

INTERNATIONAL  
STANDARD

ISO/IEC/  
IEEE  
**42020**

First edition  
2019-07

---

---

---

**Software, systems and enterprise —  
Architecture processes**

*Logiciel, systèmes et entreprise - Processus d'architecture*



Reference number  
ISO/IEC/IEEE 42020:2019(E)



© ISO/IEC 2019  
© IEEE 2019



## **COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2019

© IEEE 2019

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Tel. +41 22 749 01 11  
Fax +41 22 749 09 47  
[copyright@iso.org](mailto:copyright@iso.org)  
[www.iso.org](http://www.iso.org)

Published in Switzerland

Institute of Electrical and Electronics Engineers, Inc  
3 Park Avenue, New York  
NY 10016-5997, USA

[stds.ipr@ieee.org](mailto:stds.ipr@ieee.org)  
[www.ieee.org](http://www.ieee.org)

© ISO/IEC 2019 – All rights reserved

© IEEE 2019 – All rights reserved

Authorized licensed use limited to: Tampere University. Downloaded on March 02,2025 at 07:53:52 UTC from IEEE Xplore. Restrictions apply.

# Contents

	Page
<b>Foreword</b>	<b>vi</b>
<b>0 Introduction</b>	<b>vii</b>
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>1</b>
<b>3 Terms and definitions</b>	<b>1</b>
<b>4 Conformance</b>	<b>6</b>
4.1 General	6
4.2 Approach to conformance	6
4.3 Full conformance cases	6
4.4 Tailored conformance	7
<b>5 Process overview and application</b>	<b>7</b>
5.1 General	7
5.2 Relationship of architecture to other processes and information elements	9
5.3 Architecture Governance and Management processes	10
5.4 Architecture Conceptualization, Evaluation and Elaboration processes	10
5.5 Architecture Enablement process	12
5.6 Relationship of architecture to design	12
5.7 Architecture adaptation	13
5.8 Process application	13
5.8.1 Criteria for processes	13
5.8.2 Description of processes	14
5.8.3 General characteristics of processes	14
5.8.4 Tailoring	14
<b>6 Architecture Governance process</b>	<b>15</b>
6.1 Purpose	15
6.2 Outcomes	15
6.3 Implementation	15
6.4 Activities and tasks	15
6.4.1 Prepare for and plan the architecture governance effort	15
6.4.2 Monitor, assess and control the architecture governance activities	17
6.4.3 Establish architecture collection objectives	17
6.4.4 Make architecture governance decisions	18
6.4.5 Monitor and assess compliance with governance directives and guidance	19
6.4.6 Review implementation of governance directives and guidance	20
6.5 Work products	20
<b>7 Architecture Management process</b>	<b>20</b>
7.1 Purpose	20
7.2 Outcomes	20
7.3 Implementation	21
7.4 Activities and tasks	21
7.4.1 Prepare for and plan the architecture management effort	21
7.4.2 Monitor, assess and control the architecture management activities	22
7.4.3 Develop architecture management approach	23
7.4.4 Perform management of the architecture collection	24
7.4.5 Monitor architecting effectiveness	25
7.4.6 Prepare for completion of the architecture management plan	26
7.5 Work products	26
<b>8 Architecture Conceptualization process</b>	<b>27</b>
8.1 Purpose	27
8.2 Outcomes	27
8.3 Implementation	27

8.4	Activities and tasks.....	28
8.4.1	Prepare for and plan the architecture conceptualization effort.....	28
8.4.2	Monitor, assess and control the architecture conceptualization activities.....	29
8.4.3	Characterize problem space.....	30
8.4.4	Establish architecture objectives and critical success criteria.....	31
8.4.5	Synthesize potential solution(s) in the solution space.....	32
8.4.6	Characterize solutions and the tradespace.....	33
8.4.7	Formulate candidate architecture(s) .....	35
8.4.8	Capture architecture concepts and properties.....	36
8.4.9	Relate the architecture to other architectures and to relevant affected entities..	37
8.4.10	Coordinate use of conceptualized architecture by intended users.....	37
8.5	Work products.....	38
9	<b>Architecture Evaluation process.....</b>	<b>38</b>
9.1	Purpose.....	38
9.2	Outcomes .....	39
9.3	Implementation.....	39
9.4	Activities and tasks.....	40
9.4.1	Prepare for and plan the architecture evaluation effort.....	40
9.4.2	Monitor, assess and control the architecture evaluation activities .....	41
9.4.3	Determine evaluation objectives and criteria.....	42
9.4.4	Determine evaluation methods and integrate with evaluation objectives and criteria.....	43
9.4.5	Establish measurement techniques, methods and tools .....	43
9.4.6	Collect and review evaluation-related information.....	44
9.4.7	Analyze architecture concepts and properties and assess stakeholder value.....	44
9.4.8	Characterize architecture(s) based on assessment results .....	45
9.4.9	Formulate findings and recommendations.....	46
9.4.10	Capture and communicate evaluation results.....	46
9.5	Work products.....	47
10	<b>Architecture Elaboration process.....</b>	<b>47</b>
10.1	Purpose.....	47
10.2	Outcomes .....	47
10.3	Implementation.....	48
10.4	Activities and tasks.....	48
10.4.1	Prepare for and plan the architecture elaboration effort.....	48
10.4.2	Monitor, assess and control the architecture elaboration activities .....	49
10.4.3	Identify or develop architecture viewpoints .....	50
10.4.4	Develop models and views of the architecture(s).....	51
10.4.5	Relate the architecture to other architectures and to relevant affected entities..	52
10.4.6	Assess the architecture elaboration .....	52
10.4.7	Coordinate use of elaborated architecture by intended users.....	53
10.5	Work products.....	53
11	<b>Architecture Enablement process.....</b>	<b>54</b>
11.1	Purpose.....	54
11.2	Outcomes .....	54
11.3	Implementation.....	54
11.4	Activities and tasks.....	54
11.4.1	Prepare for and plan the architecture enablement effort .....	54
11.4.2	Monitor, assess and control the architecture enablement activities .....	55
11.4.3	Manage the architecture process enablers .....	56
11.4.4	Acquire, develop and establish enabling capabilities, services and resources .....	56
11.4.5	Deploy enabling capabilities, services and resources .....	57
11.4.6	Improve architecture enablement capabilities, services and resources .....	57
11.5	Work products.....	58
	<b>Annex A (normative) Tailoring process .....</b>	<b>59</b>
	<b>Annex B (informative) Defining metrics for architecture processes .....</b>	<b>63</b>

<b>Annex C</b> (normative) <b>Interactions with other processes and uses of architecture</b> .....	<b>66</b>
<b>Annex D</b> (informative) <b>Relationship with other standards</b> .....	<b>72</b>
<b>Annex E</b> (informative) <b>Notes on terms and concepts</b> .....	<b>73</b>
<b>Annex F</b> (informative) <b>Architecture enablement and process-enabling resources</b> .....	<b>92</b>
<b>Annex G</b> (informative) <b>Architecture governance and management</b> .....	<b>94</b>
<b>Annex H</b> (informative) <b>Mapping of processes to architecture frameworks</b> .....	<b>96</b>
<b>Bibliography</b> .....	<b>109</b>
<b>IEEE notices and abstract</b> .....	<b>112</b>

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the rules given in the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](http://www.iso.org/patents)) or the IEC list of patent declarations received (see <http://patents.iec.ch>).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Systems and Software Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

# 0 Introduction

## 0.1 Rationale for architecture processes

The complexity of human-made systems has grown to an unprecedented level, which leads to new opportunities and greater challenges for organizations that create, trade and utilize systems. To address these opportunities and challenges, it is increasingly necessary to apply concepts, principles, procedures and tools to make better architecture-related decisions, more effective architectures, better architecture strategy and increased architecture maturity. Architecture-related activities are now strategic aspects of projects and enterprises, and the use of architecture frameworks has become common practice in commercial, government, civil and military domains.

Architecture is increasingly applied to systems—and to other entities that are not traditionally considered to be systems, such as enterprises, services, data, business functions, mission areas, product lines, families of systems, software items, etc. The concept of architecture used in this document goes beyond the traditional use where the architecture entity is a system. This allows for a more generalized usage of architecture when the processes in this document are applied. These entities are becoming more complex and architecture practices are increasingly adopted to manage the complexity.

Within enterprises and the engineering disciplines, acknowledgement is increasing for the value added by architecture, both as a practice and in the realization of artifacts that guide engineering and management activities.

This document complements the architecture-related processes identified in ISO/IEC/IEEE 15288, ISO/IEC/IEEE 12207 and ISO 15704 with activities and tasks that enable architects and others to more effectively and efficiently implement architecture practices. Implementing these practices can help ensure that the architecture has greater influence on business and mission success.

## 0.2 Use of the term architecture in this document

This document uses the term architecture in a broad sense. When the word architecture is used without any qualifier the word refers to the general case where the architecture entails the fundamental concepts and properties of an architecture entity. When a qualifier is prepended to the word architecture, this indicates that the architecture applies to that entity, such as in the following cases:

- System Architecture: When the entity is a system.
- Enterprise Architecture: When the entity is an enterprise.

The following are kinds of architecture entities that can be dealt with by the architecture processes of this document: enterprise, organization, solution, system (including software systems), subsystem, business, data (as a data element or data structure), application, information technology (as a collection), mission, product, service, software item, hardware item, etc. The kind of entity can also be a product line, family of systems, system of systems, collection of systems, collection of applications, etc.

There can be cases where the word architecture is prepended by the subject of interest, not by the entity being architected, such as in the following examples: security architecture, functional architecture, physical architecture and so on. See [E.4.1](#) for more examples.

Finally, there are cases when the word architecture is prepended by the purpose of the architecture, for example integration architecture, coherence architecture, design-control architecture, etc. See [E.4.1](#) for more examples.

## 0.3 Purpose

The purpose of this document is to set the standard of performance for the governance, management, conceptualization, evaluation and elaboration of architectures, and activities that enable these processes. This document can be used as a process reference model in establishing architecture practice and be used across a range of contexts and situations. It provides guidance in conforming to

the architecture processes specified in this document, and, in a larger context, to facilitate trading in systems, products and services.

#### **0.4 Field of application**

The processes specified in this document apply in the context of:

- understanding, developing and evolving entities through their life cycle stages such as conception, development, implementation, operation, sustainment, decommissioning and disposal;
- the type of architecture to be developed;
- organization(s) acting as users, customers and providers of the solution specified by the architecture description; and
- architecting of entities.

The intent is to provide processes applicable across a wide spectrum of architecting domains (such as the enterprise, systems, services and software domains) for use by a broad range of architects and users of these practices.

When the entity is a system then it is necessary to consider that:

- Systems can vary widely in terms of purpose, domain of application, complexity, size, novelty, adaptability, qualities, locations, life spans and evolution. This document specifies processes for the development and use of architecture that involves human-made systems including one-of-a-kind systems, mass-produced systems and customized, adaptable systems either as a complete stand-alone system or systems embedded and integrated into larger, more complex and complete systems.
- Systems addressed by this document can be configured with one or more of the following kinds of system elements: hardware, software, data, humans, processes (e.g. processes for providing services to users), procedures (e.g. operator instructions), facilities, materials and naturally occurring entities.
- The processes in this document can be used to define the architecture of a system as well as to independently define the architecture of a system of systems involving that system or the architecture of an element of that system, such as a software, data or hardware item.

#### **0.5 Reference model for processes**

This document provides a process reference model defined according to the ISO/IEC TR 24774 guidelines. This process reference model is characterized by process purpose and process outcomes that result from the successful execution of the relevant tasks in each of the process activities, and the creation of relevant work products, following the process constructs of ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207. Therefore, this document is useful to support process assessment as specified in ISO/IEC 33002. ISO/IEC/IEEE 15288:2015, Annex C provides information regarding the use of processes as a process reference model.

The processes specified in this document are applicable:

- concurrently, iteratively, incrementally and recursively to an architecture entity or its elements; and
- for the management and control of portfolios, programs and projects pertaining to the entities being architected.

#### **0.6 Intended audience**

The architecture processes specified in this document apply in the context of an enterprise or an extended enterprise, as well as on individual organizations or projects within the enterprise.

This document is applicable to organizations in their roles as both acquirers and suppliers of architected entities or their elements, and is useful for a single organization in a self-imposed mode or in a multi-

party situation involving agreements between parties. Parties can be from the same organization or from different organizations and the situation can range from informal agreements to formal contracts.

The principal intended users of this document are architects and others who create, express, evaluate, communicate and document architectures. Other users include:

- clients, acquirers, designers, service providers, sub-contractors, users and operators of systems and others who need to understand architectures;
- developers and other stakeholders who need to understand, interpret and analyze architecture descriptions to establish, maintain and transform enterprises, systems or other entities;
- chief information officers, chief engineers, program/enterprise managers, auditors, independent assessors and those who oversee and evaluate architecture entities and their development;
- managers of architecting endeavors who establish, plan, monitor and control such undertakings;
- people involved in enterprise-wide activities that span development of multiple systems products, services and software, including those that seek to establish and codify architecture frameworks, architecture viewpoints and architecting methods;
- business analysts who need to understand the norms for the architecture process and process outcome sufficiently in order to verify whether a given architecture description (a) is consistent with their stakeholder needs, and (b) does not risk leaving any of their needs unsatisfied or contradicted; and
- developers of tools and methods used in support of architecting practices, architecture governance and management, and enablement of architecture process implementation.

Additional users include researchers who can use this document to provide a common framework for expressing their research discoveries related to novel methods or techniques that enable or improve the practices of architecting, architecture governance and architecture management, as well as improving the enablement of these practices.

## **0.7 Benefits from use of this document**

This document provides a process framework that:

- contributes to the identification of job roles and responsibilities in the organization, along with requisite skills and competencies;
- facilitates proper oversight, accountability, consistent governance and management, and alignment within and between architectures;
- enables proper implementation of architecture governance directives and change management of architectures; and
- facilitates the effective planning and tracking of the architecture effort.

A set of well-specified architecture activities results in:

- an architecting capability that is applicable to all architecture efforts, irrespective of size and complexity;
- a framework that provides a consistent approach for developing an architecture based on addressing stakeholder concerns and for identifying the aspects of the architecture that would be required to address those concerns;
- standardized architecture approaches that can be adopted by enterprise, system, information technology, software, product and service architects;
- an effective mechanism that facilitates the understanding and communication of the problem and corresponding solution to various stakeholders; and

- a common vocabulary that facilitates communication between stakeholders.

Various groups and individuals benefit from the use of standardized architecture processes, including:

- solution acquirers in helping them characterize the business context, evaluate providers' proposals, identify alternatives, make informed decisions, and in facilitating collaboration between providers who will work together on architecture development and governance;
- solution providers in helping them understand the problem/request, elaborate a proposal in their solution space, and define and justify their deliveries;
- solution users in helping them express the operational context, characterize their needs and evaluate providers' proposals in the context of their problem space;
- decision makers and program/project managers in helping them consider a range of options during creation and usage of architectures which are considered as a source of information and as a basis for the rationale when decisions are made; and
- other bodies such as legal, safety and security authorities, in helping them assess compliance with standards, policies, directives, treaties, regulations and laws.

## **0.8 Limitations**

No formal traceability is made between ISO/IEC/IEEE 15288, ISO/IEC/IEEE 12207, ISO 15704 and this document. Consequently, meeting all requirements in this document does not necessarily mean that all requirements related to architecture processes specified in those other documents are met.

This document does not specify a particular life cycle model to be used when applying these processes.

The ISO/IEC/IEEE 24748 series provides guidance for life cycle definition and application of life cycle processes. Although this document does not establish a management system, the intent of this specification is to be compatible with the quality management system provided by ISO 9001, the service management system provided by ISO/IEC 20000-1 (also published as IEEE Std 20000-1), and the information security management system provided by ISO/IEC 27000.

This document does not specify detailed information items in terms of format, explicit content and recording media. ISO/IEC/IEEE 15289 addresses the content for life cycle process information items (documentation).

This document does not specify any particular architecture framework or architecture documentation standard.

# Software, systems and enterprise — Architecture processes

## 1 Scope

This document establishes a set of process descriptions for the governance and management of a collection of architectures and the architecting of entities. This document also establishes an enablement process description that provides support to these other architecture processes.

The processes defined in this document are applicable for a single project, as well as for an organization performing multiple projects. These processes are applicable throughout the life of an architecture or a collection of architectures. These processes are applicable for managing and performing the activities within any stage in the life cycle of the architecture entities.

[Annex D](#) describes the relationships between this document and other standards.

## 2 Normative references

There are no normative references in this document.

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO, IEC and IEEE maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>
- IEEE Standards Dictionary Online: available at <http://ieeexplore.ieee.org/xpls/dictionary.jsp>

NOTE Definitions for other terms typically can be found in ISO/IEC/IEEE 24765, which provides the vocabulary for system and software engineering, available at [www.computer.org/sevocab](http://www.computer.org/sevocab).

### 3.1

#### activity

set of cohesive tasks ([3.23](#)) of a process ([3.16](#))

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.3]

### 3.2

#### architecting

conceiving, defining, expressing, documenting, communicating, certifying proper implementation of, maintaining and improving an *architecture* ([3.3](#)) throughout the *life cycle* ([3.11](#)) for an *architecture entity* ([3.6](#))

Note 1 to entry: The entity to be architected can be of several kinds, as illustrated in the following examples: system, *enterprise* ([3.9](#)), solution, business, data, application, information technology, mission, product, service, software, etc. See [E.4](#) for more information on this topic.

Note 2 to entry: Certifying the proper implementation of an architecture is sometimes captured as a formal statement by the architect to the client or user that the system, as built, meets the criteria as ready for use.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.1 modified — The word “system” has been replaced with “architecture entity”; the original NOTE has been removed; Notes 1 and 2 to entry have been added.]

### 3.3

#### **architecture**

fundamental concepts or properties of an entity in its environment and governing principles for the realization and evolution of this entity and its related *life cycle* (3.11) *processes* (3.16)

Note 1 to entry: *Architecture entity* (3.6) is the term used in this document when referring to the entity being architected or the entity subject to architecture processes. The fundamental concepts or properties of the architecture entity are usually intended to be embodied in the entity’s components, the relationships between components, and the relationships between the entity and its environment.

Note 2 to entry: The concept of architecture used in this document applies broadly to the entity being architected or evaluated. This allows for a more generalized usage when the elements in this document are applied.

Note 3 to entry: The entity to be architected can be of several kinds, as illustrated in the following examples: *enterprise* (3.9), *organization* (3.14), solution, system, subsystem, business, data (as a data element or data structure), application, information technology (as a collection), mission, product, service, software item, hardware item, product line, family of systems, system of systems, etc. It also spans the variety of applications that utilize digital technology such as mobile, cloud, big data, robotics, Internet of things (IoT), web, desktop, embedded systems and so on.

Note 4 to entry: Representation of the concepts or properties of an entity and governing principles is captured in architecture *models* (3.13).

Note 5 to entry: Architectures can address a wide range of *concerns* (3.8) expressed, for example, through architecture *views* (3.24) and models, as illustrated in the following examples associated with particular kinds of architectures such as: security architecture, functional architecture, physical architecture, resilience architecture, etc.

### 3.4

#### **architecture collection**

group of *architectures* (3.3) held by an *organization* (3.14) that is subject to governance and management by the organization as a whole

Note 1 to entry: The architectures in the collection can have relationships with each other (as in the case of product lines). The architectures in the collection can be based on the same reference architecture.

### 3.5

#### **architecture description**

*work product* (3.26) used to express an *architecture* (3.3)

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.3, modified — The abbreviated term has been removed.]

### 3.6

#### **architecture entity**

thing being considered, described, discussed, studied or otherwise addressed during the *architecting* (3.2) effort

**EXAMPLE** The following are kinds of architecture entities that can be dealt with by the *architecture* (3.3) *processes* (3.16): *enterprise* (3.9), *organization* (3.14), solution, *system* (3.22) (including software systems), subsystem, business, data (as a data element or data structure), application, information technology (as a collection), mission, product, service, software item, hardware item, product line, family of systems, system of systems, collection of systems, collection of applications, etc.

Note 1 to entry: When referring to the architecture itself of these architecture entities, it is common practice to place the name of the kind of entity in front of the word architecture. For example, the phrase system architecture is used when the thing being dealt with during the architecting effort is a system. Likewise, for the other kinds of entities that are being dealt with during the architecting effort.

**3.7****architecture framework**

conventions, principles and practices for use by *architecture-related* (3.3) activities that have been established within a specific domain of application or community of *stakeholders* (3.21)

EXAMPLE 1 Generalised Enterprise Reference Architecture and Methodology (GERAM) (see ISO 15704) includes an architecture description framework (called the GERA Modelling Framework).

EXAMPLE 2 Reference Model of Open Distributed Processing (RM-ODP) is an architecture framework. See the ISO/IEC 10746 series.

EXAMPLE 3 Department of Defense Architecture Framework (DoDAF), Ministry of Defence Architecture Framework (MODAF), Department of National Defense/Canadian Armed Forces Architecture Framework (DNDCAF), NATO Architecture Framework (NAF), The Open Group Architecture Framework (TOGAF®) are some architecture frameworks.

Note 1 to entry: The concept of architecture framework has been expanded in this document beyond the way this term is used in ISO/IEC/IEEE 42010 where it is used strictly with regard to the “description of architectures”.

**3.8****concern**

matter of interest or importance to a *stakeholder* (3.21)

EXAMPLE Affordability, agility, availability, dependability, flexibility, maintainability, reliability, resilience, usability and viability are examples of concerns. Survivability, depletion, degradation, loss, obsolescence are examples of concerns. The PESTEL mnemonic is a reminder of possible areas of concern: political, economic, social, technological, environmental, and legal.

**3.9****enterprise**

bold or complex endeavor

Note 1 to entry: One or more *organizations* (3.14) can participate in an enterprise. In case of multi-organization enterprises, each of the organizations brings various resources forward for use in the enterprise and they participate to the extent that they benefit from their involvement. The purpose of the enterprise is to address some challenges that these participating organizations cannot readily address on their own. Within a single organization, an enterprise may refer to a subset of the organization which is typically addressing particularly challenging or complex issues, often over a defined duration, and may undertake this with certain relaxations, tightening or otherwise authorized modifications of standard corporate *processes* (3.16) and practices (see definition of organization).

**3.10****library**

place containing collections of *work products* (3.26) and useful information items for people to read, borrow or refer to, and for machines to access and retrieve data from

Note 1 to entry: In a *repository* (3.19), work products and other items are preserved for future retrieval when needed, whereas in a library, working data is temporarily stored and retrieved as necessary.

**3.11****life cycle**

<entity> set of distinguishable *phases* (3.15) or *stages* (3.20) that an entity goes through from its conceptualization until it ceases to exist

**3.12****life cycle**

<architecture> set of distinguishable *phases* (3.15) or *stages* (3.20) that an *architecture* (3.3) goes through

Note 1 to entry: The architecture life cycle starts with the identification of a need for the architecture and ends when it is no longer needed.

### 3.13 model

abstract representation of an entity or collection of entities that provides the ability to portray, understand or predict the properties or characteristics of the entity or collection under conditions or situations of interest

Note 1 to entry: A model can use a formalism that could be based on mathematical or scientific principles and concepts. A model can be generated using an established metamodel. Metamodels are often used to facilitate development of accurate, complete, consistent and understandable models.

Note 2 to entry: A model can be used to construct or express *architecture* (3.3) *views* (3.24) of the entity. Descriptive models and analytic models are two kinds of models. A model should be governed by a model kind in accordance with ISO/IEC/IEEE 42010.

Note 3 to entry: A reference model can be used to capture a general case that is used as the basis for creating special case models for particular conditions or situations. A reference model can be used to encourage and enforce uniformity of architectures and architecture elements.

Note 4 to entry: The model can be an architecture model, *architecture entity* (3.6) model, concept model or reference model, as the case may be.

### 3.14 organization

group of people and facilities with an arrangement of responsibilities, authorities and relationships

EXAMPLE Company, corporation, firm, *enterprise* (3.9), institution, charity, sole trader, association, or parts or combination thereof.

Note 1 to entry: An identified part of an organization (even as small as a single individual) or an identified group of organizations can be regarded as an organization if it has explicitly stated responsibilities, authorities and relationships. A body of persons organized for some specific purpose, such as a club, union, corporation or society, can be an organization.

Note 2 to entry: One or more organizations will participate in an enterprise. In case of multi-organization enterprises, each of the organizations brings various resources forward for use in the enterprise and they participate to the extent that they benefit from their involvement. The purpose of the enterprise is to address some challenges that these participating organizations cannot readily address on their own. Within a single organization, an enterprise may refer to a subset of the organization which is typically addressing particularly challenging or complex issues, often over a defined duration, and may undertake this with certain relaxations, tightening or otherwise authorized modifications of standard corporate *processes* (3.16) and practices. See definition of enterprise.

### 3.15 phase

period of time in the *life cycle* (3.11) during which activities are performed that enable achievement of objectives for that phase

### 3.16 process

set of interrelated or interacting activities that transforms inputs into outputs

[SOURCE: ISO 9000:2015, 3.4.1, modified — The words “use inputs to deliver an intended result” have been replaced with “transforms inputs into outputs”; Notes 1 to 6 to entry have been removed.]

### 3.17 project

endeavor with defined start and finish criteria undertaken to create a product or service in accordance with specified resources and requirements

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.33, modified — Note 1 to entry has been removed.]

**3.18****registry**

book or *system* (3.22) for keeping an official list or record of *work products* (3.26) and the associated information items

Note 1 to entry: *Repository* (3.19) and *library* (3.10) items should be recorded in registries to enable better management and governance of these items.

**3.19****repository**

place where *work products* (3.26) and the associated information items are or can be stored for preservation and retrieval

Note 1 to entry: Repository items should be under configuration control.

Note 2 to entry: In a repository, work products and other items are preserved for future retrieval when needed, whereas in a *library* (3.10), working data is temporarily stored and retrieved as necessary.

**3.20****stage**

period within the *life cycle* (3.11) of an entity that relates to the state of its description or realization

Note 1 to entry: As used in this document, stages relate to major progress and achievement milestones of the entity through its life cycle.

Note 2 to entry: Stages often overlap.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.43]

**3.21****stakeholder**

role, position, individual or *organization* (3.14) having a right, share, claim or other interest in an *architecture entity* (3.6) or its *architecture* (3.3) that reflects their needs and expectations

**3.22****system**

combination of interacting elements organized to achieve one or more stated purposes

Note 1 to entry: A system is sometimes considered as a product or as a set of services.

Note 2 to entry: In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g., aircraft system. Alternatively, the word “system” is substituted simply by a context-dependent synonym, e.g., aircraft, though this potentially obscures a system principles perspective.

Note 3 to entry: A system element is a discrete part of a system that can be implemented to fulfill specified requirements. A system element can be hardware, software, data, humans, *processes* (3.16) (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials, and naturally occurring entities (e.g., water, organisms, minerals), or any combination.

Note 4 to entry: A system can be comprised of multiple subsystems. For example, an aircraft system can include an avionics subsystem and a radar subsystem. The distinction between a system and a subsystem is a matter of perspective, and as such the radar subsystem can be referred to as a radar system in some contexts.

[SOURCE: ISO/IEC/IEEE 15288:2015, 4.1.46, modified — in Note 1 to entry “the services it provides” is replaced by “a set of services”; Note 3 to entry is from ISO/IEC/IEEE 15288:2008; Note 4 to entry has been added.]

**3.23****task**

recommended action intended to contribute to the achievement of one or more outcomes of an *architecture* (3.3) *process* (3.16)

### 3.24

#### view

<architecture> information item expressing the *architecture* (3.3) from the perspective of specific *stakeholders* (3.21) regarding specific aspects of the *architecture entity* (3.6) and its environment

Note 1 to entry: When the term view is used without any qualifier it refers to the general case. When a qualifier is prepended to the word view, this indicates that the architecture view is specific to a particular *viewpoint* (3.25), such as illustrated in these examples:

- operational view: when the associated viewpoint is dealing with operations;
- services view: when the associated viewpoint is dealing with services.

### 3.25

#### viewpoint

<architecture> conventions for the construction, interpretation and use of *architecture* (3.3) *views* (3.24) to address specific *concerns* (3.8) about the *architecture entity* (3.6)

Note 1 to entry: When the word "viewpoint" is used without any qualifier it refers to the general case. When a qualifier is prepended to the word viewpoint, this indicates that the viewpoint applies to a specific set of concerns, such as in the following examples: operational viewpoint, capability viewpoint, services viewpoint.

Note 2 to entry: ISO/IEC/IEEE 42010 specifies that an architecture view shall be governed by its viewpoint.

### 3.26

#### work product

artifact associated with the execution of a *process* (3.16)

## 4 Conformance

### 4.1 General

This document specifies requirements, recommendations and permissions for architecture processes in *Clauses* 6 through 11. Requirements are marked by the use of the verb "shall". Recommendations are marked by the use of the verb "should". Permissions are marked by the use of the verb "may". However, despite the verb that is used, the requirements for conformance are specified as described below.

### 4.2 Approach to conformance

The processes specified in this document are suitable for use during any stage in the life cycle of an entity, such as an enterprise, a project, a system or a product. Recognizing that particular projects or organizations may not need to use all of the processes provided by this document, implementation typically involves selecting and declaring a set of process activities suitable to the enterprise, organization or project. To accommodate that flexible implementation approach, this document provides two primary ways to claim conformance — full conformance and tailored conformance.

### 4.3 Full conformance cases

It is recognized that particular projects or organizations may not need to use all of the processes provided by this document. Therefore, implementation of this document typically involves selecting and declaring a set of processes suitable to the organization or project.

A claim of full conformance declares the set of processes for which conformance is claimed. The selected processes, for which conformance is claimed, are declared. Full conformance is achieved by demonstrating that requirements for the declared set of processes, as tailored, have been satisfied.

To claim full conformance, a user of this document shall demonstrate conformance with respect to one or more of the following three criteria:

- Claiming “conformance to outcomes” asserts achievement of all required outcomes of the declared set of processes.

NOTE 1 In this situation, the provisions for activities and tasks of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

NOTE 2 Some users could have innovative process variants or life cycle models that achieve the objectives of the declared set of processes without implementing all of the activities or delivering all the work products. These users can instead assert conformance to the outcomes of the declared set of processes.

- Claiming “conformance to activities” asserts achievement of all requirements for the activities of the declared set of processes.

NOTE 3 Conformance at the task level is considered to be too constraining in most situations. The tasks specified in this document are recommendations only.

- Claiming “conformance to work products” asserts achievement of all required work products (as specified in the fifth subclause in each process description) associated with the declared set of processes.

NOTE 4 In tailoring, work products required in this document can be modified (added to, combined or retitled). The titles of work products can be tailored to satisfy requirements of an organization, its projects, or agreements based on the tailored conformance to this document. The contents of the work products correspond to the selected or tailored processes.

NOTE 5 The three criteria—conformance to outcomes, conformance to activities, and conformance to work products—are not necessarily equivalent since specific performance of activities could need, in some cases, a higher level of capability than just the achievement of process outcomes and delivery of the work products.

#### 4.4 Tailored conformance

When use of this document establishes a set of processes that do not qualify for full conformance, the selected or adapted clauses and subclauses shall be identified in accordance with the Tailoring process defined in [Annex A](#). The claim of tailored conformance shall include the modified text to replace the relevant clause or subclause. Tailoring may be performed by process, outcomes, activities, work products and tasks. Tailored conformance shall demonstrate achievement of the outcomes, activities and work products as tailored. Tailored conformance can also entail selection or modification of tasks, although the tasks are not required for conformance purposes. All the levels of tailoring are acceptable as long as consequence of a partial conformance is understood and accepted. A rationale shall be provided to show relevance of this tailoring.

### 5 Process overview and application

#### 5.1 General

This document describes each of the architecture processes in terms of a purpose, desired outcomes, and a list of activities for achieving those outcomes. Tasks are recommended for implementing those activities. [Table 1](#) provides the purpose of each process.

**Table 1 — Architecture processes and their purposes**

Clause	Process	Purpose
<a href="#">6</a>	Architecture Governance	Establish and maintain alignment of architectures in the architecture collection with enterprise goals, policies and strategies and with related architectures

**Table 1** (*continued*)

<b>Clause</b>	<b>Process</b>	<b>Purpose</b>
<a href="#">7</a>	Architecture Management	Implement architecture governance directives to achieve architecture collection objectives in a timely, efficient and effective manner
<a href="#">8</a>	Architecture Conceptualization	Characterize the problem space and determine suitable solutions that address stakeholder concerns, achieve architecture objectives and meet relevant requirements
<a href="#">9</a>	Architecture Evaluation	Determine the extent to which one or more architectures meet their objectives, address stakeholder concerns and meet relevant requirements
<a href="#">10</a>	Architecture Elaboration	Describe or document an architecture in a sufficiently complete and correct manner for the intended uses of the architecture
<a href="#">11</a>	Architecture Enablement	Develop, maintain and improve the enabling capabilities, services and resources needed to perform the other architecture processes

These architecture processes and their key interactions in terms of typical information flows for a particular team, project, organization or enterprise are depicted in [Figure 1](#). The interactions between the core processes are not labeled here but are depicted in more detail in [Figure 2](#). Not all the possible interactions are shown in [Figure 1](#) and [Figure 2](#) for implementing these processes. Typical interactions with external processes are described in [Annex C](#).

The architecture processes may execute concurrently with interactions between them and iterating over time. At the same time, governance directives and management instructions flow from Architecture Governance process to the Architecture Management process and operational plans and status flows from the Architecture Management process to the Architecture Governance process. The interactions between the core processes are described in [5.4](#).

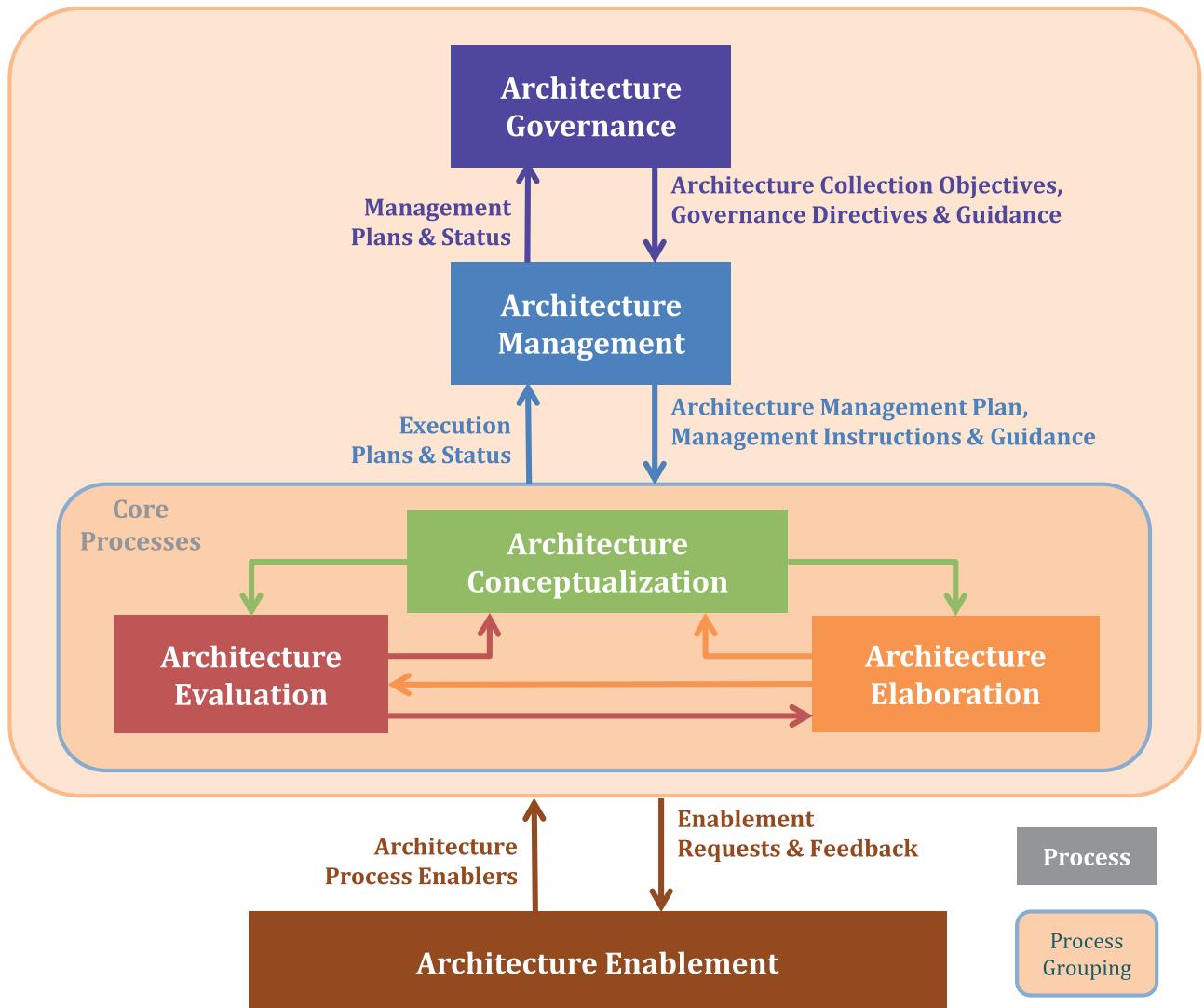


Figure 1 — Architecture processes and their interactions

## 5.2 Relationship of architecture to other processes and information elements

There are other life cycle processes that are affected by these architecture processes. For example, system requirements may be derived from the architecture and the architecture may be driven by requirements. The specific nature of how architecture and requirements are related to each other is organization dependent so cannot be specified in this document.

