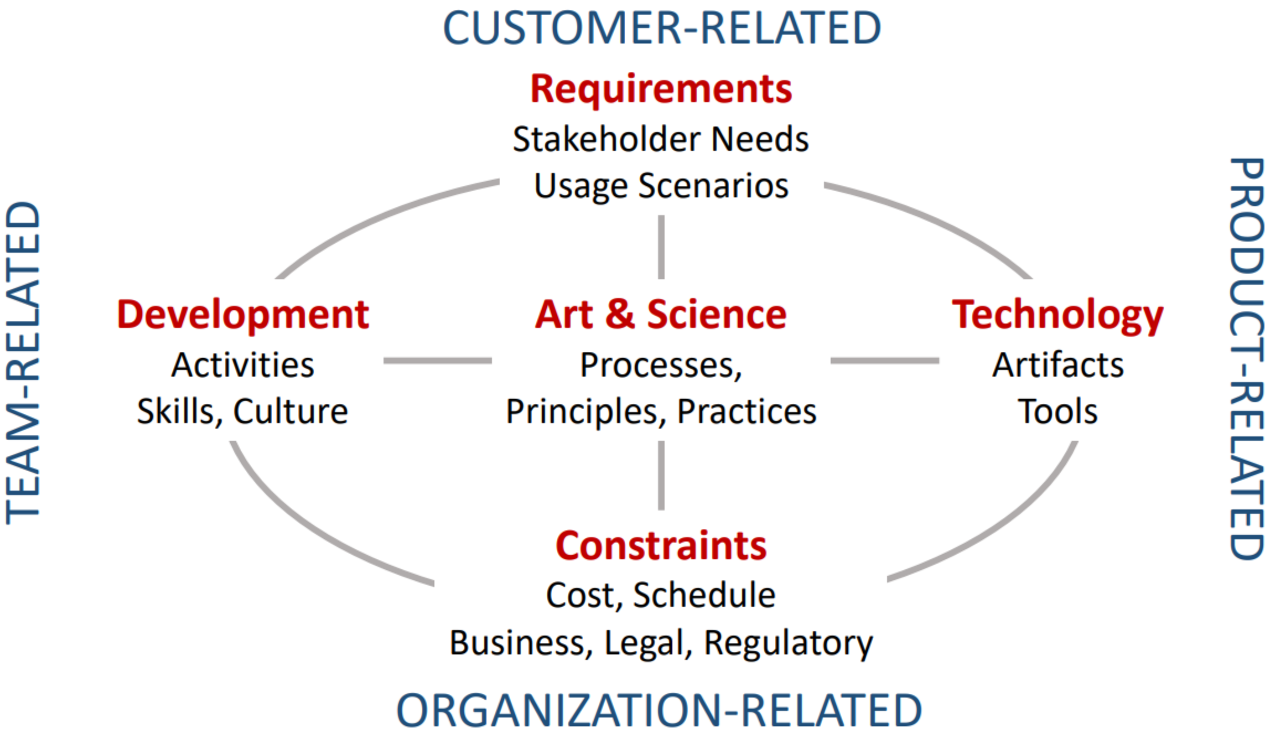
Programų kūrimo procesas

Dr. Asta Slotkienė

Software context

- The art and science of developing reliable software systems that address customer needs, subject to cost and schedule constraints



CUSTOMER-RELATED

Requirements
Stakeholder Needs
Usage Scenarios

TEAM-RELATED

Development
Activities
Skills, Culture

Art & Science
Processes,
Principles, Practices

Technology
Artifacts
Tools

PRODUCT-RELATED

Constraints
Cost, Schedule
Business, Legal, Regulatory

ORGANIZATION-RELATED

General Terminology

- **System**:
  - Combination of interacting elements organised to achieve one or more stated purposes.

    (ISO/IEC 15288:2015)
- **Software**
  - All or part of the programs, procedures, rules, and associated documentation of an information processing system

General Terminology

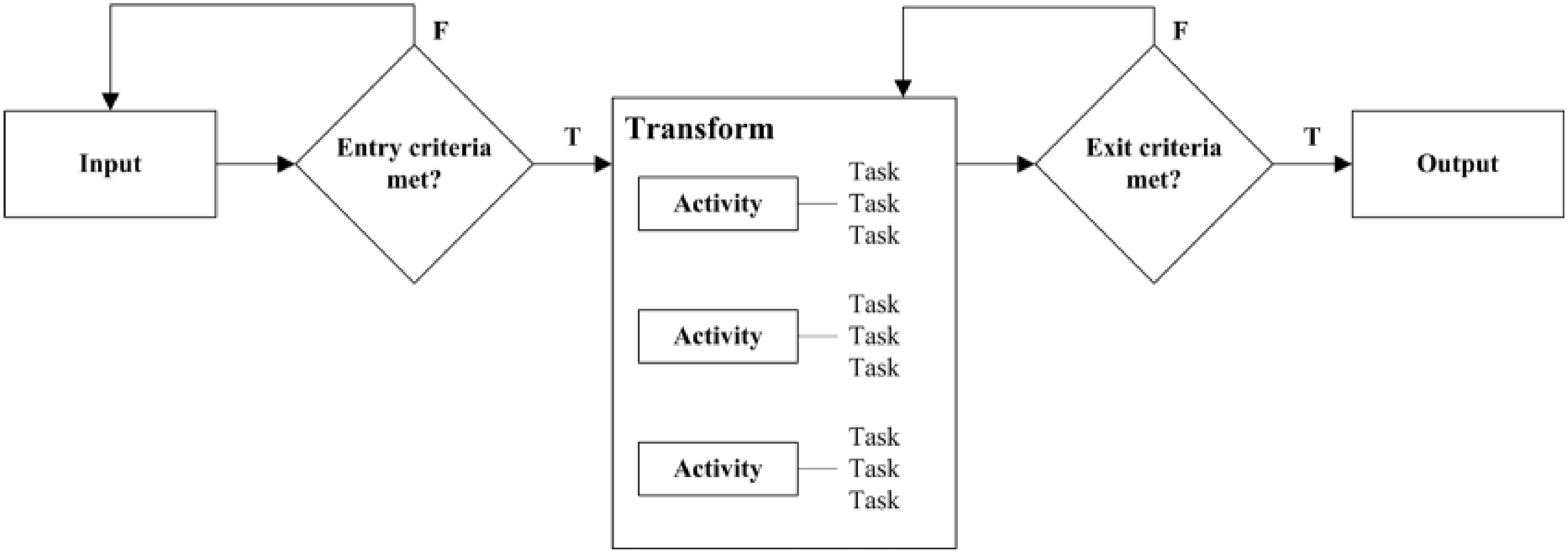
- The IEEE SE definition:
  - *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*

General Terminology

- **Process**
  - **Set of interrelated or interacting activities which use inputs to deliver an intended result**.

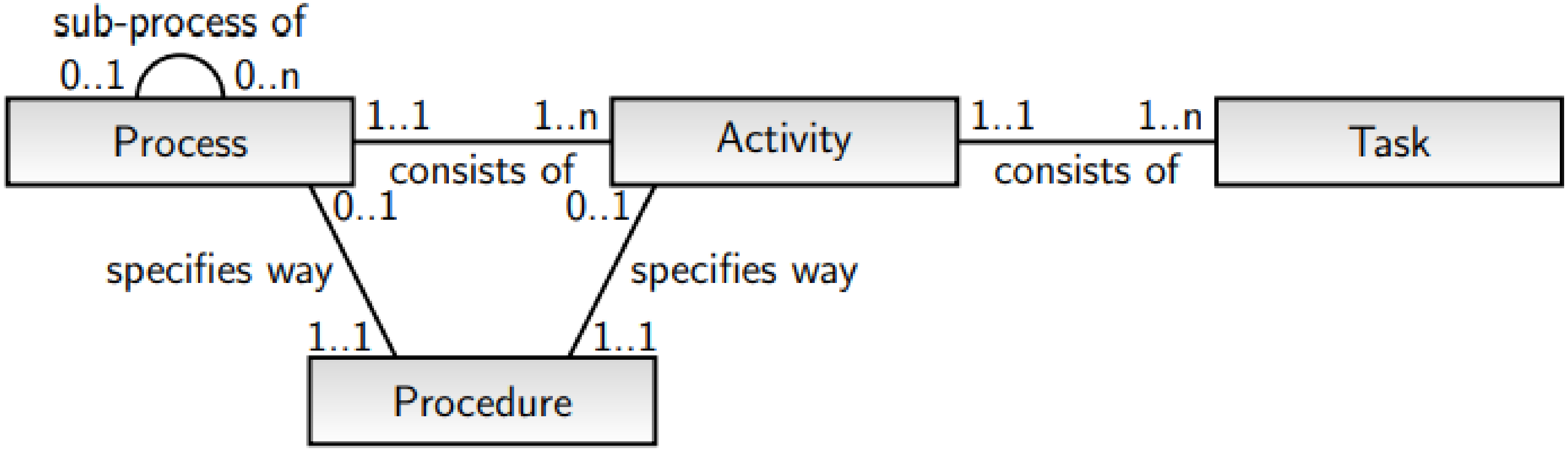
  (ISO 9000:2015)
  - Processes are therefore often described as consisting of **input, processing and output**
  - A process may be performed, enacted or executed

