redesign of the civil calendar system, advocating for a transition to a 13-month structure, with each month comprising 28 days for a total of 364 days, alongside an additional "non-day" to accurately align with the solar year. This restructured calendar is presented as a solution to the inconsistencies and inefficiencies inherent in the current Gregorian system, offering a more logical, streamlined approach to timekeeping that is better aligned with natural cycles. The article argues for the benefits of this system in simplifying scheduling and financial operations, making planning more predictable and consistent across various sectors. By introducing equal-length months, the proposal aims to eliminate the current system's irregularities, facilitating smoother economic planning, industrial scheduling, and personal organization. This calendar reform is positioned within a larger framework of societal transformation, aiming at reorganizing societal practices towards greater sustainability, efficiency, and harmony with ecological and human needs. The article outlines the rationale, advantages, and implementation strategy for this calendar system, emphasizing its role in a broader vision for an optimized, equitable society. The article also proposes a clear daily-time cycle unit and dating format.

Graphical Abstract

Image Not Yet Associated

1 Civil calendar event timing

A.k.a., Civil calendar systems.

Civil calendars play a crucial role in organizing and regulating human activities across the world. These calendars are distinct from astronomical calendars and are primarily designed for the practical and administrative needs of society. They provide a systematic way to measure and divide time into a useful scale (e.g., seconds, days, weeks, months, and years), facilitating the synchronization of various events and activities. Civil calendars are used for scheduling appointments, setting dates for holidays, planning meetings, and ensuring the smooth functioning of organizations and societies. From scheduling work shifts and planning production processes to coordinating agreement, civil calendars provide the temporal structure necessary for smooth socio-economic functioning. They serve as a common framework for event timing, allowing for the coordination of action and social occasions at specific intervals throughout the year.

Civil calendars are typically separated into the following primary sub-units:

1. Hierarchy command selection (multiple years separation).
2. One year (a.k.a., "yearly").
3. Multi-year non-command decision (a.k.a., multi-year separation).
4. Months: yearly economic production cycles.
5. Weeks: separation of work in to socio-fulfillment phases (axiomatic economic phases of work, no-work, god, flow, etc.).
    A. In the market-State, weeks are typically separate into a phase of work (for profit-survival and command), and a phase of separation from work (as profit-based and survival-centered actions).
    B. In community, lives are typically separated into the phases of activities that produce the most flow and fulfillment in our lives given our stage of human-life duration experiences:
        1. Developmental experience phases:
            i. Nurturing.
            ii. Educating.
            iii. Contribution.
            iv. Leisuring.
        2. Vector: Flourishing, as a dynamically optimizable variable-vector (encoded concept) across all four 'developmental experience' variables, optimized by the usage of the temporal sub-unit solar cycles of: daily, monthly, yearly, and multi-yearly durations of time. Here, there are 'months', in-place of 'weeks', where people have no need to take weakly breaks from their lives. In community, there is the 'flow' cycle in operation/experience over seasonal months, leading to greater experience of flow throughout all human-life phases.
6. Seasons: Solar ecological biospheric cycles.
    A. Which provide for:
        1. Vegetative growth phases.
        2. Livestock growth phases.
        3. Human-life growth phases.
            i. Nurturing, educating, contributing, leisuring.
        4. Human-flow fulfillment phases.
            i. Struggle, release, flow, restore.
        5. Habitat operation master-plan *cycles*-design/*phases*-operations.
            i. InterSystem-Team service-access operations.
                1. Common access operations.
                2. Personal access operations.
                3. User-access operations.
        6. Master planning decision cycles.

There are a set of sub-month units for the socio-economic planning of production and user lifestyles:

1. Week:
    A. "Week" days.
    B. "Week end" days.

In the context of socio-economics, there are also (sub-month) functional day units:

1. **Work days** (a.k.a., "week" days).
2. **Non-work days** (a.k.a., "week-end" days).
3. **Public holidays** (a.k.a., religious days, State enforced non-work days).

In concern to civil calendars, 3 basic issues can create significant additional complexity for individual and social planning purposes:

1. Uneven length of months.
2. Leap days (every year, an additional day or days).
3. Leap years (every 4 years, an additional day or days).

1.1 The solar year calendar

A.k.a., Solar-day cycle time, solar cycle time, astronomical solar time, local solar astronomical time.