Likewise, architecture has a relationship to risks, decisions, project planning, integration, verification, acquisition, supply, quality management, investment management and so on, but the specific nature of these relationships cannot be specified in this document. The architecture processes shall interact with other processes and information elements as described in [Annex C](#).

Iteration of the architecture processes with the Business or Mission Analysis process, System Requirements Definition process, Design Definition process and Stakeholder Needs and Requirements Definition process is often employed so that there is a negotiated understanding of the problem to be solved and a satisfactory solution is identified. The results of the architecture processes are widely used across the other life cycle processes. Architecture processes may be applied at many levels of abstraction, highlighting the relevant detail that is necessary for the decisions at that level.

### 5.3 Architecture Governance and Management processes

Architecture governance and management processes are applicable to a collection of architectures, typically in an enterprise or project context. During architecture governance and management, the stakeholders who make financial, governance and technical decisions are identified, documented and included in performance of the activities of these processes. The governance process ensures proper oversight and accountability, and identifies, manages, audits and disseminates all information related to architecture collection decisions. It is applicable for all architecture efforts in an organization.

The management process implements governance directives and guidance and captures this in an architecture management plan. The core processes are monitored and assessed against the plan and appropriate controls are implemented to direct course corrections. Instructions and guidance are issued to the core processes as a means to provide further direction beyond what the plan provides, especially with respect to directing how the architecture(s) should be developed and used.

Architecture governance specifies the governance directives and guidance that can be used to drive the appropriate evolution of architectures in the architecture collection. It also specifies the architecture collection objectives to be pursued and establishes the strategy to be adopted for achievement of these objectives. It uses the management plans and status regarding the architecture collection to monitor compliance with governance directives and guidance.

**NOTE 1** The governance framework typically comprises a set of controls over the creation and monitoring of all architecture process components, work products and outcomes.

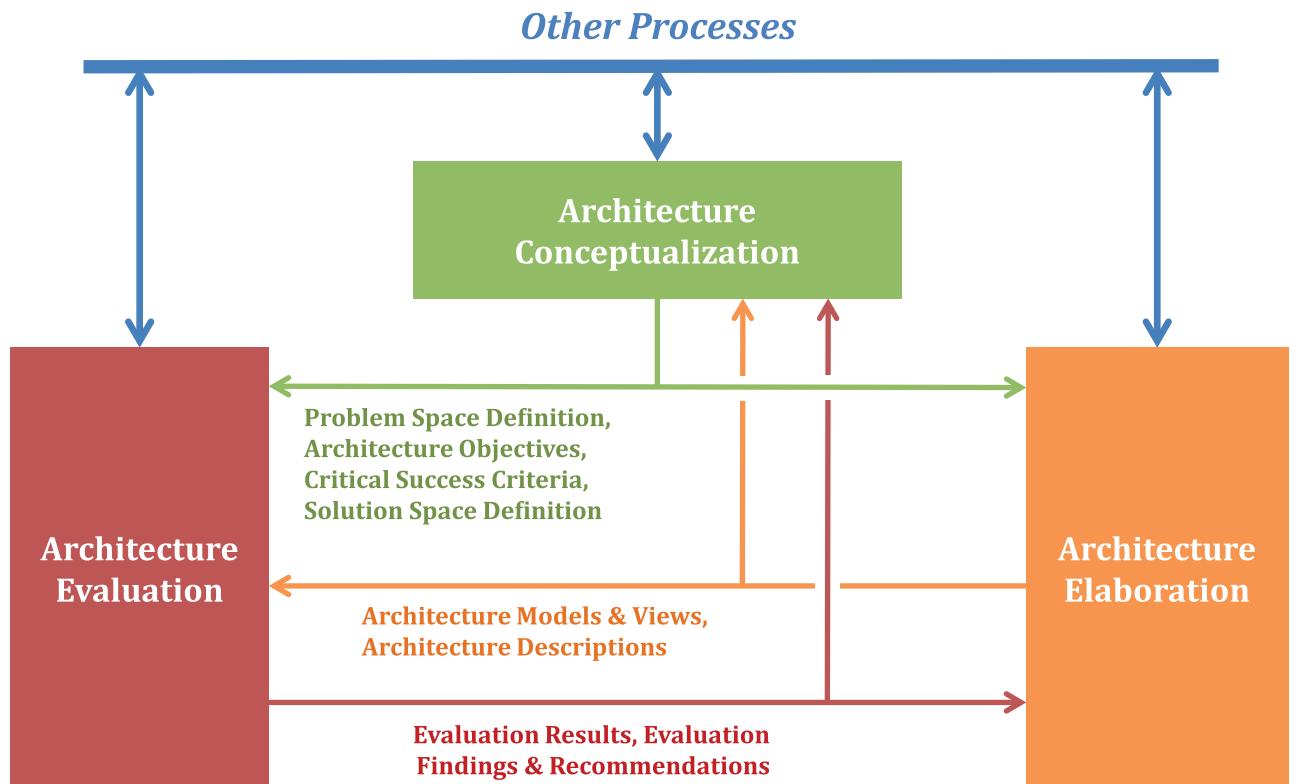
**NOTE 2** Absolute control may not be possible in cases like: SOS situations, federated architectures, consortium-based architectures, public-private partnerships, etc.

Architecture management specifies the management plans and instructions that are used to drive the core architecture processes. It uses the execution plans and status of the core processes to monitor compliance with management plans and instructions. In addition to management instructions, management guidance is developed as a means to assist those following the plans and instructions in implementing governance directives and in carrying out work to be in better alignment with management intent.

**NOTE 3** See [Annex G](#) for additional information regarding architecture governance and management.

### 5.4 Architecture Conceptualization, Evaluation and Elaboration processes

Key interactions between the core processes (Conceptualization, Evaluation and Elaboration) are illustrated in [Figure 2](#). These processes can also be triggered by other processes external to these three processes. Conceptualization specifies the objectives of the architecture and the quality measures that can be used in the assessment of its value. Value is defined in terms of the extent to which stakeholder concerns are addressed. These architecture objectives are based on the problem/opportunity identification and definition that occurs in this process. Architecture concepts are generated with value in mind and are then assessed using these quality measures.



**Figure 2 — Interactions between the core processes and with other processes**

Conceptualization aids architects and others in characterizing the problem space, synthesizing potential solutions, formulating candidate architectures and expressing these architectures in a form that is suitable for the intended uses. The name of the process being “conceptualization” does not mean that the results are necessarily at the “conceptual” level, or consist of a set of conceptual models and views. The results could include a “logical” architecture or a “physical” architecture, depending on the nature of the situation.

During early stages, it can be important to be agile and quick in conceptualizing many alternative architectures. Some of these early architecture descriptions are little more than sketches. After doing several quick rounds of evaluation and conceptualization, there can then be a smaller number of viable architectures that are worth capturing in a more complete form and storing in the repository for later use. A more complete form of architecture description would be developed during architecture elaboration.

Conceptualization can use the results of architecture elaboration, when appropriate. Architecture conceptualization only needs to describe the architecture to the level of specificity and granularity that is suitable for its intended users, which in many cases does not require significant elaboration. The elaboration of architecture views, models and descriptions can often occur later in the life cycle of the architecture, after the architecture has become more mature and the extra effort of elaboration becomes worthwhile. Elaboration can often be deferred until after several architecture alternatives have been examined for their suitability, and they are selected down to one or a few alternatives for further examination and eventual use downstream in the engineering effort.

During evaluation, there could be a recognition that alternative architecture concepts are needed to more completely search the tradespace. However, in some cases there can be only a single architecture that is being evaluated for its suitability. Value assessments can be based on analysis of relevant architecture attributes and properties of the situation, or on an assessment of how much value is delivered to, for example, an operational setting or to individual users. The assessment and analysis results, along with estimates of assessment uncertainty, are returned along with key findings and recommendations, to determine if the proposed architecture sufficiently addresses stakeholder concerns. If not, then additional cycles between conceptualization and evaluation are pursued.

When further elaboration of the architecture is needed then the architecture objectives, architecture concepts and properties are expressed in a sufficiently complete and correct manner that can be delivered and used as the basis for more complete modeling, delineation and decomposition of these concepts and properties. Elaborated models and views, along with supporting materials, can be used to provide more detailed understanding of the architecture and to check for consistency with the original concepts and alignment with the objectives.

During evaluation, there can sometimes be a need for more complete models and views. In these cases, elaboration could be requested to generate additional models and views. During evaluation these models and views can be annotated with the results of the evaluation and with comments on strengths and weaknesses of the architecture or its description.

**NOTE** These processes can be repeated for each level of decomposition, refinement, realization, abstraction, disaggregation, etc. of the architecture. In such a case, the processes need to ensure that the architecture at that level is aligned with relevant architectures above and below that level.

Any of the architecture processes can generate or contribute to an architecture description of varying scope, granularity and formality. However, the elaboration process captures the architecture description in a sufficiently complete and correct manner for the intended uses and intended users of the architecture downstream from the architecture effort.

## 5.5 Architecture Enablement process

Architecture enablement selects, modifies and develops capabilities, services and resources in support of the other processes. In this document, these things are called enablers. This process can be used to enable the development of an individual architecture or a collection of architectures, or provide enabling resources, capabilities and services to all architecture endeavors of an organization. Architecture enablement ensures that the information regarding the different enablers is uniformly organized and integrated.

The other architecture processes request enablers to assist in performing the activities, utilize the enablers and provide feedback in terms of the effectiveness of the enablers and the changes that are necessary to improve their effectiveness. These other processes also may contribute enablers to the Architecture Enablement process which can then be made available for use by other projects. Architecture enablement tracks the usage and usefulness of the enablers and the difficulties that are faced when utilizing these enablers. Architecture enablement establishes and maintains enablers for use throughout the organization.

**EXAMPLE** Enabling capabilities include, among other things, procedures, methods, tools, frameworks, architecture viewpoints, work product templates, decision support systems, storage, configuration management and reference models. Enabling services include, among other things, infrastructure, technologies, skilled personnel and automation agents. Enabling resources include, among other things, architecture repository, library, registry, communication channels and mechanisms, human and technical resources, and training and licenses for tools and methods.

The key enablers are the architecture repository, library and registry. The architecture repository provides mechanisms to store, manage and manipulate architecture work products. It also provides appropriate access to projects, organizations, and individuals. The architecture library provides services to the projects or organizations for finding and organizing information. It also catalogs and stores source materials for ready access by all those involved in an architecture effort. In support of the repository and library is an architecture registry, where the items can be referenced as to their intended uses, anticipated limitations, location in the repository or library, pointers to other relevant materials that enable proper utilization, and to their stage of development.

**NOTE** See [Annex F](#) for additional information regarding architecture enablement practices.

## 5.6 Relationship of architecture to design

Architecture processes provide one or more architecture alternatives that frame the concerns of stakeholders and addresses their key requirements. Architecture can be widely used across the life

cycle processes of the architecture entity. Architecture processes can be applied at many levels of abstraction, highlighting the relevant features that are necessary for the decisions at that level.

As defined in [3.3](#), architectures provide the fundamental concepts or properties of the architecture entity and associated governing principles. Architectures should be described using a set of views and models that are complete, consistent and correct. The completeness of an architecture view or model is determined relative to its intended use.

Design, as defined in ISO/IEC/IEEE 15288, is a technical process providing sufficient details about the architecture entity and its elements to enable an implementation consistent with architecture. An effective architecture guides the design activities in such a way that it allows for maximum flexibility in the design tradespace.

Through use of the Design Definition process in ISO/IEC/IEEE 15288, insights are gained into the relation between the requirements specified for the architecture entity and the emergent properties and behaviors of the architecture entity that arise from the interactions and relations between the elements and between the properties of those elements.

Sometimes the elements of the architecture entity are initially notional until Design Definition process in ISO/IEC/IEEE 15288 has occurred since this depends on the actual design(s) to be done. Sometimes a “reference architecture” is created using these notional elements as a means to convey architectural intent and to check for design feasibility.

Interfaces and interactions between elements are defined at a level of detail necessary to convey the architectural intent and could be further refined in the Design Definition process. The Design Definition process considers any applicable technologies and their contribution to the system solution. Design Definition provides the level of the definition necessary for realization, such as drawings, detailed design descriptions, software code, task descriptions, etc.

The Design Definition process provides feedback to the architecture processes to consolidate or confirm the concepts and properties of the architecture entity, along with the allocation, partitioning and alignment of architectural entities to elements that compose the architecture entity.

## 5.7 Architecture adaptation

All the processes are involved with adaptation of the architecture to a given situation. Conceptualization and Elaboration act directly to make the architecture evolve and adapt. Evaluation provides evidence for decision making on architecture evolution and change. Architecture-related decisions could be made in conjunction with the architecture processes, but this is usually the responsibility of some other process where the decision is most relevant. Governance and Management provide directives and instructions for architecture evolution and change. Enablement provides enabling capabilities, services and resources that support the other processes in adapting the architecture.

## 5.8 Process application

### 5.8.1 Criteria for processes

The determination of architecture processes in this document is based upon three basic principles:

- Each architecture process has strong relationships among its outcomes, activities and work products.
- The dependencies between the processes are reduced to a minimum.
- A process is capable of execution by a single organization in the life cycle.

### 5.8.2 Description of processes

Each process of this document is described in terms of the following attributes:

- The title conveys the scope of the process as a whole.
- The purpose describes the goals of performing the process.
- The outcomes express the observable results expected from the successful performance of the process.
- The activities are sets of cohesive tasks of the process.
- The tasks are recommended actions intended to contribute to the achievement of one or more outcomes (i.e. results) that come about due to execution of the process.
- The work products help achieve the desired outcomes and can be produced based on the activities of the process.

Additional detail regarding this form of process description can be found in ISO/IEC TR 24774.

### 5.8.3 General characteristics of processes

In addition to the basic process attributes described above, processes may be characterized by other attributes common to all processes. ISO/IEC 33002 identifies common process attributes that characterize six levels of achievement within a measurement framework for use in process assessment activities. ISO/IEC/IEEE 15288:2015, Annex C includes the list of process attributes that contribute to the achievement of higher levels of process capability as defined in ISO/IEC 33002.

ISO/IEC 33020 specifies a process measurement framework for assessment of process capability.

[Annex C](#) in this document specifies how system life cycle processes in ISO/IEC/IEEE 15288 should use architecture-related information. [Annex C](#) in this document also specifies how the architecture processes should use information items from ISO/IEC/IEEE 15288 processes.

### 5.8.4 Tailoring

[Annex A](#) is normative and defines the basic activities needed to perform tailoring of this document.

Tailoring may entail the changing or removal of architecture processes. It may be performed for many reasons including the scale of the architecting endeavor, the purpose of the desired architecture (such as the kind of architecture sought), and the approach to architecting which is employed. For example, if the purpose of the architecting endeavor is to devise an architecture from an extant system or system design (i.e. to employ reverse architecting) then certain processes, such as architecture conceptualization in this particular case, may not need to be performed.

In all cases where such tailoring of architecture processes is employed, the rationale behind the tailoring shall be recorded. Recording of the rationale assists stakeholders in determining the value arising from the architecting effort and the confidence they should attribute to its outcomes.

Note that tailoring may diminish the perceived value of a claim of conformance to this document. This is because it is difficult for other organizations to understand the extent to which tailoring may have removed desirable provisions. An organization asserting a single-party claim of conformance to this document may find it advantageous to claim full conformance to a smaller list of processes rather than tailored conformance to a larger list of processes.

The processes described in this document are not intended to preclude or discourage the use of additional processes that organizations find useful. In particular, governance and directives of the organization can be applied for tailoring of the architecture governance and management processes and therefore tailoring of the process outcomes and the other architecture processes.

## 6 Architecture Governance process

### 6.1 Purpose

The purpose of the Architecture Governance process is to establish and maintain alignment of architectures in the architecture collection with enterprise goals, policies and strategies and with related architectures.

**NOTE** See [Annex G](#) for additional information regarding architecture governance and management and the distinction between these related but separate processes.

### 6.2 Outcomes

As a result of the successful implementation of the Architecture Governance process:

- a) Architecture collection objectives are addressing current and anticipated business needs of the enterprise and its customers and suppliers.
- b) Architecting activities and decisions are aligned with enterprise and contextual concerns.

**NOTE 1** Concerns can include organizational standards, policies, business priorities, relevant constraints and regulations.

- c) Architecture decisions supported by rationale are made with appropriate authorities and communicated to stakeholders.
- d) The various architectures within the architecture collection are consistent with, and in alignment to, the goals, objectives, strategy and vision of the organization responsible for the architecting effort.
- e) The various architectures within the architecture collection are in alignment with each other as necessary.
- f) Agreement of the purpose, objectives and goals for the architecture collection is maintained amongst the key stakeholders.

**NOTE 2** Architecture governance outcomes are applicable across the organization while architecture management outcomes are applicable across the entire collection of architectures.

### 6.3 Implementation

The organization shall implement the activities in [6.4](#) (numbered as 6.4.N) in accordance with applicable organization policies and procedures with respect to the Architecture Governance process. The activities may be performed in any order that is deemed appropriate. The organization should implement the relevant tasks (identified as list items under each 6.4.N activity) as appropriate to the situation.

The organization should store organization policies and procedures to facilitate widespread access, enable auditing and encourage future reuse.

### 6.4 Activities and tasks

#### 6.4.1 Prepare for and plan the architecture governance effort

- a) Identify the set of architectures to be included in a single architecture collection that requires governance oversight.
- b) Identify the set of existing architectures or reference architectures which may have direct applicability to the architecture collection and can be used as guiding oversight.

- c) Identify, define or establish architecture-related standards and policies driven by organizational policies, strategy and vision.
- d) Establish roles, responsibilities, accountabilities, authorities and organizational structures to support the Architecture Governance process and reporting requirements and alignment with other relevant architectures.
- e) Establish the architecture governance organizational structure that is consistent with the defined roles, authorities, responsibilities and accountabilities.
- f) Establish guiding principles, policies and directives for performing architecture governance.

NOTE 1 These guiding principles, policies and directives delineate the procedures and work instructions to be performed by those who execute the governance activities. This is to ensure the governance directives are transparent and consistently followed. Sometimes a secretariat is used to administer these procedures and work instructions and to ensure they are properly followed.

- g) Identify, define or establish architecture-related reusable elements from the set of existing architectures or reference architectures.

NOTE 2 These reusable elements can be governance directives, guiding principles, work instructions, governance organizational structure or procedures.

- h) Identify knowledge assets (historical information, successful outcomes) that can aid in governance of the architecture collection.
- i) Establish governance forums to carry out architecture governance work instructions in accordance with guiding principles.
- j) Ensure responsibilities are appropriately assigned by utilizing a responsibility matrix.
- k) Define procedures for identifying, managing, auditing and disseminating information related to architecture governance decisions.
  - 1) Link these procedures to architecture collection strategies, policies and objectives.
  - 2) Map these procedures to resources and constraints to support strategy, planning and decision making.
- l) Plan the architecture governance effort using the Project Planning process in ISO/IEC/IEEE 15288 as a guide.

NOTE 3 ISO 21500 and ISO 21505 can be used as guidance for the planning effort.

- 1) Establish the scope of the architecture governance effort.
- 2) Establish metrics for the architecture governance effort.

NOTE 4 [Annex B](#) contains information about defining metrics for architecture processes. See [B.2.1](#) for example metrics for architecture governance.

- 3) Identify the data and information needed for the architecture governance effort.
- 4) Obtain access to enablers needed for the architecture governance effort.

NOTE 5 The enablers can be obtained from the Architecture Enablement process. When enablers are obtained from other sources, these can become candidate enablers for use by other architecture efforts through the Architecture Enablement process.

EXAMPLE 1 Architecture governance enablers could be tools and methods that support accountability gap analysis, implementation gap analysis, objectives gap analysis and capacity gap analysis.

- 5) Specify the work products and their outlines to be produced through performance of this process.
- 6) Identify and define architecture governance work elements and associated resources.

EXAMPLE 2 Architecture governance work elements could be policy documents, strategic objectives, vision, mission, values of the architecture governance board.

- 7) Develop architecture governance schedule and define associated milestones.
- m) Produce an architecture governance plan that contains the planning information.
- n) Obtain necessary approvals and funding for the plan.
- o) Collect the data and information needed for the architecture governance effort.

#### **6.4.2 Monitor, assess and control the architecture governance activities**

- a) Monitor and assess how changes to industry, governmental directives and guidance, and technological change could impact the architectures being governed.

NOTE 1 The evolution of disruptive technologies can impact the architectures being governed. E.g. Frequency Slice Processor as a new technology impacts the control system architectures.

- b) Monitor and assess how changes to industry, government and technology could impact the governing directives and guidance.
- c) Monitor and assess how changes to industry, government and technology regulations and policies could impact the governing directives and guidance.
- d) Monitor and assess metrics for the architecture governance effort.
- e) Identify and assess risks and opportunities associated with the architecture governance effort.
- f) Ensure that other processes are properly using architecture governance directives and guidance.

NOTE 2 See [C.1](#) for recommended interactions with system life cycle processes.

NOTE 3 See [C.2](#) for recommended interactions with enterprise life cycle processes.

- g) Report architecture governance activity plans and status in accordance with reporting requirements.
- h) Assess and control the architecture governance effort using the Project Assessment and Control process in ISO/IEC/IEEE 15288 as a guide.

NOTE 4 ISO 21500 and ISO 21505 can be used as guidance for assessment and control.

- i) Manage decisions about the architecture collection associated with architecture governance using the Decision Management process in ISO/IEC/IEEE 15288 as a guide.

NOTE 5 ISO 21505 can be used as guidance for strategic decision making.

- j) Manage risks associated with architecture governance using the Risk Management process in ISO/IEC/IEEE 15288 as a guide.

NOTE 6 ISO 31000, ISO 21500 and ISO 21505 can be used as guidance for risk management.

- k) Establish configuration control management mechanism to govern the changes in the architecture collection.

#### **6.4.3 Establish architecture collection objectives**

- a) Examine current and future business needs.

© ISO/IEC 2019 – All rights reserved

© IEEE 2019 – All rights reserved

Authorized licensed use limited to: Tampere University. Downloaded on March 02,2025 at 07:53:52 UTC from IEEE Xplore. Restrictions apply.

**NOTE 1** This task is an examination of information that comes from the Business or Mission Analysis process in ISO/IEC/IEEE 15288.

- 1) Examine current and future enterprise objectives that shall be achieved, such as maintaining competitive advantage.
- 2) Examine and evaluate the current and future enterprise vision, including strategies, proposals and supply arrangements (whether internal, external, or both) in support of the enterprise objectives.

**NOTE 2** The examination can consider the external or internal pressures acting upon the enterprise, such as business change, technological change, economic and social trends and political influences.

- 3) Examine specific objectives of the strategies and proposals that are being evaluated.
- b) Examine current and future mission needs for those missions supported by the enterprise.

**NOTE 3** This task is examination of information that comes from the Business or Mission Analysis process in ISO/IEC/IEEE 15288.

- 1) Examine current and future mission objectives that shall be achieved.
- 2) Examine and evaluate possible future use of entities being architected, including strategies, proposals and supply arrangements (whether internal, external or both) in support of the mission objectives.

**NOTE 4** The examination can consider the external or internal pressures acting upon the enterprise, such as technological change, economic and social trends, and political influences.

- 3) Examine specific objectives of the strategies and proposals that are being evaluated.
- c) Establish architecture strategy by making the relevant decisions about the architectures in the architecture collection.
  - 1) Discover, develop, define and evaluate the overall goals of the architecture collection as a whole.
    - i) Identify work to be performed to achieve these goals.
    - ii) Create a configuration of resources which are necessary ingredients in meeting these goals.
    - iii) Establish means of measuring governance effectiveness.
  - 2) Define architecture governance policies related to the architecture collection.
  - 3) Define a set of governance principles that apply to the architecture collection.
  - 4) Determine adherence to objectives based on governance compliance criteria and strategies.
  - 5) Establish management criteria for control of architecture governance practices, dispensations and compliance.
- d) Identify the architecture collection objectives to be pursued and desired levels of achievement for each.
- e) Establish procedures to investigate and capture how the architecture collection objectives can be changed when necessary.

#### **6.4.4 Make architecture governance decisions**

- a) Establish and issue governance direction in the form of directives and guidance for architectures in the architecture collection.
  - 1) Develop and issue governance directive(s) that drive the appropriate evolution of architectures in the architecture collection.

NOTE 1 Governance directives could be about specific architecture patterns, reference architectures, standards, frameworks, principles, tools, methods, etc. that would need to be applied to the architectures or used when architecting.

- 2) Implement a governance framework that supports this governance directive so as to define conceptual and organizational structures and provide a structured decision-making approach.

NOTE 2 The governance framework typically comprises a set of controls over the creation and monitoring of all architecture process components, work products and outcomes.

NOTE 3 Absolute control may not be possible in cases like: SOS situations, federated architectures, consortium-based architectures, public-private partnerships, etc.

- 3) Establish a decision-making mechanism that minimizes or avoids potential conflicts of interests with escalation in the organization if the problems cannot be properly addressed by architecture governance.
  - 4) Develop and issue directives on mandated standards to be used during execution of architecture processes.
- b) Assign responsibility for, and direct preparation and implementation of, directives, standards and policies that set the direction for architecting efforts.
  - c) Analyze architectural dependencies to exploit the associations amongst the architectures in the architecture collection.
- NOTE 4 Dependencies could be capabilities, resources, services, reference architectures, assumptions, principles, vocabulary, ontologies and so on.
- d) Make strategic decisions within the scope of architecture governance responsibilities and authorities using the Decision Management process in ISO/IEC/IEEE 15288 as a guide.
- NOTE 5 ISO 21505 can be used as guidance for strategic decision making.
- e) Elevate other decisions to the appropriate enterprise decision forum.
  - f) Review and communicate the decisions.

#### **6.4.5 Monitor and assess compliance with governance directives and guidance**

- a) Monitor the governance of the architecture collection by utilizing appropriate means.

NOTE 1 In order to see how well the architectures are helping to achieve the enterprise goals and objectives, architecture governance can monitor metrics from the development and use of actual entities based on the architecture.

NOTE 2 A dashboard is a common way of doing this.

- b) Ensure that the architectures comply with external obligations and internal work practices.

NOTE 3 External obligations can vary between countries where the instances of architecture entity and its life cycle activities can occur.

- c) Monitor management plans and status and assess compliance with governance directives and guidance for the architecture collection.
- d) Allow the appropriate authority to access directly an entity constructed according to a design based upon an architecture description for an architecture to verify that it conforms to obligations established by governance directives.

NOTE 4 This access is to enable verification that a system, an enterprise, a software item, a service or other kind of architecture entity conforms to obligations established by governance directives. Merely checking the architecture for conformance could be insufficient since the architectural features could lack proper translation into design or operation of the entity.

- e) Check compliance with external obligations to be applied and standards that affect architecture governance.

#### 6.4.6 Review implementation of governance directives and guidance

This activity includes the tasks necessary for closing out one iteration of the architecture governance endeavor by the organization driving the endeavor.

- a) Review all information to assert that the architecture governance work is complete and that the architecture collection objectives have been met.
- b) Establish procedures to capture, investigate and resolve the various reasons for non-implementation of governance directives and guidance.

NOTE Document and retain the investigation and the rationale to support later reviews. Resolutions may require modification of the directives and guidance.

- c) Collect lessons learned regarding architecture governance as reported from the other architecture processes and make these lessons available to future projects.
- d) Record lessons learned and communicate to all relevant stakeholders.
  - 1) Contribute to best practices for architecture governance.
  - 2) Scrutinize effectiveness of governance directives and guidance.
- e) Examine the effectiveness of the governance policies and initiate steps to revise the policies as necessary.
- f) Identify and select the changes to be made to the architecture governance directives and guidance.
- g) Incorporate the changes into the architecture governance directives and guidance.

### 6.5 Work products

The following work products shall be produced:

- architecture governance plan,
- architecture governance directives and guidance,
- architecture governance compliance status report, and
- architecture collection objectives.

## 7 Architecture Management process

### 7.1 Purpose

The purpose of the Architecture Management process is to implement architecture governance directives to achieve architecture collection objectives in a timely, efficient and effective manner.

NOTE See [Annex G](#) for additional information regarding architecture governance and management, and the distinction between these related but separate processes.

### 7.2 Outcomes

As a result of the successful implementation of the Architecture Management process:

- a) Architecting activities are in alignment with architecture governance directives and guidance.

- b) Architecting activities are identified and prioritized, maximizing value to relevant stakeholders given available resources.
- c) Inputs, including stakeholder inputs, and resources for architecting are identified, made available and utilized effectively.
- d) Architecture collection objectives are monitored and assessed for achievement.

NOTE Architecture governance outcomes are applicable across the organization while architecture management outcomes are applicable across the entire collection of architectures.

### **7.3 Implementation**

The organization shall implement the activities in [7.4](#) (numbered as 7.4.N) in accordance with applicable organization policies and procedures with respect to the Architecture Management process. The activities may be performed in any order that is deemed appropriate. The organization should implement the relevant tasks (identified as list items under each 7.4.N activity) as appropriate to the situation.

Architecture management work products should be stored in the architecture repository for future reference and audit. The repository should be used to facilitate widespread access, enable auditing and encourage future reuse.

### **7.4 Activities and tasks**

#### **7.4.1 Prepare for and plan the architecture management effort**

- a) Identify and categorize architectures into an architecture collection based on their relevance, respective purpose and scope, time validity and alignment with each other.
- b) Identify architectures, used and to be worked, in the architecture collection that require architecture management oversight.
- c) Develop a charter for management of the architecture collection.

NOTE 1 ISO 21500 and ISO 21505 can be used as guidance for developing the charter.

- 1) Identify organizational assets (people, resources, processes) that influence management of the architecture collection.
- 2) Identify knowledge assets (historical information, issues and defect resolutions, successful outcomes) that can aid in management of the architecture collection.
- 3) Identify internal and external factors and criteria that influence management of the architecture collection.
- 4) Identify in-scope and out-of-scope items that influence management of the architecture collection.
- 5) Identify skill gap of people involved in architecture management and establish training and improvement plans to address the gap.
- 6) Develop a statement of work in alignment with the charter for management of the architecture collection.

- 7) Define measurable architecture management objectives and related success criteria for the architecture collection.
- d) Develop the architecture management organizational structure.
  - 1) Identify the necessary managerial roles, responsibilities and authorities that are concerned with or involved in architecture management.
  - 2) Define an architecture management control hierarchy corresponding to the identified roles and responsibilities.
  - 3) Ensure proper delegation of management responsibilities in the architecture management control hierarchy.
  - 4) Ensure proper allocation of roles to identified role players in the architecture management hierarchy.
- e) Plan the effort for managing the architecture collection using the Project Planning process in ISO/IEC/IEEE 15288 as a guide.

NOTE 2 ISO 21500 and ISO 21505 can be used as guidance for planning of the management effort.

- 1) Establish the scope of the architecture management effort.
  - 2) Identify metrics and metrics data collection strategy for the architecture management effort.
- NOTE 3 [Annex B](#) contains information about defining metrics for architecture processes. See [B.2.2](#) for information on architecture metrics.
- 3) Collect the data and information needed for the architecture management effort.
  - 4) Obtain access to enablers needed for the architecture management effort.

NOTE 4 The enablers will usually be obtained from the Architecture Enablement process. When enablers are obtained from other sources, these can become candidate enablers for use by other projects through the Architecture Enablement process.

EXAMPLE 1 Architecture management enablers could be tools, methods and procedures for scheduling, budgeting, and cost analysis.

- 5) Specify the work products and their outlines to be produced through performance of this process.
- 6) Identify and define architecture management work elements and associated resources.

EXAMPLE 2 Architecture management work elements could be work breakdown structure, risk breakdown structure.

- 7) Develop architecture management schedule and define associated milestones.
- 8) Produce an architecture management plan that contains the planning information.
- 9) Obtain necessary approvals and funding for the architecture management plan.

#### **7.4.2 Monitor, assess and control the architecture management activities**

- a) Report architecture management activity plans and status.
- b) Monitor and assess whether architecture governance directives and guidance are being followed.
- c) Monitor and assess metrics for the architecture management effort.
- d) Identify and assess risks and opportunities associated with the architecture management effort.

- e) Ensure that other processes are properly using architecture management work instructions.
  - NOTE 1 See [C.1](#) for recommended interactions with system life cycle processes.
  - NOTE 2 See [C.2](#) for recommended interactions with enterprise life cycle processes.
- f) Report architecture management activity plans and status in accordance with reporting requirements.
- g) Assess and control the architecture management effort using the Project Assessment and Control process in ISO/IEC/IEEE 15288 as a guide.
  - NOTE 3 ISO 21500 and ISO 21505 can be used as guidance for assessment and control.
- h) Manage decisions about the architecture collection associated with architecture management using the Decision Management process in ISO/IEC/IEEE 15288 as a guide.
  - NOTE 4 ISO 21505 can be used as guidance for strategic decision making.
- i) Manage risks associated with architecture management using the Risk Management process in ISO/IEC/IEEE 15288 as a guide.
  - NOTE 5 ISO 31000, ISO 21500 and ISO 21505 can be used as guidance for risk management.

#### **7.4.3 Develop architecture management approach**

- a) Identify architecture management approaches, constraints, methods, tools and techniques according to architecture governance policies, directives and guidance.
  - NOTE 1 Management by objectives, management by measurement and management by policies are examples of management approaches.
  - NOTE 2 [Annex H](#) provides information on mapping of processes to architecture frameworks. TOGAF, DoDAF and NAF are examples of architecture frameworks that contain architecture management methods.
- b) Develop architecture management plans in accordance with governance directions for the architecture collection.
  - 1) Identify architecture collection requirements and other concerns.
  - 2) Define architecture management scope statement (scope description, acceptance criteria, architecture deliverables, exclusions, constraints, assumptions).
  - 3) Establish specific architecture management goals that address the architecture collection objectives and specify the reasons for their sufficiency.
  - 4) Prioritize the goals in terms of their importance in achieving architecture collection objectives.
  - 5) Create a work definition that provides a common framework for the overall planning and control of the architecture collection.
  - 6) Establish the resources necessary for performing the architecture management work elements and assign responsibility and authority to the management hierarchy.
  - 7) Define management measures that allow assessment of compliance with the architecture management goals.
- c) Ensure the architecture collection is established and maintained in accordance with the relevant architecture management plans.

- d) Develop the architecture management schedule in accordance with the architecture management plan(s).
  - 1) Develop the schedule for the activities identified in the work definition and define precise and measurable milestones.
  - 2) Define the activities, their dependencies and resources necessary for creating and managing the architecture collection.
  - 3) Estimate the duration of each of the activities and include them as part of the schedule.
- e) Develop a budget and associated justification to manage the architecture collection.
- f) Secure adequate types and quantities of funding and resources to properly implement the planned architecture activities.

NOTE 3 If it is not within the remit of the Architecture Management process to provide funding and resources for the planned architecture activities, then this process could champion these needs to the organizational entity that can.

- g) Adapt the architecture management plan using an agile approach according to new information.
  - 1) Adjust management activities, schedule, directions and goals in response to new information.
  - 2) Prepare adaptive management actions that respond to new problems or opportunities.
  - 3) Apply new knowledge, insights and technologies that contribute to achieving architecture management objectives.

#### 7.4.4 Perform management of the architecture collection

Architecture management is about management of the architecture collection.

NOTE 1 Often, it is necessary to manage individual architectures in the architecture collection and could require individual directions and additional tasks to specific architecture endeavors.

- a) Establish and issue management direction in the form of documented instructions and guidance for the architecture collection.
- b) Establish and issue management instructions and guidance on mandated standards and policies to be used during execution of architecture processes.
- c) Assign resources to all the identified roles in accordance with the sequence of tasks that needs to be performed.
- d) Activate the necessary tracking systems which can capture work performance information and use it to control architecture development.
- e) Perform the tasks defined in the architecture collection management plan to achieve the architecture collection objectives.
- f) Assess architecture management measures, work performance information, resource utilization, probable risks and new opportunities and manage changes to improve work performance.
- g) Provide relevant information to all stakeholders as outlined in the architecture management plan.
- h) Periodically confirm that work performance results conform to the architecture collection requirements including deviations and dispensations.