Elements of a Software Process



SWEBOK v3.0

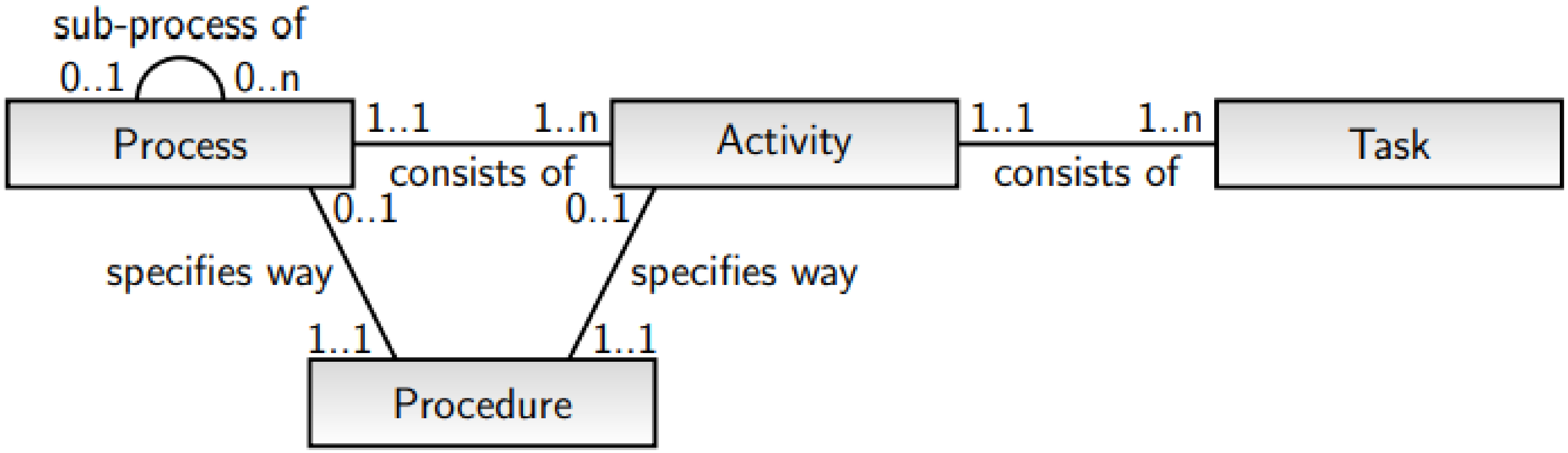
Relationship between processes and related terms



Process Terminology

- **Activity**
  - Set of cohesive tasks of a process.
- **Task**
  - Required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process

(ISO/IEC 12207:2008)

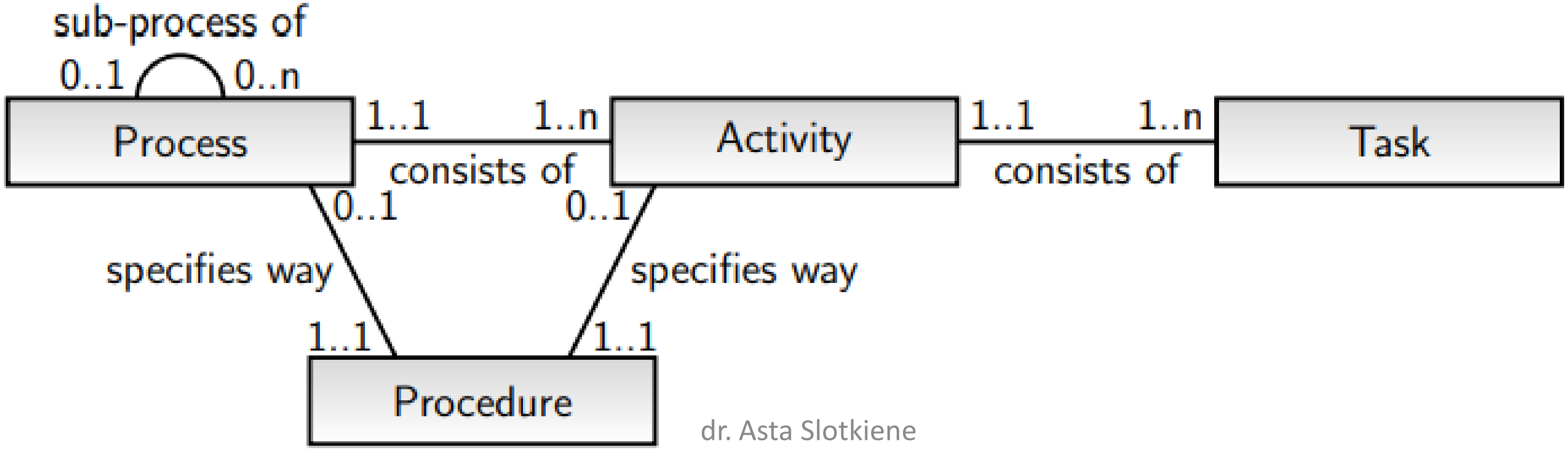


Process Terminology

- **Sub-process**
  - A process that is part of a larger process.

  (CMMI R v1.3)
- **Procedure**
  - Specified way to carry out an activity or a process.

  (ISO 9000:2015)



Example of Software Process

- **Software requirements validation** is a process used to determine whether the requirements will provide an adequate basis for software development
- Tt is a sub-process of the **software requirements process.**
- The tasks of the requirements validation activity might include
  - **requirements reviews,**
  - **Prototyping**
  - **Model validation**

Example of Software Process

- **Inputs** for requirements validation are typically a **software requirements specification** and the resources needed to perform validation (personnel, validation tools, sufficient time).

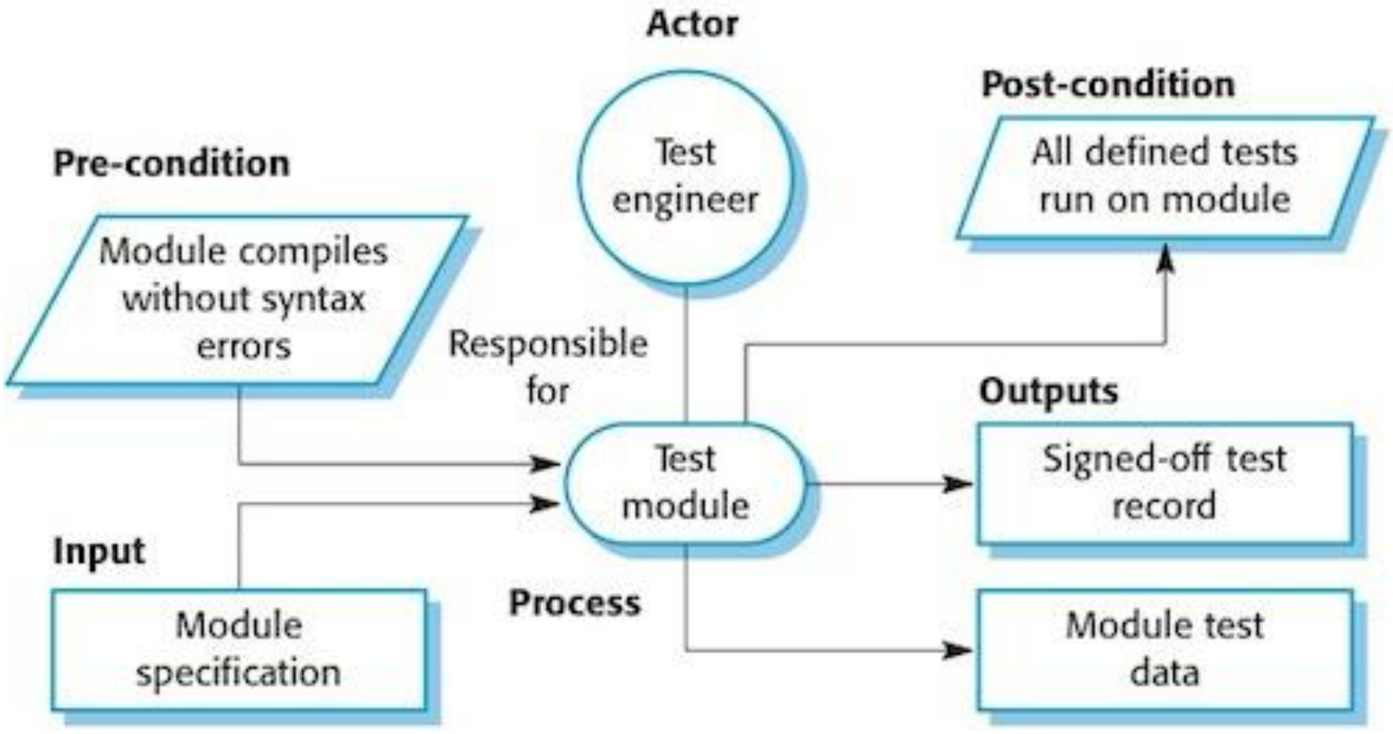
- **The output** of requirements validation is typically **a validated software requirements specification** that provides inputs to the software design and software testing processes.