The earth both rotates on its 'axis', and 'orbits', the sun (i.e., earth rotates around itself and revolves around the sun. A solar year pertains to the movement of stars and planets, and in the case of earth, it signifies a full orbit around the Sun, taking approximately 365 Earth days. This solar year corresponds to 365 day-night cycles, where each day results from the Earth's rotation on its axis, causing the alternating cycle of day and night as a specific location on Earth moves relative to the Sun.

When discussing solar-cycle time, there are three key variables:

1. **Earth orbit solar cycle:** This represents a consistent and equal time unit that repeats annually. It is based on the Earth's orbit around the sun. An earth orbit solar cycle is an equal/stable year-over-year time unit.
2. **Day-night cycle:** This refers to the regular cycle of day and night, with each day consisting of 24 hours.
3. **Derived "month" unit:** This is a time unit that divides the day-night cycle into equal parts within the solar cycle. In this context, it creates a month-like unit that aligns with the solar year. It can align with an equal or unequal number of days in each month).
   A. For instance, a thirteen-month cycle is a derived unit that spans 28 days, with an additional day added annually, and this extra day can be placed at any point throughout the year to maintain the stability of the calendar.
   B. Alternatively, the Gregorian calendar has months with 28, 29, 30, and 31 days.

NOTE: *Each solar system, planet, and moon will still have its own solar year and solar day*

To divide the total number of day-night cycles in a solar year into equal parts, we can approximate it by creating a calendar with 13 months, each consisting of 28 days, plus one additional day. This arrangement amounts to 364 days (28 days per month times 13 months) along with one extra non-day, totaling 365 days in alignment with the solar year. This structure ensures that each month comprises exactly twenty-eight days.

The solar calendar uses the objective and equal time units of:

1. Year (x1 revolution of the earth around the sun).
   A. Months (x13).
      1. Days (x28).
2. Non-day (i.e., non-month day)
   A. Year day (+1/yr).
   B. Leap day (+1 every 4yrs).

CLARIFICATION: *Leap years occur approximately every four years to account for the extra time it takes for the Earth to orbit the Sun, with some exceptions to keep the calendar synchronized with astronomical events. It is also possible to have "leap" seconds and "leap" minutes to bring clocks that have drifted back to perfect aligned timing. The concept of "leap" to denotes the adjustment made to timekeeping tools to align them more closely with a fixed time. A leap year (or intercalary year) is a year containing an extra day (or, in case of lunisolar calendars, an extra month) in order to keep the calendar year synchronized with the astronomical or seasonal year. By occasionally inserting (or intercalating) an additional day or month into the year, the drift can be corrected. A year which is not a leap year is called a common year.*

In the market-State, there is a weekly cycle of work, with a standardized four week work-break-work cycle. This four-week, seven-day work-break-work cycle is an artificial subdivision of the objective 13+1 month calendar. The week is potentially an unnecessary sub-division of the month for production (.e., economic) purposes. There are not equal weeks of seven days in a solar year. A solar year of 365-days divided by a cycle of 7-days leads to a unequal 52.41 weeks in a year cycle.

1.2 The lunar calendar(s)

A.k.a., Lunar cycle time, lunar-day cycle time, astronomical lunar time, local lunar astronomical time.

The lunar cycle, consisting of the phases of the moon, completes its course in approximately 29.5 days, culminating in a total of 354 days over 12 full cycles within a solar year. In other words, a full moon happens every 29.53 days with a lunar solar-year being approximately 354 days, which is 11 days shorter than Earth's solar year (of 365 days); therefore, depending on when the first full moon is, there will be 12-13 full moons per year. Given the solar year comprises 365 days (or 366 days in a leap year), approximately every two and a half years, a 13th full moon occurs within a single calendar year. This phenomenon disrupts the regularity of a standard balanced, thirteen-month calendar, where each month would uniformly consist of twenty-eight days, plus an additional day not assigned to any month (as the least number of additive days).

Historically, lunar cycles have played a pivotal role in timekeeping, particularly for religious faiths. Notably, ancient civilizations such as the Egyptians and Babylonians utilized the moon's phases as a calendrical system. For instance, the Babylonians, as early as the 5th century BCE, implemented a lunar calendar. This calendar was structured around 12 months, each spanning 29 or 30 days. To align with the solar year, they added an extra five or six days at the year's end, occasionally incorporating an additional intercalary month to adjust for the discrepancy between lunar cycles and the solar year.

Table 31. Measurement > Time > Calendar*: The days of the*

month for a lunar cycle are wildly different from month to month (-30 to +30), for 2018.