NOTE 2 An appropriate periodicity of confirmation can be inversely proportional to the magnitude of effort involved in performing the tasks outlined in the architecture management plan.

- i) Set strategic directions for architectures in the architecture collection that addresses policy decisions.
- j) Establish individual plans for the development or revision of architectures in the architecture collection.
- k) Manage decisions about the architectures in the architecture collection using the Decision Management process in ISO/IEC/IEEE 15288 as a guide.

NOTE 3 ISO 21505 can be used as guidance for strategic decision making

- l) Manage risks associated with architecture management using the Risk Management process in ISO/IEC/IEEE 15288 as a guide.

NOTE 4 ISO 31000, ISO 21500 and ISO 21505 can be used as guidance for risk management.

- m) Manage changes to the architectures using the Configuration Management process in ISO/IEC/IEEE 15288 as a guide.

#### **7.4.5 Monitor architecting effectiveness**

- a) Monitor and control work performance by tracking, reviewing and regulating the progress to meet the management objectives based on the architecture management plan.

- 1) Monitor work performance to identify variances from the architecture management plan.
- 2) Monitor management issues and recommend preventive action in anticipation of possible problems.
- 3) Monitor status reports and recommend corrective action when necessary.
- 4) Monitor management plan execution progress against the schedule estimates.
- 5) Monitor health of the architecture collection using the identified management measures and identify any areas that require additional attention.
- 6) Monitor the status of the architecture collection scope and manage changes to the scope baseline.
- 7) Monitor the status of the architecture collection schedule and manage changes to the schedule baseline.

- b) Monitor effectiveness of the architecture management effort.

NOTE 1 Plan, schedule, budget, approach, outcome and performance are typical monitored elements.

- c) Assess actual outcomes against planned targets and make corrective actions when necessary.
- d) Collect and communicate performance information to all relevant stakeholders at periodic intervals.

- 1) Maintain accurate and timely information concerning the architecture collection.
  - 2) Assess whether the architecture management objectives are being reached.
  - 3) Provide forecasts to update current schedule information.
- e) Define quality assurance actions and audits that confirm execution of the architecture management plans using the Quality Assurance process in ISO/IEC/IEEE 15288 as a guide.
  - f) Report architecture management activity plans and status in accordance with reporting requirements.

- g) Maintain a list of known needs and gaps and periodically perform analysis and prioritization to select architecture activities to be performed.
- h) Assess and control the architecture management effort in accordance with the Project Assessment and Control process in ISO/IEC/IEEE 15288.

NOTE 2 ISO 21500 and ISO 21505 can be used as guidance for assessment and control.

#### **7.4.6 Prepare for completion of the architecture management plan**

This activity includes the tasks necessary for closing out one iteration of the architecture management endeavor.

NOTE This could be because the architecture life-cycle has come to a closure and a new architecture is planned to be taken up.

- a) Close execution of the architecture management plan when appropriate.
- b) Review all information to determine whether any mid-stream changes are necessary and incorporate the changes into the architecture management plan, schedule, budget and the management approach.
- c) Review all information to determine whether the architecture management work is complete and the architecture collection objectives have been met, so that architecture management changes may be closed, or there are significant gaps that necessitate and drive changes to the architecture management plan, schedule, budget and approach.
- d) Assess metrics for the architecture management effort.
- e) Check compliance with standards that affect the architectures in the architecture collection.
- f) Establish procedures to investigate and capture the various reasons for management actions taken as part of architecture management.
- g) Record lessons learned and communicate to all relevant stakeholders.
  - 1) Contribute to best practices for architecture management.
  - 2) Scrutinize effectiveness of management approaches that were adopted in order to address the architecture management problem.
- h) Identify changes to be made to the architecture management plan, schedule, budget and the management approach.
- i) Select the changes to be made in the next iteration of the architecture management plan, schedule, budget and the management approach.
- j) Analyze the impact of the selected changes on the architecture management plan, schedule, budget and the management approach and adapt accordingly.
- k) Incorporate the changes into the architecture management plan, schedule, budget and the management approach.

#### **7.5 Work products**

The following work products shall be produced:

- architecture management plan,
- architecture management status report,
- architecture management work instructions and guidance,

- architecture management charter,
- execution plan, and
- execution status report.

## 8 Architecture Conceptualization process

### 8.1 Purpose

The purpose of the Architecture Conceptualization process is to characterize the problem space and determine suitable solutions that address stakeholder concerns, achieve architecture objectives and meet relevant requirements.

**NOTE 1** The name of the process being “conceptualization” does not mean that the results are necessarily at the “conceptual” level, or consist of a set of conceptual models and views. The results could include a “logical” architecture or a “physical” architecture, depending on the nature of the situation.

**NOTE 2** Identification of solutions can occur in any of the architecture processes or in any of the life cycle processes. Solutions are not limited to just the Architecture Conceptualization process. However, conceptualization is where there is a special focus on identifying solutions, but with also an emphasis on fully understanding the complete problem space. This also entails the definition and establishment of architecture objectives, as well as negotiation with key stakeholders on prioritization of their concerns.

### 8.2 Outcomes

As a result of the successful implementation of the Architecture Conceptualization process:

- a) The problem being addressed is clearly defined and understood.
- b) Architecture objectives that address the key stakeholder concerns are established.
- c) The architecture’s key concepts and properties, and the principles guiding its formulation, application and evolution, are clearly defined.
- d) The architecture is clearly conceived and key tradeoffs are understood with respect to the problem being addressed and the relevant stakeholder concerns.
- e) Tradeoffs among architecture objectives and feasibility limitations are identified, and the architecture objectives targeted to be addressed by the architecture are clearly specified.
- f) The candidate solutions for the problem are clearly defined and understood.

### 8.3 Implementation

The organization shall implement the activities in [8.4](#) (numbered as 8.4.N) in accordance with applicable organization policies and procedures with respect to the Architecture Conceptualization process. The activities may be performed in any order that is deemed appropriate. The organization should implement the relevant tasks (identified as list items under each 8.4.N activity) as appropriate to the situation.

Architecture conceptualization work products should be stored in the architecture repository for future reference and audit. The repository should be used to facilitate widespread access, enable auditing and encourage future reuse.

**NOTE** Below are a few guidelines that can assist in the implementation of this process.

- a) Customers and users can have a special role to play in the architecture development effort. They are sometimes paying for the architecting effort and usually have to live with the resulting solution for years to come. They also have a key role in helping to specify the architecture objectives and the evaluation criteria.

- b) During early stages, it is sometimes important to be agile and quick in conceptualizing many alternative architectures. Some of these early architecture descriptions can be little more than sketches. After doing several quick rounds of evaluation and conceptualization, there can then be a smaller number of viable architectures that are worth capturing in a more complete form and storing in the repository for later use. A more complete form of architecture description would be developed in the Architecture Elaboration process.
- c) An important aspect of architecture conceptualization is to allow the analysts, project participants and various stakeholders to understand the tradespace of potential options that can meet the objectives. (See [8.4.6](#) for more information on the nature of a tradespace.) Typically, a varied and distinctive mix of architectural solution concepts can be considered in order to allow the tradeoff between various constraints and concerns. In other words, in dealing with a complex problem space, an extensive exploration of the solution space is necessary as part of the conceptualization process in order to eventually select the most appropriate solution.
- d) The candidate architecture(s) can be derived from or can be already existing architecture(s) in the architecture collection under architecture management oversight or from other sources.
- e) The set of views generated for the architecture(s) do not have to be complete at this point since a minimal set of views could be sufficient to determine whether the proposed architectures are fit for purpose. The Architecture Elaboration process can fill in the necessary details where appropriate.
- f) There could be existing requirements to be considered as drivers of the candidate architecture(s). These requirements can sometimes come from execution of the requirements processes described in ISO/IEC/IEEE 15288, namely the Stakeholder Needs and Requirements Definition process and the System Requirements Definition process. These requirements can also come through the Acquisition process. These requirements could include relevant mandates and imperatives, including relevant policies, regulations and standards.
- g) In addition to requirements, another driver of the candidate architecture(s) could be needs, wants and expectations identified during execution of the Business or Mission Analysis process in ISO/IEC/IEEE 15288. Needs can either be designated as objectives or stated as requirements.
- h) Additionally, the situation context can also serve as a driver for synthesizing solutions and formulating candidate architectures. The source information about the situation context might not be stated as requirements. It can come from the assumptions in the domain or from best-practices in the domain.

## 8.4 Activities and tasks

### 8.4.1 Prepare for and plan the architecture conceptualization effort

- a) Identify the general nature of the problem area(s) that needs to be addressed.
- b) Define the expected purpose, scope, objectives and level of detail of the architecture conceptualization effort.
- c) Establish the architecture description framework(s) to be used throughout the architecting effort for developing the views and models of the architecture based on expressed stakeholder desires or analysis of stakeholder needs.
- d) Define one or more architecture conceptualization approaches that are consistent with the architecture governance and management directions and are consistent with the purpose, scope and objectives of this effort.

NOTE 1 It could be the case that more than one approach is needed to conceptualize different aspects of the architecture. For example, mathematical algorithms can be employed to generate architecture concepts for dealing with various data collection problems while employing an “expert panel” for generating concepts for dealing with security issues.

NOTE 2 There are various architecting strategies and approaches to consider when planning how to conceptualize architectures. Several examples of these are described in [E.3](#). There are various kinds, views and styles of architectures that can be considered as delineated in [E.4](#).

- e) Select or develop the required architecture conceptualization techniques, methods and tools.

- f) Plan the architecture conceptualization effort using the Project Planning process in ISO/IEC/IEEE 15288 as a guide.

NOTE 3 ISO 21500 and ISO 21505 are also useful references for planning of the architecture conceptualization effort.

- 1) Document the purpose, scope and objectives of the architecture conceptualization effort.
- 2) Establish metrics for the architecture conceptualization effort that enable a determination of when the architecture conceptualization task is complete.
- 3) Identify the data and information needed for the architecture conceptualization effort.
- 4) Obtain access to enablers needed for the architecture conceptualization effort.

NOTE 4 The enablers will usually be obtained from the Architecture Enablement process. When enablers are obtained from other sources, these can become candidate enablers for use by other projects through the Architecture Enablement process.

EXAMPLE Architecture conceptualization enablers could be tools, methods and procedures for brainstorming, collaboration, conceptual modeling, etc.

- 5) Identify and define architecture conceptualization work elements and associated resources.
  - 6) Specify the work products and their outlines to be produced through performance of this process.
  - 7) Develop architecture conceptualization schedule and define associated milestones.
- g) Produce an architecture conceptualization plan that contains the planning information.
- h) Obtain necessary approvals, resources and funding for the plan.
- i) Collect the data and information needed for the architecture conceptualization effort.
- j) Ensure personnel are trained in the use of identified techniques, methods and tools.
- k) Ensure personnel have necessary and appropriate access to relevant architecture work products, data and information.

#### **8.4.2 Monitor, assess and control the architecture conceptualization activities**

- a) Report architecture conceptualization activity plans and status.
- b) Monitor and assess whether architecture governance directives and guidance are being followed.
- c) Monitor and assess whether architecture management instructions and guidance are being followed.
- d) Monitor and assess metrics for the architecture conceptualization effort.
- e) Identify and assess risks and opportunities associated with the architecture conceptualization effort.
- f) Maintain traceability of architecture conceptualization results to the source material used during the process.
- g) Ensure that other processes are properly using architecture conceptualization products.

NOTE 1 See [C.1](#) for recommended interactions with system life cycle processes.

NOTE 2 See [C.2](#) for recommended interactions with enterprise life cycle processes.

- h) Implement corrective actions where necessary to modify the work plan or realign the work with the plan.

- i) Assess and control the architecture conceptualization effort in accordance with the Project Assessment and Control process in ISO/IEC/IEEE 15288.

NOTE 3 ISO 21500 and ISO 21505 are also useful references for assessment and control.

#### **8.4.3 Characterize problem space**

- a) Identify the potential problem area(s) that needs to be addressed.

NOTE 1 The “problem” could actually be an expectation for improvement without necessarily being considered to be a problem per se. Throughout this process the word “problem” is used to serve as the term used to indicate the thing that is being addressed by some architectural solution. In all cases, such architecture development leads to an improvement in the situation which can be expressed using the notion of problem space.

- b) Identify current and projected situation(s) in the problem space.

NOTE 2 The problem space and the solution space are abstract. In simple terms, the problem space relates to the world of end-user challenges and motivations, which is usually captured in the form of validated problem statements and specified needs. The solution space focuses on the application of products, services and technologies that will adequately address end-user problems and needs. A systemic approach is important to apply in both the solution space and the problem space to ensure that viable alternatives are not overlooked and that the most effective solution can more readily be identified and applied.

- c) Identify problems and opportunities in the current and projected situation(s).

NOTE 3 The concept of “problem” is used here in the same sense used in ISO/IEC/IEEE 15288: “difficulty, uncertainty, or otherwise realized and undesirable event, set of events, condition, or situation that requires investigation and corrective action.” A problem space is a mental representation of a problem (or set of problems) that contains knowledge of the initial state and the goal state of the problem(s) as well as possible intermediate states to be searched in order to link up the beginning and the end of the task.

- d) Identify relevant aspects of the identified situation(s).

NOTE 4 An aspect is a way in which a thing, idea, situation or domain can be observed, considered or regarded. A clear understanding of the different aspects involved can help ensure that the full nature of the problems and opportunities are well understood. It is a common error not to consider these things with respect to these different aspects. An aspect is not the same as a concern although there might be some concerns identified when considering something from a particular aspect. For example, the logistics aspect can be used to help understand and address concerns such as safety, reliability, maintainability, availability, supportability, etc.

EXAMPLE 1 PESTEL is a common approach used in strategic planning methodologies – political, economic, social, technological, environmental, legal aspects of a situation.

EXAMPLE 2 Other aspects that can be used as the basis for problem space characterization: demographic, ethical, logistical, recreational, constructional, mission, business, logical, mathematical, strategic, operational, tactical, philosophical, ontological, axiological, epistemological, constructivist, positivist.

- e) Examine current and future business and mission needs with respect to these problems and opportunities.

NOTE 5 This task entails the examination of information that could potentially come from execution of the Business or Mission Analysis process described in ISO/IEC/IEEE 15288. The findings from this problem space analysis could be useful as feedback to the Business or Mission Analysis process.

NOTE 6 Refer to the Business or Mission Analysis process in ISO/IEC/IEEE 15288 for more information on the distinction between business needs and mission needs.

- f) Identify stakeholders and their concerns corresponding to each of these problems and opportunities.

NOTE 7 The stakeholders could be those concerned about the architecture itself or those concerned about the systems or other entities the architecture is dealing with. Therefore, architecture concerns can be different from entity concerns and vice versa.

- g) Identify and analyze any formal and informal requirements that apply to this situation and document which requirements are relevant to current effort and how they would flow to the evaluation criteria.
- h) Identify value, if any, that stakeholders will receive when the stakeholder concerns are addressed.
- i) Identify and prioritize quality measures associated with these stakeholder concerns that can be used as the basis for development and evaluation of the architecture.

NOTE 8 Quality attributes can be related to any kind of entity. Quality attributes for software and computer systems have been specified in ISO/IEC 25010, otherwise referred to as SQuaRE (systems and software quality requirements and evaluation). Quality measures are not only a function of the problem space but could also derive from things in the solution space. Quality measures relative to problem space can be for example like “ends objectives,” whereas quality measures relative to solution space can be for example like “means objectives.”

- j) Identify and characterize how the problems and opportunities affect different stakeholders and their priorities in addressing these items.
- k) Identify and characterize complexities of each problem and opportunity, its cause and effect, and how it is being addressed currently in each of the identified situations.
- l) Identify and characterize the cause and effect relationships for the identified problem(s).

EXAMPLE 3 Inference networks can be used to characterize cause-effect relationships and can take various forms depending on the situation such as problem tree, influence diagram, Bayesian network, causal loop diagrams, fishbone diagrams, etc.

- m) Formulate a clear statement of the problem(s), including risks, opportunities which could be addressed and constraints which are to be satisfied in the solutions.
- n) Produce a problem space definition report.

#### **8.4.4 Establish architecture objectives and critical success criteria**

- a) Use identified problems and related risks and constraints as the basis for gathering, analyzing and negotiating requirements.
- b) Classify requirements by stakeholder needs to help identify who is interested in each problem and opportunity.
- c) Determine boundary conditions, root causes, drivers and relevant scenarios for each identified problem and opportunity.
- d) Determine gaps or shortfalls of current or planned solutions in addressing the problem and opportunity.
- e) Identify relevant assumptions, degrees of freedom, constraints, conditions and challenges.
- f) Identify and define architecture objectives that address the problem(s) and opportunities with respect to the stakeholder concerns, requirements or quality attributes, or that take advantage of identified opportunities.

EXAMPLE A generic example of an objective to take advantage of an opportunity is for the architecture of interest to support reuse of an entity across various technologies, protocols, platforms, operational venues, market segments, etc.

- g) Develop a quality model that captures the relevant quality measures and defines the relationships between the measures.

NOTE 1 There are generic quality models such as in ISO/IEC 25010 that can be used to help inform development of the specific quality model in this activity. The quality model can show how all the measures are related to each other and can sometimes indicate the relative priorities for the various quality measures in the model. (See [E.6](#) for further details).

- h) Define critical success criteria that can be used to assess the extent to which the problems(s) are resolved and to inform exploration and selection of alternatives.

NOTE 2 These success criteria are directly related to aspects of the problem(s) in the problem space, and are primarily concerned with achieving desired “ends”. As such, these are sometimes called ends objectives. In some domains, the ends objectives are related directly to desired “effects” and are the basis for so-called effects-based analysis. These criteria are different but related to the “means objectives” to be defined during solution space analysis.

NOTE 3 Ends objectives are sometimes called “fundamental” objectives. Information on the distinction between ends objectives and means objectives, and how to apply them, can be found in the Bibliography. See Handbook of Decision Analysis, Parnell 2013<sup>[43]</sup> and OMG Business Motivation Model<sup>[34]</sup> for example.

- i) If relevant, include relative weighting of success criteria, using a normalized scale, to help distinguish more critical items from less critical items.
- j) If necessary, translate the critical success criteria into evaluation criteria that can be used during architecture evaluation and in support of decision making.

#### 8.4.5 Synthesize potential solution(s) in the solution space

- a) Define more specific objectives, where necessary, to be achieved in addressing the problems and opportunities and relate these to the established architecture objectives and success criteria.
- b) Identify existing or previous solutions to determine if these can be used as potential solution(s) for the current or projected situation under examination.

NOTE 1 Solutions can be found in current or future architectures and can sometimes address more than one problem or opportunity. The collection of architectures is sometimes called an “architecture landscape”.

- c) Identify problem mitigation strategies that can achieve the specific objectives and serve as potential solution(s).
  - 1) Consult with subject matter experts for relevant technologies, design patterns and solution approaches.
  - 2) Perform technology scan for relevant technologies.
  - 3) Perform problem/solution pattern scan for relevant solutions to similar problems.
  - 4) Perform natural system metaphor scan for possible naturally occurring solutions to similar problems.
  - 5) Perform risk assessment of the identified strategies.

NOTE 2 The architecting strategies and approaches described in [Annex E](#) suggest solution approaches that could help in finding the correct problem mitigation strategies.

- 6) Identify relevant architectural patterns, heuristics, and tactics.
- NOTE 3 Domain architectures, reference architectures, and architectural patterns can be used in forming potential solutions or parts of solutions. Further details on these can be found in [Annex E](#).
- d) Review the resulting relationships between problem mitigation strategies and problem causes to assure the completeness of the potential solution(s).

NOTE 4 Strategies might apply to multiple objectives. It could be important to clearly specify the relationships between strategies and objectives to facilitate the examination of solution completeness.

- e) Formulate purpose statement(s) for each potential solution.

NOTE 5 There might be a different purpose statement for each solution since the solution might not be addressing the entire problem or all aspects of the problem.

- f) Characterize strengths, weaknesses and tradeoffs for each potential solution.

NOTE 6 The solution might not be addressing the entire problem or all aspects of the problem. Therefore, it is important to understand where the solutions fall short and the tradeoffs that are to be considered when choosing among alternative solutions.

- g) Identify needs, wants and expectations for each potential solution.

NOTE 7 Expectations are what someone regards as likely to happen, which might have nothing to do with what they need or want. Expectations can drive the solutions as much as what might be needed or wanted by a person.

- h) If needs, wants or expectations drive the solutions, then negotiate with those stakeholders to determine which of the needs, wants or expectations are to be translated into requirements on the solution and the relative priority of each.

- i) Determine consequences and obligations associated with the needs, wants and expectations, and translate these into requirements or objectives on each of the solutions.

NOTE 8 Since there can be more than one solution to the problem that can achieve the objectives defined above in task (a), this is where the objectives (or requirements) are established for each of those solutions.

- j) Identify relevant critical success factors and key performance indicators for each potential solution.

NOTE 9 Success factors could be different for each potential solution. They depend on the problem perspective on which the solution is based. Success factors are relative to the purpose for that particular solution. The proposed solutions might also be addressing different aspects/parts of the problem. Success factors and performance indicators can be based on the identified stakeholder concerns or quality attributes from problem space analysis.

#### **8.4.6 Characterize solutions and the tradespace**

NOTE 1 This activity characterizes the solutions and tradeoffs from the technical perspective, considering aspects such as ease of realization, compatibility with other solutions, trends in the technology space, etc. It is complementary to architecture evaluation, which examines the proposed solutions in terms of value to stakeholders and quality attributes delivered by each solution. The previous activity is focused on the "creation" or identification of potential solutions while this activity is focused on the characterization of those solutions. These can be done simultaneously, but often they are done by different people since some who are very creative and innovative might not be as analytical and as precise as needed during the characterization activity.

NOTE 2 Tradespace is the range and extent of parameters, properties and characteristics that are relevant in satisfying architecture objectives and stakeholder concerns. The tradespace often does not include a solution that will completely satisfy the architecture objectives and stakeholder concerns. Tradespace analysis is used for analyzing the relevant constraints, conditions and challenges involved in large complex problems with multiple stakeholders and multiple objectives. It involves the identification and understanding of tradeoffs involved in choosing between competing solutions. Tradeoffs can be within each solution and between solutions (and the problems/opportunities they are intended to address).

- a) Examine the context in which stakeholders perceive value and formulate value propositions for each potential solution.
- b) Identify strengths, weaknesses, opportunities and threats for each potential solution.
- c) Identify other important aspects related to each potential solution including, but not limited to, the following.
  - 1) Identify and characterize risks for each potential solution.
  - 2) Identify cost and schedule considerations for each potential proposed solution.

- 3) Identify areas for potential reuse of existing architecture elements and the risks associated with this reuse.
  - 4) Identify assumptions with respect to each potential solution.
  - 5) Identify additional problems that might be caused by each potential solution.
  - 6) Determine remaining gaps or shortfalls after implementing the proposed solutions.
- d) Harmonize elements of each potential solution to ensure that it can be realized in a coherent and cohesive manner.

NOTE 3 If architectural patterns or tactics have been employed (see [E.3.3](#) for further details) then such harmonization could be necessary to ensure they are properly applied to the situation at hand.

- e) Identify and characterize the tradeoffs involved in achieving the architecture objectives:
- 1) between and within proposed solutions,
  - 2) between these solutions and the status quo, and
  - 3) between proposed solutions and other possible solutions (i.e. those not being proposed).

NOTE 4 Tradeoffs can be within each solution and between solutions. There are also tradeoffs between proposed solutions and the status quo (which can be considered as one of the alternative “solutions” to be considered). Typical tradeoffs to consider are the following: cost vs performance, cost vs schedule, weight vs speed, accuracy vs timeliness, acquisition cost vs operating cost, ease of use vs security, flexibility vs predictability, agility vs robustness, risk vs reward, etc.

- f) Identify and characterize negative and positive influences and interactions between proposed solutions and pre-existing/planned solutions.

NOTE 5 The collection of tradeoff considerations, along with the influences and interactions between solutions, is called in this document the “tradespace”. More information on how to explore the tradespace can be found in the Bibliography. See for example, Trade-off Analytics, Parnell 2017<sup>[34]</sup>.

- g) Formulate a roadmap for implementing the proposed solution(s).

NOTE 6 The purpose of the roadmap is to help in identifying possible pathways to achieving the solution. Some of these pathways can reveal otherwise unanticipated difficulties that need to be considered when formulating candidate architectures in the next activity. It might also reveal some cases where the solution is not achievable in a timely or efficient manner.

- h) Define success criteria that can be used to assess the extent to which the proposed solution(s) address the specified problem(s) and to inform exploration and selection of alternatives.

NOTE 7 These success criteria are directly related to aspects of the solution(s) in the solution space, and are primarily concerned with the “means” by which a solution addresses the identified problems. As such, these are sometimes called means objectives. These criteria are different but related to the “ends objectives” defined during problem space analysis.

Ends objectives are sometimes called “fundamental” objectives. Information on the distinction between ends objectives and means objectives, and how to apply them, can be found in the Bibliography. See for example, Handbook of Decision Analysis, Parnell 2013<sup>[43]</sup> and OMG Business Motivation Model<sup>[34]</sup>.

- i) Identify issues and areas for improvement in the proposed solution(s).
- j) Establish and capture the desired functional and non-functional characteristics based on the identified solution(s) with respect to the purpose of each solution that corresponds to the stakeholder concerns, relevant requirements and constraints, and quality attributes identified during problem space analysis.

NOTE 8 Quality attributes can be related to any kind of entity. Quality attributes for software and computer systems have been specified in ISO/IEC 25010, otherwise referred to as SQuaRE (systems and software quality requirements and evaluation).

#### 8.4.7 Formulate candidate architecture(s)

- a) Identify the solution(s) to be architected.
- b) Devise structural, behavioral and organizational concepts and properties, as appropriate, that support the desired functional, non-functional and other (e.g. operational, business, environmental impact) characteristics.

EXAMPLE 1 These concepts and properties could be expressed in the form of information-technology-like constructs such as information flows, control flows, data structures, operational rules, event/trace diagrams, state transition diagrams, timelines, roadmaps, etc.

EXAMPLE 2 These concepts and properties could be expressed in other forms such as risk models, financial models, economic models, simulation models, sensitivity models, queuing models (as well as other kinds of continuous and discrete event simulation models), geospatial models, management models, business models, social- and environmental-impact models, value stream models, etc.

NOTE 1 Domain architectures, reference architectures and architectural patterns can be used in forming potential solutions or parts of solutions. Further details on these can be found in [Annex E](#).

NOTE 2 Architectural patterns and related tactics, along with heuristics, (see [E.3.3](#) for further details) could be employed to deliver desired characteristics.

- c) Identify and characterize the tradeoffs between candidate architectures for each solution.
- d) Identify key characteristics that provide insight into the architecture and use the key characteristics to define the context and scope of the architecture(s).

NOTE 3 The key characteristics are based on identified stakeholder concerns, relevant requirements, quality attributes, architecture objectives and other relevant factors. A mapping between such drivers and the key characteristics can be developed to aid in traceability.

NOTE 4 Consideration can be given to various architecture solution concepts as delineated in [E.2.2](#) and to various architecture life concepts and life cycle models as delineated in [E.2.3](#) and [E.2.4](#).

- e) Formulate principles, implications, guidelines, protocols and standards for each candidate architecture.
- f) Decompose, when necessary, and allocate the key characteristics to components, processes and other kinds of elements that make up each candidate architecture.
- g) Identify the processes and activities that, when arranged or performed in a specific order, will provide or enable the identified characteristics.
- h) Identify rules governing the components, their composition, aggregation, interaction and interdependence that ensure that each candidate architecture provides or enables the desired solution characteristics.
- i) Determine that each candidate architecture provides the desired solution characteristics or enables them to be realizable.
- j) Identify issues and areas for improvement in the architecture(s).

NOTE 5 If a promising architecture with high stakeholder value does not close (i.e. it does not meet all the requirements) due to a constraint imposed by a requirement, it is helpful to discuss with relevant stakeholders to determine whether the requirement can be relaxed or whether some other compromise can be reached that allows the promising, high-value architecture to close. This could involve securing a waiver on a requirement or changing the requirement.

- k) If more than one architecture is devised, select the best architecture(s) for use downstream.

**NOTE 6** The Architecture Evaluation process can be used to facilitate selection of the most suitable architecture(s). The results of problem space analysis can be used as a basis for the evaluation criteria. Often an initial screening of candidate architectures is conducted prior to sending these for evaluation to avoid unnecessary effort in evaluating candidates that are duplicative, too costly or risky, unsuitable or infeasible, etc.

**NOTE 7** Sometimes the selection of the architecture(s) going forward is made through decisions outside the scope of the architecture processes. In that case, the candidate architectures are presented to decision makers along with an assessment of these architectures. This might entail involvement of the Decision Management Process or the Portfolio Management Process described in ISO/IEC/IEEE 15288.

#### **8.4.8 Capture architecture concepts and properties**

**NOTE 1** This activity could use the results of the Architecture Elaboration process when appropriate. Architecture conceptualization only needs to describe the architecture to the level of specificity and granularity that is suitable for its intended users, which in many cases does not require significant elaboration. The elaboration of architecture views, models and descriptions will often occur later in the life cycle of the architecture, after the architecture has become more mature and the extra effort of elaboration becomes worthwhile. Elaboration could also occur only after several architecture alternatives have been examined for their suitability, and they are selected down to one or a few alternatives for further examination and eventual use downstream in the engineering effort.

- a) Identify the uses and users of architecture views and models.
- b) Define the purpose, scope, breadth and depth for the necessary architecture views and models.
- c) Identify key aspects to be addressed by the architecture views and models.
- d) Identify relevant stakeholders and concerns with respect to the architecture views and models.
- e) Specify the form(s) of expression for the necessary architecture views and models suitable for their intended users for this stage of development.
- f) Identify, select, develop or modify enablers to support generation of the necessary models and views, such as the following items:
  - 1) architecture viewpoints, modeling methods and view generation methods, model kinds,
  - 2) architecture frameworks, modeling templates, view templates and metamodels, and
  - 3) modeling tools and techniques.

**NOTE 2** These enablers could have been already selected or developed by the Architecture Enablement process. If these are selected or developed by the Architecture Conceptualization process then these can be provided to Architecture Enablement to be placed in the organization repository for use by other architecture efforts.

- g) Capture architecture decisions and architectural characteristics in the specified form.
- h) Capture key architectural concepts, properties of interest, rationales, conditions, constraints and assumptions in the specified form.
- i) Capture architectural guidelines, principles, protocols and standards in the specified form.
- j) Capture components, their composition, interdependence and their interactions in the specified form.
- k) Capture identified processes, activities and tasks that aid in achievement of architecture and solution characteristics.
- l) Develop an architecture description consisting of relevant viewpoints, views, models, model correspondences and express them in the specified form with a level of detail, correctness and completeness suitable for their intended use.

**NOTE 3** See ISO/IEC/IEEE 42010 for more information on these architecture description concepts.

#### 8.4.9 Relate the architecture to other architectures and to relevant affected entities

NOTE 1 It is common that a new architecture is replacing or subsuming existing or planned design elements. In such a case, the new architecture can be mapped to the design, for example, to help understand the impact of implementing the architecture. It could also be important to map the architecture to other relevant elements such as policies, processes, doctrine, organizations, training, logistics, personnel, facilities, etc.

- a) Identify relevant entities and other architectures that relate to architecture elements and the nature of these relationships.
- b) Define the correspondences, interfaces and interactions between the relevant entities and other architectures with each other and with the architecture being expressed.
- c) Partition, align and allocate requirements to architecture elements and related entities.

NOTE 2 During this task, sometimes additional factors that drive the architecture will be found, which might result in changes to the architecture. Also, this provides an opportunity to help clarify the meaning and intent of requirements and to ensure that all the relevant requirements can be met by the architecture.

- d) Map affected entities and other architectures to relevant architecture concepts, properties and other attributes.
- e) Formulate principles and precepts expected to be used during execution of the life cycle processes for the architecture entity.

NOTE 3 Architecture information can be used in other systems life cycle processes as specified in [C.1](#).

EXAMPLE 1 The architecture could dictate as a matter of principle for example that elements are to use a single place to store all data rather than having their own storage location.

NOTE 4 A precept is “a general rule intended to regulate behavior or thought” that might not rise to the level of being a principle.

EXAMPLE 2 An example precept is that designers are to look at the future evolution of the architecture to determine possible design features that can be devised to better accommodate future architecture characteristics (i.e., sometimes called “hooks” in the design).

- f) Formulate principles and precepts for design and evolution of the architecture entity.

#### 8.4.10 Coordinate use of conceptualized architecture by intended users

- a) Identify intended users of architecture conceptualization information, including relevant architecture descriptions, models and data.

NOTE 1 Users of this information could be different than the original set of users identified during the initial generation of architecture models and views. These users could be those doing evaluation or elaboration of the architecture, architects of related systems and other collaborators and reviewers who provide inputs and feedback to the conceptualization of the architecture, those managing the collection so they can gain some understanding of the nature of an architecture, analysts who need to understand the architecture as the basis for their analysis, managers who need to use the architecture for planning and scoping a project, design engineers who need to use the architecture to get an early start on conceptual design and to give feedback on the architecture that will drive their design, and so on.

NOTE 2 Interactions with project processes are delineated in [B.1](#), which could suggest potential users of architectural information.

- b) Prepare architecture conceptualization information and data, along with supporting material, for use by others.
- c) Deliver architecture conceptualization information and data to intended users and other interested parties.

**NOTE 3** This could occur as a direct delivery to intended recipients, as a posting to the architecture repository, or in a formal release through some organizational release process. The particular mechanism for the delivery could be a factor in determining how best to package the information and what steps need to be taken to validate the information and data before it is released.

**NOTE 4** Sometimes the conceptualized architecture is not mature or complete enough for use by those who need to know about the architecture or to use the architecture information in performing their duties. In such cases, it could be necessary to prepare a more complete set of architecture views and models through the Architecture Elaboration process before this information is suitable for use downstream.

- d) Monitor use of architecture conceptualization information to collect feedback on the architecture and on the form and contents of the architecture work products.
- e) Communicate architecture conceptualization information to interested parties.
- f) Incorporate feedback into the architecture descriptions, views and models.
- g) Incorporate feedback into the architecture conceptualization effort when this is not otherwise captured in architecture descriptions, views and models.

## **8.5 Work products**

The following work products shall be produced:

- architecture conceptualization plan,
- architecture conceptualization status report,
- problem space definition report,
- architecture objectives,
- quality model, and
- architecture views and models.

**EXAMPLE** Work products from Architecture Conceptualization can be captured, for example, as problem formulations (concepts, abstractions, scope and relationships definition), problem structures (cause/effect relationships, objectives hierarchies, needs, wants and expectation), solution structures (components, connectors), and principles and guidelines. Solution elements could also include such things as policies, procedures, rules, organizations, people and a variety of other entities and practices.

**NOTE** During early stages it is sometimes important to be agile and quick in conceptualizing many alternative architectures. Some of these early architecture descriptions will be little more than sketches. After doing several quick rounds of evaluation, there might then be a smaller number of viable architectures that are worth capturing in a more complete form and storing in the repository for later use. The more complete form of architecture description would be developed in the Architecture Elaboration process.

# **9 Architecture Evaluation process**

## **9.1 Purpose**

The purpose of the Architecture Evaluation process is to determine the extent to which one or more architectures meet their objectives, address stakeholder concerns and meet relevant requirements.

**EXAMPLE** Architecture evaluations can be performed to answer questions such as:

- a) Is the architecture sufficient to address expected operational uses and situations?
- b) Is the architecture sufficiently flexible and extensible to address the evolving needs?
- c) Is the quality of the architecture acceptable to stakeholders?

- d) Is the architecture addressing stakeholder concerns?
- e) Is the architecture addressing stated objectives? and
- f) Can the architecture be implemented successfully?

## 9.2 Outcomes

As a result of the successful implementation of the Architecture Evaluation process:

- a) The basis for evaluation findings and recommendations are clearly communicated and understood by the relevant decision makers and key stakeholders.
- b) Relationship between stakeholder concerns and evaluation findings and recommendations is well established.
- c) The projected costs, risks, opportunities and tradeoffs associated with implementing the architecture(s) are understood and well founded.
- d) Stakeholders are able to understand the extent to which the architecture addresses their concerns and intended operational uses, and where and why there are shortfalls.
- e) Value of the architecture(s) to relevant stakeholders is understood by those doing the architecting.

**NOTE** The “value of the architecture” referred to above is a determination of the extent to which the architecture addresses stakeholder concerns and intended operational uses. This implies that the entity that is being architected, when implemented, will be capable of providing some degree of satisfaction if that entity is implemented in accordance with the architecture description.