Example of Software Process

- A software process also include:
  - roles
  - competencies,
  - IT support,
  - software engineering techniques
  - software engineering tools,
  - work environment needed to perform the process,
  - approaches and measures used to determine the efficiency and effectiveness of performing the process.

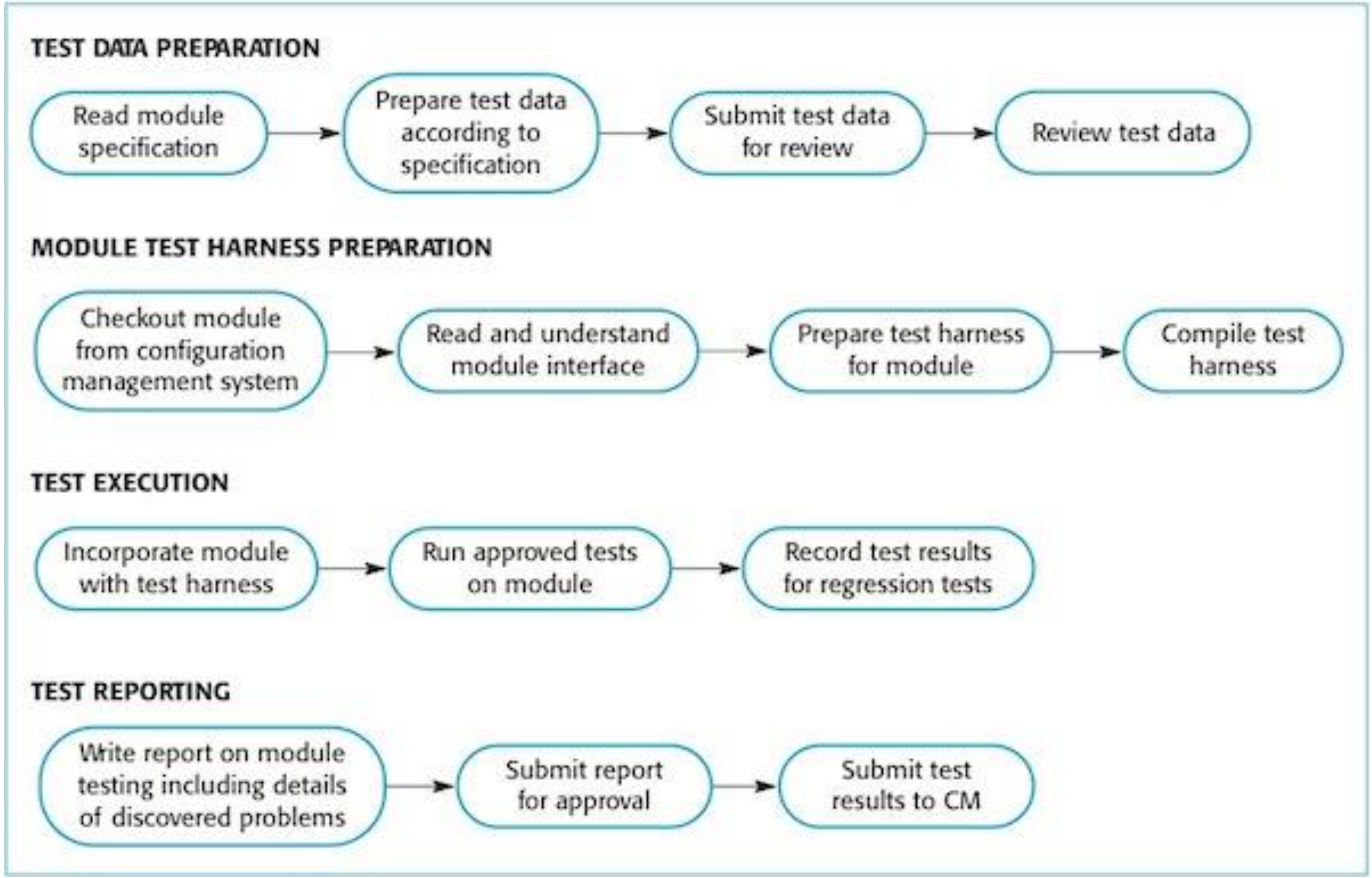
In addition, a software process may include interleaved technical, collaborative, and administrative activities

Example of Software Process



Ian Sommerville 2008

Example of Software Process



TEST DATA PREPARATION

Read module specification → Prepare test data according to specification → Submit test data for review → Review test data

MODULE TEST HARNESS PREPARATION

Checkout module from configuration management system → Read and understand module interface → Prepare test harness for module → Compile test harness

TEST EXECUTION

Incorporate module with test harness → Run approved tests on module → Record test results for regression tests

TEST REPORTING

Write report on module testing including details of discovered problems → Submit report for approval → Submit test results to CM

Ian Sommerville 2008

Software Process Terminology

- Life cycle
  - Evolution of a system, product, service, project or other human-made entity **from conception through retirement.** (ISO/IEC 12207:2008)
- Life cycle model
  - **Framework of processes** and activities concerned with the life cycle, which can be organised into stages, and which acts as a common reference for communication and understanding.

(ISO/IEC 12207:2008)

Software Process Terminology

- Software process
  - A process that forms part of the life cycle of software.

  Life cycle model
  - Framework of processes and activities concerned with the life cycle, which can be organised into stages, and which acts as a common reference for communication and understanding.

  (ISO/IEC 12207:2008)
  - requirements analysis,
  - design,
  - implementation,
  - maintenance and operations,
  - quality assurance
  - project planning

Software Process Terminology

Software development process

1. Process by which user needs are translated into a software product.
2. A software process that is performed (fully or at least partly) during the development part of the software life cycle.

Software Process Terminology

Software development life cycle (SDLC)

- Life cycle that includes the software development processes used to **specify software requirements and transform them into a deliverable software product.**

(Based on SWEBOK)

Software Process Terminology

- Software (product) life cycle (SPLC))
  - Life cycle that includes the software development life cycle
    - plus **additional software processes that provide for deployment, maintenance, support, evolution, retirement, and all other inception-to-retirement processes** for a software product,
    - including the **software configuration management and software quality assurance processes** that are applied through out a software product life cycle.

