

# 中华人民共和国国家标准

GB/T 8566—2022/ISO/IEC/IEEE 12207:2017

代替GB/T 8566—2007

## 系统与软件工程 软件生存周期过程

Systems and software engineering—Software life cycle processes

(ISO/IEC/IEEE 12207:2017,IDT)

2022-10-12发布

2023-05-01实施



国家市场监督管理总局  
国家标准化管理委员会

发布

目 次

前言 .....III

引言 .....IV

1 范围 ..... 1

    1.1 概述 ..... 1

    1.2 目的 ..... 1

    1.3 应用领域 .....1

    1.4 限制 ..... 2

2 规范性引用文件 ..... 2

3 术语和定义、缩略语 .....2

    3.1 术语和定义 .....2

    3.2 缩略语 ..... 9

4 符合性..... 10

    4.1 预期用法 ..... 10

    4.2 完全符合 ..... 11

    4.3 剪裁符合..... 11

5 关键概念和应用..... 11

    5.1 导引..... 11

    5.2 软件系统概念 ..... 11

    5.3 组织和项目概念 ..... 15

    5.4 生存周期概念 ..... 15

    5.5 过程概念 ..... 17

    5.6 过程组 ..... 17

    5.7 过程应用.....19

    5.8 过程参考模型 ..... 20

6 软件生存周期过程 ..... 20

    6.1 协定过程组 ..... 20

    6.2 组织的项目使能过程组 ..... 24

    6.3 技术管理过程组 ..... 30

    6.4 技术过程组 ..... 43

附录 A(规范性)剪裁过程 ..... 79

附录B(资料性)过程信息项示例 .....81

附录C(资料性)用于评估目的的过程参考模型 .....85

附录D(资料性)过程集成和过程构建 ..... 87

附录 E(资料性)过程视图 ..... 89

附录F(资料性) 软件系统架构建模..... 96



附录G（资料性）将软件生存周期过程应用于系统之系统..... 98

附录H（资料性） 敏捷的应用 ..... 101

附录 NA（资料性） 本文件与GB / T 8566—2007的差异 ..... 103

参考文献 ..... 107

## 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件与GB/T 22032—2021《系统与软件工程 系统生存周期过程》共同构成了软件与系统工程领域的基础性标准。

本文件代替GB/T 8566—2007《信息技术软件生存周期过程》。与GB/T 8566—2007相比，除结构调整和编辑性改动外，主要技术变化如下：

- 术语和定义做了调整和补充(见第3章，2007年版的第3章)；
- 软件生存周期过程模型作了重大调整和变化(详见附录NA)。

本文件等同采用ISO/IEC/IEEE 12207:2017《系统与软件工程 软件生存周期过程》。

本文件做了下列最小限度的编辑性改动：

- 删除附录I(资料性)与ISO/IEC/IEEE 12207:2008的过程映射，本文件代替的2007版本采用ISO/IEC 12207:1995,本文件采用ISO/IEC/IEEE 12207:2017,ISO/IEC/IEEE 12207:2008为中间版本，对国家标准没有意义，并不涉及技术内容的变更；
- 增加了附录NA(资料性)。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本文件起草单位：浙江省电子信息产品检验研究院、中国电子技术标准化研究院、江苏赛西科技发展有限公司、深圳赛西信息技术有限公司、中国电子科技集团公司第五十四研究所、北京航天自动控制研究所、浙江中控技术股份有限公司、中国航发商用航空发动机有限责任公司、山东山科数字经济研究院有限公司、中国航天系统科学与工程研究院、北京软件和信息服务交易所有限公司、上海宝信软件股份有限公司、大连理工大学、中国电子科技集团公司第十研究所、山东省计算中心(国家超级计算济南中心)、广东益安人防工程科技有限公司、上海计算机软件技术开发中心、北京赛迪认证中心有限公司、成都四方伟业软件股份有限公司、山东正中信息技术股份有限公司、北京华宇软件股份有限公司、上海市软件评测中心有限公司。

本文件主要起草人：张君、张旻旻、季永炜、李文鹏、孙纪敏、赵浩强、郭晓慧、刘永召、祝钦、李刚、杨桂枝、于铁强、张星星、黄钰梅、宋明秋、董李梅、赫畅、孙金洋、王公韬、马烈、胡芸、颜怀柏、宋丽华、祁雨奇、庄园、王春晓、董冠涛、米坤、孟艳、李艳、韩德隆、李敏。

本文件及其所代替文件的历次版本发布情况为：

- 1988年首次发布为GB/T 8566—1988《计算机软件开发规范》；
- 1992年第一次修订为GB/T 856—1992《信息技术软件生存期过程》；
- 2001年第二次修订为GB/T 8566—2001《信息技术 软件生存周期过程》，2007年第三次修订；
- 本次为第四次修订。