## 9.3 Implementation

The organization shall implement the activities in [9.4](#) (numbered as 9.4.N) in accordance with applicable organization policies and procedures with respect to the Architecture Evaluation process. The activities may be performed in any order that is deemed appropriate. The organization should implement the relevant tasks (identified as list items under each 9.4.N activity) as appropriate to the situation.

Architecture evaluation work products should be stored in the architecture repository for future reference and audit. The repository should be used to facilitate widespread access, enable auditing and encourage future reuse.

**NOTE** Below are a few guidelines that can assist in the implementation of this process.

- a) This clause specifies the requirements on architecture evaluation activities while ISO/IEC/IEEE 42030 specifies the requirements on an architecture evaluation framework and its elements.
- b) Architecture evaluation is most useful when it focuses on areas that are high risk or on particularly strong concerns held by key stakeholders.
- c) The process activities are based on the concepts and principles specified in ISO/IEC/IEEE 42030 on an architecture evaluation framework: evaluation objectives, value assessment objectives, architectural analysis objectives and factors for evaluation, assessment and analysis.
- d) This process is for evaluating one or more architectures (i.e. the fundamental concepts or properties of an architecture entity) rather than just evaluation of an architecture description (AD) (e.g. views and models) for those architectures. The evaluation could use an AD as the basis for getting information about the architecture but this is not the only way.
- e) Customers and users can have a special role to play in the architecture development effort. They are sometimes paying for the architecting effort and usually have to live with the resulting solution for years to come. They also have a key role in helping to specify the architecture objectives and the success criteria.
- f) Conduct the architecture evaluation effort in such a manner as to avoid potential bias or conflict of interest. One common way of doing this is by using an independent party to perform the evaluation.

## 9.4 Activities and tasks

### 9.4.1 Prepare for and plan the architecture evaluation effort

a) Formulate judgments to be made by architecture evaluation based on the potential decision(s) that can be supported by the architecture evaluation effort.

b) Define the expected purpose, scope, objectives and level of detail of the architecture evaluation effort.

EXAMPLE 1 Scope can include things such as selection of the best alternative among several, identification of alternatives that meet a minimum criterion set, determination of best alternatives to include in a portfolio.

c) Review stated purpose, scope and objectives of the architecture evaluation effort with the sponsor, architect and other interested parties.

d) Define one or more architecture evaluation approaches that are consistent with the architecture governance and management directions and are consistent with the purpose, scope and objectives for this effort.

NOTE 1 In some cases, the approaches defined here could be a function of the evaluation criteria defined in [9.4.3](#). Also, approval of the evaluation plan can sometimes depend on the evaluation criteria to be used. Therefore, the remainder of this planning activity could depend on completion of the criteria determination.

e) Select the evaluation approach to be used.

f) Select or develop the requisite architecture evaluation techniques, methods and tools.

NOTE 2 The items selected or developed here could be a function of the evaluation methods defined in [9.4.4](#) and the measurement techniques defined in [9.4.5](#). Also, approval of the evaluation plan can sometimes depend on these methods and techniques to be used. Therefore, the remainder of this planning activity could depend on determination of these methods and techniques.

g) Select or develop one or more architecture evaluation framework(s).

NOTE 3 This framework is different than the architecture description framework typically used as the basis for generation of architecture views and models. The architecture evaluation framework can consist of generalized evaluation objectives and criteria, value models, assessment measurements and methods, scorecard templates, business case heatmap templates, dashboard constructs, etc.

h) Collect any relevant regulatory requirements that dictate when and where an evaluation is to be performed and possibly who should be involved in the effort.

i) Plan the architecture evaluation effort using the Project Planning process in ISO/IEC/IEEE 15288 as a guide.

NOTE 4 ISO 21500 and ISO 21505 are also useful references for planning the architecture evaluation effort.

1) Document the purpose, scope and objectives of the architecture evaluation effort.

2) Establish metrics for the architecture evaluation effort.

3) Identify the data and information needed for the architecture evaluation effort.

4) Obtain access to enablers needed for the architecture evaluation effort.

NOTE 5 The enablers will usually be obtained from the Architecture Enablement process. When enablers are obtained from other sources, these can become candidate enablers for use by other projects through the Architecture Enablement process.

EXAMPLE 2 Architecture evaluation enablers could be tools, methods and procedures for value modeling, optimization, value measurement, meeting facilitation, etc.

5) Identify and define architecture evaluation work elements and associated resources.

- 6) Specify the work products and their outlines to be produced through performance of this process.
- 7) Identify subject matter experts necessary to support the architecture evaluation effort.

NOTE 6 This could require access to the architecture development lead and other people involved in architecture conceptualization and elaboration. Stakeholders and implementers are sometimes needed to participate in the effort.

- 8) Develop architecture evaluation schedule and define associated milestones.
- j) Produce an architecture evaluation plan that contains the planning information.
- k) Obtain necessary approvals, resources and funding for the plan.
- l) Collect the data and information needed for the architecture evaluation effort.
- m) Determine how and when to involve stakeholders in the architecture evaluation effort.
- n) Ensure personnel are trained in the use of identified techniques, methods and tools.
- o) Ensure personnel have necessary and appropriate access to relevant architecture work products, data and information.

#### **9.4.2 Monitor, assess and control the architecture evaluation activities**

- a) Report architecture evaluation activity plans and status.
- b) Monitor and assess whether architecture governance directives and guidance are being followed.
- c) Monitor and assess whether architecture management directives and guidance are being followed.
- d) Monitor and assess metrics for the architecture evaluation effort.
- e) Identify and assess risks and opportunities associated with the architecture evaluation effort.
- f) Maintain traceability of architecture evaluation results to the source material used during the process.
- g) Ensure traceability and integration of information that was assessed in tasks above.
- h) Ensure that other processes are properly using architecture evaluation products.

NOTE 1 See [C.1](#) for recommended interactions with system life cycle processes.

NOTE 2 See [C.2](#) for recommended interactions with enterprise life cycle processes.

- i) Implement corrective actions where necessary to modify the work plan or realign the work with the plan.
- j) Assess and control the architecture evaluation effort in accordance with the Project Assessment and Control process in ISO/IEC/IEEE 15288.

NOTE 3 ISO 21500 and ISO 21505 are also useful references for assessment and control.

- k) Manage risks associated with architecture evaluation in accordance with the Risk Management process in ISO/IEC/IEEE 15288.

NOTE 4 ISO 31000, ISO 21500 and ISO 21505 are also useful references for risk management.

#### 9.4.3 Determine evaluation objectives and criteria

**NOTE 1** The evaluation criteria consist of both the value assessment criteria and the architectural analysis criteria. Value assessment criteria are the conditions that are to be met by or the tests that are to be passed by the entity being assessed. This assessment is a determination of the extent to which stakeholder concerns and architecture objectives are going to be met. Architectural analysis criteria are the conditions that are to be met by or the tests that are to be passed by the entity being analyzed in terms of those concepts and properties that contribute to the value assessment activity. This analysis is a determination of the extent to which stakeholder needs and requirements are going to be met.

Since the evaluation could be about determination of architecture suitability for some use different than the original purpose of the architecture, the evaluation objectives may be different from the objectives used in conceptualization of the architecture(s). In that case, the conceptualization objectives should be examined to determine to what extent they apply to this evaluation.

The evaluation objectives and criteria should be based on the results of problem space analysis performed during architecture conceptualization or elsewhere, when applicable.

- a) Identify relevant mandates and imperatives, including relevant policies and standards.
- b) Identify relevant stakeholders and their concerns for the architecture(s) being evaluated.

**NOTE 2** The stakeholder concerns to be considered during the architecture evaluation might be different from the stakeholder concerns addressed during the original conceptualization of the architecture. The evaluation, for example, might be tasked with determining if the architecture can be suitable for some other purpose than that originally envisioned.

- c) Define value assessment objectives and criteria that contribute to key success factors, key indicators and decisions that need to be made.

**NOTE 3** During Architecture Evaluation, those value assessment criteria defined during Architecture Conceptualization can be evaluated for relevance in the evaluation effort, and if necessary, additional criteria can be added or existing criteria can be modified to reflect the evaluation context.

- d) Define architectural analysis objectives and criteria that support the value assessment objectives and criteria.

**NOTE 4** Consider the agreed-upon quality attributes for the architecture as candidates for architectural analysis criteria.

- e) Determine value assessment and architectural analysis objectives and criteria structure and relationships.

**NOTE 5** The relationships between value assessment and architectural analysis criteria helps determine the extent to which stakeholder concerns are addressed. These relationships can often be structured in such a way to facilitate doing the value assessment and correlating this with the analysis results. See ISO/IEC/IEEE 42030 for more details on this.

- f) Determine relationships between value assessment objectives and criteria and architectural analysis objectives and criteria and elements of value or utility (e.g. value function, utility curve).

**NOTE 6** This task will determine how the two sets of objectives and criteria map to the “value curves” that represent the figures of merit for the architecture. See Trade-off Analytics, Parnell 2017<sup>[42]</sup> for example.

- g) Examine evaluation objectives and criteria with respect to requirements and validate them against stakeholder concerns and intended operational uses.

- h) Inform each stakeholder about the validation of evaluation objectives and criteria that are traceable to their respective concerns, needs or requirements.

#### 9.4.4 Determine evaluation methods and integrate with evaluation objectives and criteria

- a) Select or develop value assessment and architectural analysis methods that support the defined value assessment and architectural analysis objectives and criteria.

NOTE 1 These methods might have been developed by the Architecture Enablement process. If so, then they will be located in the architecture repository. If these methods are developed here, then they can be supplied to architecture enablement as candidate items to be made appropriate for reuse across the organization.

- b) Identify and define factors for evaluation, assessment and analysis with respect to the selected or developed methods.
- c) Review assessment and analysis objectives, criteria and methods and associated factors, scales and weights with the sponsor and architect.
- d) Identify sources of information for use during application of the value assessment and architectural analysis objectives and criteria.

NOTE 2 Some information will come from analysis, but other information could come from other sources, such as prior evaluation efforts, operational experience, industry databases, system verification activities and research activities.

#### 9.4.5 Establish measurement techniques, methods and tools

NOTE 1 Many architecture evaluations can be performed adequately without invoking a substantial multi-tiered structure of value assessment and architectural analysis layers. At a minimum, the evaluation will assess each alternative against the objectives and criteria. It does not always need to be quantitative, and does not always need to apply measurement scales. More information on how to perform evaluations using measurement techniques, methods and tools can be found in the Bibliography. See for example Trade-off Analytics, Parnell 2017<sup>[42]</sup>.

NOTE 2 Weights are not absolutely required since this is nearly always a multi-objective problem, and the goal will usually be to present to the stakeholders the tradeoffs between those objectives, not necessarily to mathematically determine the "best" solution.

- a) When appropriate, utilize scales and weights as a means to measure factors and properties of the architecture.
  - 1) Define analysis scales for measuring against the analysis objectives and criteria, if appropriate.
  - 2) Define assessment scales for measuring against the assessment objectives and criteria, if appropriate.
  - 3) Specify weights for assessment and analysis objectives and criteria, if appropriate.

NOTE 3 Some methods do not use weights, while others depend on them to achieve more accurate results. There are different kinds of weights, such as importance weights, swing weights and criticality weights. The methods chosen will usually specify the kinds of weights to be used. See Trade-off Analytics, Parnell 2017<sup>[42]</sup> for example.

- 4) Determine where on these scales the architecture is now and identify desired point(s) for future levels of achievement.
- b) Identify appropriate measures for the relevant architecture concepts and properties.
- c) Identify metrics to be determined from the measures.

NOTE 4 There is an overlap between measures and metrics. Both can be qualitative or quantitative, but what distinguishes them is important. Measures are concrete, usually measure one thing, and are quantitative in nature (e.g. I have five apples). Metrics describe a quality and require a measurement baseline (e.g. I have five more apples than I did yesterday). Measures and metrics can be useful for setting program priorities, allocating resources, and measuring performance. See ISO/IEC/IEEE 15939 for information on the measurement process.

- d) Define relationships between measures, metrics and evaluation objectives and criteria.
- e) Identify sources of information for obtaining values for these measures and metrics.
- f) Identify techniques, methods and tools appropriate for these measures, metrics and evaluation objectives and criteria.
- g) Estimate the likely accuracy, errors and degrees of uncertainty in results when using these measures, metrics and evaluation objectives and criteria.

#### **9.4.6 Collect and review evaluation-related information**

- a) Identify relevant information for the chosen value assessment and architectural analysis methods.
 

NOTE If possible, reuse existing data and results from previous evaluations of this kind, if the information available is still valid.
- b) Collect all relevant and necessary information, including required architecture views and models.
- c) Create additional information if not readily obtainable (i.e. non-existent, inaccessible) and if its creation is feasible within the available time without causing disruptions.
- d) Examine and qualify collected artifacts in terms of completeness, correctness and consistency.
- e) Develop an understanding of the architecture, the architecture quality attributes, key decisions and concerns about the architecture or associated architecture entities.

#### **9.4.7 Analyze architecture concepts and properties and assess stakeholder value**

- a) Identify the architecture or the architecture alternatives that will be subject to the evaluation.
 

NOTE 1 The architecture(s) can come from the architecture conceptualization or elaboration activities.

NOTE 2 The status quo is sometimes one of the alternatives that could be considered.

NOTE 3 Development of these alternatives could be outside the scope of the evaluation activity. However, sometimes the evaluation activity determines that there is an insufficient number, variety or extent of alternatives that have been predefined and that additional alternatives need to be generated.
- b) Eliminate alternatives that are similar to each other and that do not provide a discriminating case with respect to the evaluation objectives and criteria.
- c) Identify where alternatives fail to meet identified mandates and imperatives and propose changes to the architectures (or new alternatives to consider) that would meet these mandates and imperatives.
- d) Use the assessment and analysis method(s) as specified in the evaluation work plan to assess architecture for the identified purpose.
- e) Use selected evaluation methods to determine concepts and properties of the architecture (alternatives) with respect to the evaluation objectives and criteria.

EXAMPLE Examples of methods to determine these concepts and properties include elements such as analysis, observation, simulation, prototyping, experimentation, inspection, audit, review, walk-through and expert judgment.

- f) Identify areas for potential or planned reuse of existing architecture elements and the risks associated with this reuse, when appropriate.
- g) Identify and characterize costs, risks and opportunities, when appropriate.

**NOTE 4** Risks and costs will have been identified during architecture conceptualization for each potential solution under consideration. The risks and costs identified here for the architecture under evaluation might be related to the previously identified risks and costs for the proposed solutions. It is sometimes necessary to assess potential implementations of the architecture to determine other risks and costs that might arise. This could involve interaction with a development project or organization to devise potential implementations that can be used as the basis for risk identification and assessment and the basis for cost estimation and projection.

- h) Identify the causes of risk and propose modifications to the architecture to mitigate these risks.
- i) Evaluate the architecture or the architecture alternatives against the identified quality attributes, stakeholder concerns and architecture objectives.
- j) Characterize the accuracy of, the degrees of uncertainty associated with and the extent of errors in the measurements used and other results obtained during the evaluation.
- k) Perform sensitivity analysis to help understand which factors are dominant.
- l) Produce an architectural analysis results report.
- m) Produce a value assessment results report.

#### 9.4.8 Characterize architecture(s) based on assessment results

- a) When multiple alternatives are being considered, develop screening criteria to use as a filter to facilitate dismissal of alternatives from being further evaluated.
- b) Screen alternatives from further consideration.

**NOTE 1** Similarity is not the only reason for screening. This activity is for examining tradeoffs using the results of the prior analysis and assessment. Alternatives that don't "highlight" the tradeoff criteria are screened. Two alternatives that present the exact same tradeoffs are not examined. A set of alternatives that "span" the tradespace is also examined.

- c) Identify and characterize tradeoffs with respect to quality attributes, stakeholder concerns, architecture concepts and properties, costs, risks and opportunities.

**NOTE 2** This activity analyzes the proposed solutions in terms of value to stakeholders and quality attributes delivered by each solution. It is complementary to architecture conceptualization, which characterizes the solutions and tradeoffs from the technical perspective, considering aspects such as ease of realization, compatibility with other solutions, trends in the technology space, etc.

**NOTE 3** Tradespace is the range and extent of parameters, properties and characteristics that are relevant in potentially satisfying architecture objectives and stakeholder concerns. The tradespace often does not include any solution that will completely satisfy the architecture objectives and stakeholder concerns. Tradespace analysis is used for analyzing the relevant constraints, conditions and challenges involved in problems where you have multiple stakeholders and multiple objectives. It involves the identification and understanding of tradeoffs involved in choosing between competing solutions. Tradeoffs can be within each solution and between solutions (and the problems/opportunities they are intended to address).

- d) Assess whether, and the extent to which, relevant mandates and imperatives, including relevant policies and standards, are met by the architecture(s).

**NOTE 4** Sometimes the architecture meets the objectives and addresses stakeholder concerns, but does not meet a mandate or imperative. It might be possible to get relief from meeting such mandates or imperatives, so it would be good to examine this possibility.

**NOTE 5** It is sometimes better to eliminate architecture alternatives that fail to meet mandates and imperatives earlier in the process to avoid expending considerable resources in the further evaluation of these items. However, it could also be helpful to see how far off these are from meeting the mandates and imperatives.

- e) When multiple alternatives are being considered, assess results of this analysis to determine, as appropriate, the best architecture(s) among the alternatives.

- f) Assess results of this analysis to determine, as appropriate, the quality of the architecture, or the extent to which the architecture achieves architecture objectives, meets relevant requirements or addresses stakeholder concerns.
- g) Assess results of this analysis to determine, as appropriate, where the architecture(s) fail to meet objectives and satisfy stakeholder concerns.
- h) Review analysis and assessment results with the sponsor, architect and other interested parties.

#### **9.4.9 Formulate findings and recommendations**

- a) Identify and characterize findings from the evaluation.
- b) Analyze the findings.
- c) Validate the findings with subject matter experts and other relevant parties, as appropriate.
- d) Assess implications of findings.
- e) Develop recommendations.
- f) Identify how the findings and recommendations can contribute to evolution of the evaluated architectures, evolution of other architectures and evaluation of other architectures.
- g) Identify how the findings and recommendations can contribute to relevant organization and project decisions and milestone determinations.
- h) Review findings and recommendations with the sponsor, architect and other interested parties.

#### **9.4.10 Capture and communicate evaluation results**

- a) Identify the audience for communicating the evaluation results.
- b) Select the most relevant result and elaborate on key findings and recommendations.
- c) Develop an evaluation report summarizing the findings and recommendations and describing how these were developed.
- d) Obtain approval for report, if appropriate.
- e) Present findings and recommendations to decision makers, if relevant.
- f) Present to key stakeholders and architects.

**NOTE** If a promising architecture with high stakeholder value does not close (i.e. it does not meet all the requirements) due to a constraint imposed by a requirement, then discuss with relevant stakeholders to determine whether the requirement can be relaxed or whether some other compromise can be reached that allows the promising, high-value architecture to close. This could involve securing a waiver on a requirement or changing the requirement.

- g) Capture responses from these presentations (e.g. issues, action items, risks, observations, perspectives).
- h) Capture resolutions of issues raised during reviews and presentations for future reference.
- i) If required or requested, iterate relevant parts of the evaluation.
- j) If evaluation is updated, review changes with the sponsor, architect and other interested parties.
- k) Update report, if necessary, based on feedback from presentations.
- l) Archive report and responses received during presentations.

## 9.5 Work products

The following work products shall be produced:

- architecture evaluation plan,
- architecture evaluation report,
- architecture value assessment results, and
- architecture analysis results.

# 10 Architecture Elaboration process

## 10.1 Purpose

The purpose of the Architecture Elaboration process is to describe or document an architecture in a sufficiently complete and correct manner for the intended uses of the architecture.

NOTE 1 This clause specifies the requirements on architecture elaboration activities while ISO/IEC/IEEE 42010 specifies the requirements on architecture description elements.

NOTE 2 The elaboration process is not necessarily working on the “logical” or “physical” architecture. Likewise, the conceptualization process is not necessarily working on the “conceptual” architecture. Each of these processes can work on any or all three of these “kinds” of architecture (or even none of these three kinds). The kinds of architectures that are relevant depend on the situation at hand. See [Annex E](#) for discussion on these and several other kinds of architectures.

## 10.2 Outcomes

As a result of the successful implementation of the Architecture Elaboration process:

- a) Architecture viewpoints and metamodels are suitable for developing the appropriate architecture views and models.
- b) Architecture views and models are captured to an appropriate level of detail for their intended purposes, using selected architecture modeling/description languages and notations to the extent necessary for their intended use.
- c) Architecture views and models are accurately and completely expressed to capture the fundamental concepts and properties of the architecture to the extent necessary for their intended use.
- d) Architecture views and models are adequately expressed to capture the principles and precepts of the architecture that guide development and evolution of the entity being architected.
- e) Architecture description is under configuration control and made available to all relevant parties.
- f) Architecture description continues to be aligned with changes to the architecture entity during its development.
- g) Alignment of the architecture with relevant requirements and design characteristics is achieved.
- h) Alignment of the architecture with other relevant architectures is achieved.

NOTE Alignment is a shared responsibility between architecture governance and management, as well as between conceptualization and elaboration. As the architecture views and models are refined during elaboration it is possible that the alignment established earlier by these other processes is diminished. Where misalignment is noticed during elaboration, it is important to make this known to the other processes to ensure that appropriate mitigation can occur, where possible and appropriate.

## 10.3 Implementation

The organization shall implement the activities in [10.4](#) (numbered as 10.4.N) in accordance with applicable organization policies and procedures with respect to the Architecture Elaboration process. The activities may be performed in any order that is deemed appropriate. The organization should implement the relevant tasks (identified as list items under each 10.4.N activity) as appropriate to the situation.

The Architecture Elaboration process may need to handle system architectures (in the sense used in ISO/IEC/IEEE 15288), software architectures (in the sense used in ISO/IEC/IEEE 12207), and enterprise architectures (in the sense used in ISO 15704) depending on the situation at hand.

Architecture elaboration work products should be stored in the architecture repository for future reference and audit. The repository should be used to facilitate widespread access, enable auditing and encourage future reuse.

**NOTE** Below are a few guidelines that can assist in the implementation of this process.

- a) Architecture description concepts considered in this document are those described in ISO/IEC/IEEE 42010: stakeholder concerns, viewpoints, model kinds, views, models, architecture description.
- b) The Architecture Elaboration process is typically applied to one or more architecture descriptions produced by the Architecture Conceptualization process. However, there could be cases where this process is applied during the “reverse” architecting of a system or other entity. Another case could be creation of a product line architecture based on knowledge of several existing or imagined systems or other entities.
- c) Architecture Elaboration process can be applied to current architectures (to articulate and expound what the current architecture is), to planned architectures (to define the architectural basis on which they are formed), and to future envisioned architectures (to set architectural target levels against which planned architectures should make progress through the inclusion of relevant features and/or the provision of evolutionary capabilities).
- d) ISO/IEC/IEEE 15288 specifies activities for dealing with systems that are designed and built by projects and programs within an organization. In the context of implementing the processes in this document, the activities in ISO/IEC/IEEE 15288 have limited value for engineering the enterprise itself.
- e) The views in an architecture description can be created in any of the architecture processes. However, the more complete and refined set of views are generally created in the Architecture Elaboration process. These views created in the Architecture Elaboration process are usually composed of architecture models that conform to selected viewpoints and model kinds. ISO/IEC/IEEE 42010 provides requirements and guidance on architecture description.

## 10.4 Activities and tasks

### 10.4.1 Prepare for and plan the architecture elaboration effort

- a) Identify the intended users of the architecture description to be generated by this elaboration effort.
- b) Identify the question(s) to be addressed by the architecture elaboration effort.
- c) Determine the set of stakeholder concerns and architecture objectives that will be addressed during the architecture elaboration effort.
- d) Define the expected purpose, scope, objectives and level of detail of the architecture elaboration effort.
- e) Define one or more architecture elaboration approaches that are consistent with the architecture governance and management directions and are consistent with the purpose, scope and objectives of this effort.
- f) Select or develop the requisite architecture elaboration techniques, methods and tools.

NOTE 1 If the development is going to be extensive, it might be appropriate to have the Architecture Enablement process develop these items. But in any case, it might be prudent to nominate these items for standardized use throughout the organization by providing these to the Architecture Enablement process for possible further development and for placing them in the architecture library and registry.

- g) Identify, select, develop or modify architecture frameworks, viewpoints, modeling templates, view generation methods and metamodels to be used for generating the necessary models and views.

NOTE 2 Frameworks, viewpoints, modeling templates, view generation methods and metamodels could specify modeling formalisms, languages and notations that enable the modeling effort to be more effective in providing key insights and communicating clearly.

- h) Plan the architecture elaboration effort using the Project Planning process in ISO/IEC/IEEE 15288 as a guide.

NOTE 3 ISO 21500 and ISO 21505 are also useful references for planning of the architecture elaboration effort.

- 1) Document the purpose, scope and objectives of the architecture elaboration effort.
- 2) Establish metrics for the architecture elaboration effort.
- 3) Identify the data and information needed for the architecture elaboration effort.
- 4) Obtain access to enablers needed for the architecture elaboration effort.

NOTE 4 The enablers will usually be obtained from the Architecture Enablement process. When enablers are obtained from other sources, these can become candidate enablers for use by other projects through the Architecture Enablement process.

EXAMPLE Architecture elaboration enablers could be tools, methods and procedures for model development, view creation, document production, change control, etc.

- 5) Identify and define architecture elaboration work elements and associated resources.
  - 6) Specify the work products and their outlines to be produced through performance of this process.
  - 7) Develop an architecture elaboration schedule and define associated milestones.
- i) Produce an architecture elaboration plan that contains the planning information
  - j) Obtain necessary approvals, resources and funding for implementing the plan.
  - k) Collect the data and information needed for the architecture elaboration effort.
  - l) Ensure personnel are trained in the use of identified techniques, methods and tools.
  - m) Ensure personnel have necessary and appropriate access to relevant architecture work products, data and information.

#### **10.4.2 Monitor, assess and control the architecture elaboration activities**

- a) Report architecture elaboration activity plans and status.
- b) Monitor and assess whether architecture governance directives and guidance are being followed.
- c) Monitor and assess whether architecture management directives and guidance are being followed.
- d) Monitor and assess metrics for the architecture elaboration effort.
- e) Identify and assess risks and opportunities associated with the architecture elaboration effort.
  - 1) Implement risk mitigation efforts for risks deemed sufficiently critical to warrant such action.

- 2) Implement opportunity pursuit efforts for opportunities deemed sufficiently worthy to warrant such action.
- f) Maintain traceability of architecture elaboration results to the source material used during the elaboration effort.
- g) Ensure that the architecture description is maintained.
- h) Ensure that the architecture description is under proper change control.
- i) Provide developed or modified viewpoints to the Architecture Enablement process for possible use as an organization standard.
- j) Ensure that the elaboration effort is consistently using architecture elaboration products.
- k) Ensure that other processes are properly using architecture elaboration products.

NOTE 1 See [C.1](#) for recommended interactions with system life cycle processes.

NOTE 2 See [C.2](#) for recommended interactions with enterprise life cycle processes.

- l) Implement corrective actions where necessary to modify the work plan or realign the work with the plan.
  - m) Manage risks associated with architecture elaboration using the Risk Management process in ISO/IEC/IEEE 15288 as a guide.
- NOTE 3 ISO 31000, ISO 21500 and ISO 21505 are also useful references for risk management.
- n) Manage changes to the architecture descriptions, views and models using the Configuration Management process in ISO/IEC/IEEE 15288 as a guide.

#### 10.4.3 Identify or develop architecture viewpoints

- a) Select, adapt or develop viewpoints and model kinds based on stakeholder concerns.

NOTE 1 Relevant viewpoints might have already been identified or developed during the Architecture Conceptualization process. These are examined to see if they can be used as-is or if they need to be modified or expanded for use during elaboration.

NOTE 2 An architecture viewpoint governs the view(s) that will be created and addresses a particular set of stakeholder concerns. It is usually best to have the people who develop the viewpoint be those who are most familiar with the nature of the problem being resolved and the relevant stakeholder concerns being addressed. Where possible, generalize the viewpoint to enable its use in other architecting efforts of this kind. ISO/IEC/IEEE 42010 can be used to assist in developing viewpoints.

NOTE 3 If the development is going to be extensive, it might be appropriate to have the Architecture Enablement process develop these items. But in any case, it might be prudent to nominate these items for standardized use throughout the organization by providing these to the Architecture Enablement process for possible further development and for placing them in the architecture repository.

- b) Identify expected users of architecture elaboration information, including relevant architecture descriptions, models and data.
- c) Establish or identify potential architecture framework(s) to be used in developing models and views.

NOTE 4 Relevant architecture frameworks might have already been established or identified during the Architecture Conceptualization process. These are examined to see if they can be used as-is or if they can to be modified or expanded for use during elaboration.

NOTE 5 Frameworks and viewpoints usually come with metamodels that could be modified or configured for this particular purpose. Metamodels (and related data schemas) can be used to inform the relative completeness and consistency of an architecture description by identifying constituent architecture entities and relationships that are useful and possibly necessary to capture and describe.

- d) Capture rationale for selection of framework(s), viewpoints, templates, metamodels and model kinds.
- e) Define purpose and scope of each model and view to be developed.
- f) Select, modify or develop supporting modeling methods and tools.
- g) Select, develop or modify architecture frameworks, modeling templates and metamodels to be used for generating the necessary models and views.
- h) Select, modify or develop relevant metamodel specifications and templates.

#### **10.4.4 Develop models and views of the architecture(s)**

The extent and variety of the models and views should be limited to those models and views that adequately address the specified concerns, questions and the purpose, scope, breadth and depth specified in the architecture elaboration plan. The models and views should comply with, or conform to, the selected framework and viewpoint. The models and views should fulfill stated stakeholder desires for specific views to be produced.

- a) For each viewpoint, define the architectural context and boundaries in terms of interfaces and interactions with external entities to establish the purpose and scope of the viewpoint.
- b) For each viewpoint, identify the kinds of entities and their relationships to be modeled that will address key stakeholder concerns and address architecture objectives.

NOTE 1 Elements in the model of the architecture will represent things that exist as particular and discrete units. They are neither necessarily tangible objects nor visible items. Examples of modeled things include such items as organizations, facilities, activities, roles, personnel, techniques, processes, policies, rules, principles, objectives, capabilities, nodes, links, systems, system elements, interfaces, data elements, layers, protocols, hardware items, software items, etc.

- c) Modify the metamodels and view templates associated with selected architecture frameworks to ensure the relevant modeling elements and relationships can be accommodated.
- d) Identify specific entities and relationships between these entities to be modeled that will address key stakeholder concerns and address architecture objectives.
- e) Allocate relevant concepts, properties, characteristics, behaviors, functions, features or constraints to the entities and relationships to be modeled.

NOTE 2 This allocation could have been done in the Architecture Conceptualization process. If so, this allocation is checked to see if it is still relevant and of sufficient detail to use here. If not, more details are added as appropriate for the intended use of the elaborated architecture description.

- f) Identify relevant principles and precepts that will guide evolution of the architecture entity.
- g) Select, adapt or develop models of the architecture.
- h) Compose views from the models in accordance with identified viewpoints to express how the architecture addresses stakeholder concerns, architecture principles and precepts, and architecture objectives, and meets stakeholder and system requirements.
- i) Identify risks and opportunities that have become apparent in the architecture views and modify the architecture to address significant risks and opportunities where appropriate.
- j) Harmonize the architecture models and views with each other.
- k) Generate or modify the architecture description based on relevant models and views.
- l) Modify the architecture description based on feedback from issue resolution arising during various stages in the development life cycle of the relevant architecture entity.
- m) Place architecture description in the architecture repository.

### 10.4.5 Relate the architecture to other architectures and to relevant affected entities

**NOTE 1** It is common that a new architecture is replacing or subsuming existing or planned design elements. In such a case, the new architecture will be mapped to the design, for example, to help understand the impact of implementing the architecture. It could also be important to map the architecture to other relevant elements such as policies, processes, doctrine, organizations, training, logistics, personnel, facilities, etc.

- a) Identify related entities and other architectures that relate to architecture elements and the nature of these relationships.
- b) Define the interfaces and interactions between the related entities and other architectures with each other and with the architecture being elaborated.
- c) Identify areas for potential reuse of existing architecture elements and the risks associated with this reuse.
- d) Partition, align and allocate requirements to architecture elements and these related entities.
- e) Map related entities and other architectures to relevant architecture concepts, properties and other attributes.
- f) Formulate principles and precepts expected to be used during execution of the life cycle processes for the architecture entity.

**NOTE 2** Architecture information can be used in other systems life cycle processes as specified in [C.1](#).

- g) Formulate principles and precepts for design and evolution of the architecture entity.

### 10.4.6 Assess the architecture elaboration

- a) Assess each architecture view and model against architecture elaboration objectives, purposes and questions.
- b) Validate elaborated architecture against the intent of the conceptualized architecture and identify and pursue resolution of any discrepancies or disconnects.
- c) Assess the suitability of the architecture description.

**NOTE** This assessment of the architecture description is separate from evaluation of the architecture that happens during the Architecture Evaluation process. However, these activities are sometimes done concurrently or in close cooperation since the quality of the architecture description could impact the effectiveness of the architecture evaluation.

- 1) Assess the architecture description to determine if it meets the needs of intended users.
- 2) Assess the architecture description to determine if it is comprehended by potential users and relevant stakeholders.
- 3) Assess the architecture description for consistency, completeness and correctness.
- 4) Assess whether the architecture description fully and accurately reflects the architecture concepts.
- 5) Assess whether architecture views are consistent with the selected viewpoints.
- d) Verify the consistency of architecture work products with descriptions and depictions of the architecture entity or other associated entities of interest (e.g. system design drawings, concept of operations, construction drawings, user manuals, etc.).
- e) Update the architecture description to address identified redundancies, gaps and shortfalls.
- f) Place updated architecture description in the architecture repository.
- g) Place updated architecture description under change control, when appropriate.

- h) Identify views and models that can be generalized for reuse by other projects.

#### **10.4.7 Coordinate use of elaborated architecture by intended users**

- a) Identify users of architecture elaboration information, including relevant architecture descriptions, models and data.

NOTE 1 Users of this information could be those doing evaluation or elaboration of the architecture, those managing the collection so they can gain some understanding of the nature of an architecture, analysts who need to understand the architecture as the basis for their analysis, stakeholders who have particular concerns relevant to certain models and views, managers who need to use the architecture for planning and scoping a project, design engineers who need to use the architecture to get an early start on design and to give feedback on the architecture that will drive their design, and so on.

NOTE 2 Interactions with project processes is delineated in [B.1](#), which could suggest potential users of architecture-related information.

- b) Prepare architecture elaboration information and data and supporting material for use by others.
- c) Maintain the architecture elaboration information and data and supporting material as it is being used by users to clarify intent, correct errors found, tailor views and models for particular uses, and incorporate lower level architecture information where appropriate.
- d) Maintain change control of architecture elaboration-related data items and inform users of relevant changes.
- e) Deliver architecture elaboration information and data to intended users.

NOTE 3 This delivery could occur as a direct delivery to intended recipients, as a posting to the architecture repository, or in a formal release through some organizational release process. The particular mechanism for the delivery could be a factor in determining how best to package the information and what steps need to be taken to validate the information and data before it is released.

- f) Monitor use of architecture elaboration information to collect feedback on the architecture and on the form and contents of the architecture work products.
- g) Communicate architecture elaboration information to interested parties.
- h) Incorporate feedback into the architecture descriptions, views and models.

### **10.5 Work products**

The following work products shall be produced:

- architecture elaboration plan,
- architecture elaboration status report,
- architecture viewpoints,
- model kinds,
- architecture views,
- architecture models, and
- architecture descriptions.

NOTE These work products are usually developed by either the Architecture Conceptualization or Architecture Elaboration processes. However, in some cases where these products can be made in a generalized manner for use by multiple architecting efforts, then these products could be generated instead by the Architecture Enablement process so they can ensure the products are created in such a manner that they are suitable for use in the various relevant situations.

## 11 Architecture Enablement process

### 11.1 Purpose

The purpose of the Architecture Enablement process is to develop, maintain and improve the enabling capabilities, services and resources needed to perform the other architecture processes.

**NOTE** This could involve the acquisition or development of these capabilities, services and resources, as appropriate.

**EXAMPLE** Enabling capabilities include, among other things, procedures, methods, tools, frameworks, architecture viewpoints, work product templates, decision support systems, storage, configuration management and reference models. Enabling services include, among other things, infrastructure, technologies, skilled personnel and automation agents. Enabling resources include, among other things, architecture repository, library, registry, communication channels and mechanisms, human and technical resources, and licenses for tools and methods.