(Based on SWEBOK)

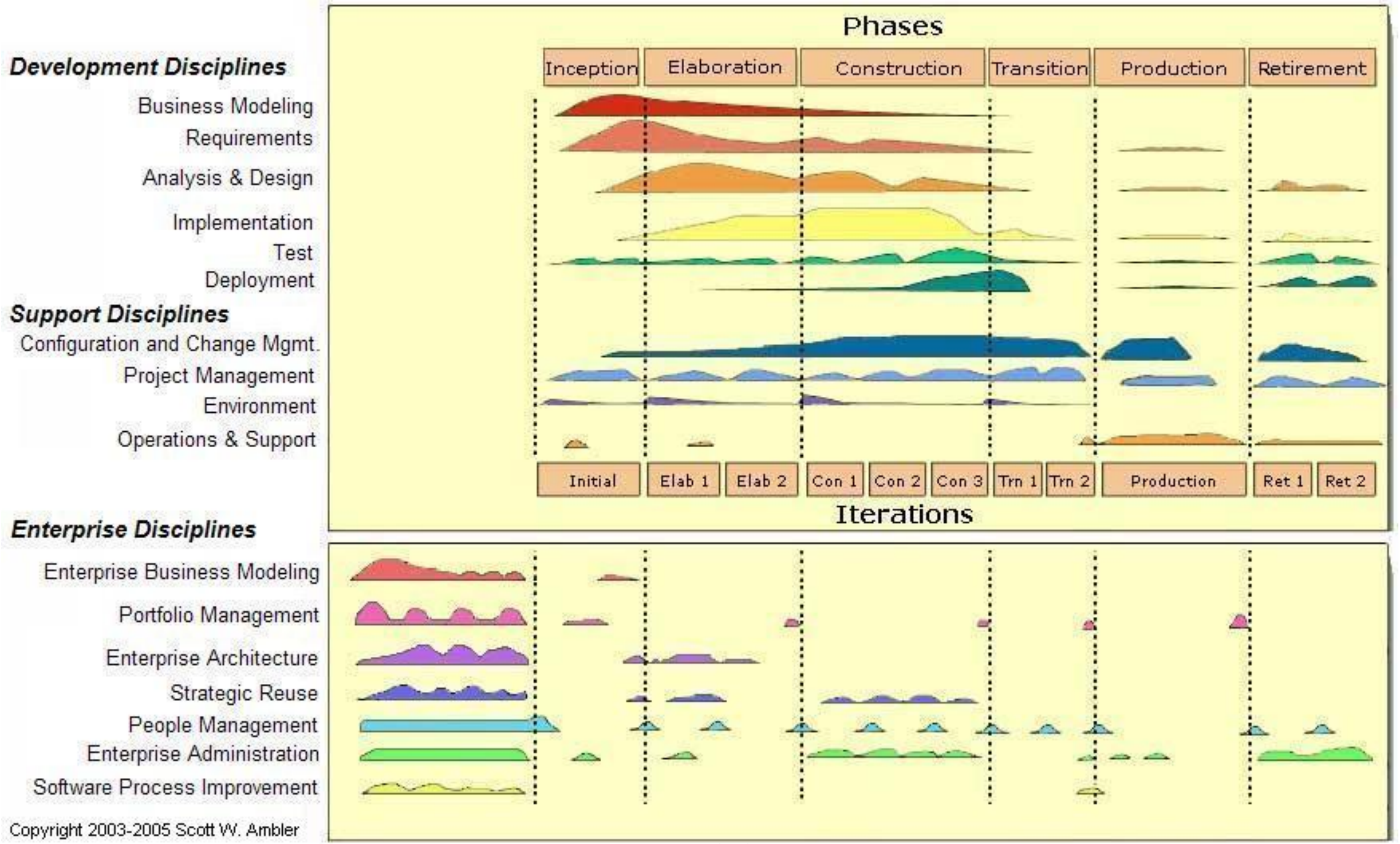
Software processes and software development

- Software development is a complex task, consisting of many individual activities that **need to be coordinated**.
- This is where **software processes and life cycle models** come into play, providing an infrastructure to coordinate and manage these various activities.

SWEBOK 3.0

1. Software requirements
2. Software design
3. Software construction
4. Software testing
5. Software configuration management
6. Software quality

The Rational Unified Process



The Rational Unified Process

- Business modelling
  - The business processes are modelled using business use cases.
- Requirements
  - Actors who interact with the system are identified and use cases are developed to model the system requirements.
- Analysis and design
  - A design model is created and documented using architectural models; component models; object models and sequence models.
- Implementation
  - The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.

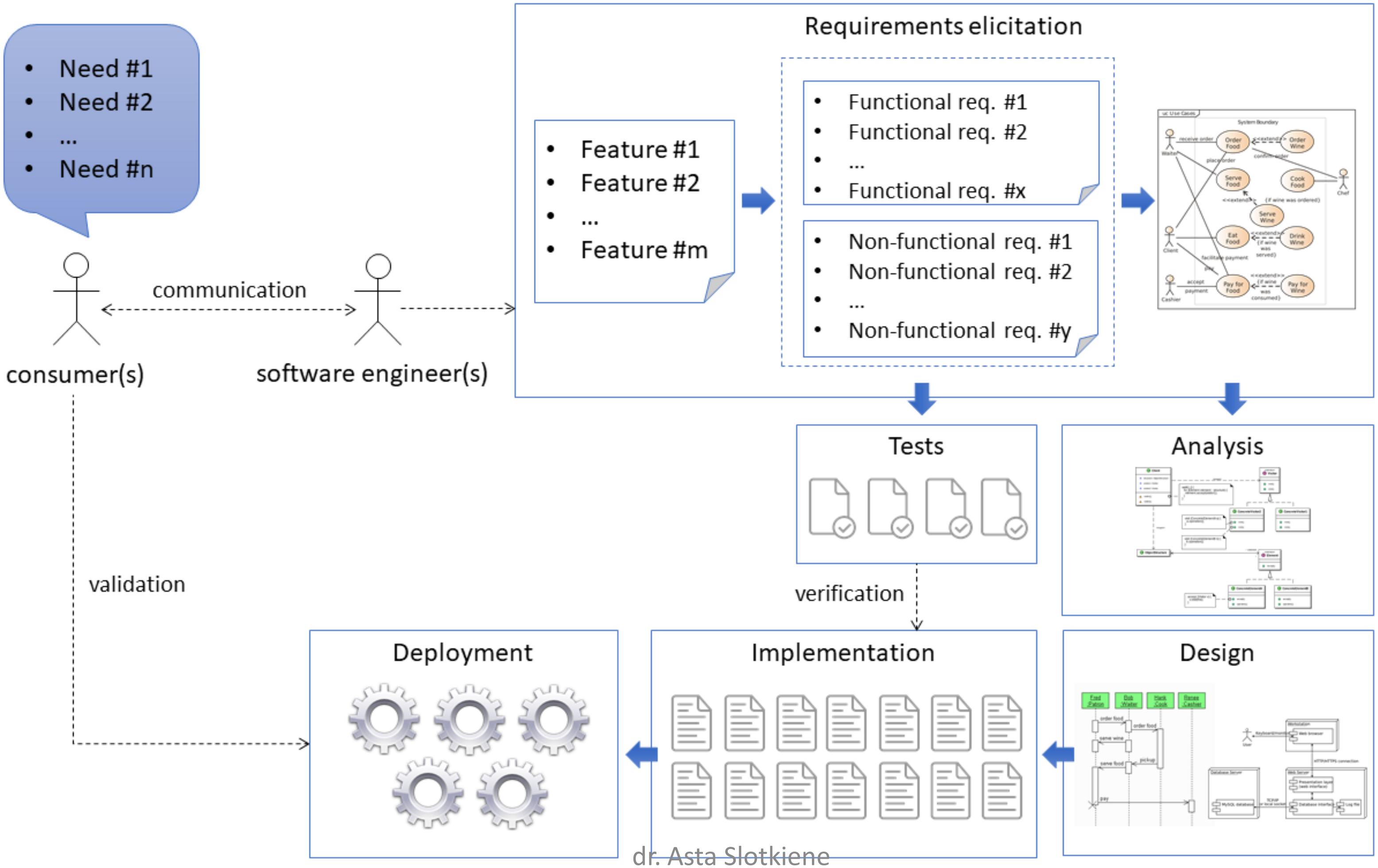
The Rational Unified Process

- Testing
  - Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
- Deployment
  - A product release is created; distributed to users and installed in their workplace.
- Configuration and change management
  - This supporting workflow managed changes to the system (see Chapter 25).
- Project management
  - This supporting workflow manages the system development
- Environment
  - This workflow is concerned with making appropriate software tools available to the software development team.