Month #	Date	Gregorian Delta Days
1	01-January	-2
2	31-January	+30
3	01-March	-30
4	31-March	+30
5	29-April	-2
6	29-May	0
7	28-June	-1
8	27-July	-1
9	26-August	-1
10	24-Sept-ember	-2
11	24-Oct-ober	0
12	23-Nov-ember	-1
13	22-Dec-ember	-1

In the Gregorian calendar from month to month, the delta for the date of the full moon varies by as much as -30 days to +30 days, which is a very broad range; although the more normalized range falls between -2 and +1, which is a more modest range date range of 4 (-2, -1, 0, 1).

In concern to the international fixed calendar, the lunar year is shorter than (354 days) and the lunar month is near the same (29.53 days). With a consistent 28 day month it is easier to predict the changes of the date of the next full moon from month-to-month, which amounts to same date next month + 1 day or +2 days. The international fixed calendar is more in sync with the lunar pattern than is the Gregorian calendar.

1.3 The international fixed calendar

> *A.k.a., Equal month calendar, the thirteen month calendar, uniform month calendar, the indigenous calendar, world calendar, the equal-month solar calendar, the Cotsworth plan, the Eastman plan, the fixed solar year calendar, the solar international calendar, international perpetual calendar, the international fixed civil calendar.*

The international fixed calendar (IFC) represents is a proposed calendar that can overcome the shortcomings of the conventional Gregorian calendar. This alternative calendar offers a unique approach to organizing time, aiming to simplify datekeeping, enhance global coordination, and provide a more rational and balanced calendar system. It offers an intuitive method for time organization, with the primary objectives of streamlining datekeeping, promoting international synchronization, and establishing a more logical and equitable calendar structure. At its core, this calendar entails each month consisting of precisely 28 days, resulting in a total of 13 months within a year. Summarily, this is a calendar that separates the solar year into 13 months, each with 28 days, with one day at the end of each year belonging to no month or week. This configuration results in a 364-day calendar year and one non-day between years. Essentially, the calendar has an equal number of days in every month, and the same number of months every yearly cycle. All months would have exactly four weeks.

If the month is to be a proper unit of account for time, then all months should be equivalent in days (28 days); the days of the month should not change for the month unit to have precise meaning in concern to time (i.e., to have functional temporal meaning). To equally partition months throughout a single 365 day year, there will be 13 months, each with an equal 28 days. And, there would be one "non-day" per year that is not part of the thirteen 28-day month cycle to bring the calendar baseline calendar year to 365 days. Thus, a year is composed of 365 days that are equally divided into 13 months of 28 days, with one "non-day" per year (to have a total of 365 days). The non-days (once per year and once every four years) are placed anywhere in the calendar between two months. Obviously, the "non-day" is still a real-world experiential day, it is just that it is not counted as a day of any month in the 365 day year cycle (thus, it is not a day of the week either):

- ((364 days = (13 months) (28 days)) + 1 non-day = 365) + 1 leap day/4 years = 366 days every 4th year

The international fixed calendar has the following unique characteristics:

1. The extra month is placed:
   A. Between June and July and is named Sol (for the Sun).
2. The extra, in-between day (364 + 1day) is placed:
   A. Between December and January. This day is called "new years" day (or, year day), and it does not count in the normal cycle of the week or the month.
3. A leap day is required to eliminate drift. Every four years there is a leap day that occurs:
   A. Right after June, before the month of sol starts (Cotsworth plan).
   B. Mid-month in Sol, on the summer solstice, between Saturday 14th and Sunday 15th, and it would be the longest day of the year (in Northern hemisphere).

Just like the "year" day, "leap" day doesn't count as a day. In this scenario, both year day and leap day will always fall on the weekend between Saturday and Sunday.

> **NOTE:** *To recalculate a specific day, first determine how many days into the year the specific day is in the old calendar (e.g, 72 days, or 182 days), and then map that one to the new calendar.*

Hence, the international fixed calendar has three categories of day units:

1. **"Month" days** (28 days per month).
   A. Where "weeks" are four equal separations within the 28 days of every month:
      • 28 days = (7 days)(4 weeks)
2. **"New year" days** (a.k.a., non-day, year-day; is not a day of the month, are not a month-day; happens once per cycle). This is not a day of the week or month.
3. **"Leap year" days** (a.k.a., leap day, leap non-day; is not a day of the month, are not a month-day; happens once every 4 cycles). It isn't the day of a week or month. This is not a day of the week or month.