### 11.2 Outcomes

As a result of the successful implementation of the Architecture Enablement process:

- a) Enabling capabilities, services and resources are available when and where they are needed and are accessible by those who need them.
- b) Enabling capabilities are suitable for and accessible to the architecture process activities.
- c) Enabling services are suitable for and accessible to the architecture process activities.
- d) Enabling resources are suitable for and accessible to the architecture process activities.
- e) Personnel have the requisite knowledge and skills for proper use of the enabling capabilities, services and resources.

### 11.3 Implementation

The organization or project shall implement the activities in [11.4](#) (numbered as 11.4.N) in accordance with applicable organization policies and procedures with respect to the Architecture Enablement process. The activities may be performed in any order that is deemed appropriate. The organization should implement the relevant tasks (identified as list items under each 11.4.N activity) as appropriate to the situation.

Architecture enablement work products should be stored in the architecture repository for future reference and audit. The repository should be used to facilitate widespread access, enable auditing and encourage future reuse.

## 11.4 Activities and tasks

### 11.4.1 Prepare for and plan the architecture enablement effort

- a) Identify the enabling capabilities, services and resources needed for support to governance and management of the architecture collection.
- b) Identify the enabling capabilities, services and resources needed for support to conceptualization, evaluation and elaboration of architectures.
- c) Identify the guidelines, policies, strategies and constraints for deploying the enabling capabilities, services and resources.

**NOTE 1** These guidelines, policies, strategies and constraints could be obtained from the architecture governance directives and architecture management guidance.

- d) Identify and define the necessary roles and responsibilities of people involved in the architecture enablement effort.
- e) Plan the architecture enablement effort using the Project Planning process in ISO/IEC/IEEE 15288 as a guide.

NOTE 2 ISO 21500 and ISO 21505 can be used as guidance for planning of the architecture enablement effort.

- 1) Establish the scope of the architecture enablement effort.
  - 2) Establish metrics for the architecture enablement effort.
  - 3) Collect the data and information needed for the architecture enablement effort.
  - 4) Obtain access to capabilities, services and resources needed for the architecture enablement effort.
  - 5) Specify the work products and their outlines to be produced through performance of this process.
  - 6) Identify and define architecture enablement work elements and associated resources.
  - 7) Develop architecture enablement schedule and define associated milestones.
  - 8) Develop necessary control and communication plans for architecture enablement.
- f) Produce an architecture enablement plan that contains the planning information.
  - g) Obtain necessary approvals, funding and resources for the plan.

#### **11.4.2 Monitor, assess and control the architecture enablement activities**

- a) Report architecture enablement activity plans and status.
- b) Monitor and assess whether architecture governance directives and guidance for architecture enablement are being followed.
- c) Monitor and assess whether architecture management directives and guidance for architecture enablement are being followed.
- d) Monitor and assess metrics for the architecture enablement effort.
- e) Identify the enablement issues arising from changes to governance directives and guidance.
- f) Identify the enablement issues arising from changes to management instructions and guidance.
- g) Identify and assess risks and opportunities associated with the architecture enablement effort.
- h) Monitor and assess whether architecture enablers are being utilized properly.
- i) Monitor and assess whether the architecture repository and library are being utilized properly.
- j) Implement corrective actions to modify the work plan or realign the work with the plan where necessary.
- k) Assess and control the architecture enablement effort using the Project Assessment and Control process in ISO/IEC/IEEE 15288 as a guide.

NOTE ISO 21500 and ISO 21505 can be used as guidance for assessment and control.

### 11.4.3 Manage the architecture process enablers

- a) Manage decisions about architecture process enablers using the Decision Management process in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 as a guide.

NOTE 1 ISO 21505 can be used as guidance for strategic decision making.

- b) Manage risks associated with architecture process enablers using the Risk Management process in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 as a guide.

NOTE 2 ISO 31000, ISO 21500 and ISO 21505 can be used as guidance for risk management.

- c) Manage changes to the architecture process enablers using the Configuration Management process in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 as a guide.

- d) Manage the architecture repositories, libraries and registries using the Information Management, Knowledge Management and Configuration Management processes in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 as a guide.

- e) Manage the quality of architecture process enablers using the Quality Management and Quality Assurance processes in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 as a guide.

- f) Manage infrastructure associated with architecture process enablers using the Infrastructure Management process in ISO/IEC/IEEE 15288 or ISO/IEC/IEEE 12207 as a guide.

### 11.4.4 Acquire, develop and establish enabling capabilities, services and resources

- a) Identify capabilities, services and resources that can be leveraged for facilitating achievement of architecture vision, strategy, goals and objectives.

- b) Develop a catalog of capabilities, services and resources that can be used by the other architecture processes and put this catalog in the architecture registry.

- c) Develop and establish an architecture repository that can be used to store architecture-related information and data.

- d) Establish access control measures for the architecture-related information and data in the architecture repository.

- e) Establish access control measures for the architecture-related information and data in the architecture library.

- f) Establish access control measures for the architecture-related information and data in the architecture registry.

- g) Identify and establish reusable architecture frameworks and viewpoints that can be used for conceptualization, evaluation and elaboration of architectures.

- h) Establish work product templates that can be used by the other architecture processes.

- i) Define measurement systems that can aid in measurement of progress in the execution of the architecture processes.

- j) Develop the requisite information and information flows needed for governing and managing the architecture collection.

- k) Select, develop and establish architecture frameworks, architecture description languages, modelling templates and architecture viewpoints that can be used by the architecting processes.

NOTE 1 Architecture frameworks and viewpoints are often key enablers for development of architecture descriptions. Refer to ISO/IEC/IEEE 42010 for requirements and guidance on developing architecture descriptions, architecture viewpoints and architecture frameworks. [Annex H](#) provides information on mapping of processes to architecture frameworks.

- l) Identify areas of improvement and additional training to be provided in order to drive the architecture vision, strategy, goals and objectives.
- m) Establish activities, events and controls that can be applied for architecture enablement.

NOTE 2 Events are milestones, reviews, audits, key decision points, quality gates, etc. Controls are checklist, entry criteria, exit criteria, decision trees, etc.

#### **11.4.5 Deploy enabling capabilities, services and resources**

- a) Deploy the architecture repository for maintaining the architecture work-products.
- b) Align activities, services, resources, capabilities and information cohesively for provision of effective architecture enablement.
- c) Deploy capabilities, services, activities, events and controls for support to the other architecture processes.
- d) Deploy the appropriate organizational resources, capabilities, assets, activities and services for provision of architecture enablement.
- e) Deploy architecture viewpoints, modelling templates and architecture frameworks.
- f) Deploy architecture work product templates.
- g) Deploy information structures and information flows necessary for architecture enablement.
- h) Deploy mechanisms to collect relevant data and information needed for the architecture enablement effort.
- i) Deploy mechanisms to collect relevant data and information needed for managing the architecture repository.
- j) Provide training on enabling capabilities, services and resources.
- k) Certify personnel on requisite knowledge skills for relevant enabling capabilities, services and resources.

#### **11.4.6 Improve architecture enablement capabilities, services and resources**

- a) Identify gaps and shortfalls in
  - 1) architecture frameworks, modelling templates and viewpoints,
  - 2) enabling capabilities, services and resources,
  - 3) mechanisms used to collect data and information “related to” or “used in” architecture enablement,
  - 4) architecture repository contents and structure,
  - 5) architecture library contents and structure,
  - 6) architecture registry contents and structure,
  - 7) work product templates and guidelines,
  - 8) information structures and flows related to performing the other architecture processes,
  - 9) enablement activities, events and controls,
  - 10) architecture governance enablers,
  - 11) architecture management enablers,

- 12) architecture conceptualization enablers,
  - 13) architecture evaluation enablers,
  - 14) architecture elaboration enablers, and
  - 15) decision support systems.
- b) Examine gaps and shortfalls to identify potential improvements.
  - c) Propose improvements to the enabler development activity.
  - d) Monitor development of the improved enablers to ensure the expected improvements are achieved.

## **11.5 Work products**

The following work products shall be developed:

- architecture enablement plan,
- architecture enablement status report,
- architecture framework,
- architecture viewpoint,
- catalog of enabling capabilities,
- catalog of enabling services,
- catalog of enabling resources, and
- architecture work product templates.

## Annex A (normative)

### Tailoring process

NOTE This annex is an adaptation of ISO/IEC/IEEE 15288:2015, Annex A.

#### **A.1 General**

This Annex provides requirements for the tailoring of architecture processes in this document.

NOTE 1 Tailoring is not a requirement for conformance to this document. In fact, tailoring is not permitted if a claim of "full conformance" is to be made. If a claim of "tailored conformance" is made, then this process is applied to perform the tailoring.

NOTE 2 Additional guidance for tailoring can be found in the ISO/IEC/IEEE 24748 series on the application of life cycle processes.

#### **A.2 Overview of architecture processes**

##### **A.2.1 Process usage**

The architecture processes defined in this document can be used by any organization when acquiring, using, creating or supplying a system, as well as when operating, evolving or transforming an enterprise. They can be applied at any level in an enterprise and at any stage in the life cycle of the architecture or associated systems.

The functions these processes perform are defined in terms of specific purposes, outcomes and the set of activities and tasks that constitute the process.

##### **A.2.2 Introduction to process ordering**

Each architecture process in [Figure 1](#) can be invoked, as required, at any time throughout the life cycle. The order that the processes are presented in this document does not imply any prescriptive order in their use. However, sequential relationships are introduced by the definition of a life cycle model. The detailed purpose and timing of use of these processes throughout the life cycle are influenced by multiple factors, including social, economic, organizational and technical considerations, each of which can vary during the life of an architecture. An individual architecture life cycle is thus a complex aggregation of processes that will normally possess concurrent, iterative, recursive and time dependent characteristics. Concurrent use of processes can exist within an organization (e.g. when the architecture is being elaborated at the same time that a architecture is conceptualization is improving), and between organizations (e.g. when an organization unit governs and manages contracted architecting activities).

##### **A.2.3 Process iteration**

When the application of the same process or set of processes is repeated on the same level of an architecture structure, the application is referred to as iterative. The iterative use of processes is important for the progressive refinement of process outputs, e.g. the interaction between successive architecture evaluation efforts can incrementally build confidence in the suitability of the architecture. Iteration is not only appropriate but also expected. New information is created by the application of a process or set of processes. Typically this information takes the form of questions with respect to architecture objectives, stakeholder needs, analyzed risks or opportunities. Such questions should be resolved before completing the activities of a process or set of processes.

## A.2.4 Process recursion

The recursive use of processes, i.e. the repeated application of the same process or set of processes applied to both the whole architecture and successively to several parts of the architecture structure, is a key aspect of the application of this document. The outcomes from one process application are used as inputs to invoked processes at the next recursion level in order to get a more detailed or mature architecture structure. Such an approach adds value to successive architectures in the architecture structure.

## A.2.5 Life cycle stages

The changing nature of the influences on the architecture (e.g. operational environment changes, new opportunities for architecture entity implementation, modified structure and responsibilities in organizations) requires continual review of the selection and timing of process use. Process use in the life cycle can be dynamic, responding to the many external influences on the architecture. The life cycle approach also allows for incorporating the changes in the next stage. The life cycle stages assist the planning, execution and management of architecture processes in the face of this complexity in life cycles by providing comprehensible and recognizable high-level purpose and structure. The life cycle approach also allows for incorporating the changes in the next stage when defined criteria are satisfied.

The discussion in this clause on iterative and recursive use of the architecture processes is not meant to imply any specific hierarchical, vertical or horizontal structure for the architecture, system-of-interest, enabling system, organization or project.

## A.2.6 Process instantiation

Where justified by product quality risks, detailed descriptions of process instances in the context of the specific product may also be created. Instantiation of processes involves identifying specific success criteria for a process instance, derived from the product requirements, and identifying the specific activities and tasks needed to achieve the success criteria, derived from the activities and tasks identified in this document. Creating detailed descriptions of process instances enables better management of product quality risks by establishing the link between the process and the specific product requirements.

Further elaboration of these concepts can be found in the ISO/IEC/IEEE 24748 series on the application of architecture processes over a life cycle.

## A.2.7 Process reference model

ISO/IEC/IEEE 15288:2015, Annex C defines a Process Reference Model (PRM) at a level of abstraction higher than that of the detailed requirements contained in the main text of this document. The PRM is applicable to an organization that is assessing its processes in order to determine the capability of these processes. The purpose and outcomes are a statement of the goals of the performance of each process. This statement of goals permits assessment of the effectiveness of the processes in ways other than simple conformity evaluation.

**NOTE** ISO/IEC 33004 can be used as guidance for specification of the PRM, consideration of process assessment and associated maturity model.

## A.3 Tailoring process steps

### A.3.1 Purpose

The purpose of the Tailoring process is to adapt the processes of this document to satisfy particular circumstances or factors that:

- surround an organization that is employing this document in an agreement;
- influence a project that is required to meet an agreement in which this document is referenced;

- c) reflect the needs of an organization in order to supply products or services.

### A.3.2 Outcomes

As a result of the successful implementation of the Tailoring process:

- a) Modified or new life cycle processes are defined to achieve the purposes and outcomes of a life cycle model.
- b) Justification for tailoring is provided in support of ensuring a successful outcome to the architecture processes.

### A.3.3 Activities and tasks

If this document is tailored, then the organization or project shall implement the following tasks in accordance with applicable policies and procedures with respect to the Tailoring process, as required.

- a) Identify and record the circumstances that influence tailoring. These influences include, but are not limited to:
  - 1) instability of, and variety in, operational environments;
  - 2) risks, commercial or performance, to the concern of interested parties;
  - 3) novelty, urgency, size and complexity;
  - 4) starting date and duration of utilization;
  - 5) integrity issues such as safety, security, privacy, usability, availability;
  - 6) emerging technology opportunities;
  - 7) profile of budget and organizational resources available;
  - 8) non-availability of the services of enabling products, services or other enabling item;
  - 9) roles, responsibilities, accountabilities and authorities in the overall life cycle of the system;
  - 10) the need to conform to other standards.
- b) In the case of properties critical to the architecture entity, take into account the life cycle structures recommended or mandated by standards relevant to those properties.
- c) Obtain input from parties affected by the tailoring decisions. This includes, but may not be limited to:
  - 1) the stakeholders for the architecture entity(ies);
  - 2) the interested parties to an agreement made by the organization;
  - 3) the contributing organizational functions.
- d) Make tailoring decisions in accordance with the enterprise and project management processes to achieve the purposes and outcomes of the selected life cycle model.

**NOTE 1** Organizations establish standard life cycle models as a part of the Life Cycle Model Management process. It is sometimes appropriate for an organization to tailor processes of this document in order to achieve the purposes and outcomes of the stages of a life cycle model to be established.

**NOTE 2** Projects select an organizationally-established life cycle model for the project as a part of the Project Planning process. It is sometimes appropriate to tailor organizationally adopted processes to achieve the purposes and outcomes of the stages of the selected life cycle model.

**NOTE 3** In cases where projects are directly applying this document, it is sometimes appropriate to tailor processes of this document in order to achieve the purposes and outcomes of the stages of a suitable life cycle model.

- e) Select the life cycle processes that require tailoring and add, modify or delete relevant outcomes, activities or tasks.

**NOTE 4** Irrespective of tailoring, organizations and projects can always implement processes that achieve additional outcomes or implement additional activities and tasks beyond those required for conformance to this document.

**NOTE 5** An organization or project sometimes encounters a situation where there is the desire to modify a provision of this document, but modification is a potential source of unanticipated consequences on other processes, outcomes, activities or tasks. Consequently, when necessary, modification is performed by deleting the provision (making the appropriate claim of tailored conformance) and, with careful consideration of consequences, implementing a process that achieves additional outcomes or performs additional activities and tasks beyond those of the tailored standard.

- f) Record the rationale for the tailoring in terms of the reason for tailoring, the architecture process impacted and the nature and extent of such impact.

The nature and extent of such impact shall be captured in terms of the change, restriction or removal of architecture processes. It may be expressed in terms of processes, outcomes, activities and work products.

- g) Provide explanation for why each life cycle process was tailored or why each outcome, activity or task was deleted.
- h) Provide sufficient information for future determination of the purpose and rationale for making these decisions.

## Annex B (informative)

### Defining metrics for architecture processes

#### **B.1 General**

It would be difficult to provide a definitive list of process metrics that would be applicable in all situations. Therefore, this annex provides guidance on the information that can be considered when process metrics particular to an organization are defined.

Two definitions can be considered for metrics:

- A composite of two or more measurements resulting in a value that defines a characteristic of the process. [SEI]
- A quantifiable entity that allows the measurement of the achievement of a process goal. Metrics should be SMART—specific, measurable, actionable, relevant and timely. Complete metric guidance defines the unit used, measurement frequency, ideal target value (if appropriate) and also the procedure to carry out the measurement and the procedure for the interpretation of the assessment. [ISO/IEC 33001]

#### **B.2 Guidelines for developing architecture process metrics**

##### **B.2.1 Metrics for governance effort**

- Business resources indicators like key performance indicators, key business indicators, value chain, financial measures, business excellence scores.
- Architectures status and evolution/intentions with regards to business objectives.
- Associations amongst the architectures of interest: models identifying commonalities, differences and dependencies.

##### **B.2.2 Metrics for management effort**

- Indicators allowing monitoring of architecture tasks and enabling planning evolution include:
  - Scope (breadth of coverage, level of details, partitioning characteristics).
  - Schedule (time period, delivery schedule dates, provision for risk, etc.).
  - Resource utilization (staffing availability, manpower loading limitations, facility availability dates, capacity restrictions, and use of particular materials or reusable hardware or software units).
- Indicators allowing monitoring whether governance directives and guidance are being followed include:
  - Status or progress of management process via critical success factors, and numeric and graphical key performance indicators.
  - Decision-making aids used to help to define actions when variations and trends are pointing out some risks.

### B.2.3 Metrics for conceptualization effort

- Indicators allowing monitoring whether directives and guidance are being followed include:
  - Governance directives and guidance.
  - Management directives and guidance.
- Check-lists to verify if conceptualization techniques, methods and tools are available, and if personnel are trained in their use.
- Status or progress of conceptualization process via critical success factors, and numeric and graphical key performance indicators.
- Indicators of risks and opportunities include:
  - Indicators of risks with regards to likelihood and severity, and opportunities to be anticipated.
  - Monitoring aids used to trigger warning and alert when variations and trends are pointing out some risks.
  - Indicators of opportunities.
  - Dependencies within the process.
  - Dependencies between this process and other processes.

### B.2.4 Metrics for evaluation effort

- Indicators allowing monitoring whether directives and guidance are being followed include:
  - Governance directives and guidance.
  - Management directives and guidance.
- Check-lists to verify if evaluation techniques, methods and tools are available, and if personnel are trained in their use.
- Status and progress of evaluation process via critical success factors, and numeric and graphical key performance indicators.
- Indicators of risks and opportunities include:
  - Indicators of risks with regards to likelihood and severity, and opportunities to be anticipated.
  - Monitoring aids used to trigger warning and alert when variations and trends are pointing out some risks (e.g. unsatisfied stakeholders).
  - Indicators of opportunities.
  - Dependencies within the process.
  - Dependencies between this process and other processes.

### B.2.5 Metrics for elaboration effort

- Indicators allowing monitoring whether are directives and guidance being followed include:
  - Governance directives and guidance.
  - Management directives and guidance.
- Check-lists to verify if elaboration techniques, methods and tools are available, and if personnel are trained in their use.

- Status and progress of elaboration process via critical success factors, and numeric and graphical key performance indicators.
- Indicators of risks and opportunities include:
  - Indicators of risks with regards to likelihood and severity, and opportunities anticipated.
  - Dependencies within the process.
  - Dependencies between this process and other processes.
- Monitoring aids used to trigger warning and alert when variations and trends are pointing out some risks e.g. deviation with respect to architecture description rules, consistency with requirement elaboration process, etc.
- Indicators of opportunities: bypassing techniques, customization, etc.

### B.2.6 Metrics for enablement effort

- Indicators allowing monitoring whether directives and guidance are being followed include:
  - Governance directives and guidance.
  - Management directives and guidance.
- Check-lists to verify if enablement information and communications technology (ICT) resources and tools (e.g. licenses, documentation) are available and used.
- Check-list to verify performance of training, coaching and mentoring.
- Status of enablement process via critical success factors, and numeric and graphical key performance indicators.
- Indicators of risks and opportunities include:
  - Indicators of risks with regards to likelihood and severity, and opportunities to be anticipated.
  - Monitoring aids used to trigger warning and alert when variations and trends are pointing out some risks. For example: users administration, interface consistency with other tools (import/export format), impact of software upgrades (software bloat), storage capacity, overcritical access delay, recurrent software bugs, etc.
  - Indicators of opportunities (e.g. software bypassing, information and communications technology (ICT) infrastructure enhancement, etc.).
  - Dependencies within the process.
  - Dependencies between this process and other processes.

## Annex C (normative)

# Interactions with other processes and uses of architecture

### C.1 Relationship with system and software life cycle processes and stages

The architecture processes should interact with the following categories of system life cycle processes (per ISO/IEC/IEEE 15288) and software life cycle processes (per ISO/IEC/IEEE 12207):

- agreement processes;
- organizational project-enabling processes;
- technical management processes;
- technical processes.

The links are identified in three cases:

- When system or software life cycle processes provide information to architecture-related processes. For example, the validation process in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 may identify interfaces required for validation and fulfillment of constraints related to the architecture.
- When architecture processes provide information to a system or software life-cycle process. For example, the Architecture Elaboration process in this document might provide a validated architecture description package to the design definition process.
- When an architecture process in this document is implemented inside a particular system life cycle process in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207.
- When a life cycle process is realized within an architecture process, such as an example of doing life cycle tradeoffs, competitive designs, etc.

**Table C.1** describes how architecture should be used during each stage of a system life cycle as described in ISO/IEC/IEEE 15288. The relationship between system life cycle stages and these architecture processes is discussed in [A.2.5](#).

When the systems engineering process is applied to things like a product line or an enterprise, these system life cycle stages may not be appropriate. For this description, the stages are those described in the INCOSE SE Handbook<sup>[38]</sup> and ISO/IEC/IEEE 24748-1:2018, Figure 5.

**Table C.1 — Architecture use along a system life cycle**

15288 Stages	Architecture usage
Concept	Selling the program, procurement of funding, discovery of needs with stakeholders, opportunity assessment, operational analysis, exploration of system concepts and concepts of operations, exploration of support concepts, problem definition, understand feasibility and alternatives
Development	Requirements definition, concept of operations development, system analysis, system design, verification planning, support system design, organizational interface design, support data mapping, etc.
Production	Problem resolution, production planning, integration planning
Utilization	Operational planning, training of users, logistics planning, defect resolution
Support	Problem resolution, anomaly investigation, evolution planning, define support workflows, support function interactions

**Table C.1 (continued)**

<b>15288 Stages</b>	<b>Architecture usage</b>
Retirement	Reuse planning, decommissioning decision, repurposing analysis

System and software life cycle processes in ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 should use architecture-related information as specified in [Table C.2](#).

**Table C.2 — Uses of architecture by ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 12207 processes**

<b>15288 &amp; 12207 Clause</b>	<b>System life cycle process</b>	<b>Uses of architecture by this process</b>
<b>6.1</b>	<b>Agreement processes</b>	
6.1.1	Acquisition	Basis of supplier evaluation.
6.1.2	Supply	Basis of solution to be supplied.
<b>6.2</b>	<b>Organizational project-enabling processes</b>	
6.2.1	Life cycle model management	Identification of systems and system elements, system transition points.
6.2.2	Infrastructure management	Identification of infrastructure needed by systems.
6.2.3	Portfolio management	Identification of systems and system elements, system transition points, system inter-dependencies.
6.2.4	Human resource management	Determination of necessary knowledge, skills and expertise.
6.2.5	Quality management	Identification of systems and system elements.
6.2.6	Knowledge management	Identification of architecture features to be used for tagging information items in knowledge repository, management of architectures and architecture information.
<b>6.3</b>	<b>Technical management processes</b>	
6.3.1	Project planning	Identification of systems and system elements, system transition points, system attributes and measure, system inter-dependencies.
6.3.2	Project assessment and control	Identification of systems and system elements, system transition points, system attributes and measures.
6.3.3	Decision management	Architecture evaluation recommendations. Identification of systems and system elements, system transition points, system attributes and measures.
6.3.4	Risk management	Identification of systems and system elements, system transition points, system attributes and measures, system inter-dependencies.
6.3.5	Configuration management	Identification of systems and system elements, system interfaces, system configurations and options.
6.3.6	Information management	Identification of architecture features to be used for tagging information items to be managed.
6.3.7	Measurement	Identification of systems and system elements, system transition points, system attributes and measures.
6.3.8	Quality assurance	Identification of systems and system elements, system transition points, system attributes and measures.
<b>6.4</b>	<b>Technical processes</b>	
6.4.1	Business or mission analysis	Understanding of current and planned architectures and related systems.
6.4.2	Stakeholder needs and requirements definition	Understanding of current and planned architectures and related systems. Identification of architecture features and functions.

**Table C.2** (*continued*)

<b>15288 &amp; 12207 Clause</b>	<b>System life cycle process</b>	<b>Uses of architecture by this process</b>
6.4.3	System requirements definition	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions.
6.4.4	Architecture definition	Basis for definition of architecture, architecture conceptualization, architecture elaboration.
6.4.5	Design definition	Basis for design of system and non-system solutions. NOTE Design definition can employ architecture at lower levels within a system hierarchy, e.g. at the system element level.
6.4.6	System analysis	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions. NOTE Problem analysis will occur as part of the Architecture Conceptualization process.
6.4.7	Implementation	Understanding of intended use of architecture-related systems.
6.4.8	Integration	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions. NOTE Integration addresses the composition of systems from their constituent elements and the integration of systems into their operational context/environment. Such considerations are identified, conceptualized and elaborated by architecting. So, architecting can be applied for this purpose.
6.4.9	Verification	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions.
6.4.10	Transition	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions.
6.4.11	Validation	Identification of systems and system elements, system transition points, system attributes and measures. Identification of architecture features and functions.
6.4.12	Operation	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions.
6.4.13	Maintenance	Identification of systems and system elements, system transition points, system attributes and measures. Understanding of current and planned architectures and related systems. Identification of architecture features and functions.
6.4.14	Disposal	Identification of systems and system elements, system transition points. Understanding of current and planned architectures and related systems.

Architecture processes in this document should use system and software life cycle-related information as specified in [Table C.3](#).

**Table C.3 — Information used by ISO/IEC/IEEE 42020 architecture processes**

<b>15288 &amp; 12207 Clause</b>	<b>System life cycle process</b>	<b>Information used by architecture processes</b>
<b>6.1</b>	<b>Agreement processes</b>	
6.1.1	Acquisition	Acquisition plans, supplier evaluation criteria.
6.1.2	Supply	Proposed solution attributes.
<b>6.2</b>	<b>Organizational project-enabling processes</b>	
6.2.1	Life cycle model management	Life cycle models of systems and system elements.
6.2.2	Infrastructure management	Infrastructure features, functions and services.
6.2.3	Portfolio management	Portfolio evaluation criteria, program and project dependencies.
6.2.4	Human resource management	Knowledge, skills and expertise of current or planned personnel.
6.2.5	Quality management	Quality assessment criteria.
6.2.6	Knowledge management	General information from knowledge repository.
<b>6.3</b>	<b>Technical management processes</b>	
6.3.1	Project planning	Project plans.
6.3.2	Project assessment and control	Project assessment data.
6.3.3	Decision management	Architecture-related decisions, decision criteria.
6.3.4	Risk management	Identification of system risks and risk mitigation plans.
6.3.5	Configuration management	Identification of configuration items. Baselined requirements and requirements changes (proposed and actual).
6.3.6	Information management	General information from information repository.
6.3.7	Measurement	Measurement parameters and values.
6.3.8	Quality assurance	Quality assurance criteria.
<b>6.4</b>	<b>Technical processes</b>	
6.4.1	Business or mission analysis	Business needs, gaps and shortfalls. Mission needs, gaps and shortfalls.
6.4.2	Stakeholder needs and requirements definition	Identification of stakeholders and their concerns. Prioritization of concerns. Definition of perceived needs, gaps and shortfalls.
6.4.3	System requirements definition	Proposed and approved system requirements. Identification of key requirements constraints, conditions and challenges.
6.4.4	Architecture definition	Architecture plans and roadmaps. Architecture descriptions.
6.4.5	Design definition	Design plans, tools and roadmaps. Design descriptions. Design evaluation results.
6.4.6	System analysis	System analysis results. System analysis tools, methods, capabilities and limitations.
6.4.7	Implementation	Implementation plans and roadmaps. Identification of key implementation constraints, conditions and challenges.
6.4.8	Integration	Integration plans and roadmaps. Identification of key integration constraints, conditions and challenges.
6.4.9	Verification	Verification plans and roadmaps. Identification of key verification constraints, conditions and challenges. NOTE Verification approaches can impose issues such as architecture controllability and observability.
6.4.10	Transition	Transition plans and roadmaps. Identification of key transition constraints, conditions and challenges.
6.4.11	Validation	Validation plans and roadmaps. Identification of key validation constraints, conditions and challenges.
6.4.12	Operation	Operations plans and roadmaps. Identification of key operations constraints, conditions and challenges.

**Table C.3 (continued)**

<b>15288 &amp; 12207 Clause</b>	<b>System life cycle process</b>	<b>Information used by architecture processes</b>
6.4.13	Maintenance	Maintenance plans and roadmaps. Identification of key maintenance constraints, conditions and challenges.
6.4.14	Disposal	Disposal plans and roadmaps. Identification of key disposal constraints, conditions and challenges.

## C.2 Relationship with enterprise processes

One or more organizations participate in an enterprise to perform architecture processes to improve the ability for achieving mission and business objectives and desired outcomes.

[Table C.4](#) describes how architecture should be used during each stage or phase of life cycle of an enterprise or any of its entities. For this description, the phases are those described in ISO 15704, which are called GERA (generalized enterprise reference architecture) life cycle phases, for any enterprise or entity.

**Table C.4 — Architecture use along an enterprise life cycle**

<b>Phases</b>	<b>Architecture usage</b>
Identification	Selling the venture, procurement of funding.
Concept	Discovery of needs with stakeholders, identification of the enterprise assets. NOTE     Architecture can be used to structure or restructure the enterprise itself, e.g. conceptualization.
Requirements	Finding and definition of the requirements, identification of enterprise projects.
Design	Enterprise analysis, preliminary and detailed design, definition of enterprise projects.
Implementation	Problem resolution, production planning, integration planning, set up of enterprise projects.
Operation	Operational planning, training, logistics planning, monitoring of enterprise projects, anomaly investigation, problem resolution, evolution planning.
Decommissioning	Reuse planning, decommissioning decision, repurposing analysis.

The architecture processes will interact with the enterprise engineering process (specified in ISO 15704:2000 A.3.2, on enterprise engineering methodologies), taking into account aspects such as:

- human factors;
- project management;
- economic, financial and commercial considerations;
- new and changing standards.

The links are identified in three cases:

- When enterprise life cycle processes provide information to architecture-related processes. For example, when the enterprise strategy impacts the architecture governance activities.
- When architecture processes provide information to the enterprise life cycle process. For example, the architecture definition process updates the vision of the asset management of the enterprise.
- When an architecture process is implemented inside a particular enterprise life cycle process. For example, when the architecture definition activities are performed to elaborate an enterprise architecture aiming to structure the enterprise itself and its projects.



## Annex D (informative)

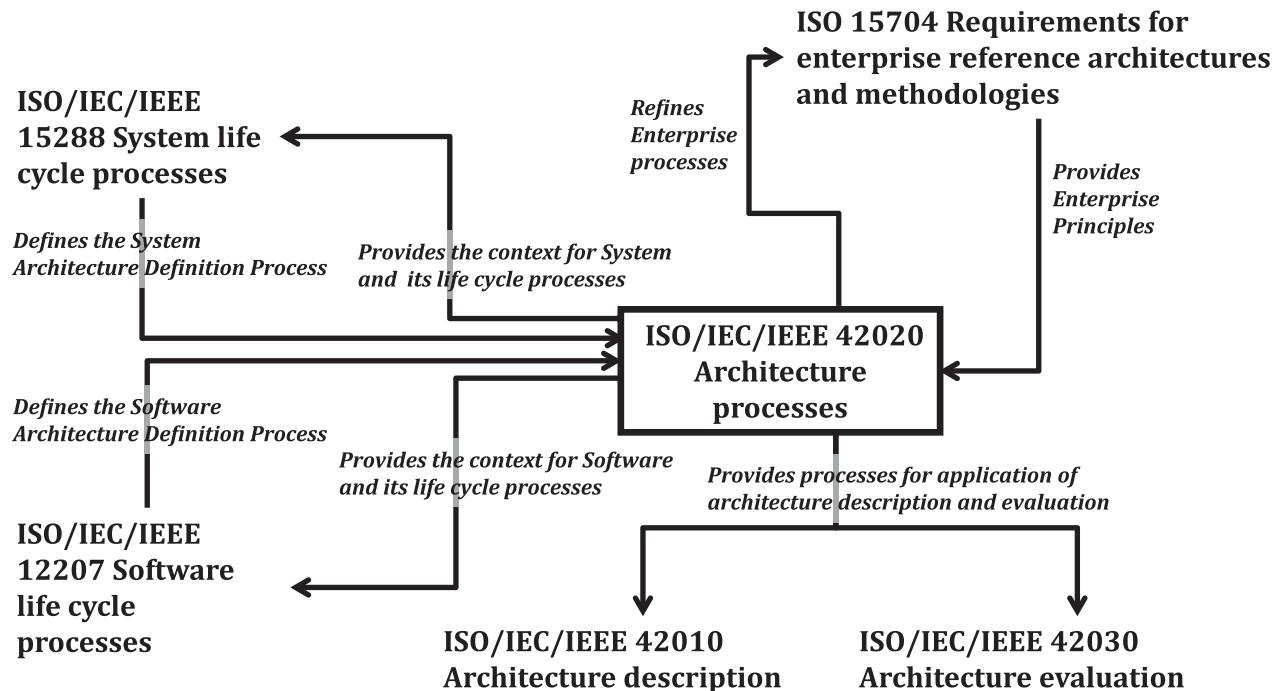
### Relationship with other standards

The standards related to architecture are listed in Bibliography.

**NOTE** The JTC1/SC7 Architecting Guidance Study Report provides the main references about architecture and architecting.

**Figure D.1** describes the main relationships between this document and other ISO standards related to architecture and related activities. This document does the following:

- refines the Architecture Definition process of the ISO/IEC/IEEE 15288 for systems;
- refines the Architecture Definition process of ISO/IEC/IEEE 12207 for software systems;
- frames the processes of these two standards;
- refines the Enterprise Reference Architecture of ISO 15704 and the process description provided in the GERAM annex;
- is considered with the enterprise principles defined by ISO 15704;
- provides processes for application of Architecture Description defined by ISO/IEC/IEEE 42010;
- provides processes for application of Architecture Evaluation defined by ISO/IEC/IEEE 42030.



**Figure D.1 — Main relationships between ISO/IEC/IEEE 42020 and other ISO standards**

## Annex E (informative)

### Notes on terms and concepts

#### **E.1 General**

This annex complements [Clauses 3](#) and [5](#) with additional information about key terms and concepts used in this document.

#### **E.2 Architecture concepts**

##### **E.2.1 Metaphors**

A companion document, ISO/IEC/IEEE 42010, lists six overlapping applications of architecture. In terms of metaphors, the six are: architecture as concept; architecture as property; architecture as blueprint; architecture as literature; architecture as language; and architecture as decision. The present document adds a seventh metaphor: architecture as constraint. The constraint is on the design space available to the design definition process. While it is true that an architecture should be as design-agnostic as possible (see ISO/IEC/IEEE 15288:2015, 6.4.4.1, NOTE 2), at the same time it is also true that there may be reasons for an architecture to constrain the design space. The major reasons for constraint are:

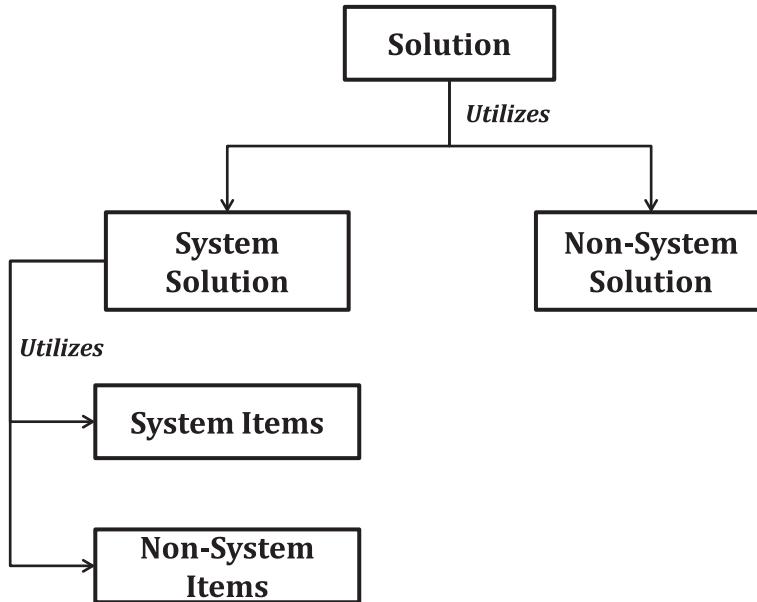
- to close off parts of the design space that are showing to be prone to undesirable emergent properties;
- to avoid limitations on entity evolution - to avoid painting oneself into a corner;
- to avoid limitations on entity re-use;
- to avoid designs that would cause incompatibilities among implementations on platforms or technologies of different capabilities - communication and network architectures, and indeed platform architectures, are cases in point;
- to avoid limitations on the flexibility to respond to future, evolving stakeholder needs.