Software process models
vs
Software life cycle models

- Software life cycle models **define the main steps** (phases or stages) in the software life cycle and **their sequence**
- **Software process models provide more detail, breaking** the main steps down into sub-steps, and adding information about the results generated and the roles involved

Software engineering generic development process



- Need #1
- Need #2
- …
- Need #n

consumer(s)

communication

software engineer(s)

validation

Requirements elicitation

- Feature #1
- Feature #2
- …
- Feature #m

- Functional req. #1
- Functional req. #2
- …
- Functional req. #x

- Non-functional req. #1
- Non-functional req. #2
- …
- Non-functional req. #y

Tests

Analysis

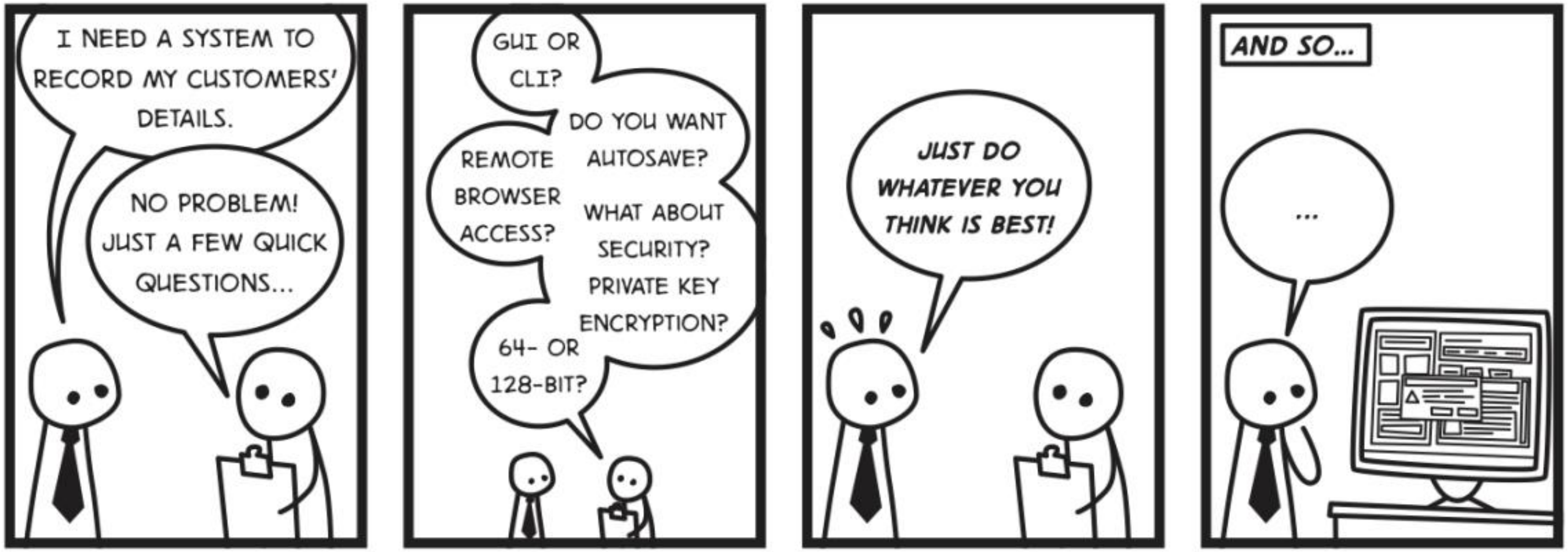
verification

Deployment

Implementation

Design

Why software engineering?


I NEED A SYSTEM TO RECORD MY CUSTOMERS' DETAILS.
NO PROBLEM! JUST A FEW QUICK QUESTIONS...
GUI OR CLI?
DO YOU WANT AUTOSAVE?
REMOTE BROWSER ACCESS?
WHAT ABOUT SECURITY? PRIVATE KEY ENCRYPTION?
64- OR 128-BIT?
JUST DO WHATEVER YOU THINK IS BEST!
AND SO...
...

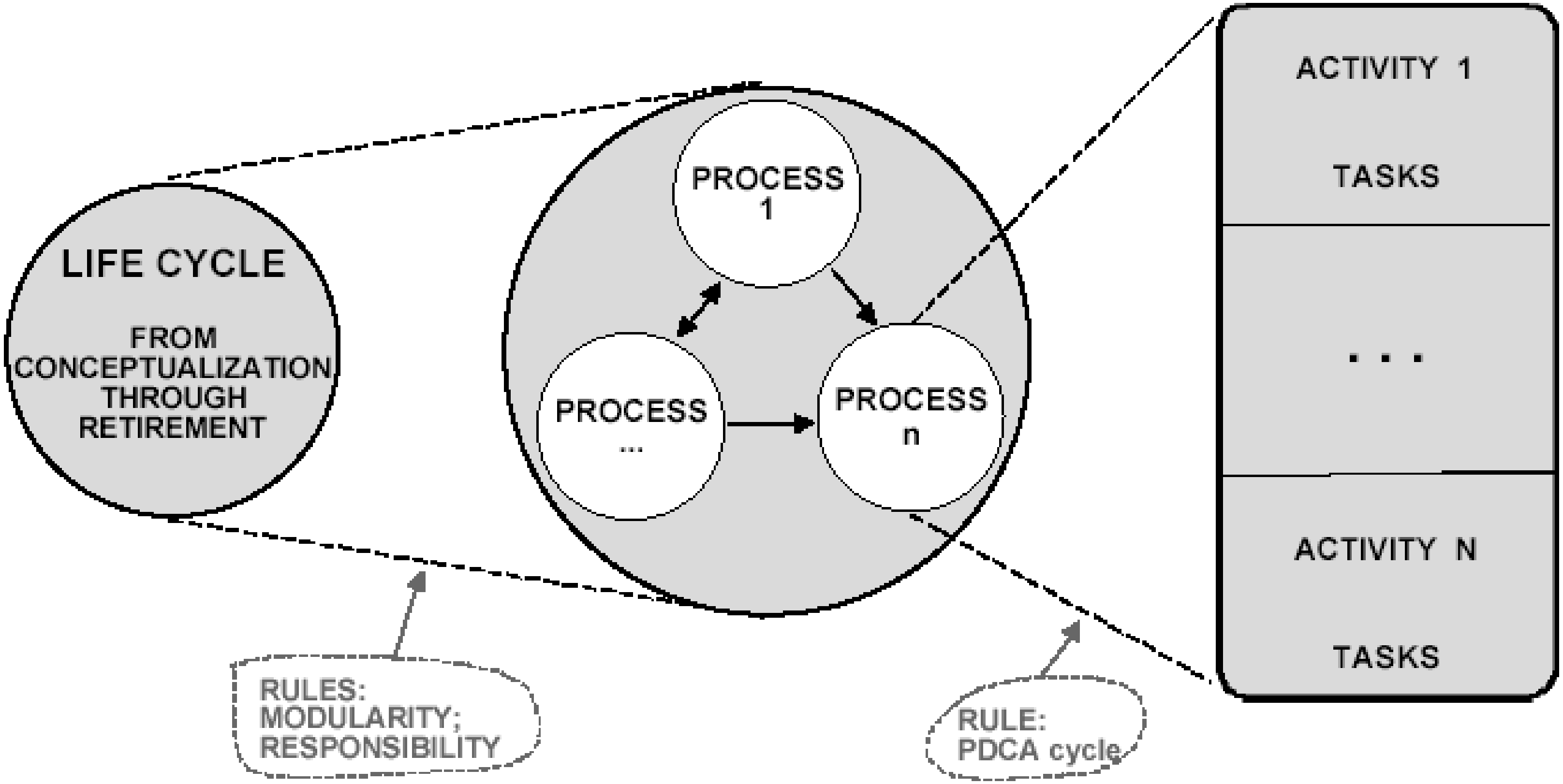
ISO/IEC 12207 – *Systems and software engineering – Software life cycle processes*