CLARIFICATION: *Leap years (which keep the calendar in sync with the year) should not be confused with leap seconds (which keep clock time in sync with the day).*

Each month will have 28 days meaning each day of each month will be the same day of the week forever, meaning that the 1st of each month will always be the first day of the week (as a consequence, "birthdays" will remain on the same day of the week). Since each month consists of exactly four weeks, the first day of each month and every seventh day after that for the rest of the month is deemed to be a Sunday, the second day of each month and every seventh day after that for the rest of the month is deemed to be a Monday, and so on. Therefore, each month begins on a Sunday and ends on a Saturday. It could also start on Monday and end on Saturday, if that was the chosen pattern.

Table 32. Measurement > Time > Calendar*: All months in the international fixed calendar look like this and have 28 days, with the day of the week starting on Sunday:*

Week	1	2	3	4	5	6	7
	Sun	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**
1	01	02	03	04	05	06	07
2	08	09	10	11	12	13	14
3	15	16	17	18	19	20	21
4	22	23	24	25	26	27	28

Table 33. Measurement > Time > Calendar*: All months in the international fixed calendar look like this and have 28 days, with the day of the week starting on Monday:*

Week	1	2	3	4	5	6	7
	Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
1	01	02	03	04	05	06	07
2	08	09	10	11	12	13	14
3	15	16	17	18	19	20	21
4	22	23	24	25	26	27	28

To remove the "Friday the 13th" issue it is possible to start the calendar on Monday instead of Sunday, which:

1. Gets rid of Friday the 13th.
2. Allows each year, month, and week to start on a Monday which makes more sense since Monday is the beginning of the work week and has the full weekend at the end of the week.
3. The intercalary days will be found at the complete end of the week and month so it can be the 29th day of the month and still part of the weekend.

Table 34. Measurement > Time > Calendar*: If the year were separated into quarters; then, end dates for quarters all end on the same day of the week each and every year and the quarters are even. Normally, only the 4th quarter has a different number of days (+1 day) due to the solar year having 365 days per year and not the evenly divisible 364 days. In Leap Years Q3 has 1 extra day due to the addition of Leap Day on June 29 (if that is where the Lead Day is to be placed). Dates for the basic 13 week quarters end on the same day of the week each and every year. Only the 4th quarter has a different number of days due to the solar year having 365 days per year and not the evenly divisible 364 days. Here, there is an intercalary day, Year Day. In Leap Years Q3 will have 92 days. The quarters end on the same day of the week each and every quarter.*

Quarter	End Date	Days in Quarter
Q1	Sunday, April 7	91
Q2	Sunday, June 14	91
Q3	Sunday, August 21	91 (+1 Leap day every 4 years = 92 every 4 years)
Q4	Sunday, December 28	92 (91 + 1 non-day)

The benefits of this are intuitive, obvious, numerous, and significant:

1. With 13 twenty-eight day months, every month is identical in terms of the positioning of its weekly sub-unites in an ordered count (from 1-28). All months would have the same number of days (28), the same number of working days (except public holidays), and the same number of Sundays and Saturdays.
   A. The first of the month always falls on a Sunday; the month will always start on Sunday.
   B. The second of very month will always be a Monday. Monday will always be the 2nd, 9th, 16th, and 23rd of every month.
   C. The last day of the month always falls on a Saturday; the month would always end on Saturday.
   D. Each weekday would always occur on the same four fixed dates of the month.
2. Thus, a public holiday would always occur on the same week-day every year. Users will always know what day of the week every holiday and day of the month is.

3. Quart-years and half-years would be of the same length.
4. All the solstices and equinoxes will fall on the same day every year (and they will always be on a weekend).
5. Year-specific calendars would not longer be needed; one fixed monthly calendar would be sufficient (this economizes information and physical resources). The same calendar format can be used every year, and new calendars layouts do not have to be printed each year.
6. Planning or scheduling events becomes easier because dates are consistent every month and every year with the days of the month falling on the same days of the week.
7. Finances are easier to maintain and plan due to equal month lengths. Where goods carry a price and laborers are paid, calculating pay per month is no longer erratic. Calculating pay using the Gregorian calendar is erratic on a month-by-month basis, which makes monthly financial planning difficult. Where payment is by the month, payment is proportionately the same each month.
8. Production calculations per month have a stable number of days (and hours), allowing for a constant to which availability, uptime, and other production variables can be calculated simply. In production planning and analysis, using a stable number of days per month for calculations enables a consistent framework for assessing various production variables such as availability, uptime, maintenance schedules, and overall output. This approach simplifies the complexity inherent in production coordination. In other words, there is the same number of production days and hours each month.
9. Trends and analysis become easy to track using this format.
    A. 13 months with the same length (4 weeks of 28 days)
    B. Days of the month falling on the same days of the week
    C. Holidays and other events fall on the exact same date each month, each and every year.