In many cases, needs for constraining a design space are discovered by the “downstream” processes, particularly (but not only) the processes of implementation, verification, validation and operation. This document's architecture processes are able to respond to these discoveries.

Architectures as constraint are limited by the counter-balancing desire to maximize the available design space. Within its limits, an architecture as constraint can be exhaustive and exceedingly precise. Architectures as constraint also tend to be long-lived, leading to organizational forms such as an architecture maintenance board.

##### **E.2.2 Architecture solution concepts**

Solution is a term used very often in scientific and technical activities. [Figure E.1](#) gives an overview of the solution concepts.

**Figure E.1 — Solution concepts**

A solution is an answer to a problem that addresses concerns of stakeholders.

**EXAMPLE 1** Solution can be business, information technology, mission, capability, service, architecture building block (as defined by the TOGAF framework) and reference architecture (as defined by ISO 15704).

This solution may utilize system solutions and/or non-system solutions.

**NOTE 1** The term “system” is used in ISO/IEC/IEEE 42010 to refer to entities whose architectures are of interest. The term is intended to encompass, but is not limited to, entities within the following domains:

- systems as described in ISO/IEC/IEEE 15288: “systems that are man-made and may be configured with one or more of the following: hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g. operator instructions), facilities, materials and naturally occurring entities”;
- software products and services as described in ISO/IEC/IEEE 12207;
- software-intensive systems as described in IEEE Std 1471:2000: “any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole” to encompass “individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregations of interest”.

**EXAMPLE 2** System solution can be system of systems, class of systems, individual systems and solution building block (as defined by the TOGAF framework).

**EXAMPLE 3** Non-system solutions can be product line, family of systems, data, technology, policy, process and mission thread.

System Solutions may utilize system items and/or non-system items.

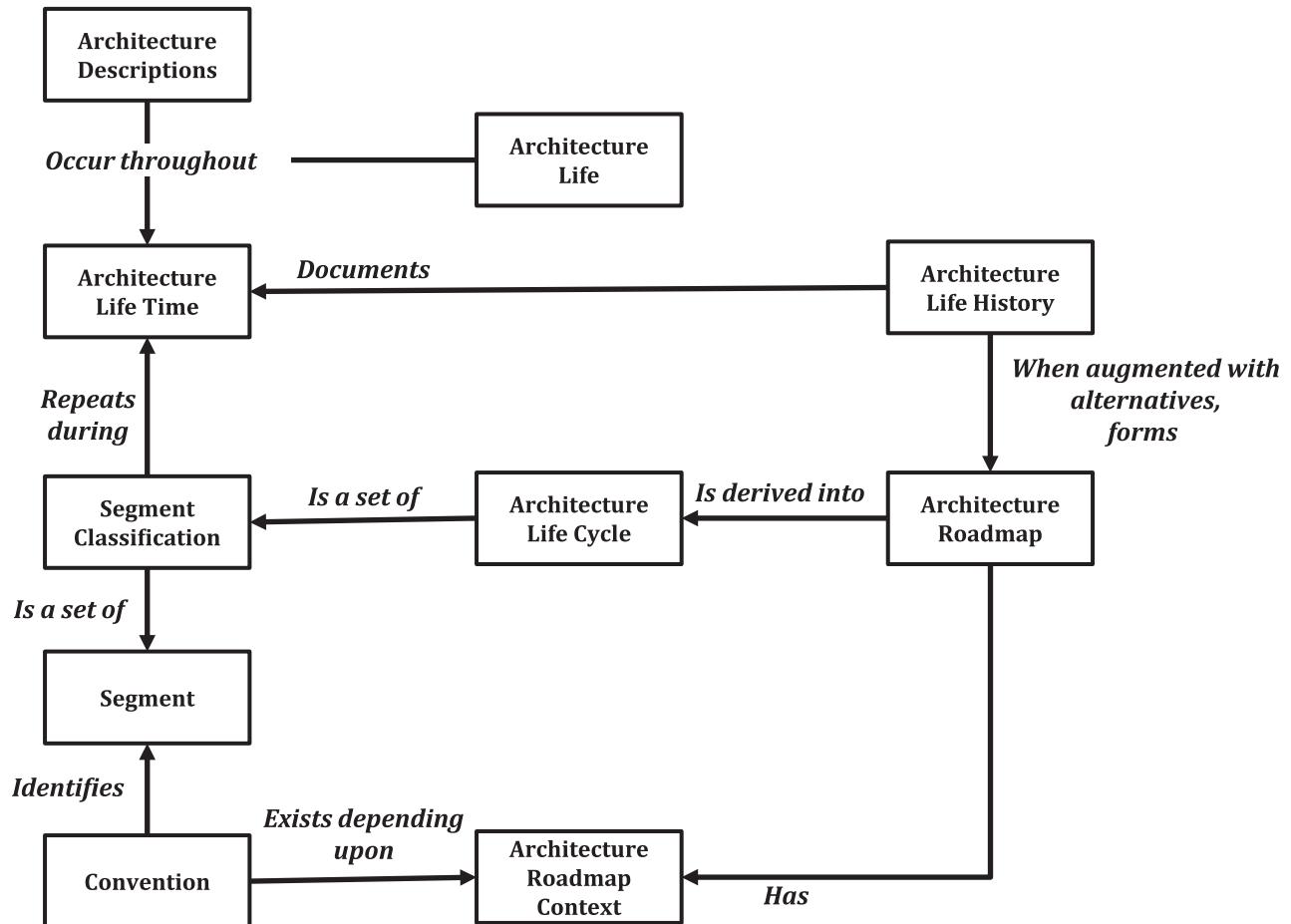
**EXAMPLE 4** System item can be class of systems, system of systems, individual systems, product system, service system and natural system.

**EXAMPLE 5** Non-system item can be hardware item, firmware, software item, database, personnel, role, resource, service and natural resources (e.g. water, air, animals).

**NOTE 2** Architecture can exhibit any part of the solution being considered as entity-of-interest.

### E.2.3 Architecture life concepts

[Figure E.2](#) identifies key architecture life concepts and their relationships.



**Figure E.2 — Architecture life concepts**

Architecture life concepts refer to the characterization of entity architecture as documented by architecture descriptions that occur throughout the entity's life span. Each entity for which an architecture is said to exist, should have a life time described by a life history associated with the evolution of the architecture, in whatever form that reality takes, to end of life. This architecture life history, when augmented with alternatives considered along the way, forms a roadmap for the architecture of the entity described, whether that entity is a single system-of-interest or some grouping of systems as yet lacking particular clarification.

Since many entities may utilize the same architecture, and since that architecture may change over time for particular entities or may be reused at some other time, the architecture itself have a life history distinct from the life history of the entities utilizing that architecture.

To classify commonly encountered segments along that roadmap several conventions exist depending upon roadmap context. Collectively, these conventions identify distinct segments found in most roadmaps as a life cycle, i.e. while a particular entity may progress from one segment to the next during its life span without repetition of a segment, all entities of that particular kind transition through the same set of segments. Life cycle refers to the set of distinct segment classifications that commonly occur within a context even when one or more of those segments repeats during the entity's life span. In this regard, the life cycle metaphor is conceptual since no life cycle instance in nature repeats a metamorphic segment for a particular individual.

Life cycle is a set of distinguishable phases or stages that an entity goes through from its conceptualization until it ceases to be used (See [3.11](#)). Phases are period of time in the life cycle during which activities are performed (See [3.15](#)) while stages are periods within the life cycle that relates to the state of its description or realization (See [3.20](#)).

Two life cycle classification schemes have found utility in International Standards associated with system and enterprise architecture. Independently developed at about the same time, ISO/IEC/IEEE 15288 is the result of work associated with the standardization of engineered systems while ISO 15704 is the result of work associated with the standardization of industrial automation systems. Since initial publication, amendment and revision for both standards now position them for broader and more general application than originally published, the focus of ISO/IEC/IEEE 15288 is still systems and the focus of ISO 15704 is still enterprises.

#### E.2.4 Life cycle models

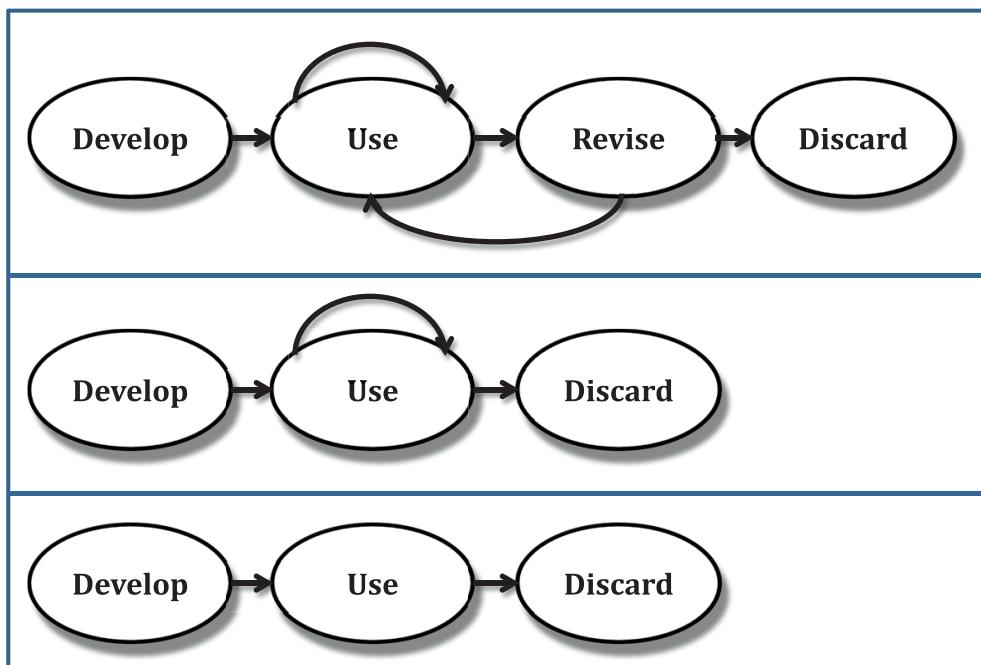
Every entity has an associated life cycle. According to ISO/IEC/IEEE 15288 a life cycle is the “evolution of a system, product, service, project or other human-made entity from conception through retirement”. A life cycle can be described using an abstract functional model that represents the conceptualization of a need for the entity, its realization, utilization, evolution and disposal. Such life cycle models may be defined as a “framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding”.

An entity progresses through its life cycle as the result of tasks performed and managed by people in organizations, using processes and practices for the execution of these tasks. A life cycle model is expressed in terms of processes, their outcomes, relationships and sequencing/concurrency. ISO/IEC/IEEE 15288 defines a set of processes, termed life cycle processes, which can be used to describe a system’s life cycle. Further details may be found in [Annex C](#).

An architecture can also be viewed as having a life cycle that is distinct from the associated entity life cycle.

Every architecture goes through various distinct stages of development, use and revision before it is ultimately discarded. It is therefore useful to think of an architecture as having a life cycle, which is not necessarily aligned completely with any particular entity. This perspective is important because it allows the identification of the appropriate infrastructure needed to manage architecting products. The nature of the architecture life cycle is determined by the purpose of the architecture.

Some typical architecture life cycles featuring varying degrees of use (and revision) are indicated in [Figure E.3](#) below.



**Figure E.3 — Architecture life cycle options**

## E.3 Architecting strategies and approaches

### E.3.1 Architecting strategy

#### E.3.1.1 Introduction to architecting strategy

An architecting strategy defines the starting point for conducting an architecting activity. It defines the background and reasons for trying to form a system concept. While there is no complete list, there are a number of common cases. The situation, the approach and the other points made here are distinct but not independent. Arbitrary combinations will often not make practical sense, but neither does each solely depend on the other.

Note that these strategies are described in terms of being applied to a “system” architecture although they can be also applied to enterprise architectures, as well as to products and services that are not otherwise considered to be systems (e.g. software item).

#### E.3.1.2 New development, (sometimes known as a “greenfield” approach)

This is the classic case where there is no prior system, the system (or other kind of entity) being architected will be all new. In the most extreme case the sponsor will desire a system delivering an unprecedented capability implemented with a technology with limited or non-existent precedent (consider the development of nuclear submarines in the 1950's).

#### E.3.1.3 New product in a product-line

In this case there are pre-existing systems with capabilities and technology quite similar to the desired system. The new system is intended to make a preceded extension of the pre-existing related systems that have been genericized into a product line.

#### E.3.1.4 Legacy evolution

Here the architect has an existing system or collection of systems (the “legacy”) and is tasked to evolve or extend that legacy. The goals may be to add capability, reduce operating cost, eliminate obsolete technology, or something else. A key aspect is that the legacy exists and cannot be abandoned.

An outstanding example of legacy evolution is planned periodic technology refresh such as submarine combat system technology with alternating biennial refresh of the hardware and software baselines in even and odd years to minimize logistics costs for hardware and provide a more capable computing infrastructure to support future evolution.

#### E.3.1.5 Legacy revolution

In this case the legacy exists but the sponsors’ intent is to radically depart from that legacy. The usual case here is that while the legacy works there is a belief that the adoption of a radical departure (usually a radical departure in both technical approach and concept of operations) will enable large changes in capabilities or costs. The situation is not entirely a new development because the legacy capability exists and the “revolutionary” system will presumably have to interface to it in some fashion, but the deliberate intent is to abandon some large fraction of existing infrastructure.

#### E.3.1.6 Incremental start

Here the sponsor wants to build something new, but the uncertainty about what will be the best fit is great. Instead of making a large-scale and irrevocable bet on the future system the intent is to make an incremental step, to be followed by later steps that move toward a superior system. The architecting activity embraces the uncertainty and develops both initial steps and options for subsequent steps that account for known (and possibly unknown) uncertainties in user demand or technology.

### **E.3.1.7 Product-line start**

This is a particular case of a new start where the intent is to build a product-line rather than a single system. The result should be a genericized solution used to produce an indeterminate number of future, related systems, related by reliance on a common base of design, technology, or production, and possibly variable elements.

## **E.3.2 Architecting approaches**

### **E.3.2.1 General**

Several different approaches to architecting (i.e. conceptualization, evaluation and elaboration) exist. These are categorized primarily in terms of the strategy (starting points) for forming the system architecture. The approach which should be adopted depends upon the complexity of the system of interest, its novelty, realization mechanisms and/or the uncertainty in the stakeholder needs.

Note that a particular system architecting effort sometimes requires the use of more than one of these approaches.

Some examples of architecting approaches are given in the following subclauses.

### **E.3.2.2 Forward**

In forward architecting the architecting process moves from consideration of the problem space to the solution space, initially at an architectural level of consideration. Within forward architecting several more specific approaches may be employed, either singly or in combination, for example top-down or bottom-up. Architecture evaluation is also likely to be employed to ensure that the architecture is sound in meeting its intended purpose.

### **E.3.2.3 Bottom-up**

In bottom-up architecting the starting points are the artifacts, capabilities and/or services that are available and/or realizable which are then composed and formed into a system architecture exhibiting the desired (or desirable) emergent properties.

Note that in studying an existing system there is a clear distinction between producing an architecture description document and comprehending the pre-existing architecture of the system. One may have a clear comprehension of the architecture of the system of interest without having developed a complete document. Conversely, one may have a comprehensive document (albeit one badly written) and still have no clear comprehension of any organizing structure or principles of the system (perhaps because there are no such organizing abstractions).

### **E.3.2.4 Middle-out**

In the middle-out approach an arbitrary level of abstraction in the system hierarchy is used as the starting point. Reasoning about, and architecting of, the system then progresses both upwards (towards the goals) and downwards (towards artifacts/capabilities/services).

### **E.3.2.5 Outer-in**

This approach to system architecting starts at both the top (system goals) and bottom (artifacts/capabilities/services) and works towards the middle. It entails balancing and harmonizing desirable and achievable system properties.

### **E.3.2.6 Reverse**

Reverse architecting is an aspect of reverse engineering for making the architectures of existing (or designed) systems explicit. It involves the extraction, abstraction and presentation of system

information. The devising of “as is” architectures can be devised in this manner. Reverse architecting may address the goals of the existing systems, their components or both. Architecture evaluation may form part of reverse architecting in ascertaining for example as to whether an existing architectural solution is suitable for continued usage.

#### **E.3.2.7 Top-down**

In top-down system architecting the starting points are the system goals. The approach proceeds through conceptualization to form the system architecture, stopping when appropriate levels of definitional formality and detail have been achieved. The devising of “to-be” architectures generally involves some top-down architecting.

#### **E.3.2.8 Zigzagging**

An approach to decomposition which entails moving from the functional domain to the physical domain (and the process domain) since, according to axiomatic design, decomposition of functional requirements and design parameters (and process variables) cannot be achieved by remaining in a single domain.

### **E.3.3 Useful architecting mechanisms**

#### **E.3.3.1 General**

Various mechanisms can be used to devise a system architecture. Some commonly used mechanisms are described in the following subclauses.

#### **E.3.3.2 Reference architectures**

Reference architectures are defined in [E.4.1.4](#). They can be instantiated and specialized to yield potential architecture solutions.

#### **E.3.3.3 Architectural Patterns**

Architectural patterns are patterns as applicable to architectures and architectural artifacts; they tend to be more specific in scope or aspect than reference architectures. They are more abstract in nature and have broader applicability than design patterns. Patterns may be implemented as a set of tactics.

They apply to functional components and modules, their structure, organization and interaction. They include rules and guidelines for organizing the relationships between the constituent elements, serving as templates for functional composition. They may be used to identify commonality and re-usability in systems and to deliver key functionality.

#### **E.3.3.4 Tactics**

Tactics are primitive techniques architects employ to achieve particular system characteristics or enshrine these characteristics in architecture principles. They are simpler and more primitive than patterns and are abstract in nature. Architectural patterns and styles incorporate and are implemented using multiple tactics. Tactics may be employed, for example, to ease modifiability of a system or to ensure performance levels are achieved.

#### **E.3.3.5 Heuristics**

Within the scope of this document, heuristics are guidelines for architecting. They are derived from experience. Not all heuristics apply in all circumstances. Heuristics exist for different aspects of architecting such as partitioning functionality, aggregating functionality, evaluating systems architectures, etc.

## E.4 Architecture kinds, views and styles

### E.4.1 Architecture kinds

#### E.4.1.1 General

Different kinds of architecture can be considered according to their purpose, domains of application and roles within entity and architecture life cycles. Architecting may require the use (including development and/or application) of architectures of several kinds.

**NOTE** Descriptions expressing a viewpoint, an aspect, an abstraction level or a perspective are sometimes called architecture kinds. Examples are security architecture, logical architecture and physical architecture.

In discussing architecture kinds it is important to keep in mind the distinction in ISO standards (especially ISO/IEC/IEEE 42010) between architecture descriptions (as documents) and architectures (as conceptual things).

Some commonly used architecture kinds are given in the following subclauses.

#### E.4.1.2 Enterprise architecture

An enterprise architecture is either the architecture of an enterprise<sup>[46]</sup> or the architecture of an entity from an enterprise point of view. In both cases, it has a defined overall business objective and may include one or more participating organization.

Enterprise architecture is the organizing logic for business processes and information technology infrastructure reflecting the integration and standardization requirements of the company's operating model. The operating model is the desired state of business process integration and business process standardization for delivering goods and services to customers. See Innovating in Information Systems, Weill 2007<sup>[44]</sup>.

#### E.4.1.3 Overarching or strategic architecture

An overarching architecture provides a strategic architectural context for a collection of entities and their associated architectures, including the interactions between these entities and any dependencies between the architectures. It concentrates on high-level objectives at the level of capabilities, systems of systems, or portfolios of projects and of necessity addresses such considerations at a comparatively high level of abstraction given the breadth of coverage.

**NOTE** Notion of overarching architecture is described in the NATO Architecture Framework. A similar architecture kind is called "strategic architecture" in TOGAF.

**EXAMPLE** An overarching architecture can be done for maritime surveillance of a country. This architecture orients programs and projects which occur in this scope, focusing particular domains of activities, like maritime search and rescue.

#### E.4.1.4 Reference architecture

A reference architecture is used by a community of interest as a shared and agreed reference description that can be used for that community's business purposes. It is usually generic and is instantiated as architectures specific for individual business purposes. Reference architectures are used to:

- a) aid understanding of the forms of likely solutions to problems within a particular domain, and
- b) maximize the possible commonality in forms of solutions to similar problems within such a domain.

When the entity of interest is intended as a generalized case to be used as a guide for use in the architecting effort, then this kind of architecture is sometimes called a "reference architecture". In this case, the entity itself is not intended to be instantiated but is used as the basis for creating a realizable architecture of some entity that is a more concrete example of the abstract reference entity. An example

of a reference architecture of this kind is the Open Systems Interconnection model (OSI model) specified in ISO/IEC 7498-1.

**NOTE** Notion of reference architecture is described in many architecture frameworks, engineering methodologies and guides.

**EXAMPLE** See OASIS Reference Architecture for Service Oriented Architecture

#### **E.4.1.5 Domain architecture**

A generic, organizational structure or design for systems in a domain (based on IEEE Std 1517-1999).

**NOTE** The domain architecture contains the architectural forms that are capable of satisfying requirements within a specific domain. A domain architecture 1) can be adapted to create designs for systems within a domain, and 2) provides a framework for configuring assets within individual systems.

#### **E.4.1.6 Baseline architecture**

Baseline architecture is the definition of the architecture being defined for a given point of time. They serve as synchronization key-points for review and provision of architectures as references. Particular cases are:

- Current architecture (or “as-is” architecture) is the definition of the architecture currently in use
- Target architecture (or “to-be” architecture) gives the expected definitive definition.

**NOTE** A definition of target architecture is provided below.

#### **E.4.1.7 Target architecture**

Target architecture is a description of an envisioned architecture concerning the ultimate evolution of the entity of interest.

#### **E.4.1.8 System architecture**

System architecture addresses the architecture of a system. It can be defined for one or more epochs according to a roadmap.

**NOTE 1** System is used here to cover anything studied with a systemic approach.

**NOTE 2** System architecture definition will apply the directives given by the relevant Overarching Architecture, if any.

**NOTE 3** A system architecture can be derived totally or partially from one or several Reference Architectures.

**NOTE 4** If the system architecture is defined for one or more epochs according to a roadmap, there is one (or several) system Baseline Architecture and one system Target Architecture.

#### **E.4.1.9 Product line architecture**

According to ISO/IEC/IEEE 24765, a product line is:

- from the commercial viewpoint, a group of products or services sharing a common, managed set of features that satisfy specific needs of a selected market or mission;
- from the engineering viewpoint, a collection of systems that are potentially derivable from a single domain architecture.

Architecture description of a product line formulates a set of products addressing similar problems and exhibiting an appropriate degree of architectural and solution commonality.

#### **E.4.1.10 System of systems architecture**

The architecture of a system that meets the criteria for a “system-of-systems”. Systems of systems will be distinguished from large but monolithic systems by the operational and managerial independence of their components, their evolutionary nature, emergent behavior and a geographic extent that limits the interaction of their components to information exchange. See Architecting Principles for Systems-of-Systems, Maier 1996<sup>[45]</sup>.

#### **E.4.1.11 Product-service system architecture**

The architecture of a system that meets the criteria for a “product-service system”. A product service-system will be a system of products, services, networks of “players” and supporting infrastructure that continuously strives to be competitive, satisfy customer needs and have a lower environmental impact than traditional business models. The product/service ratio in this system will vary depending on the function fulfillment or economic value. The architecture of the product-service system will be capable of jointly satisfying a user’s need while taking into consideration the shift from techno-productive dimension (focus on functionality) to the social and cultural dimension (focus on value).

**NOTE** Service-Oriented Architecture (SOA) is sometimes identified as an architecture kind but is more accurately termed an architecture style. In fact, any of the architecture kinds considered can be defined with a service orientation.

#### **E.4.1.12 Data architecture**

A data architecture is composed of models, policies, rules or standards that govern which data is collected, and how it is stored, arranged, integrated, and put to use in data systems and in organizations.

### **E.4.2 Architecture views**

#### **E.4.2.1 General**

Architecture views express the architecture from the perspective of specific concerns about the architecture entity. Typically, several such (complementary) views are formed during architecting. Some commonly used views are given in the following subclauses.

#### **E.4.2.2 Contextual (view of) architecture**

The contextual view of an architecture deals with contextual factors and drivers including, for example, PESTEL (political, economic, social, technical, environmental or legal) aspects or description of DOTMLPFI (Doctrine, Organization, Training, Material, Leadership, Personnel, Facilities, Interoperability) aspects.

#### **E.4.2.3 Conceptual (view of) architecture**

The conceptual view of an architecture provides the main ideas or concepts from outside of the architecture. This provides the fundamental view that can be used as a starting point for further architecture efforts.

#### **E.4.2.4 Functional (view of) architecture**

ISO/IEC/IEEE 42010 conformant architecture description document will normally contain a view capturing the functions of the system of interest (it is not a 42010 normative requirement but would almost always be included). A functional architecture can be said to be the essential or organizing functional structure, the way abstracted inputs are transformed into outputs by the system to achieve its mission.

#### **E.4.2.5 Logical (view of) architecture**

The logical view of an architecture in an architecture description is typically an integrated model of both the system's functions and retained data, where the abstraction level is chosen to be directly relevant to users rather than implementation. The logical view of the architecture defines data as it makes sense in the problem domain, and defers definition of how it can be represented to other views. So, for example, the logical view would define data and functional transformations in terms of positions, currency amounts, or objects of user interest and not in terms of XML records or database fields.

**NOTE** What constitutes a logical or physical architecture is often somewhat subjective and a spectrum of such architectures can be so employed.

#### **E.4.2.6 Physical (view of) architecture**

A physical view of the architecture description is an arrangement of system elements and physical interfaces that provides the design solution for a product, service or enterprise, and is intended to satisfy logical architecture elements and system requirements.

#### **E.4.2.7 Technical (view of) architecture**

A technical view of the architecture description is defined in terms of technical recognizable objects that compose a systems implementation and the interfaces among them. It is implementable through technologies.

#### **E.4.2.8 Organizational (view of) architecture**

The organizational view represents the responsibilities and authorities on all entities identified in the other enterprise views (processes, information and resources). It caters for the structure of the enterprise organization by organizing the identified organizational units into larger units such as departments, divisions, sections, etc. See ISO 15704:2000 A.3.1.5.3.2.

The organization-related aspects have to do with decision level, responsibilities and authorities, the operational ones relate to the capabilities and qualities of humans as enterprise resource elements. See ISO 15704:2000 A.3.1.1.

### **E.4.3 Architecture styles**

#### **E.4.3.1 General**

An architecture (or architectural) style is a set of principles and/or a generic pattern that provides a canonical guidance for architecting. It can be defined by the architecture elements, their topological layout, connectors and interaction mechanisms, and applicable constraints. Architecture styles may describe deployment patterns, structure and design issues, and communication factors. Their use improves structuring and understanding (through the use of established mechanisms and vocabularies), promotes design reuse (through the development of entities based upon proven forms of solution) and supports the consideration of pertinent technical issues.

**NOTE** Architecture styles are distinct from the notion of architecting styles. Architecting styles refer to ways of architecting which are codified according to the architecture's primary purpose including its extent of influence. See Styles of Architecting, Evans 2014<sup>[46]</sup>.

Some examples of architecture styles particularly as applicable to computer-based systems are given in the following subclauses.

#### **E.4.3.2 Client server**

This architecture distributes data and processing physically across different types of system element. Servers provide specific services such as printing, data management, etc. Clients call on these services. Networks allow clients to access servers.

#### **E.4.3.3 Component-based architecture**

This style decomposes system functionality into reusable cohesive functional or logical components that expose well-defined communication interfaces.

#### **E.4.3.4 Data-driven architecture**

Data-driven architectures are concerned with the acquisition, manipulation and dissemination of data. They may be considered as being composed of pipelines of filters (which perform functional transformations of input data to produce data output) and pipes (which convey streams of data).

#### **E.4.3.5 Event-driven architecture**

Event-driven architectures promote the production, detection, consumption of, and reaction to events, where an event is a significant change in system state. Event-driven systems comprise event emitters (or agents), event consumers (or sinks) and event channels.

#### **E.4.3.6 Layered architecture**

Layered architectures hierarchically structure functionality as several layers of increasing abstraction typically ranging from a problem focus at the top level to realization considerations at the lowest level. Interaction between layers is often restricted to adjacent layers.

#### **E.4.3.7 Object-oriented architecture**

An object is a collection of functions (called methods) and associated data. Object-orientation is an analysis and design paradigm based on the division of responsibilities for a system into individual reusable and self-sufficient objects, each containing the data and the behavior relevant to the object.

#### **E.4.3.8 Publish-subscribe oriented architecture**

Publish-subscribe is a style of information exchange in which certain elements offer data to other elements through published messages. Elements requiring such input data subscribe to the relevant messages.

#### **E.4.3.9 Repository architecture**

In this style of architecture the information exchange between system elements is physically realized through a central data repository which can be accessed (and where appropriate, contributed to) by all system elements.

#### **E.4.3.10 Service-oriented architecture (SOA)**

SOA is an architecture style that supports the service-orientation paradigm by exposing (and consuming) functionality from distributed systems as independent services in the form of stateless functions and using contracts and messages. SOA promotes reuse at the macro (service level) rather than micro (e.g. object) level.

Any of the architecture kinds described above can be defined with a service orientation. With this approach, services are preferred to expressed outcomes and interactions between entities (actors and constituents of the solution), with consideration of various operational, system, application and technical views.

**NOTE** Entities can support multiple architecture styles (heterogeneous architecture) for example to support different architectural concerns (information centricity, process/flow-orientation) but this increases solution complexity. The use of a single architecture style (homogeneous architecture) could ease design but could compromise certain required (or desired) system capabilities

## E.5 Architecture motivation model

Enterprise activities, including architecture ones, will be driven by a set of motivation elements:

- business aspiration including vision, goal, objectives and mission;
- business means including strategy, policies, rules and guidance;
- business constraints including laws, regulation and influencers;
- existing and expected business assets including products, tools and people.

**EXAMPLE** An example of an architecture motivation model is provided by OMG<sup>[34]</sup>.

These motivation elements can be used to form a dashboard for monitoring of the architecture activities:

- Aspiration elements are elaborated and used by governance.
- Means are drivers for management.
- Constraints and assets have to be considered for any process.

Criteria are derived from the dashboard for analysis and assessment of architectures activities and of the architectures themselves.

## E.6 Quality

### E.6.1 General

This clause provides information on quality attributes, quality models and quality measures that could be useful in applying this document. The following items are covered in this clause:

- What is “Quality”?
- Architecture quality attributes.
- Boehm’s quality models and ontology.
- Standards on System and Software Quality Requirements and Evaluation (SQuaRE) – the ISO/IEC 25000 family.

The inclusion of these items does not imply endorsement of these particular ways of addressing quality. Exclusion of other items is not intended to imply their shortfalls. The intention is to include those items that can be related to the conceptual elements in this document.

### E.6.2 What is “Quality”?

The importance of the concept of “Quality” during architecting has been underscored in numerous articles, journal papers, case studies, books, and in practice. “Quality” is a subjective term for which each person or sector has its own definition. There is a wide variety of opinions on the nature of “quality”; some of these are summarized below:

- Quality has a pragmatic interpretation as the non-inferiority or superiority of something.
- Quality is the characteristics of a product or service that bear on its ability to satisfy stated or implied needs<sup>[42]</sup>.
- Quality is a product or service free of deficiencies<sup>[42]</sup>.
- Quality means “fitness for purpose”<sup>[51]</sup>.

- Quality means “conformance to requirements”<sup>[49]</sup>.
- Quality means “uniformity around a target”<sup>[53]</sup>.
- Quality means “the loss a product imposes on society after it is shipped”<sup>[53]</sup>.
- Quality in a product or service is not what the supplier puts in. It is what the customer gets out and is willing to pay for<sup>[50]</sup>.
- Degree to which a set of inherent characteristics fulfills requirements<sup>[2]</sup>.
- Quality is “excellence of the system in a chosen dimension and is the basis for satisfying its stated purpose”<sup>[55]</sup>.
- Quality characteristics of a system are a set of essential and distinguishing attributes that have a pragmatic interpretation of the system’s inferiority or superiority<sup>[54]</sup>.

In business, engineering and manufacturing, quality has a pragmatic interpretation as the non-inferiority or superiority of something; it is also defined as fitness for purpose. Quality is a perceptual, conditional and somewhat subjective attribute and may be understood differently by different people. Consumers may focus on the specification quality of a product/service, or how it compares to competitors in the marketplace. Producers might measure the conformance quality, or degree to which the product/service was produced correctly<sup>[51]</sup>.

Support personnel may measure quality in the degree that a product is reliable, maintainable or sustainable. A quality item (an item that has quality) can perform satisfactorily in service and is suitable for its intended purpose<sup>[52]</sup>.

### E.6.3 Architecture quality attributes

Architecture quality attributes are the extent to which the architecture can deliver value to its stakeholders. It is a set of essential and distinguishing attributes that have a pragmatic interpretation of the architecture’s inferiority or superiority. It is a function of:

- a) architecture process outcomes,
- b) impact of the architecture on various stakeholders,
- c) measure of extent of achievement of stakeholder concerns, and
- d) measure of capabilities of the architecture.

While ATAM does deal with quality attributes, these are attributes of the architecture entity, not the architecture itself. See more on ATAM in [Annex D](#).

Architecture quality attributes are the overall factors that affect behavior, structure, design and experience of architectures. They represent areas of concern that potentially impact the structure and behavior exhibited by the realized system. The extent to which the architecture handles a combination of quality attributes indicates the success of the architecting effort and overall quality of the realized system. The taxonomy for each architecture quality attribute would be:

- a) Measures: The parameters by which the attributes are measured.
- b) Factors: Policies and mechanisms of the system and its environment that impact the stakeholder concerns.
- c) Methods: Techniques for addressing concerns and processes for realizing the quality attributes during productions.

While conceptualizing architecture to address the architecture quality attributes, it is necessary to consider potential impact of each of the quality attributes on other stakeholder concerns. While tradeoff analysis techniques aid architects in prioritizing architecture quality attributes, architectural tactics

describe techniques to achieve particular system characteristics or enshrine these characteristics in architecture principles. The importance of each architecture quality attribute depends on the context and the stakeholder's concerns for which the specific architecture is conceptualized. [Table E.1](#) provides an example list of architecture quality attributes.

**Table E.1 — Architecture quality attributes**

Quality Attribute	Description
Coherence <sup>[56]</sup>	Being logical and consistent
Completeness <sup>[56]</sup>	Ability to form a whole
Elegance <sup>[60]</sup>	Form and function are graceful and stylish
Hierarchy <sup>[61]</sup>	Levels of abstractions
Modularity <sup>[58]</sup>	Separation of concerns
Variability <sup>[57]</sup>	Expandable in preplanned ways
Subsetability <sup>[57]</sup>	Support the production of a subset
Conceptual integrity <sup>[56]</sup>	Architecture unification
Commonality <sup>[57]</sup>	Sharing in preplanned ways
Durability <sup>[59]</sup>	Stand up robustly and remain in good condition
Utility <sup>[59]</sup>	Useful and function well for people
Beauty <sup>[59]</sup>	Delight people and raise their spirits
Robust <sup>[60]</sup>	Strong and not be vulnerable to changes
Feasible <sup>[60]</sup>	Should be able to implement
Flexible <sup>[60]</sup>	Adapt to changing conditions
Verifiable <sup>[60]</sup>	Perform as designed
Traceable <sup>[60]</sup>	Architectural elements can be traced in any direction
Cohesion <sup>[58]</sup>	Forming a unified whole

#### E.6.4 Boehm's quality model and ontology

Boehm et al.<sup>[41]</sup> attempt to qualitatively define quality by a set of attributes and metrics. They utilize a hierarchical quality model structured around high-level quality characteristics, intermediate level quality characteristics and primitive quality characteristics for this purpose. Each of these characteristics contributes to the overall quality level. They consider high level quality characteristics to represent the basic high-level requirements, intermediate level quality characteristics to represent the quality factors (portability, reliability, efficiency, usability, testability, understandability and flexibility), and primitive level quality characteristics to represent the quality metrics (device independence, accuracy, completeness, robustness, consistency, accountability and so on) that measure a given primary characteristic.