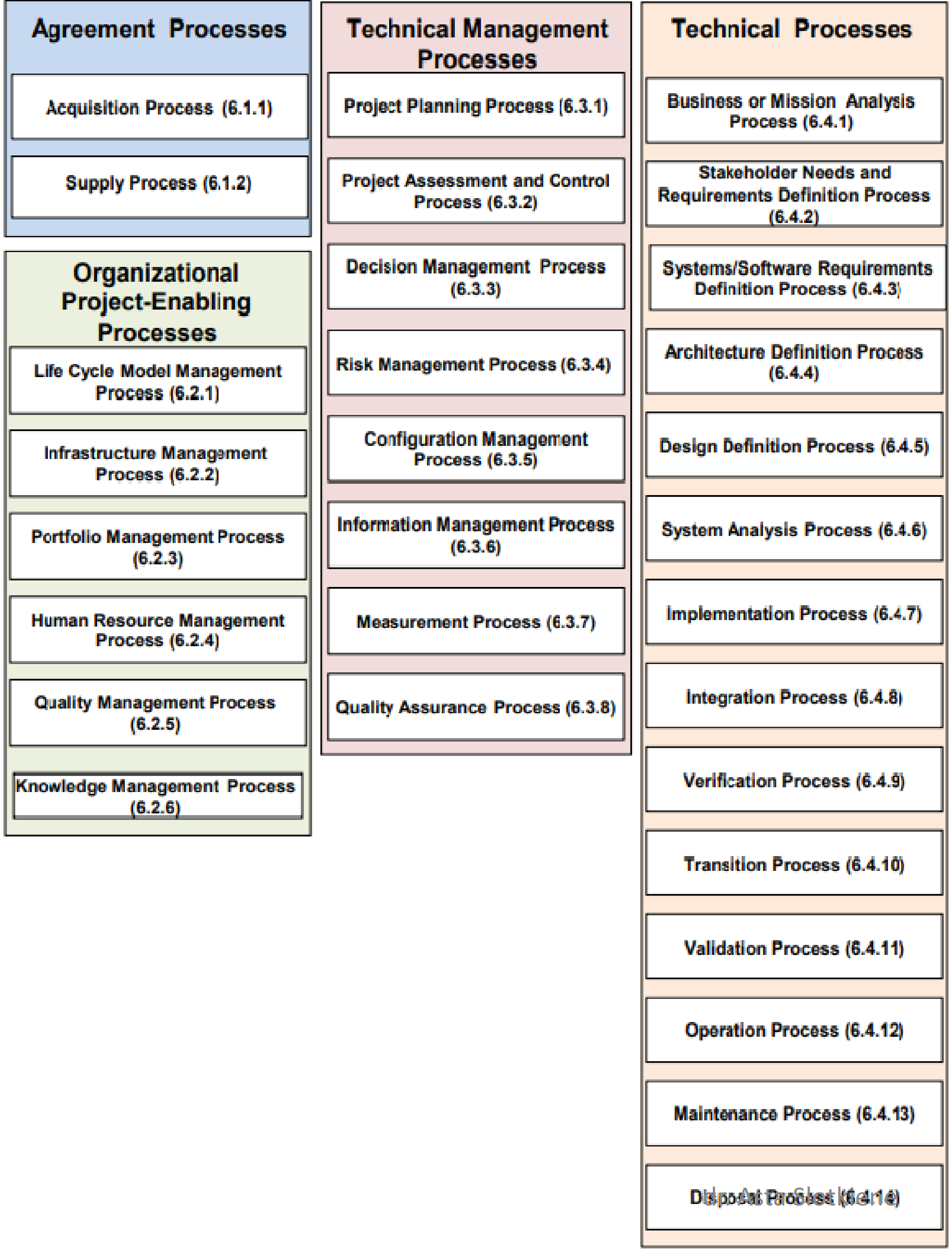
- **ISO/IEC/IEEE 12207** *Systems and software engineering – Software life cycle processes*[is **an international standard for software lifecycle processes.**
- First introduced in 1995, it aims to be a **primary standard that defines all the processes required for developing and maintaining software systems**, including the outcomes and/or activities of each process.

ISO/IEC 12207 – *Systems and software engineering – Software life cycle processes*

- ISO/IEC/IEEE 12207:2017 divides software life cycle processes into four main process groups:
  1. Agreement
     - address **the cooperation and agreement with other organisations**
  2. Oganizational project-enabling processes
     - processes are performed on the level of the development organisation and **provide the environment needed to perform projects.**
  3. Technical management
     - **Project processes**
  4. Technical processes
     - describe the various processes or phases within a software product life cycle, **from stakeholder requirements definition to software disposal.**

ISO/IEC 12207:2005 – Information Technology – Software life cycle processes





Agreement Processes

- Acquisition Process (6.1.1)
- Supply Process (6.1.2)

Organizational Project-Enabling Processes

- Life Cycle Model Management Process (6.2.1)
- Infrastructure Management Process (6.2.2)
- Portfolio Management Process (6.2.3)
- Human Resource Management Process (6.2.4)
- Quality Management Process (6.2.5)
- Knowledge Management Process (6.2.6)

Technical Management Processes

- Project Planning Process (6.3.1)
- Project Assessment and Control Process (6.3.2)
- Decision Management Process (6.3.3)
- Risk Management Process (6.3.4)
- Configuration Management Process (6.3.5)
- Information Management Process (6.3.6)
- Measurement Process (6.3.7)
- Quality Assurance Process (6.3.8)

Technical Processes

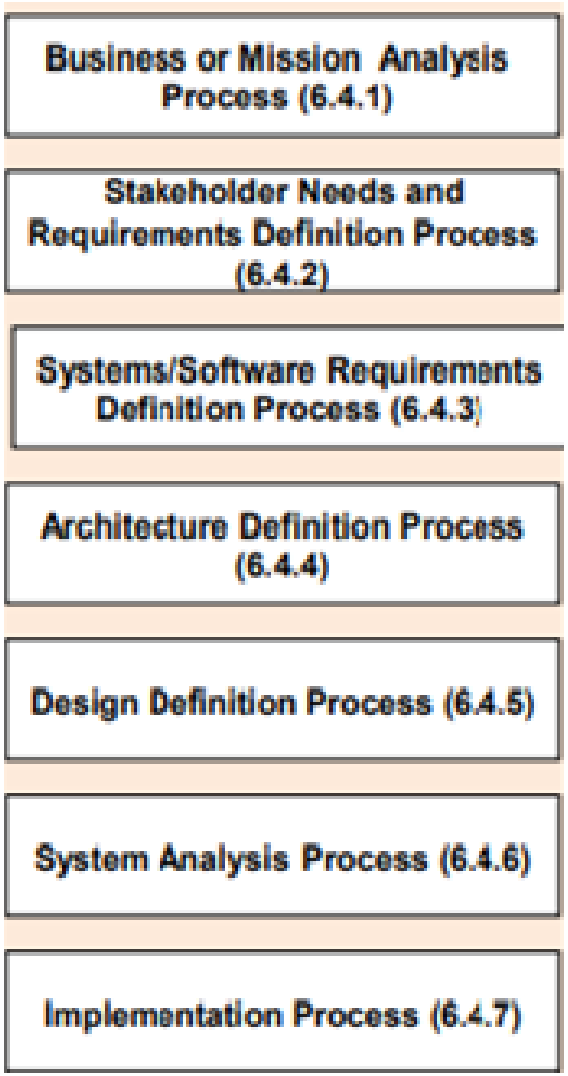
- Business or Mission Analysis Process (6.4.1)
- Stakeholder Needs and Requirements Definition Process (6.4.2)
- Systems/Software Requirements Definition Process (6.4.3)
- Architecture Definition Process (6.4.4)
- Design Definition Process (6.4.5)
- System Analysis Process (6.4.6)
- Implementation Process (6.4.7)
- Integration Process (6.4.8)
- Verification Process (6.4.9)
- Transition Process (6.4.10)
- Validation Process (6.4.11)
- Operation Process (6.4.12)
- Maintenance Process (6.4.13)
- Disposal Process (6.4.14)

ISO/IEC 12207 – *Systems and software engineering – Software life cycle processes*

Technical processes

- to define the **requirements** for a software system,
- to transform the requirements into an **effective product,**
- to permit consistent **reproduction of the product** where necessary,
- to **dispose of the product when it is retired from service**

Technical processes



Business or Mission Analysis Process (6.4.1)

Stakeholder Needs and Requirements Definition Process (6.4.2)