The only negative to the calendar is:

- There is no guarantee that a given week day is always seven days from the same week day.

The logical place to start and end a year are the points where the earths hemisphere is maximally tiled toward and away from the sun (i.e., summer and winter solstice). Seasons are already delineated along this axis with the mid spring and vernal equinoxes; and the winter solstice is the start of the new year are already. So it is well reasoned that any new calendar system (taking over from the Gregorian) ought to also start a new year on the winter solstice.

A solstice is where the Earth's tilt creates either a really long day or really long night which occurs within the depths of the 2 most extreme seasons (summer and winter). In other words, the moments of maximum tilt [of the earth's hemispheres] away from the sun are the seasonal solstices:

1. **Summer solstice (a.k.a., summer *solstice*)** - the longest day of the year in the northern hemisphere and shortest day of the year in the southern hemisphere).
2. **Winter solstice (a.k.a., winter *solstice*)** - the shortest day of the year in the northern hemisphere and longest day of the year in the northern hemisphere).

An equinox is where the Earth's axis tilt creates a day and night of equal length, which occurs in the seasons which are transitional (spring and fall). In other words, the moments of equal tilt [of the earth's hemispheres away from the sun are the seasonal equinoxes:

1. **Autumnal equinox (a.k.a., fall *equinox*)** - the day when the sun crosses the celestial equator from north to south, signaling the start of autumn. Similar to the vernal equinox, during the autumnal equinox, the duration of day and night is nearly equal, but it marks the transition to cooler temperatures and the approach of winter in the northern hemisphere, and summer in the southern hemisphere.
2. **Vernal equinox (a.k.a., spring *equinox*)** - the day when the sun crosses the celestial equator from south to north, marking the beginning of spring. On this day, the length of day and night is approximately equal, and it signifies the start of warmer and more favorable weather conditions in the northern hemisphere, and colder weather in the southern hemisphere.

Equinoxes and solstices mark different points in Earth's orbit around the Sun and are key to understanding the changing seasons. Equinoxes occur twice a year, in March and September. During an equinox, the Earth's axis is not tilted toward or away from the Sun, resulting in nearly equal day and night lengths all over the world. The March equinox marks the start of spring in the Northern Hemisphere and autumn in the Southern Hemisphere. The September equinox signals the beginning of autumn in the Northern Hemisphere and spring in the Southern Hemisphere. Solstices also happen twice a year, in June and December. They occur when the Earth's axis is tilted maximally toward or away from the Sun. The

June solstice, when the North Pole tilts toward the Sun, results in the longest day of the year in the Northern Hemisphere (summer solstice) and the shortest day in the Southern Hemisphere (winter solstice). Conversely, the December solstice, when the South Pole tilts toward the Sun, brings about the longest day in the Southern Hemisphere (summer solstice) and the shortest day in the Northern Hemisphere (winter solstice). In short, equinoxes are about balance between day and night, while solstices are about the extremes of daylight and darkness.

To transition from the Gregorian to the fixed international at the end of a year would require dropping ten days (from January (up to 10 January) to start timing with the fixed calendar correctly. The first fixed calendar year would be started on January eleventh, or the last Gregorian calendar year is ended on December 21st. Also, schedule the adoption of the new calendar to be a year (in the four year leap day cycle) that the winter solstice falls on a Saturday. Thus, there be no need to skip any days of the week.

Table 35. Measurement > Time > Calendar: *The international fixed calendar in relation to the Gregorian calendar dates. Note that the Cotsworth calendar has "Sol" between June and July. Here, the international fixed calendar month prefixes do not align with the month count.*

Month	International fixed calendar	Gregorian calendar STARTS	Gregorian calendar ENDS
1	January	January 1	January 28
2	February	January 29	February 25
3	March	February 26	March 25*
4	April	March 26	April 22*
5	May	April 23	May 20*
6	June	May 21	June 17* (leap day)
7	Sol	June 18	July 15
8	July	July 16	August 12
9	August	August 13	September 9
10	September	September 10	October 7
11	October	October 8	November 4
12	November	November 5	December 2
13	December	December 3	December 30
Non-day	Year Day	December 31	
Leap-day	* One day earlier on leap years; these dates are a day earlier in a leap year		