# 引 言

软件系统的复杂性已经增加到前所未有的程度。这为创建和使用系统的组织带来了新的机遇，但也带来了更多的挑战，这些挑战存在于软件系统的整个生存周期以及架构层次上的所有细节。本文件提供了一个公共过程框架，以便采用软件工程方法描述人工创建的系统生存周期。软件工程是成功实现软件系统的一种跨学科方法和手段，它关注定义利益相关方的需要以及开发周期之前所要求的功能，它关注建立需求文档，它关注执行设计集成和系统验证，同时也考虑全局性的问题。软件工程将所有的规程和专业组集成到一个团队工作中，形成一个从概念到生产、到操作、到维护的结构化开发过程，同时也考虑全部利益相关方的业务和技术需要，以提供满足用户和其他利益相关方要求的高质量产品为目标。该生存周期跨越了从概念到系统退役的整个过程，为系统的获取和供应，提供了相应的过程，同时也有助于改进创建、使用和管理现代软件系统各方之间的沟通和合作，使它们可以按一种集成化的、高内聚的模式工作。该框架还提供了对生存周期过程的评估和改进。

本文件中的过程形成了一个全面的集合，组织可从中构建适合其产品和服务的软件生存周期模型。根据组织目的，可选择并应用一个适当的子集来实现该目的。

本文件可在下列一种或多种模式下使用。

- a) 组织使用——帮助建立所需过程的环境。这些过程可由方法、过程、技术、工具和经过培训的人员组成基础结构来支持。然后组织可使用上述环境来执行和管理自身的项目，并通过它们的生存周期阶段来开发软件系统。在这种模式下，本文件用于评估已声明的、已建立的环境是否符合其规定。
- b) 项目使用——帮助选择、构建和使用一个已建立的环境元素来提供产品和服务。在这种模式下，本文件用于评估项目是否符合已声明和已建立的环境。
- c) 需方和供方使用——帮助制定关于过程和活动的协议。通过该协议，本文件中的过程和活动被选定、协商、同意和执行。在这种模式下，本文件用于指导协议的制定。
- d) 过程评估者使用——作为过程参考模型，用于过程评估的绩效，以支持组织的过程改进。

考虑到软件和系统范围的区别，本文件中的软件生存周期过程模型中的“特定项目包管理过程”与GB/T 22302—2021系统生存周期过程模型中的“项目群管理过程”指的是同一过程，但在表述上存在差异。

# 系统与软件工程 软件生存周期过程

## 1 范围

### 1.1 概述

本文件使用良好定义的术语，为软件生存周期过程建立了一个公共框架，以供软件产业界引用。该框架包含过程、活动和任务，可用于软件系统、产品和服务的获取、供应、开发、运行、维护或处置期间。这些生存周期过程是通过所有与系统有关的各方参与，以实现顾客满意为最终目标来完成的。

本文件适用于软件系统、产品和服务，以及任何系统中的软件部分的获取、供应、开发、运行、维护和处置(无论在组织内部还是外部执行)。软件包括固件的软件部分，还包括为软件产品和服务提供环境所需的系统定义那些部分。

本文件还提供了可用于一个组织或一个项目内来定义、控制和改进软件生存周期过程的过程。

本文件中的过程、活动和任务还可应用于一个包含软件的系统的获取期间，其中，既可以单独使用，也可以和GB/T 22032—2021结合使用。

GB/T 22032—2021主要关注那些很少使用或不使用软件的人造系统，与GB/T 22032-2021的使用环境相比，本文件主要关注的是一个连续统一体的软件人造系统。现实中，很少遇见一个没有软件的复杂系统，且所有软件系统均需通过物理系统的部件(硬件)来运行，或只作为关注焦点的软件系统的一部分，或只作为一个使能系统或基础设施。因此，是否把本文件应用于软件生存周期，还是把GB/T 22032-2021应用于软件生存周期，这一选择依赖于所关注的系统。两个标准中的过程具有相同的过程目的和过程输出，但是分别在执行软件工程或系统工程的活动和任务中有所不同。

### 1.2 目的

本文件的目标是在系统生存周期中提供一个已定义过程集合，来促进需方、供方和其他利益相关方之间的沟通。

本文件适用于软件系统、产品和服务的需方、供方、开发方、集成方、操作方、维护方、管理者、质量保证管理者和用户。它既可由单方作为自我改进工作采用，也可用于多方的情况。各方可来自于同一个组织，也可来自不同的组织，各方之间的关系可以是非正式合同或正式合同。