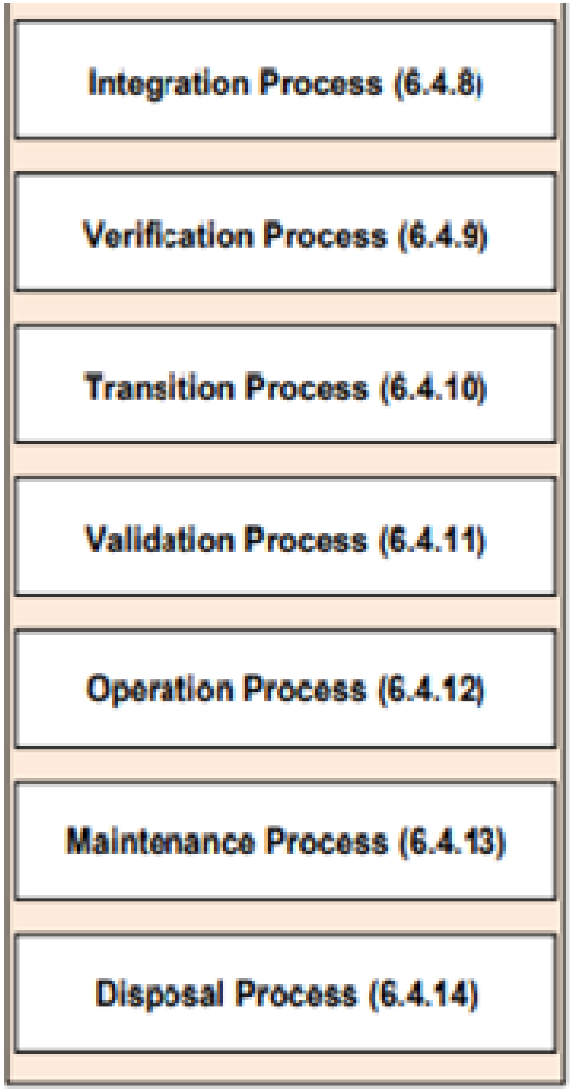
Systems/Software Requirements Definition Process (6.4.3)

Architecture Definition Process (6.4.4)

Design Definition Process (6.4.5)

System Analysis Process (6.4.6)

Implementation Process (6.4.7)



Integration Process (6.4.8)

Verification Process (6.4.9)

Transition Process (6.4.10)

Validation Process (6.4.11)

Operation Process (6.4.12)

Maintenance Process (6.4.13)

Disposal Process (6.4.14)

Business or Mission Analysis process

- The purpose of the Business or Mission Analysis process is to define:
  - the business or mission problem or opportunity,
  - characterize the solution space,
  - determine potential solution class(es) that could address a problem or take advantage of an opportunity.

Stakeholder Needs and Requirements Definition process

- The purpose of the Stakeholder Needs and Requirements Definition process is to define:
  - the stakeholder requirements for a system that can provide the capabilities needed by users and other stakeholders in a defined environment.

System/Software requirements definition process

- The purpose of the System/Software Requirements Definition process is **to transform the stakeholder, user- oriented view of desired capabilities into a technical view of a solution that meets the operational needs of the user.**

Architecture Definition process

- The purpose of the Architecture Definition process is
  - to generate system architecture alternatives,
  - to select one or more alternative(s) that frame stakeholder concerns and meet system requirements,
  - to express this in a set of consistent views.

Design Definition process

- The purpose of the Design Definition process is:
  - to provide sufficient **detailed data and information** about the system and its elements
  - to enable the implementation consistent with **architectural entities** as defined in models and views of the system architecture.

System Analysis process

- The purpose of the System Analysis process is **to provide a rigorous basis of data and information for technical understanding** to aid decision-making across the life cycle.

Implementation process

- The purpose of the Implementation process is to realize a specified system element.

- This process **transforms requirements, architecture, and design, including interfaces, into actions that create a system element** according to the practices of the selected implementation technology, using appropriate technical specialties or disciplines.

Integration process

- The purpose of the Integration process is to **synthesize a set of system elements into a realized system** that satisfies system/software requirements, architecture, and design.

- This process **assembles the implemented system elements.**

Verification process

- The purpose of the Verification process is to provide **objective evidence that a system or system element fulfils its specified requirements and characteristics**.
-
- This process **provides the necessary information to determine resolution of identified anomalies.**

Verification process

- The Verification process identifies the anomalies:
    - errors, defects, or faults in any information item (e.g., system/software requirements or architecture description),
    - implemented system elements,
    - life cycle processes using appropriate methods, techniques, standards or rules.

Transition process

- The purpose of the Transition process is to establish a capability for a system to provide services specified by stakeholder requirements in the operational environment.
  - This process **moves the system in an orderly, planned manner into the operational status, such that the system is functional, operable and compatible with other operational systems.**
  - It installs a verified system, together with relevant enabling systems, e.g., planning system, support system, operator training system, user training system, as defined in agreements.

Validation process

- The purpose of the Validation process is **to provide objective evidence** that the system, when in use, fulfils its business or mission objectives and stakeholder requirements, achieving its intended use in its intended operational environment.
    - The objective of validating a system or system element is to acquire confidence in its ability to achieve its intended mission, or use, under specific operational conditions. Validation is ratified by stakeholders.
    - This process provides the necessary information so that identified anomalies can be resolved by the appropriate technical process where the anomaly was created.

Operation process

- The purpose of the Operation process is to use the system to deliver its services.
  - In order to sustain services, **it identifies and analyzes operational anomalies** in relation to agreements, stakeholder requirements and organizational constraints.

Maintenance process

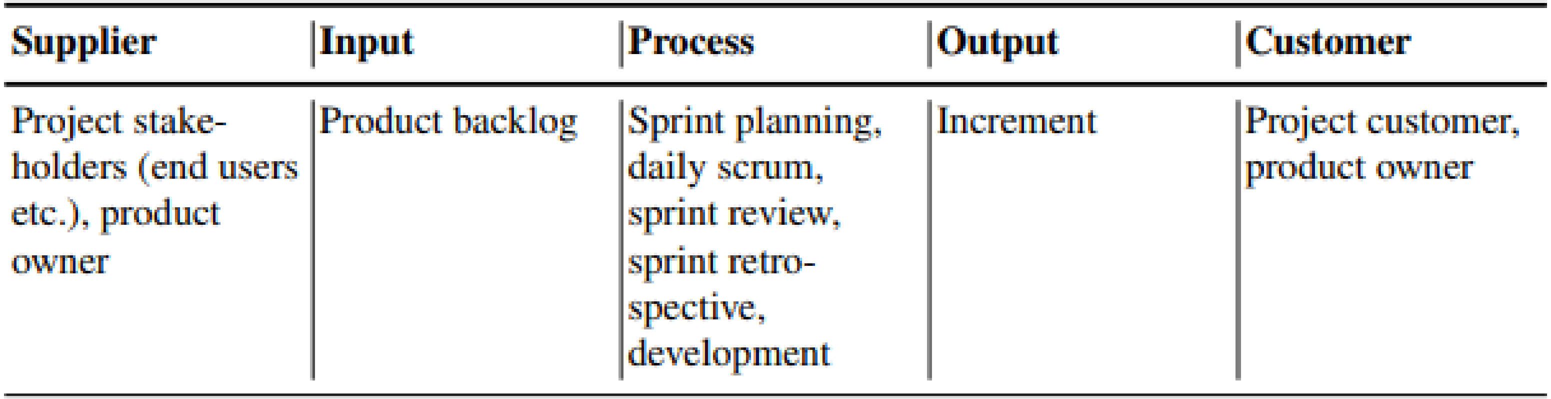
- The purpose of the Maintenance process is to sustain the capability of the system to provide a service.
  - This process monitors the system’s capability to deliver services, records incidents for analysis, takes corrective, adaptive, perfective and preventive actions and confirms restored capability.

Disposal process

- The purpose of the Disposal process is to end the existence of a system element or system for a specified intended use, appropriately handle **replaced or retired elements**, and to properly attend **to identified critical disposal needs** (e.g., per an agreement, per organizational policy, or for environmental, legal, safety, security aspects).