Table 36. Measurement > Time > Calendar: *An alternative to the month naming in the International Fixed Calendar system. Here, the Roman additions to the calendar have been removed and replace with the months the Roman's took out. The Romans replaced Quintilis (replaced by July) and Sextilis (replaced by August). The 5th, 6th, 7th, 8th, 9th, 10th, and 13th months are all named correctly per their prefix. This is a corrected version of the calendar months with proper prefixes and the repositioning of March at the start of the year, and January following December. Originally in the Roman Julian Calendar March used to be the first month of the year before January and February were added later. Here, January and February will be added to the end of the year, just prior to the last correctly prefixed month of Triember. Of note, January is named after the ancient Roman god Janus of both beginnings and endings, so it could alternatively be placed as the final month of the year (not shown below).*

Month	Rearranged International fixed calendar	Prior to the Gregorian Calendar
1	March	January
2	April	February
3	May	March
4	June	April
5	Quintilis (5th month)	May
6	Sextilis (6th month)	June
7	September (7th month)	Quintilis (replaced by July)
8	October (8th month)	Sextilis (replaced by August)
9	November (9th month)	September (7th month)
10	December (10th month)	October (8th month)
11	January	November (9th month)
12	February	December (10th month)
13	Triember (13th month)	*No 13th month*

1.3.1.7 The Gregorian calendar

A.k.a., The international standard calendar, the inefficient calendar, the early 21st century calendar.

In the Gregorian calendar, which is the most widely used civil calendar system in the early 21st century, a standard year consists of 365 days. Months do not divide equally into days in the Gregorian calendar. In a leap year, an extra day is added, making it 366 days. The Gregorian calendar, the current standard calendar in most of the world, adds a 29th day to February in all years evenly divisible by 4, except for centennial years (those ending in -00), which receive the extra day only if they are evenly divisible by 400. Thus 2000 was a leap year, but 1700, 1800, and 1900 were not.

Table 37. Measurement > Time > Calendar: *The early 21st century Gregorian calendar months.*

Month Number	Month Name	Prefix Meaning	# of days in month
1	January	N/A	31
2	February	N/A	28 / 29
3	March	N/A	31
4	April	N/A	30
5	May	N/A	31
6	June	N/A	30
7	July	N/A	31
8	August	N/A	31

Month Number	Month Name	Prefix Meaning	# of days in month
9	September	7	30
10	October	8	31
11	November	9	30
12	December	10	31

NOTE: *The non-prefixed months in the above calendar are either named after gods, or after Roman Caesars (Julius Caesar for July and Augustus Caesar for August).*

The Gregorian calendar is the most widely used calendar in the world today. It is the calendar used in the international standard for Representation of dates and times: ISO 8601:2004. It is a solar calendar based on a 365-day common year divided into 12 months of irregular lengths.

Table 38. Measurement > Time > Calendar: *The Gregorian calendar quarters are.*

Quarter	End Date	Days in Quarter
Q1	Friday, March 31	90
Q2	Friday, June 30	91
Q3	Sunday, September 30	93
Q4	Sunday, December 31	92

This early 21st century calendar is irrational in the following ways:

1. To evenly split the year into months with equal numbers of days (28 days + 1 non-day), then the calendar should have 13 months, not 12 months.
   A. Months have different numbers of days.
2. Four of the months are irrationally placed considering their numerical indicating prefixes:
   A. "Sept" (in September) is the prefix for 7, and yet, September is the 9th month of the year.
   B. "Oct" (in October) is the prefix for 8, and yet October is the 10th month of the year.
   C. "Nov (in November) is the prefix for 9, and yet, November is the 11th month of the year.
   D. "Dec" (in December) is the prefix for 10, and yet December is the 12th month of the year.

INSIGHT: *Using the Gregorian calendar, people say and believe, that there are 4 weeks in a month, when in fact, February is the only month with exactly 4 weeks*

The Gregorian calendar introduces unnecessary additional cognition and the potential for planning difficulties and mistakes. A useful calendar should follow a simple and straightforward pattern and not add additional complexity and resource requirements beyond baseline.

Using the Gregorian calendar, it doesn't make any sense to say, "Schedule the next meeting for the same time next month." Using the Gregorian calendar, it is not clear what the statement, "the same time next month" means. Calendar planning becomes unnecessarily complex plan when, from year to year, the days of the week for a given date change. Each 'month' is a parameter of production, wherein, there will be variances in production. Differences in production per month mean that optimal production calculations ought include months which are themselves uniform distributed, as in, into 13 'month' time-units of 28 'days' each.