Boehm and Nupul<sup>[48]</sup> present an ontology for reasoning about a system's qualities. They espouse the view that functional requirements specify what the system should do and hence it is additive in nature, while non-functional requirements (system qualities) specifies how well the system performs its functions and hence it is multiplicative and system-wide in nature. They utilize a variation of the IDEF5 ontology structure comprising the elements Class, Individual, Referent, Relation, State and Process to express the ontology of system qualities. These class hierarchies are organized in terms of stakeholder value propositions, and child-class system qualities as means for achieving the parent class system quality end objectives. They also espouse the view that class hierarchies do not necessarily maintain one-to-many relationships and there are many cases where many-to-many relationships exist, especially when one or more system qualities impacts one or more top level system qualities. [Table E.2](#) presents a typical upper level of system quality hierarchy.

**Table E.2 — Upper level of system quality hierarchy<sup>[48]</sup>**

Stakeholder value-based system quality ends	Contributing system quality means
Mission Effectiveness	Stakeholders-satisfactory balance of Physical Capability, Cyber Capability, Human Usability, Speed, Endurability, Maneuverability, Accuracy, Impact, Scalability, Versatility, Interoperability
Resource Utilization	Cost, Duration, Key Personnel, Other Scarce Resources; Manufacturability, Sustainability
Dependability	Security, Safety, Reliability, Maintainability, Availability, Survivability, Robustness
Flexibility	Modifiability, Tailorability, Adaptability

## E.6.5 The ISO/IEC 25000 family of standards on quality

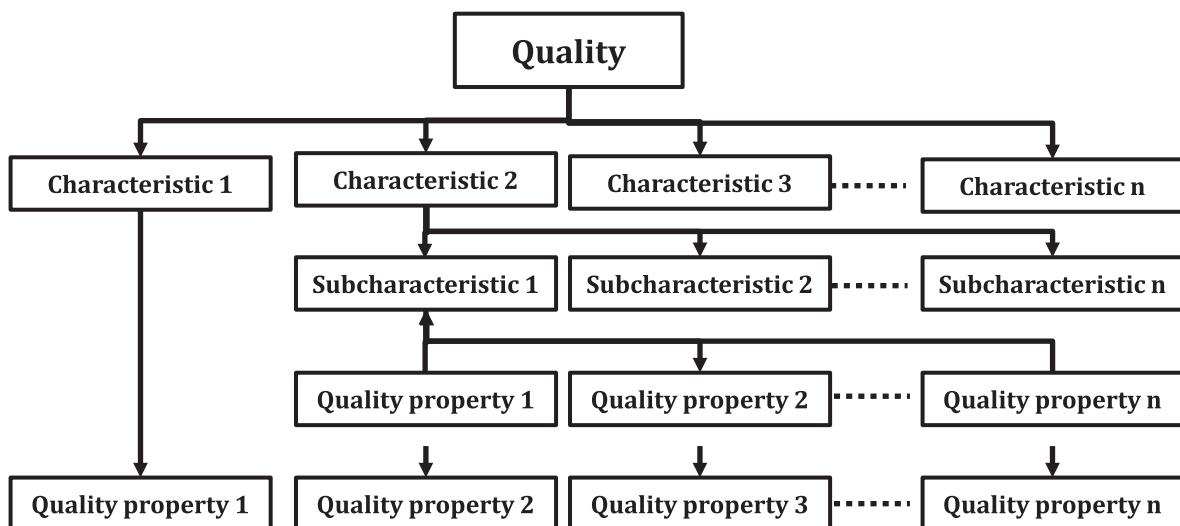
### E.6.5.1 General

The ISO/IEC 25000 family of standards, also known as SQuaRE (System and Software Quality Requirements and Evaluation), has the goal of creating a framework for the evaluation of software product quality. Some of these are discussed in this Annex:

- ISO/IEC 25000, SQuaRE — Quality model framework
- ISO/IEC 25010, SQuaRE — System and Software Quality models
- ISO/IEC 25012, SQuaRE — Data Quality model
- ISO/IEC 25020, SQuaRE — Measurement reference model and guide

### E.6.5.2 ISO/IEC 25000 Quality model framework

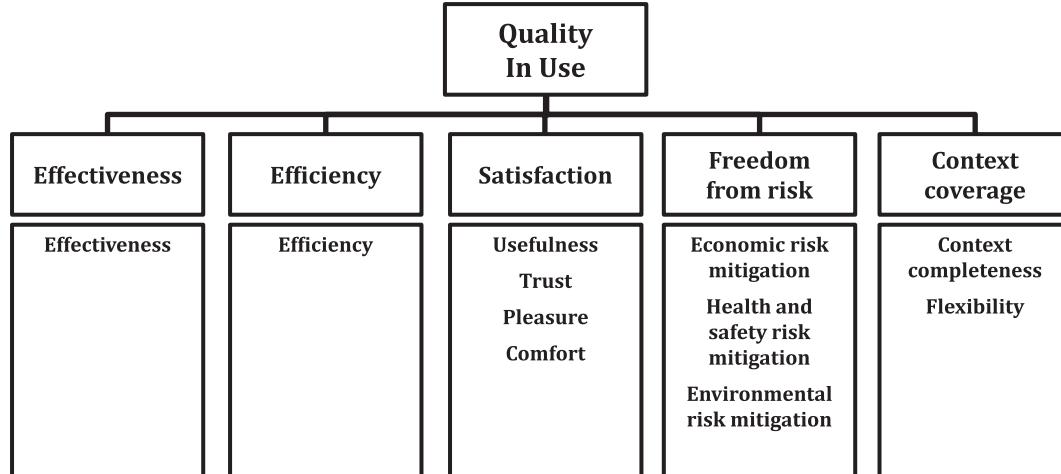
This framework categorizes product quality into characteristics, which in some cases are further subdivided into sub-characteristics. A sub-characteristic in some cases can be divided into sub-sub-characteristics. This results in a quality breakdown structure as illustrated in [Figure E.4](#).

**Figure E.4 — ISO/IEC 25000 Quality model framework**

### E.6.5.3 ISO/IEC 25010 System and software quality models

#### E.6.5.3.1 Quality in use model

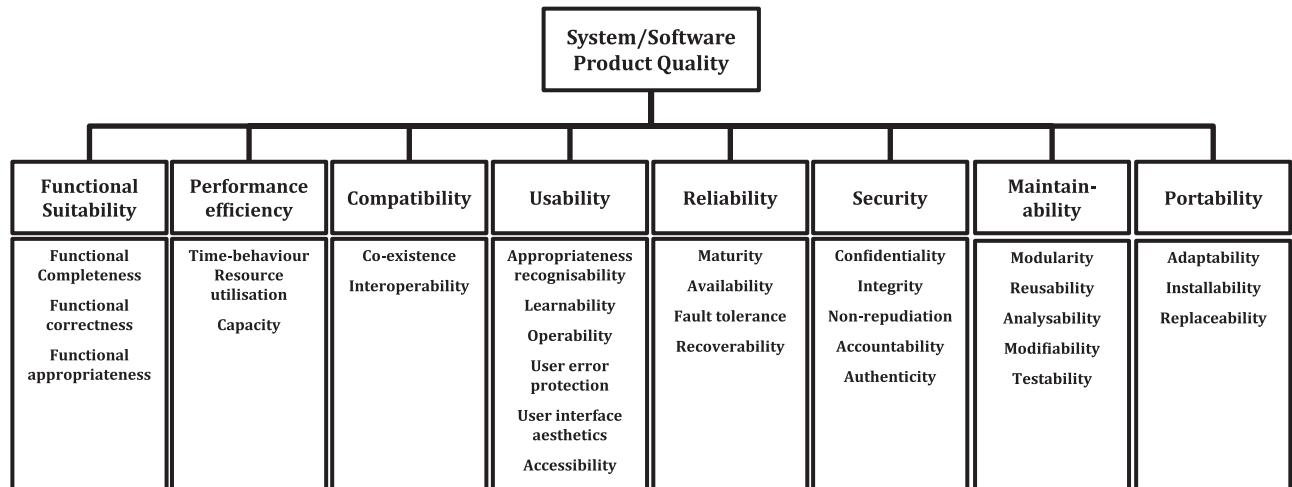
This quality in use model defines five characteristics related to outcomes of interaction with a system. It characterizes the impact that the product has on stakeholders. This model is presented in [Figure E.5](#).



**Figure E.5 — ISO/IEC 25010 Quality in use model**

#### E.6.5.3.2 System/software product quality model

This product quality model categorizes a system/software product quality properties into eight characteristics that focuses on the target system. This model is presented in [Figure E.6](#).



**Figure E.6 — ISO/IEC 25010 System/software product quality model**

#### E.6.5.4 ISO/IEC 25012 Data quality model

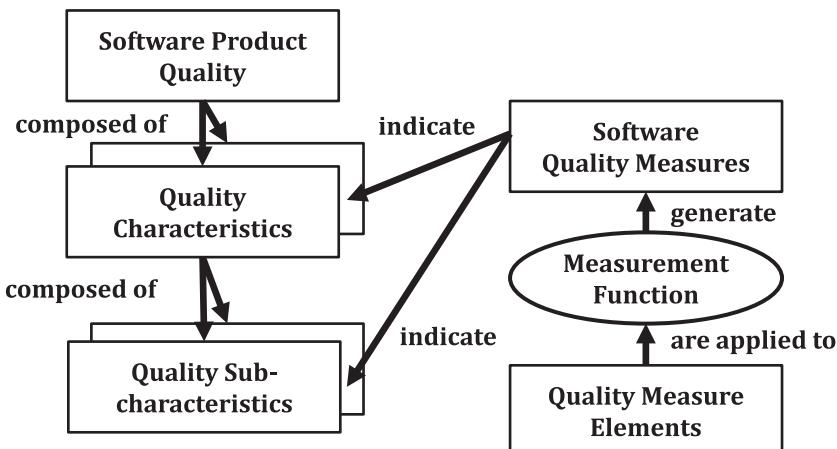
This data quality model, as illustrated in [Table E.3](#), categorizes data quality attributes into fifteen characteristics that are considered from two points of view: inherent and system dependent. While the inherent data quality refers to the degree to which the quality characteristics of data have the potential to satisfy needs when data is used in specified conditions, system dependent data quality refers to the degree to which data quality is reached and preserved within a system when data is used under specific conditions.

**Table E.3 — ISO/IEC 25012 Data quality model**

Characteristics	Inherent	System dependent	Characteristics	Inherent	System dependent
Accuracy	X		Efficiency	X	X
Completeness	X		Precision	X	X
Consistency	X		Traceability	X	X
Credibility	X		Understandability	X	X
Currentness	X		Availability		X
Accessibility	X	X	Portability		X
Compliance	X	X	Recoverability		X
Confidentiality	X	X			

#### E.6.5.5 ISO/IEC 25020 System and software product quality measurement reference model

This product quality measurement reference model describes the relationship between a quality model, its associated quality characteristics (and sub-characteristics), and system and software product attributes with the corresponding software quality measures, measurement functions, quality measure elements and measurement methods. This model is presented in [Figure E.7](#).

**Figure E.7 — ISO/IEC 25020 System/software product quality measurement reference model**

An illustration of this model as given in ISO/IEC 25021 is presented in [Table E.4](#).

**Table E.4 — ISO/IEC 25021 product quality measurement reference model**

SNo	Element	Particulars
1	Quality measure element name	Number of records
2	Objective	To determine data quality of target data
3	Property to quantify	Record is a set of related data items treated as a unit
4	Relevant quality measures	Measure of accuracy
5	Measurement method	Review and analyze data records
6	List of sub-properties	Data Item: Lowest component of a group of data File: A set of related records
7	Input for the quality measure element	Physical files of a database
8	Numerical rules	Adding total records

**Table E.4** (*continued*)

SNo	Element	Particulars
9	Context of the quality measure element	Measure the accuracy and completeness to a group of data
10	Measurement constraints	Verify the impact of technology on the number of records generated for the same information

## Annex F (informative)

# Architecture enablement and process-enabling resources

### **F.1 Architecture enablement**

Architecture enablement is needed for establishing and maintaining consistent practices, standard approaches, reusable items and uniform ways of communication, and for proper utilization of these things. The enablers consist of a collection of tools, techniques, technologies, skills, practices, frameworks, methods and processes used in architecture processes in support of the accomplishment of an organization's objectives. It is implemented for a given area of responsibility to guide the proper selection, development, utilization and improvement of enablers (tools, technologies, approaches and so on). This process maintains information about the various work-products and also provides checks and balances to ensure that the information about these work-products is of high quality.

Architecture enablement has oversight over the performance of architecture process operations and decision making, as well as the efficient organization of people, capabilities, processes and other resources to achieve the architecture goals and objectives. It enables users to quickly understand available information so that they can make better and faster decisions and efficiently achieve architecture goals and objectives. Architecture enablement involves dealing with interactions, interconnections, activities, outcomes and work-products and how they need to be structured to produce the desired governance, management and architecting functions.

Architecture enablement includes the following elements:

- Establishing an architecture repository that provides for the storage and archiving of architecture process artifacts and work products. The repository can be used to store different classes of architecture work products that facilitates coordination and cooperation between the architecture process stakeholders.

**NOTE** The architecture repository referred to in this document is not necessarily isomorphic with architecture repositories in commercial use.

- Establishing an architecture library concerning enabling capabilities, services and resources that can be used by or deemed to be useful to those who perform role specific architecture activities. The library can provide guidelines, templates, patterns and other forms of source material that can be leveraged by the architecture processes.
- Establishing an architecture registry that keeps information about the artifacts and work products contained in the repository and library as well as the changes made to them. The registry can be used to discover and use current and relevant information items for an ongoing architecture endeavor.

### **F.2 Architecture process enabling resources**

Examples of resources that can be used in performing the architecture processes are listed below.

- Architecture patterns: A pattern addresses a specific architecture problem, in a context, to provide a solution maximizing reuse and permitting a range of tradeoffs.
- Architecture kinds: Several kinds may be needed to fully express the essential properties and concepts (see [E.4.1](#)).
- Architecture styles: An idiom for organizing an architecture to achieve certain properties (see [E.4.2](#)).

- Model kinds: Conventions for a type of modeling, to address specific types of concerns (see ISO/IEC/IEEE 42010).
- Architecture description languages: Any defined form of expression for use in architecture descriptions (see ISO/IEC/IEEE 42010).
- Architecture viewpoints: Work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns (see ISO/IEC/IEEE 42010).
- Architecture frameworks: Conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders (see ISO/IEC/IEEE 42010).
- Architecture methods: Practice, technique, or procedure with rules to guide architecture processes.
- Skills and knowledge associated with specific roles identified to perform architecture-related activities:
  - The skills required by each role.
  - The depth of knowledge required to fulfill the role successfully.
- Norms and standards associated with the activities and the work products.
- Tools and languages sustaining the activities and allowing the formulation of work products and their related information.

NOTE Catalogs can be used to collect homogeneous sets of metadata, resources and related information. The repositories can be implemented with catalogs used as references for governance, management and usage.

## Annex G (informative)

### **Architecture governance and management**

#### **G.1 Architecture governance**

Architecture governance is needed for consistent management, cohesive standards and policies, proper guidance, uniform processes and appropriate decision-rights. It is implemented for a given area of responsibility to ensure proper oversight and accountability. This enables the organization to identify, manage, audit and disseminate all information related to architecture decisions, management actions in response to these decisions, contracts affecting the architecture(s) and implementation of architecture changes.

Architecture governance has oversight of the architecture objectives for the architecture collection to ensure their consistency with organizational goals and objectives, among other things. Each set of architecture objectives is considered with respect to factors of maintenance, servicing and upgrade with minimal disruption of the everyday operations. In particular, consideration is given to available internal and external resources in order to determine when general resources can be adapted for specific needs and to determine where specific solutions can be generalized to support wider re-use.

Architecture governance is the practice and orientation by which architectures are managed and controlled at an organization-wide level. It includes the following:

- a) Formulating directives and guidelines of all the architectural components and processes, to ensure effective conceptualization, evaluation, elaboration, implementation and evolution of the collection of architectures within the organization.
- b) Ensuring compliance with industry and governmental standards and regulatory obligations.
- c) Establishing processes that support effective adherence to the directives, guidelines, policies, standards and other regulatory obligations.
- d) Developing practices that ensure accountability to the governance board for the architectural decisions for a collection of architectures.

Architecture governance is typically performed at higher levels of the organization providing oversight over business units, programs, projects, etc. Architecture governance has responsibilities for legal compliance, alignment with organizational goals and objectives, optimum utilization of resources, maintaining focus on the long term vision, responding to changes in the marketplace and user community, anticipating new forces and scenarios that will likely arise, maximizing shareholder gains, etc.

Architecture governance acts on an architecture collection in order to check the alignment between them and for compliance with the organizational mandates and expectations. Usually the collection consists of several architectures that are related to each other, but the collection could consist of a single architecture if appropriate.

Each activity is governed by principles. An organizational authority should be in charge of checking that the activities are performed according to these principles. This authority is sometimes called a "Design Authority", identified for governance according to architecture principles with an escalation approach when necessary.

## G.2 Architecture management

Architecture management is needed for centralized management of current and proposed collection of architectures. Architecture management establishes, maintains and uses a coherent set of guidelines, principles and management regimes that provides direction and instructions for the design and development of an architecture. Architecture management involves identifying potential risks and mitigating them in accordance with the governance directives. The objectives of architecture management are to determine, manage and control the risks, processes and resources necessary for architecting the collection of architectures while taking into account constraints, conflicts, strategic objectives and governance directives.

Architecture management is concerned with managing the architectures, not the activities of architecting. The main focus of architecture management is on managing the implementation and evolution of the architecture(s) to maximize alignment with strategic goals and objectives. The Architecture Management process does not manage the development of the architecture but rather its evolution and its implementation in the design, build, deployment, operations, maintenance, decommissioning, etc. of one or more systems related to the architecture. Architecture management is responsible for implementing the guidance and direction from architecture governance where this is accomplished by giving management guidance and direction to the other architecture processes. Architecture management provides plans and status to architecture governance on how well the architectures are evolving and being implemented.

Architecture management has oversight over the architectural decisions for the collection of architectures to ensure their consistency. Each set of architecture decisions will be considered with respect to factors of risks, evolution, cost, budget, and with minimal disruption of schedule and effort. In particular, consideration will be given to coordination, communication and control in order to determine when shared resources can be used efficiently and effectively.

Architecture management is the practice by which a set of architectural decisions are managed and controlled at a collection level. Architecture management includes the following elements:

- a) Formulating an architecture management charter that defines the statement of work for a collection- of architectures.
- b) Ensuring compliance with project, industry and governmental quality requirements.
- c) Establishing management hierarchies and management plan that support architectural decision making.

## Annex H (informative)

### **Mapping of processes to architecture frameworks**

#### **H.1 General**

This annex provides information on how the elements of various architecture frameworks relate to the processes in this document. The following frameworks are covered in this annex:

- TOGAF Framework
- Pragmatic Enterprise Architecture Framework (PEAF)
- Generalized Enterprise Reference Architecture and Methodology (GERAM) framework
- Department of Defense Architecture Framework (DoDAF) adjustment to reflect the terminological development
- RM-ODP (Reference Model – Open Distributed Processing) framework

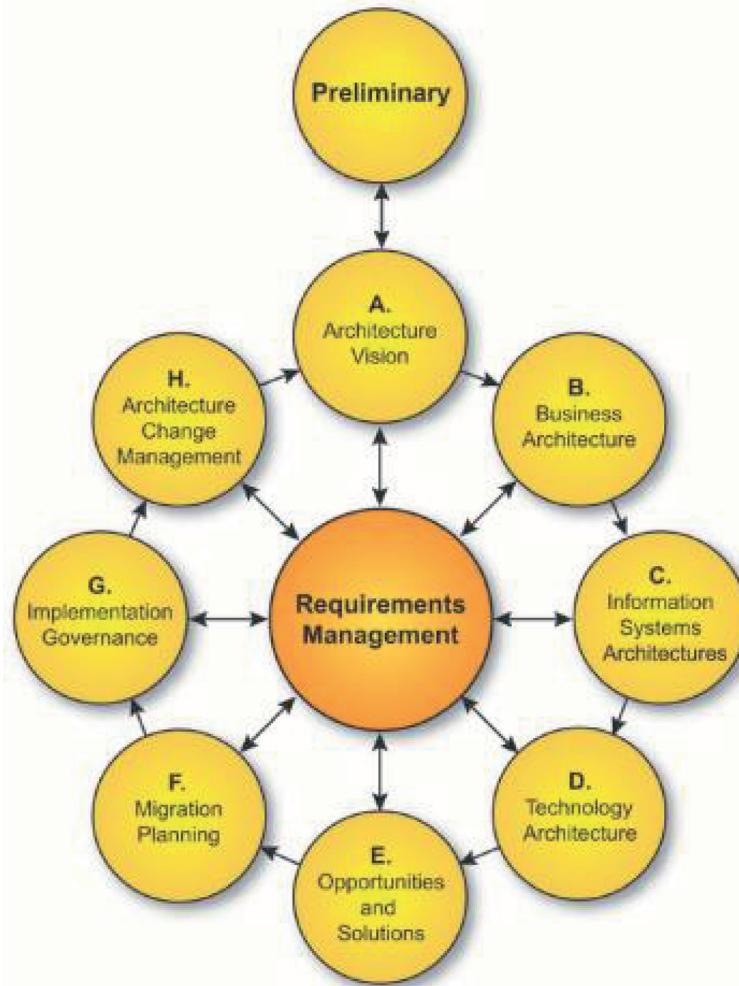
The inclusion of these frameworks does not imply endorsement of these particular frameworks. Exclusion of other frameworks is not intended to imply shortfalls of those frameworks. The intention is to include those frameworks that contain processes similar in nature to the processes in this document so that the relationships can be better understood, or that include elements that can be related to the processes in this document.

Other well-known architecture frameworks like NAF, AUS-DAF, UPDM, UAF and Archimate provide formalisms, and sometimes notations, but do not provide description of architecture processes. These frameworks can be used during architecture elaboration in order to provide a formalized way of describing the architectures. The Architecture Elaboration process as described in [Clause 10](#) does not make any assumption on formalisms or notations.

#### **H.2 TOGAF framework**

##### **H.2.1 Framework overview**

TOGAF<sup>[36]</sup>, an Open Group standard, is a framework that provides a detailed method and a set of supporting tools for developing enterprise architectures. It includes a process for developing the architecture description called the Architecture Development Method (ADM) (see [Figure H.1](#)), as well as general principles for doing architecting and for architecture governance.



**Figure H.1 — Architecture development cycle, TOGAF v9.1, 2011<sup>[36]</sup>**

The ADM (part II of the TOGAF framework<sup>[36]</sup>) together with the rest of the guidance detailed in subsequent parts of the TOGAF framework cover the set of processes and guidance provided in this document as shown in [Table H.1](#).

## H.2.2 Mapping to framework elements

While the coverage mapping is not one-to-one (e.g. several sections of the TOGAF framework map to more than one process provided by this document and vice versa), it is intended as a quick reference for users of the TOGAF framework<sup>[36]</sup> who may wish to document how coverage of the ISO/IEC/IEEE 42020 processes is achieved. An X indicates that full or partial coverage of this ISO/IEC/IEEE 42020 process is achieved via activities as described in the corresponding phase or in Parts III, V, or VII of the TOGAF Framework<sup>[36]</sup>.

**NOTE 1** This document uses the definition for phase from ISO/IEC/IEEE 24765: "a collection of logically related project activities, usually culminating in the completion of a major deliverable." In the TOGAF framework<sup>[36]</sup>, the phases refer to iterative states used to group activities around specific content of the architecture description (e.g. business, technology, etc.).

**NOTE 2** Refinement and updates to architecture governance and architecture management documents can occur during any of the TOGAF phases<sup>[36]</sup>.

**Table H.1 — Mapping of processes to the TOGAF framework<sup>[36]</sup>**

<b>TOGAF Architecture Development Method (ADM) Phase</b>	<b>Architect- ture Gov- ern- ance</b>	<b>Architect- ure Man- age- ment</b>	<b>Architect- ure Con- ceptu- aliza- tion</b>	<b>Architect- ure Evalua- tion</b>	<b>Architect- ure Elabo- ration</b>	<b>Architect- ure En- able- ment</b>
CH 6 Preliminary Phase	X	X		X		X
Phase A: Architecture Vision	X	X	X			
Phase B: Business Architecture		X			X	
Phase C: Information Systems Architectures		X			X	
Phase D: Technology Architecture		X			X	
Phase E: Opportunities and Solutions		X		X	X	
Phase F: Migration Planning	X	X		X	X	
Phase G: Implementation Governance		X		X		
Phase H: Architecture Change Management	X	X		X		
Ch 26 Business Scenarios and Business Goals			X	X		
Ch 27 Gap Analysis				X		
Part V: Enterprise Continuum and Tools, Chs 38-32						X
Ch 46 Establish an Architecture Capability						X
Ch 47 Architecture Board	X					
Ch 48 Architecture Compliance	X	X		X		
Ch 49 Architecture Contracts		X				
Ch 50 Architecture Governance	X	X				
Ch 51 Architecture Maturity Models	X					X
Ch 52 Architecture Skills Framework				X		X
NOTE This table is sourced from TOGAF® <sup>&lt;?&gt;</sup> Version 9.1 <sup>[36]</sup> .						

### H.2.3 Items in the TOGAF framework not addressed in this document

The scope of this document addresses roadmaps, life cycles and baselines; but the Architecture Governance process is limited to cover the relevance of the architecture with regards to application of architecture and consequently does not cover completely the implementation governance.

### H.2.4 Items in this document not addressed in the TOGAF framework

This document is concerned with the architecting effort, including architecture management and dealing with stakeholders for whom the architecture is being developed, evaluation of the architecture, etc.

ADM is only concerned with the steps needed to develop the Architecture Description. To augment this, the TOGAF framework includes additional information in Parts III, IV, and V, specifically, Chapter 26 covers Business Scenarios and Business Goals, and Chapter 27 covers Gap Analysis in Part III, while Part IV covers the Architecture Content Framework, and Part V covers the Enterprise Continuum and Tools.

## H.3 PEAF framework

### H.3.1 Framework overview

The Pragmatic Enterprise Architecture Framework (PEAF)<sup>[31]</sup> is an element of the “Pragmatic Family of Frameworks” (PF<sup>2</sup>) designed to help improve the maturity of how enterprises carry out their business.

PEAF instantiates the methods, artifacts, cultural and environmental sections defined in the “Pragmatic Ontology for Enterprise Transformation” (POET) framework in order to set the context for strategizing and roadmapping enterprise architecture, as part of enterprise transformation.

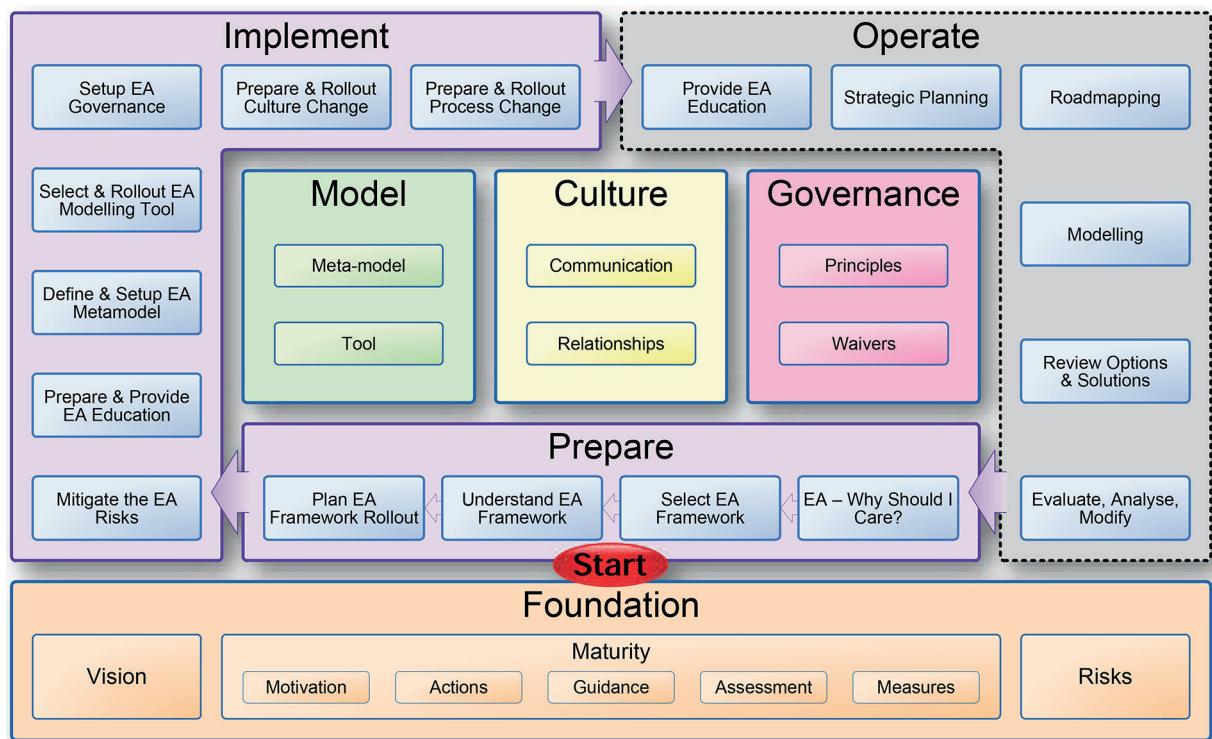
POET defines an ontology for enterprise transformation with information existing at different levels of Idealization/Realization: Motivation, Actions, Guidance, Measures and Assessment. Height fundamental phases of transformation are identified: Strategizing, Roadmapping, Initiating, Elaborating, Transitioning, Using and Governance & Lobbying.

The enterprise architecture context described by PEAF relies on: Processes, Disciplines, Levels, Input and Output. Viewpoints to describe this context are: Contextual, Conceptual, Logical, Physical and Operational.

PEAF artifacts are metamodels used for enterprise planning and governance in order to develop: Business Model, Roadmap Model, Operating Model, Capability Model and Enterprise Context. From these basic models come two aggregate and overlapping artifacts: Enterprise Strategy and Transformation Strategy.

### H.3.2 Mapping to framework elements

PEAF proposes to perform enterprise transformation based on methods, artifacts, culture and environment with actions and foundation described in the following figure.



**Figure H.2 — PEAF functional areas and foundation (PEAF v3, August 2014[31])**

The following table relates the PEAF actions with the processes defined in this document.

**Table H.2 — Mapping of processes to the PEAF framework**

PEAF functional domain	PEAF actions	Architecture Governance	Architecture Management	Architecture Conceptualization	Architecture Evaluation	Architecture Elaboration	Architecture Enablement
Prepare	Strategizing (Why should I care?)						
	Roadmapping (Select EA Framework)						?
	Initiating (Understand EA Framework)			?			
	Elaborating (Plan EA Framework Rollout)		X				
Implement	Constructing (Prepare Culture Change)	X					
	Constructing (Setup EA Governance)	X					
	Constructing (Prepare Process Change)	?					
	Constructing (Prepare Education)	?					
	Constructing (Define & Setup EA Metamodel)			X			
	Constructing (Develop EA Change)			X			
	Constructing (Select EA tools)						X
Operate	Transitioning (Rollout EA Changes: strategic planning; EA roadmapping)	X					
	Transitioning (Provide EA Education)	?					
	Transitioning (Rollout EA Modeling)			X			
	Transitioning (Manage Value and Evolution: Review Options & solutions; Evaluate, Analyze and Modify)				X		

NOTE “?” occurrences show potential extension of the scope of this document.

### H.3.3 Items in the PEAF framework not addressed in this document

The scope of this document does not include rationale data for enterprise architecture setup, culture and education related activities and data, roadmaps and plans for transitioning the enterprise from baseline to target architectures.

### H.3.4 Items in this document not addressed in the PEAF framework

PEAF very lightly addresses architecture management and does not include the elaboration activities. The scope is specifically dedicated to enterprises transformation and does not consider architecture/transformation of any kind of architecture entity.

NOTE 1 Frameworks like TOGAF can complement PEAF for management of enterprise architectures.

NOTE 2 Frameworks like DoDAF, NAF and UAF can complement PEAF with formalism and architecting concepts for architecture entities.

## H.4 GERAM framework

### H.4.1 GERAM framework overview

The GERAM framework is a generalization of frameworks and defines a number of fundamental concepts that any architecture framework needs to cover. GERAM is lightweight in the sense that it defines placeholders for necessary components, but leaves the population of these to the collective development of the enterprise architecture body of knowledge. For example, GERAM defines the concept of methodologies, but acknowledges that depending on the industry domain and a number of other factors there can be many legitimate and useful methodologies.

The requirements that frameworks need to satisfy are the normative part of ISO 15704, and the GERAM framework is an annex that demonstrates how these requirements can be met. Historically, the GERAM framework was developed first, then the normative part of ISO 15704 was extracted from it. Similarly, ISO 15704 defines enterprise entity life cycle (consisting of phases, each being a set of life cycle processes considering the enterprise entity on a given level of abstraction), and life history (consisting of stages in time, similar to stages in ISO/IEC/IEEE 15288): whereupon GERAM defines eight phases, the normative part of ISO 15704 only requires that life cycle phases be defined and leaves that subdivision to individual frameworks.

According to GERAM's philosophy an enterprise is implemented as a socio-technical system of systems, that are embodied in concrete (and sometimes also virtual) enterprise entities, such as business units, corporate headquarters, programs, projects, various entities that implement supporting systems, infrastructure service entities, virtual organizations, networks of organizations, etc.

When the architecture of an enterprise entity is devised, a fundamental (architectural) decision is made about the nature of how the entity implements the system (the way design parameters map to the functions of the system), and the nature (and timing) of this mapping decides various non-functional systemic properties of the implemented system.

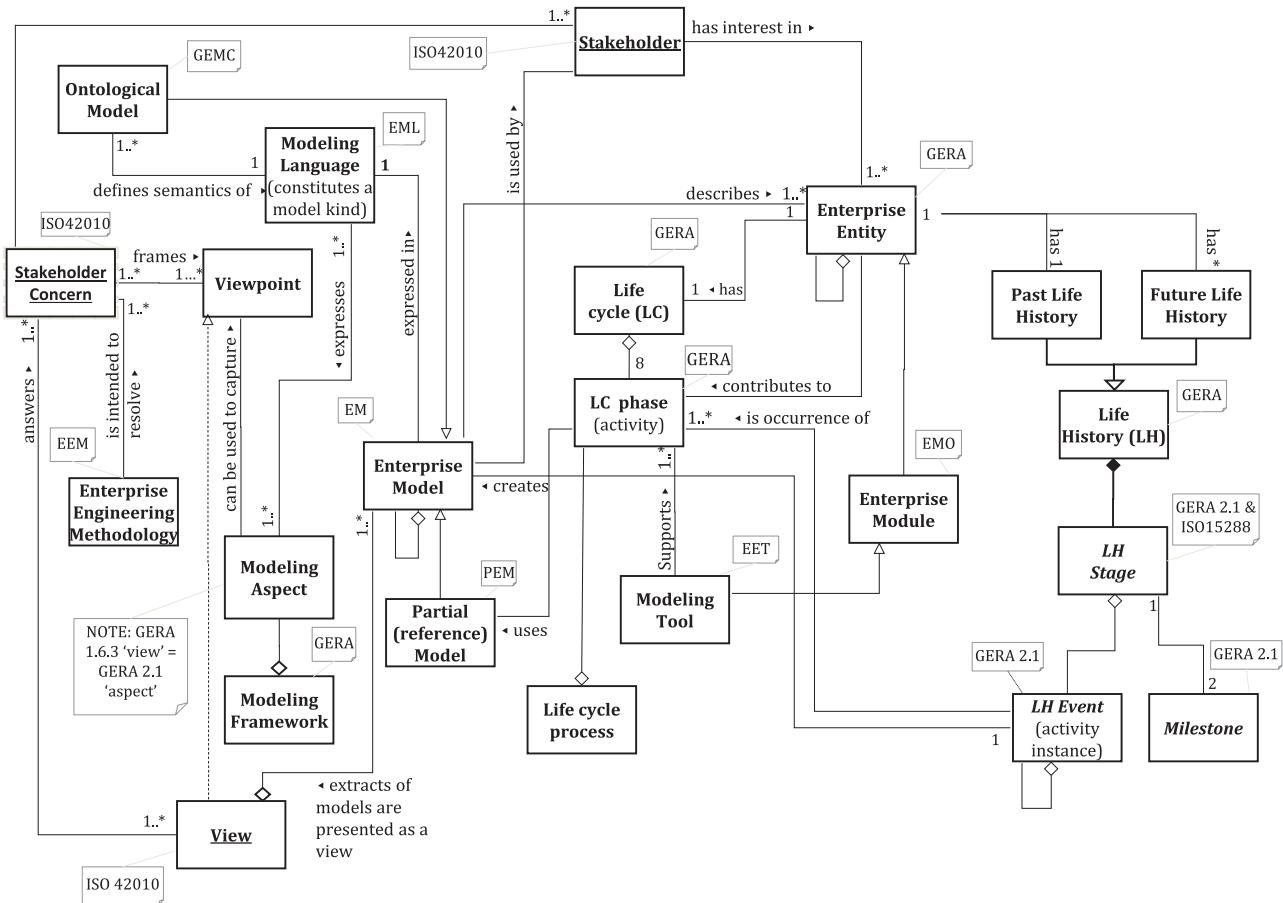
GERAM also defines a modeling framework, defining an open-ended list of "aspects" such as functional, information, resource, organizational, economic, etc. Each "aspect" can be populated by various kinds of models for the purpose of supporting various life cycle processes. We call these "aspects" here to make this description independent from the outcome of current terminological developments both in ISO 15704 and the ISO 42000 family of standards.