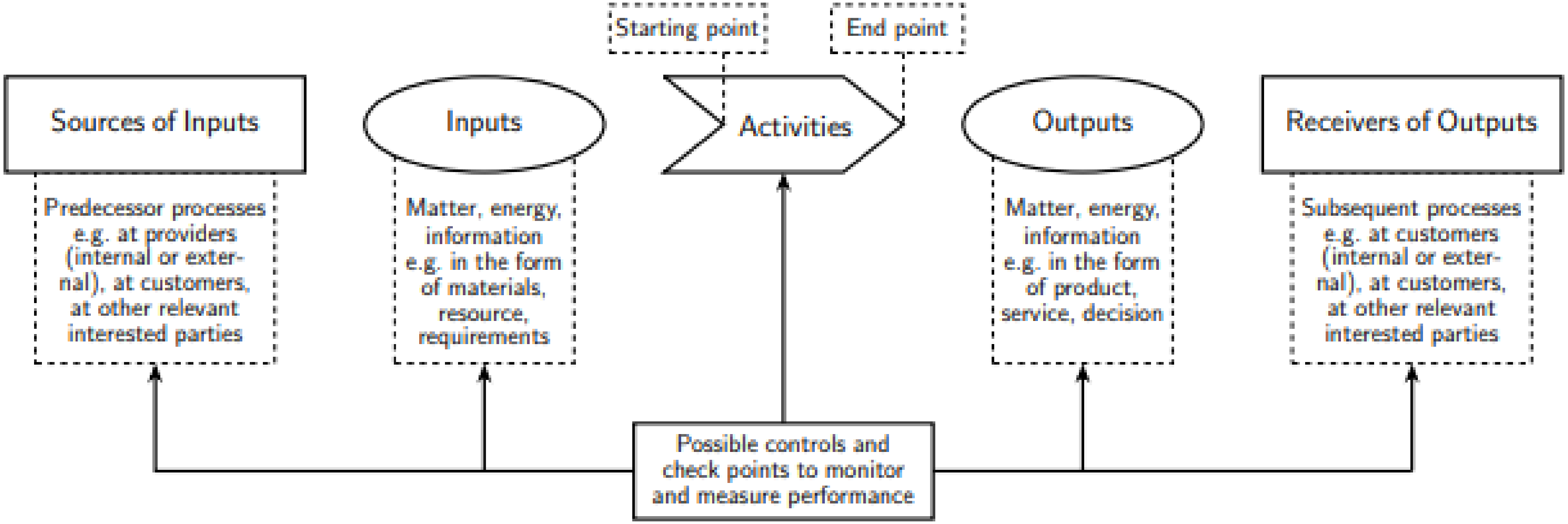
Representation of processes according SIPOC

- SIPOC is a process notation that is commonly used in Six Sigma

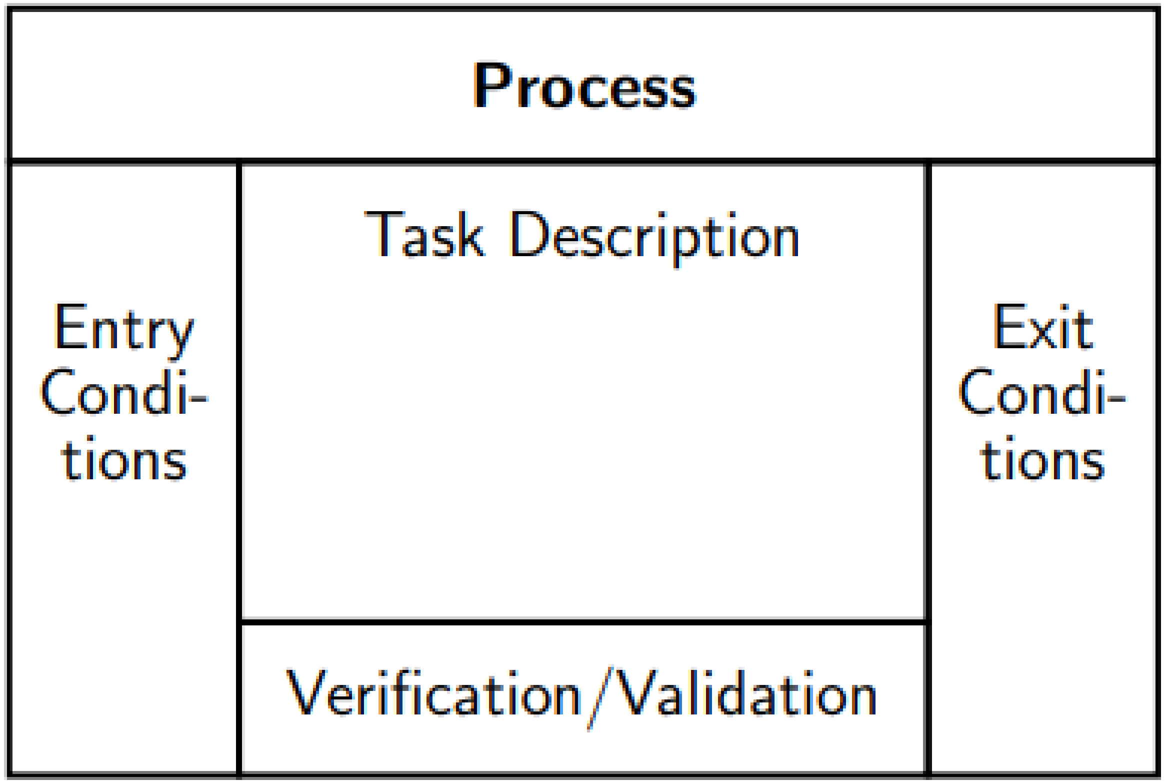


Supplier	**Input**	**Process**	**Output**	**Customer**
Project stakeholders (end users etc.), product owner	Product backlog	Sprint planning, daily scrum, sprint review, sprint retrospective, development	Increment	Project customer, product owner

Representation of processes according to ISO 9001


Starting point
End point
Sources of Inputs
Inputs
Activities
Outputs
Receivers of Outputs
Predecessor processes e.g. at providers (internal or external), at customers, at other relevant interested parties
Matter, energy, information e.g. in the form of materials, resource, requirements
Matter, energy, information e.g. in the form of product, service, decision
Subsequent processes e.g. at customers (internal or external), at customers, at other relevant interested parties
Possible controls and check points to monitor and measure performance

Representation of processes according The Entry-Task-Verification-Exit (ETVX) notation



Process

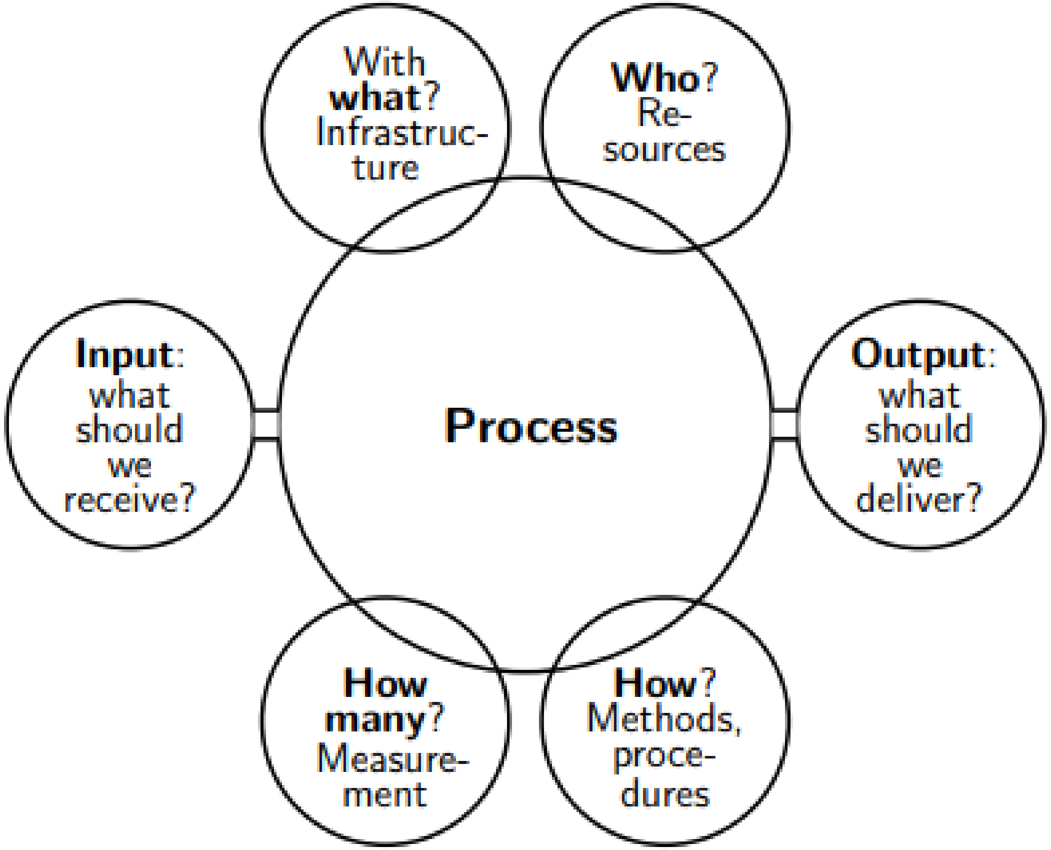
Entry Conditions

Task Description

Verification/Validation

Exit Conditions

Representation of processes according Turtle diagrams


With **what**? Infrastructure
Who? Resources
Input: what should we receive?
Process
Output: what should we deliver?
How many? Measurement
How? Methods, procedures