1.4 Other calendar systems

A year is 12 main months, each beginning with the same weekday and composed of 5 weeks. In between every 3 moths is a 1-2 day holiday (for a total of 5-6 holiday days per year), which land near one of the two seasonal equinoxes and two seasonal solstices of each year, and which are not assigned to a weekday or month. The logic for this calendar system is:

- 360 days + 5 days + leap
  - 360 days = (12 months) • (30 days
    - 12 months per year (each with 30 day)
    - 5 weeks per month (each with 6 days)
  - 365 days = ((12) • (30)) + 5 non-days
    - 366 days = (((12) • (30)) + 5 non-days) + leap year non-day/4 years)

Hence, this calendar system has:

1. For a total of 360 days:
   A. 12 months per year.
   B. 5 Weeks per month.
   C. 6 days per week.
2. For a total of 365 to 366 days (as days in an actual solar year):
   A. Plus, 5 non-days to 365 days, plus 1 day on leap years to 366 days; neither of which are part of a month or a day of the week. These days could be put at the end of the year cycle, or scattered throughout the year (and/or aligned with solstices or equinoxes).

1.5 The day and week cycle

The Egyptians passed their idea of a 7-day week onto the Romans, who also started their week with the Sun's day, dies solis.

It is possible to divide months of any number of days into various balanced week periods, for instance:

1. A 28 day month cycle of 13 months could be divided equally into:
   A. 4 Weeks, 7 Days each.
   B. 2 Weeks, 14 Days each.

2. A 30 day month cycle of 12 months could be divided equally into, for instance:
   A. 4 Weeks, 10 Days each.
   B. 5 Weeks, 6 Days each.
   C. 6 Weeks, 5 Days each.
   D. 2 Weeks, 15 Days each.

Table 39. Measurement > Time > Calendar: *The early 21st century calendar days of the standard 7-day week. The names of the standard, conventional days of the week originate from the names of the classical planets from Greek Astrology which are also the named after Greek/Roman gods.*

Day Number	Day Name	Planet Association	God Association
1	Sunday	Sun	Helios/Sol
2	Monday	Moon	Luna/Selene
3	Tuesday	Mars	Ares/Mars
4	Wednesday	Mercury	Hermes/ Mercury
5	Thursday	Jupiter	Zeus/Jupiter
6	Friday	Venus	Aphrodite/ Venus
7	Saturday	Saturn	Kronos/ Saturn

Different calendars that use this standard have the first day of the week starting on different days, usually the first day of a week starts on either Sunday or Monday.

2 The daily-time cycle unit

A.k.a., The daily clock, the daily solar clock, the sun clock, the daily sun clock, the earth-axis solar-oriented clock.

In early 21st century society there two primary unit daily [time] clocks:

1. **The 24-hour clock (a.k.a., military time)**: This clock is known by many names including: Under the 24-hour clock system, the day begins at midnight, 00:00, and the last minute of the day begins at 23:59 and ends at 24:00, which is identical to 00:00 of the following day. 12:00 can only be mid-day. This is the clock unit system universally used the current planetary transportation and logistics system (which has several names, including: Zulu time; Greenwich Mean Time (GMT); and Universal Standard Time).
   A. 00:00 and 24:00 are midnight.
   B. 12:00 is midday.
   C. 23:59:59 is 1 millisecond before midnight.
   D. 00:00:01 is 1 millisecond after midnight.
2. **The midday centric clock (a.k.a., mid-day clock, AM/PM clock)**: This clock is divided into two 12 hour segments. The first 12 hours of the day are signified by “a.m.”, which is the acronym for, “anti meridian”, which is Latin for, “before midday. The second 12 hours of the day are signified by “p.m.”, which is the acronym for “post meridian”, which is Latin for “after midday”. The two segments are as follows:
   A. 12:00am –11:59am is before midday (AM, ante meridiem).
   B. 12:00pm-11:59pm is after midday (PM, post meridiem).

Given what is known, the 24-hour clock is more logical and more intuitive than a clock with two specific segments (i.e., the mid-day centric clock), which adds an additional unit of measure. The 24-hour clock only has the unit ‘Time’. In order to have a complete comprehension of a value given by the 12-hour am/pm clock, two units must be given: the time unit, and the am/pm (after/before midday). The mid-day clock should be done away with entirely in favor of the 24-hour clock in order to increase efficiency and reduce unintentional time clashing.