The modeling framework defined three categories of models:

- Particular Models (describing an entity of interest),
- Partial (or Reference-) Models (describing reusable models that can be specialized and instantiated to build Particular models), and
- Generic Models (describing the semantics of the models populating the first two).

Generic Models can be defined on various levels of formalization, such as illustrated text, meta-models or formal ontological theories expressed in a suitably selected logic.

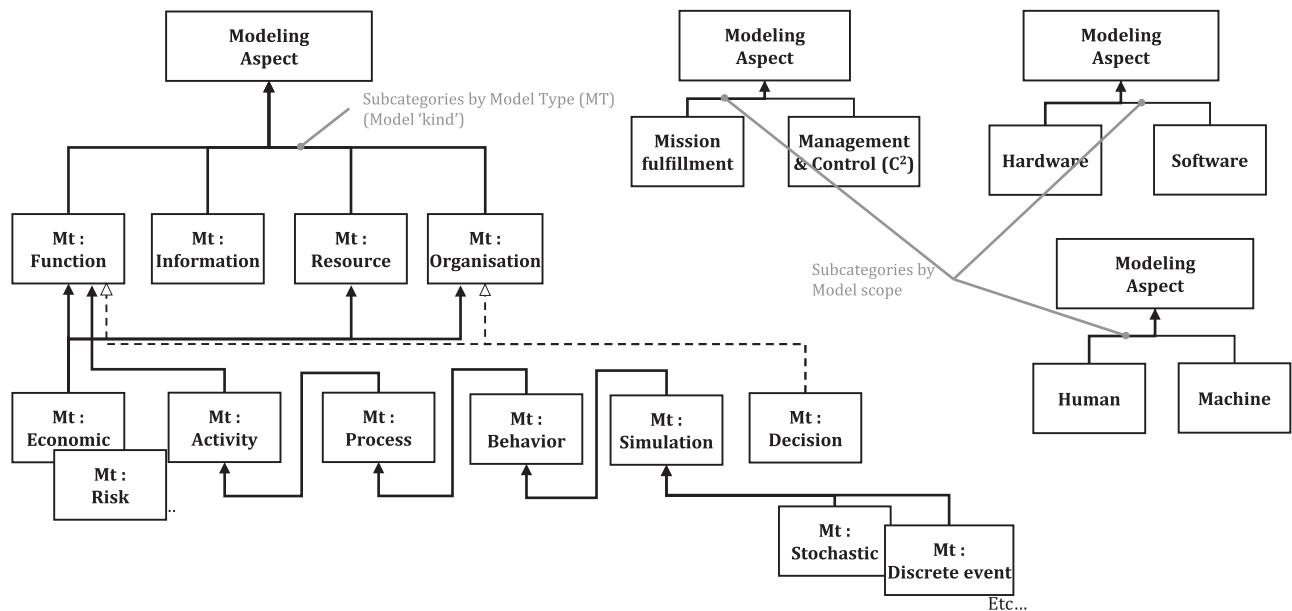
[Figure H.3](#) and [Figure H.4](#) show respectively version 1.6.3 (current) and version 2.1 (being part of the new edition of ISO 15704 under preparation, therefore it is not yet final) of the GERAM meta-model; the basic difference is the adjustment to reflect the terminological development of the ISO 42000 family of standards.



NOTE 1 GERAM metamodel V2.1 elements in italics are not explicitly mentioned in GERAM V1.6.3 text (but implied). Underlined concepts are included to show compatibility with the ISO 42000 family of standards.

NOTE 2 For all relations the participation and cardinality constraint is 0...\* unless otherwise noted.

**Figure H.3 — GERAM1.6.3 and GERAM meta-model V2.1**



NOTE 1 This is an illustration of GERA's typology of models according to model type and scope: a) Multiple categorisations are possible. b) The figure illustrates the detail of the typology models that capture the "Function" aspect. c) The combination of these aspects determines the scope and kind of a model. d) The Model Type determines the types of questions about the entity of interest that the model can be used to answer.

NOTE 2 Aspect was called "view" in V1.6.3.

NOTE 3 NB model scope can span multiple enterprise entities.

**Figure H.4 — GERAM Modeling Aspect concept V2.1**

#### H.4.2 Mapping this document to GERAM framework elements

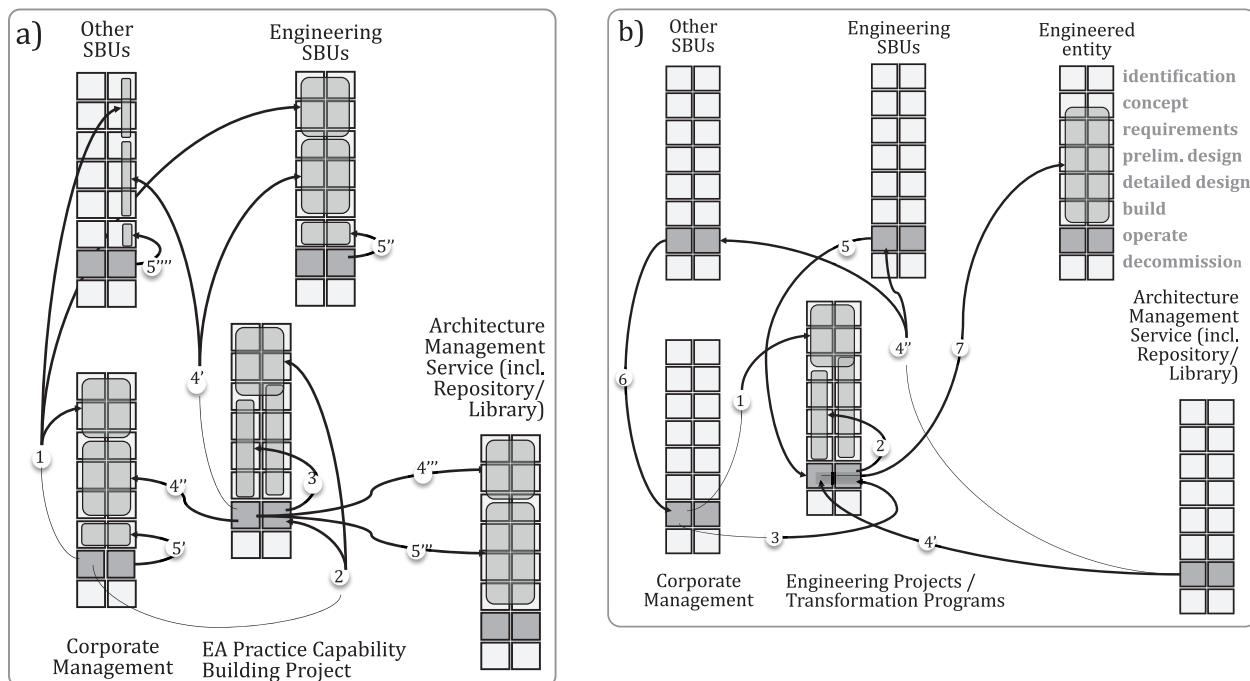
In order to illustrate the role of this document's Architecture Processes (as a reference model) [Figure H.5](#) describes a "dynamic business model" of a typical enterprise (decomposed into its constituent entities). The figure shows enterprise entities, the relationships among their life-cycles and the role of reference models. [Figure H.5 a\)](#) shows the case of EA practice adoption, and [Figure H.5 b\)](#) shows EA practice (including architecture processes) in operation.

When describing or (re)designing various entities in the above, a large number of reference models are used, helping to instantiate various (e.g. process-, information-, organizational-, financial-, decisional-, structural- etc.) designs.

Architecture Processes need to be distributed across the entities in question in two senses:

- As part of the introduction of architecture practices, these process reference models need to be adopted as part of a portfolio activity (possibly through an EA maturity improvement program and its projects). The reference models would be adopted, adapted, particularized and distributed among organizational roles. There exist multiple different ways, depending on whether EA is a centralized function or distributed across business units, supported by some central service, for example. Also, the reference models can either be adopted through policy instruments or through introducing them as standard procedures (depending on the skill levels and experience of the roles among which the processes or parts thereof are distributed). Depending on these decisions the introduction of the ISO/IEC/IEEE 42020 Architecture Processes may only have to be defined on the level of the requirements and preliminary design life cycle phases (without any detailed design being necessary as the rest of the details are to be filled by highly skilled personnel who take the respective roles), or additional detailed design level procedures would be defined, which are then followed by personnel filling the respective roles.

- As shown in [Figure H.5 a](#)), in this establishment stage of EA practice, a typical distribution of Architecture Processes includes activities and tasks being allocated to and rolled out to roles in Corporate Management, in Business Units (SBUs), as well as in Program management, Project mission fulfillment, Project management, as well as possibly external entities (e.g., consulting and other service providers). As shown in [Figure H.5](#), the build life cycle phase is concerned with [Figure H.5 a](#)) establishing the required human architecting, governance and management skills and competencies (through hiring & training individuals, forming committees, and appointing personnel to roles), [Figure H.5 b](#)) selecting, commissioning and deploying of tools (in support architecting, modeling, communication, management and governance) and of their respective repositories.



**Figure H.5 — a) Deploying Architecture Processes (establishment stage) and b) Applying Architecture Processes (operation stage)**

The establishment stage uses this document as a reference model [see [Figure H.5 a](#))]: (1) Corporate Management decides on the need to establish Architecture Processes in its architecture practice, this has consequences to the policies and principles that govern how SBUs (and Corporate Management) do business; (2) Corporate Management defines the mandate of an EA Practice Capability Building Project (and in turn participates in project supervisory capacity) and appoints project management; (3) Project management works out the detail of the project; (4'...4'') the project uses this document as a reference model to design the changes necessary in Engineering SBUs and other SBUs, as well as specifies the need for an architecture management service entity that incorporates architecture management, library and repository services (including technology and human roles), as well as designs the necessary localized processes and organizational roles necessary to build Architecture Governance; (5'...5''') respective entities roll out the above, including the commissioning and deployment of the tools that support architecture processes.

As part of (the operation stage of) EA practice, Architecture Processes are applied. [Figure H.5 b](#)) illustrates the typical life cycle relationships through which corporate strategic portfolio managers and transformation programs can exercise direction (by operating governance, management, control, communication & coordination processes). The figure also illustrates the typical use of the architecting processes proper (including architecture development, elaboration and evaluation).

In the operation stage uses established ISO/IEC/IEEE 42020 processes as follows [see [Figure H.5 b](#))]: (1) any strategic engineering project, or transformation program's mandate is defined by Corporate Management (through its established architecture governance roles), including the mandate to

use architecture definition, elaboration and evaluation processes; (2) Project management defines the details of the project (program), including the use of the mandated processes; (3) Corporate Management through its Architecture Governance processes participates in the supervision of projects/programs; (4'...4'') Architecture management services (including architecture management and supporting services for architecture work) provides operational support to all of the other entities involved; (5) Engineering SBUs participate in these projects/programs, and as part of that participation perform architecture definition, elaboration and evaluation processes; (6) interests of other SBUs not involved in architecture work are still represented by contributing to architecture governance; (7) For example, as the Engineering Project operates it covers part of the life cycle of the Engineered entity (as defined in the project's mandate). As part of this, it develops its architecture(s). (In this example the identification and concept development is assumed to be out of scope – e.g. it was already developed by an acquirer).

#### **H.4.3 Items in GERAM not addressed in this document**

This document does not explicitly address the introduction of architecture processes into architecture practice [see [Figure H.5 a\)](#)]. It is to be noted, that this document is not defining information, organizational or structural models: it is up to the introduction effort to standardize these (or not, and leave the details to be decided on a case-by-case basis in various transformation projects or programs).

#### **H.4.4 Items in this document not addressed in GERAM**

The details of the architecture processes described in this document (as a reference model) are out of scope of GERAM, because GERAM only stipulates the need for reference modes (referring the details to standards such as this document and other industry reference models); GERAM does not prescribe any preferred set. It is part of the introduction of EA practice to identify, evaluate for suitability, select, adapt as appropriate, adopt and finally deploy process reference models that suit best the characteristics of the given industry and organization.

### **H.5 DoDAF framework**

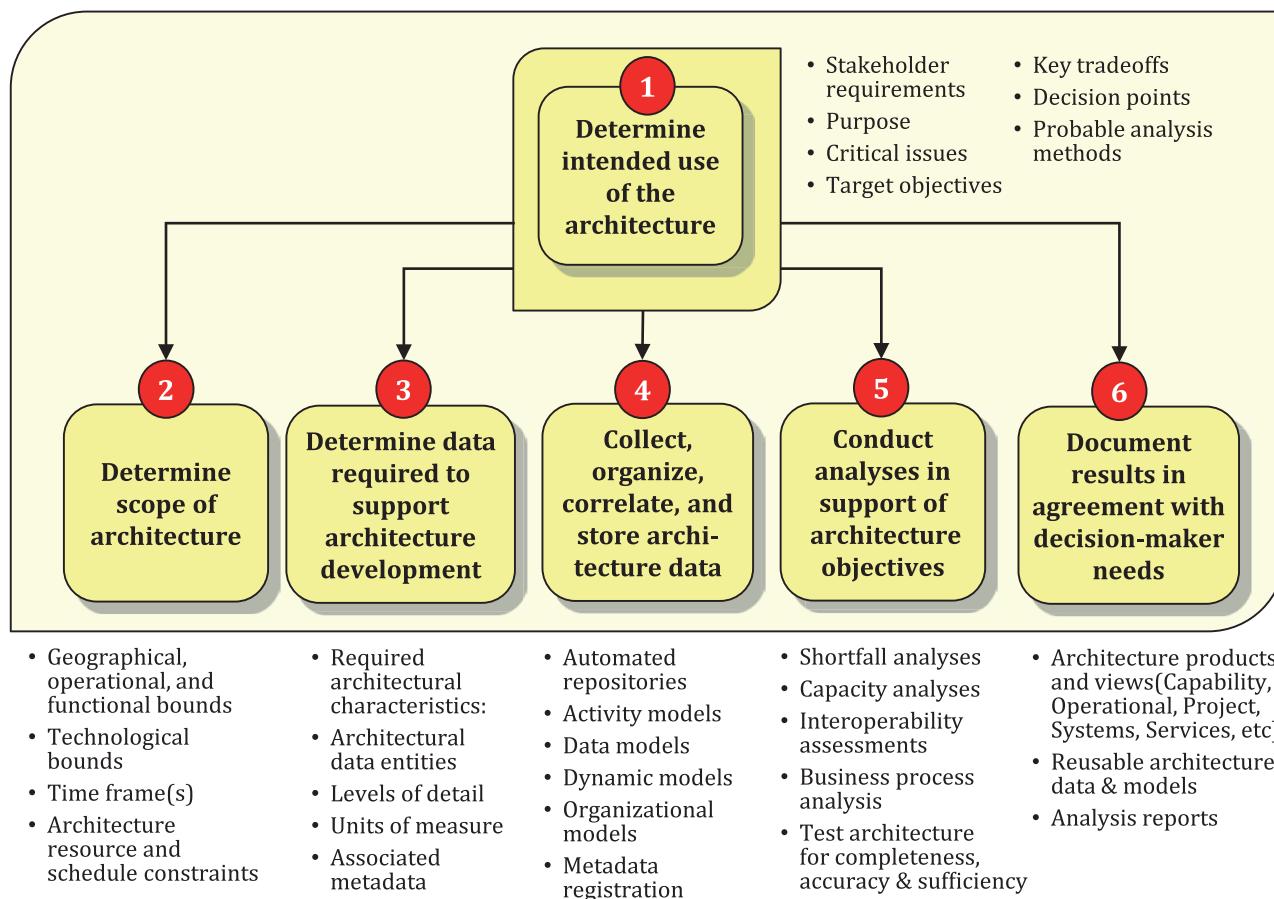
#### **H.5.1 Framework overview**

The US Department of Defense Architecture Framework (DoDAF)<sup>[40]</sup> provides guidance and rules for developing, representing and understanding architectures. Architecture descriptions based-on DoDAF can be compared and related across programs, mission areas and, ultimately, the enterprise, thus, establishing the foundation for analyses that supports decision-making processes.

The major DoDAF elements are:

- Architecture Development describes a method for building DoDAF-compliant architectures. The method is data-centric rather than view-centric. The data-centric approach ensures concordance between the views (formerly called “products”) and also ensures that all essential entity relationships are captured to support a wide variety of analysis tasks.
- DoDAF Meta Model (DM2) establishes and defines the constrained vocabulary to be used for architecture development, facilitating the understandability of the views and exchanging of data between collaborative environments.
- DoDAF formalism defines a way of representing an architecture by dividing the problem and solution spaces into manageable pieces, according to the stakeholders’ viewpoints, further defined as DoDAF-specified Models. Views are instances of specific models or combinations of models.
- As presented in DoDAF V2.0<sup>[39]</sup>, volume 2 (defining the formalism), Section 1 – Introduction:
  - Version 1.0 and 1.5 of the DoDAF used the terms “Product” and “Products” to describe the visualizations of architectural data. In Version 2, the term “Model” is generally used instead, unless there is a specific reference to term “Products” of earlier versions. DoDAF-described “Models” that have been populated or created with architecture data are called “Views”.

- The term “View” is used when the DoDAF-described “Models” are customized or combined for the decision-maker’s need. When creating a view that is not strictly based on one of the DoDAF-specified models, this kind of view is called a “Fit-for-Purpose” view.
- In addition, to align with ISO 15704, ISO 19439, and ISO/IEC/IEEE 42010 terminology where appropriate, “Views” in DoDAF V1.0 and 1.5 are changed to “Viewpoints” in DoDAF V2.0 (e.g., from Operational View to Operational Viewpoint, from System View to System Viewpoint).
- The DoDAF 6-step architecture development process provides guidance to the architect and Architectural Description development team and emphasizes the guiding principles. [Figure H.6](#) depicts this six-step process as described in [\[40\]](#).



**NOTE** The process illustrated here is a synthesis of information from multiple versions of DoDAF. In particular, Step-6 was titled “Document results in accordance with the Architecture Framework” in the first versions of DoDAF, meaning that the results are elaborated according to established templates; but are not always fulfilling all the decision maker needs; i.e. the results are formalizing the best compromise found during the trade-off analysis done during the Step-5.

**Figure H.6 — DoDAF six-step architecture process**

### H.5.2 Mapping to framework elements

The DoDAF 6-step process cannot be mapped directly to one or more processes provided by this document and vice versa. The reason for the lack of a direct mapping is that this document promotes a functional decomposition of the architecture process, while DoDAF uses a decomposition into stages in time, whereupon each stage (called 'step') some or all architecture processes may be iterated. In the following table, an X indicates that full or partial coverage of the ISO/IEC/IEEE 42020 process is achieved via activities as described in the corresponding step in the DoDAF architecture process.

**Table H.3 — Mapping of ISO/IEC/IEEE 42020 processes to the DoDAF framework<sup>[39]</sup>**

<b>DoDAF 6-steps</b>	<b>Architecture Governance</b>	<b>Architecture Management</b>	<b>Architecture Conceptualization</b>	<b>Architecture Evaluation</b>	<b>Architecture Elaboration</b>	<b>Architecture Enablement</b>
Step 1: Determine intended use of the architecture	X		X		X <sup>a</sup>	
Step 2: Determine scope of architecture	X	X	X		X <sup>b</sup>	
Step 3: Determine data required to support architecture development		X	X		X	
Step 4: Collect, organize, correlate and store architecture data			X	X	X	X
Step 5: Conduct analyses in support of architecture objectives			X	X <sup>c</sup>		
Step 6: Document results in agreement with decision-maker needs		X	X	X <sup>d</sup>	X	X

<sup>a</sup> The Elaboration process is usually provided with the intended use of the architecture when the elaboration task is initiated. However, the Elaboration process needs to identify and understand the intended uses of the architecture views, models and descriptions, which could be different than intended uses of the architecture itself.

<sup>b</sup> The scope of the architecture views and models developed in the Conceptualization process may be different than the scope of the architecture views and models produced by the Elaboration process.

<sup>c</sup> The Evaluation process also includes value assessment which is separate from architectural analysis. However, DoDAF uses the term “analysis” in a broader sense to include both architectural analysis and value assessment.

<sup>d</sup> During the Evaluation process, more details about the architecture can be determined or discovered, so these extra details could be captured during the Evaluation process itself rather than necessarily cycling through one of the other processes to add this detail. Furthermore, details might be added to enable the assessment and analysis activities to be performed adequately.

### H.5.3 Items in the DoDAF framework not addressed in this document

The DoDAF framework provides a very precise formalism for architecture description with a predefined set of viewpoints and models. This document does not specify particular viewpoints and models.

### H.5.4 Items in this document not addressed in the DoDAF framework

This document is concerned with the architecting effort, including architecture management and dealing with stakeholders for whom the architecture is being developed, evaluation of the architecture, etc. The DoDAF 6-step Architecture Development process is more oriented towards the architecture data to be produced with regard to satisfaction of the stakeholders.

## H.6 RM-ODP framework

### H.6.1 Framework overview

Reference Model – Open Distributed Processing (RM-ODP)<sup>[2]</sup> is a reference model based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification of the architecture. Many RM-ODP concepts, possibly under different names, have been around for a long time and have been rigorously described and explained in exact philosophy and in system-thinking. Some of these concepts—such as abstraction composition and emergence—have recently been provided with a solid mathematical foundation in category theory.

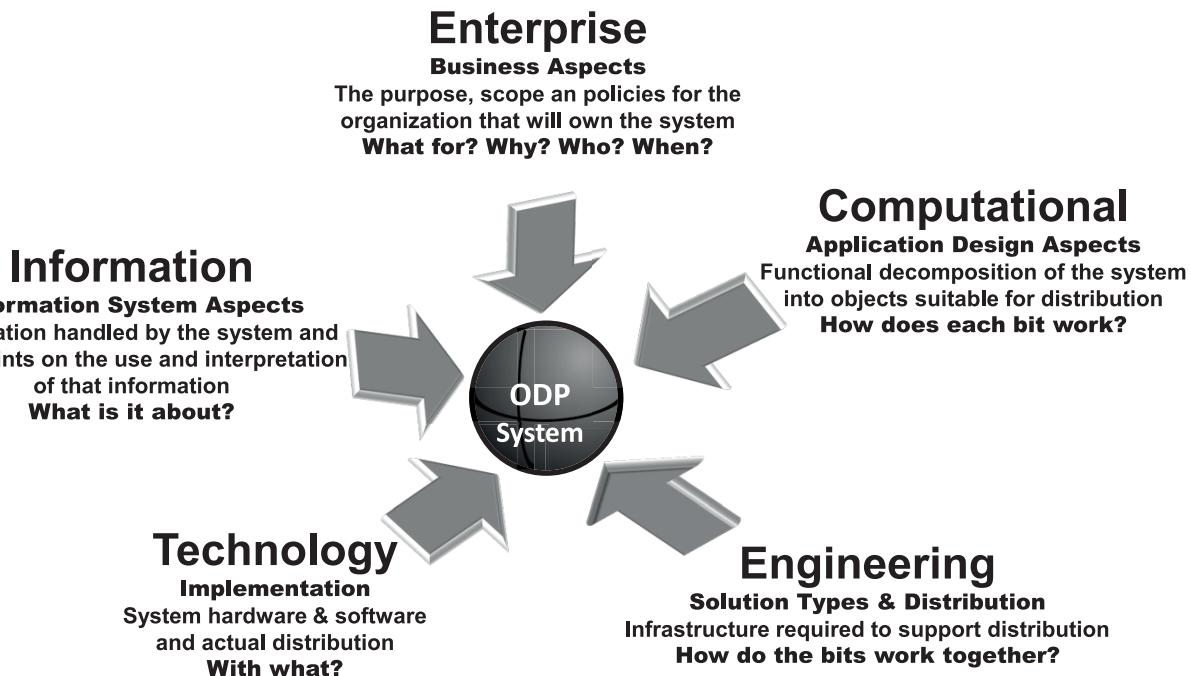
RM-ODP has four fundamental elements:

- an object-modeling approach to system specification;

- the specification of a system in terms of separate but interrelated viewpoint specifications;
- the definition of a system infrastructure providing distribution transparencies for system applications; and
- a framework for assessing system conformance.

[Figure H.7](#) shows the RM-ODP framework. As shown, the RM-ODP consists of 5 viewpoints: Enterprise viewpoint, Information viewpoint, Computational viewpoint, Engineering viewpoint, Technology viewpoint. Each viewpoint prescribes its own architecture constituents.

The RM-ODP family of recommendations and international standards defines a system of interrelated essential concepts necessary to specify open distributed processing systems and provides a well-developed enterprise architecture framework for structuring the specifications for any large-scale systems including software systems.



**Figure H.7 — RM-ODP framework**

## H.6.2 Mapping to framework elements

The RM-ODP is a framework for specification of system's architecture. This does not correspond to architecture process perspective directly. Therefore, it is impossible to indicate mapping.

## H.6.3 Items in the RM-ODP Framework not addressed in this document

The RM-ODP prescribes specification notion/description for systems. Especially, in ISO/IEC 19793, concrete elements of all viewpoints are defined as UML profile. Therefore, System specifications are constructed using these elements in diagrams. Furthermore, the RM-ODP represents from the high abstraction specification (Enterprise specification) to detail one (Technology/Engineering specification).

## H.6.4 Items in this document not addressed in the RM-ODP Framework

This document prescribes the Architecture process for system development. However, the RM-ODP doesn't include development process. Furthermore, the RM-ODP mainly focuses on Enterprise Architecture, on the contrary, this document covers Enterprise/System Architecture.

## Bibliography

- [1] ISO/IEC 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*
- [2] ISO 9000:2015, *Quality management systems — Fundamentals and vocabulary*. International Organization for Standardization
- [3] ISO/IEC 10746 (all parts), *Information technology — Open distributed processing — Reference model*
- [4] ISO/IEC/IEEE 12207, *Systems and software engineering — Software life cycle processes*
- [5] ISO/IEC/IEEE 15288:2015, *Systems and software engineering — System life cycle processes*
- [6] ISO/IEC/IEEE 15289, *Systems and software engineering — Content of life-cycle information items (documentation)*
- [7] ISO/IEC/IEEE 15939, *Systems and software engineering — Measurement process*
- [8] ISO 15704:2000, *Industrial automation systems — Requirements for enterprise-reference architectures and methodologies*
- [9] ISO 19439, *Enterprise integration — Framework for enterprise modelling*
- [10] ISO/IEC 19793, *Information technology — Open Distributed Processing — Use of UML for ODP system specifications*
- [11] ISO/IEC 20000-1, *Information technology — Service management — Part 1: Service management system requirements*
- [12] ISO 21500, *Guidance on project management*
- [13] ISO 21505, *Project, programme and portfolio management — Guidance on governance*
- [14] ISO/IEC/IEEE 24748 (all parts), *Systems and software engineering — Life cycle management*
- [15] ISO/IEC/IEEE 24765, *Systems and software engineering — Vocabulary*
- [16] ISO/IEC/TR 24774, *Systems and software engineering — Life cycle management — Guidelines for process description*
- [17] ISO/IEC 25000, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*
- [18] ISO/IEC 25010, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*
- [19] ISO/IEC 25012, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model*
- [20] ISO/IEC 25020, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality measurement framework*
- [21] ISO/IEC 27000, *Information technology — Security techniques — Information security management systems — Overview and vocabulary*
- [22] ISO 31000, *Risk management — Guidelines*
- [23] ISO/IEC 33001, *Information technology — Process assessment — Concepts and terminology*

- [24] ISO/IEC 33002, *Information technology — Process assessment — Requirements for performing process assessment*
- [25] ISO/IEC 33004, *Information technology — Process assessment — Requirements for process reference, process assessment and maturity models*
- [26] ISO/IEC 33020, *Information technology — Process assessment — Process measurement framework for assessment of process capability*
- [27] ISO/IEC 38500, *Information technology — Governance of IT for the organization*
- [28] ISO/IEC/TR 38502, *Information technology — Governance of IT — Framework and model*
- [29] ISO/IEC/IEEE 42010, *Systems and software engineering — Architecture description*
- [30] ISO/IEC/IEEE 42030, *Enterprise, systems and software — Architecture evaluation framework*
- [31] IEEE Std 1471:2000, *IEEE Recommended Practice for Architectural Description for Software-Intensive Systems*
- [32] IEEE Std 1517-1999 (R2004) *IEEE Standard for Information Technology — Software Life Cycle Processes — Reuse Processes, and modified by generalizing in terms of systems rather than software systems*
- [33] Pragmatic Enterprise Architecture Framework, V3.3 June 2017, Pragmatic EA Ltd, ISBN: 978-1-908424-10-5
- [34] BUSINESS MOTIVATION MODEL, Version 1.0, OMG Document Number: formal/2008-08-02, standard document. URL: <http://www.omg.org/spec/BMM/1.0/PDF>
- [35] MDA Guide version 1.0.1, omg/03-06-01, June 2003
- [36] THE OPEN GROUP ARCHITECTURE FRAMEWORK, (TOGAF®) Version 9.1, van Haren Publishing, ISBN-10: 9087536798
- [37] The DoD Architecture Framework Version 2.02, DoD Deputy Chief Information Officer. URL: <http://dodcio.defense.gov/Library/DoD-Architecture-Framework/>
- [38] HANDBOOK Systems Engineering, a guide for system life cycle processes and activities, International Council on Systems Engineering, INCOSE-TP-2003-002-04 2015
- [39] The DoDAF Architecture Framework Version 2.0, 28 May 2009
- [40] The DoDAF Architecture Framework Version 2.02. URL: <http://dodcio.defense.gov/Library/DoD-Architecture-Framework/>
- [41] BOEHM B. W., BROWN J. R., KASPAR H., LIPOW M., MCLEOD G., MERITT M., Characteristics of Software Quality, Edition 2, North-Holland Pub. Co., 1978 [Broy, 2009] Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description, An overview on description concepts, models and methods
- [42] PARRELL Gregory S., (ed.) (2017). Trade-off Analytics: Creating and Exploring the System Tradespace (Wiley Series in Systems Engineering and Management, January 2017)
- [43] PARRELL Gregory S., BRESNICK Terry, TANI Steven N., JOHNSON Eric R., (2013). Handbook of Decision Analysis (Wiley Handbooks in Operations Research and Management Science, April 2013) by
- [44] WEILL P., (2007), Innovating in Information Systems: What do the most agile firms in the world do? Sixth e-Business Conference, Barcelona Spain, URL: [http://www.scirp.org/\(S\(i43dyn45teexjx455qlt3d2q\)\)/reference/ReferencesPapers.aspx?ReferenceID=382108](http://www.scirp.org/(S(i43dyn45teexjx455qlt3d2q))/reference/ReferencesPapers.aspx?ReferenceID=382108)

- [45] MARK W., MAIER (1996), Architecting Principles for Systems-of-Systems, DOI: 10.1002/j.2334-5837.1996.tb02054.x
- [46] EVANS D., (2014), Styles of Architecting - A smarter approach to architecting the Defence Enterprise, Niteworks White Paper, URL: [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/692432/NWP\\_-\\_Styles\\_of\\_Architecting.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/692432/NWP_-_Styles_of_Architecting.pdf)
- [47] American Society for Quality, Glossary – Entry: Quality, retrieved 2017-08-01
- [48] BOEHM B., KUKREJA N., (2015), An initial Ontology for System Qualities, Proceedings of 25th INCOSE Symposium, Seattle, USA
- [49] CROSBY Philip, (1979). Quality is Free. New York: McGraw-Hill. ISBN 0-07-014512-1
- [50] DRUCKER P. F., (1986). Innovation and entrepreneurship: Practice and principles. New York: Harper Row
- [51] JURAN, JOSEPH M and JOSEPH A DEFEO (2010), Quality Control Handbook, New York, McGraw-Hill, 6th Edition
- [52] KIRAN D R, (2017), Total Quality Management: Key Concepts and Case Studies, Edition 1, Elsevier, Butterworth Heinemann publications
- [53] TAGUCHI G., (1992). Taguchi on Robust Technology Development. ASME Press. ISBN 978-99929-1-026-9
- [54] KUMAR A., DOJI S. L, NIKHIL R Z, and SWAMINATHAN N (2016), Value based System Architecting: Illustrated by Designing a Task Automation System, Proceedings of 26th INCOSE Symposium, Edinburgh, UK
- [55] KUMAR A., DOJI S. L, NIKHIL R Z, and JOSE K R (2017), Value based Architecture of Digital Product-Service Systems, Proceedings of 27th INCOSE Symposium, Adelaide, Australia
- [56] LEN Bass, PAUL Clements, RICK Kazman, 2003, Software Architecture in Practice, 2nd Edition, Addison Wesley Publications, ISBN: 0-321-15495-9
- [57] PAUL Clements, RICK Kazman, MARK Klein, 2002, Evaluating Software Architectures: Methods and Case Studies, 1st Edition, Addison Wesley Publications, ISBN-10: 0-201-70482-X
- [58] The qualities of architecture, John Critchley, March 10th, 2008
- [59] VITRUVIUS BC, De architectura, Marcus Vitruvius Pollio (1st century BC) (Transl. Morris 2106 Hicky Morgan, 1960), The Ten Books on Architecture. Courier Dover Publications. ISBN 0-2107 486-20645-9
- [60] Characteristics of Good Architecture, Enterprise Architect User Guide v13.0, Sparx Systems, 2018, URL: [http://sparxsystems.com/enterprise\\_architect\\_user\\_guide/13.0/guidebooks/ea\\_characteristics\\_of\\_good\\_architecture.html](http://sparxsystems.com/enterprise_architect_user_guide/13.0/guidebooks/ea_characteristics_of_good_architecture.html)
- [61] SIMON A, HERBERT, 1962, The Architecture of Complexity, *Proceedings of the American Philosophical Society*, Vol. 106, No. 6. (Dec. 12, 1962), pp 467-482

## **IEEE notices and abstract**

### **Important Notices and Disclaimers Concerning IEEE Standards Documents**

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

### **Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents**

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

**IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.**

### **Translations**

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

## Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

## Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board  
445 Hoes Lane  
Piscataway, NJ 08854 USA

## Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

## Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

## Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

## Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at <http://ieeexplore.ieee.org/xpl/standards.jsp> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

## **Errata**

[Errata](#), if any, for all IEEE standards can be accessed on the IEEE-SA Website: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

## **Patents**

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

## **Abstract and keywords**

This document complements the architecture-related processes identified in ISO/IEC/IEEE 15288, ISO/IEC/IEEE 12207 and ISO 15704 with activities and tasks that enable architects and others to more effectively and efficiently implement architecture practices. Implementing these practices can help ensure that the architecture has greater influence on business and mission success. It specifies a coherent set of processes for governance, management, conceptualization, evaluation and elaboration of architectures, and activities that enable these processes. Users of this document can apply these processes in the context of: (1) understanding, development and evolution of entities through their life cycle stages such as conception, development, implementation, operation, sustainment, decommissioning, and disposal; (2) organization(s) acting as users, customers and providers of the solution specified by the architecture description; and (3) architecting of entities.

**Keywords:** architecture, architecture collection, architecture entity, architecture conceptualization, architecture elaboration, architecture enablement, architecture evaluation, architecture governance, architecture management, concern, design, enterprise, life cycle, life cycle model, model, outcomes, phase, process reference model, process tailoring, stage, stakeholder, system, software, tradeoff, view, viewpoint.



---

---

**ICS 35.080**

**ISBN 978-1-5044-5902-0 STD23727 (PDF); 978-1-5044-5903-7 STDPD23727 (Print)**

Price based on 111 pages

© ISO/IEC 2019 – All rights reserved

© IEEE 2019. All rights reserved. This document is licensed under a Creative Commons License. The document is provided "as is". The document is not guaranteed to be error-free. Tampere University is not responsible for any damages resulting from the use of the document. The document is not intended to be used for commercial purposes.