3 Dating format

As a part of this process there is a more efficient and logical world-wide dating standard: year month day, such as:

- 2024 June 04
- 2024 06 04

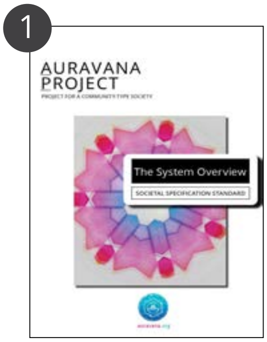
Events, records, and files ought to be organized with the year first, then month, then day. There is already an international standard ISO standard (8601) for this and some countries are already using this standard. This format already works well for organizing computer directories and files.

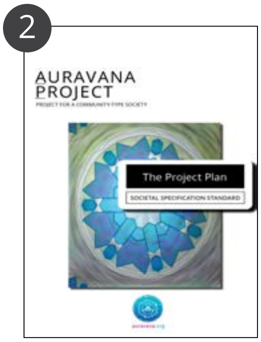
The Auravana Project exists to co-create the emergence of a community-type society through the openly shared development and operation of a information standard, from which is expressed a network of integrated city systems, within which purposefully driven individuals are fulfilled in their development toward a higher potential life experience for themselves and all others. Significant project deliverables include: a societal specification standard and a highly automated, tradeless habitat service operation, which together orient humanity toward fulfillment, wellbeing, and sustainability. The Auravana Project societal standard provides the full specification and explanation for a community-type of society.

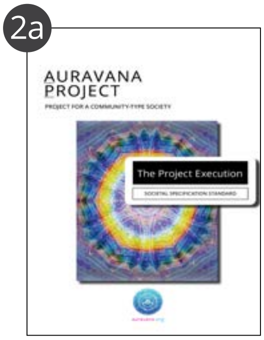
This publication is the Project Plan for a community-type society. A societal-level project plan describes the organized thinking and execution of a socio-technical environment; the operation of community. This project plan identifies humanity's project to create a global community-type society for the fulfillment of that which everyone has mutually in common. This is a planned project for a configuration of society that may be tested in its results at optimally meeting all human life requirements at the global scale. This is a planning and work proposal for an open-source, societal-level project. This document describes and explains a unified approach to actions and results that is likely, given what is known and accessible, to improve all of humanity. This is the plan for societal navigation that specifies an approach, direction, and execution to socio-technical life. The project plan has three core sections: (1) Approach to project execution, (2) Direction of project execution, and (3) Execution of project execution. The standard details the complete, plannable information set for the society's operation, including its approach to action, its direction of action, and its execution and adaptation of action. Herein, these concepts, their relationships and understandings, are defined and modeled. Discursive reasoning is provided for this specific configuration of a project plan, as opposed to the selection and encoding of other configurations. A project plan provides for the formalized project-based development operation of a society, organized in time and with available resources, coordinated to become a societal service system for human fulfillment and ecological well-being.

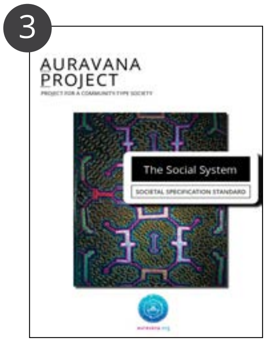
Fundamentally, this standard facilitates individual humans in becoming more aware of who they really are.

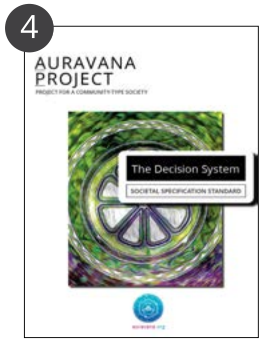
All volumes in the societal standard:

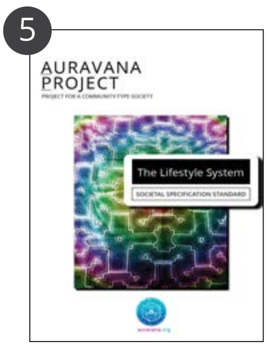


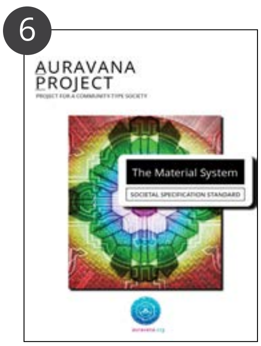


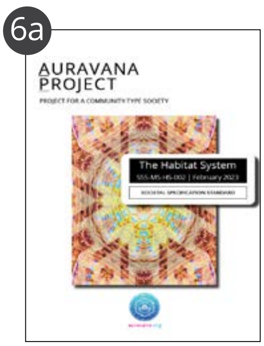












aurava.org

$15.00
ISBN 979-8-9861436-1-3
51500>