本文件的过程可用于作为创建业务环境(例如，方法、规程、技术、工具和专业人员)的基础。附录A规定了对这些软件生存周期过程进行剪裁的规范性要求。

### 1.3 应用领域

本文件应用于完整的软件系统、产品和服务的生存周期，包括概念、开发、生产、使用、支持和退役，同时也应用于它们的获取和供应，无论是在组织内部还是外部运行。本文件定义的生存周期过程可同时地、迭代地、递归地应用于软件系统，也可递增地应用于软件系统元素。

在软件系统的目的、应用领域、复杂性、规模、新颖性、适应性、数量、位置、生存时间与演进等方面，软件系统是千差万别的。本文件描述了包含人工软件系统的生存周期过程。因此，它既可应用于单件生产、面向广泛的商业或公共发行，以及可定制可适应的软件系统，也可应用于完整的单机软件系统和可嵌入/集成为更大更复杂的完整系统中的软件系统。

本文件提供了根据过程目的和过程输出特征而展现的过程参考模型，而过程目的和过程输出来源于活动和任务的成功执行。附录B列出了与不同过程相关工作产品和信息项的例子。因此本文件作

为参考模型，用于支持过程评估(参见 ISO/IEC 33002:2015)。附录C 提供了作为过程参考模型和关于软件生存周期使用的信息。附录D 描述了使用过程参考模型的过程结构。

## 1.4 限制

本文件并不规定具体的软件生存周期模型、开发方法学、方法、建模方法或者技术。本文件的用户负责选择项目的生存周期模型，并把本文件的过程、活动和任务映射到模型。各方也可选择和应用适合该项目的合适的方法学、方法、模型和技术。

本文件没有建立管理体系，也未要求使用任何管理体系标准。然而还是期望与GB/T 19001规定的质量管理体系、ISO/IEC 20000-1(IEEE std 20000-1)规定的服务管理体系以及GB/T 29246规定的信息安全管理体系兼容。

本文件没有详述关于命名、格式、明确内容、记录媒介的信息项。生存周期过程信息项(文档)内容参见ISO/IEC/IEEE 15289。

## 2 规范性引用文件

本文件没有规范性引用文件。

## 3 术语和定义、缩略语

### 3.1 术语和定义

下列术语和定义适用于本文件。

#### 3.1.1

**需方 acquirer**

从供方获得或采购某一产品或服务的利益相关方。

注：需方的同义术语，常用的有买方、顾客、所有者、购买者或内部/组织赞助方。

#### 3.1.2

**获取 acquisition**

获得某一系统、产品或服务的过程。

#### 3.1.3

**活动 activity**

某一过程中高内聚的任务集合。

#### 3.1.4

**敏捷开发 agile development**

以迭代开发、频繁检查和调整、增量交付为手段，依靠跨功能团队协同和持续与利益相关方沟通反馈促进需求和解决方案不断演进的软件开发方法。

[来源：ISO/IEC/IEEE 26515:2011]

#### 3.1.5

**协定 agreement**

据以维持工作关系并得到相互确认的条款与条件。

示例：合同，协定备忘录。

#### 3.1.6

**架构 architecture**

**体系结构**

(系统)在其环境中的一些基本概念或性质，体现在其元素、关系，以及设计与演进原则中。

[来源: ISO/IEC/IEEE 42010:2011,3.2]

### 3.1.7

**架构框架 architecture framework**

**体系结构框架**

在特定应用领域和/或利益相关方的团体中,为描述架构所建立的约定、原理和实践。

**示例1:**GB/T 18757中的通用企业参考架构、方法论(GERAM)是一种架构框架。

**示例2:**ISO/IEC 10746开放分布式处理参考模型(RM-ODP)是一种架构框架。

[来源: ISO/IEC/IEEE 42010:2011,3.4]

### 3.1.8

**架构视图 architecture view**

从特定系统关注的视角来表达某一系统架构的工作产品。

[来源: ISO/IEC/IEEE 42010:2011,3.5]

### 3.1.9

**架构视角 architecture viewpoint**

为架构视图的构造、解释和使用,建立约定的工作产品,以便构建特定系统的关注焦点。

[来源: ISO/IEC/IEEE 42010:2011,3.6]

### 3.1.10

**审核 audit**

**审计**

为评估工作产品或工作产品集是否符合规范、标准、合同协定或其他准则而进行的独立检查。

[来源: GB/T 11457—2006,3.213]

### 3.1.11

**基线 baseline**

经正式批准的配置项。它与媒介无关,是在该配置项的生存周期中的特定时间节点确定并固化的。

[来源: IEEE Std 828—2012,2.1]

### 3.1.12

**业务过程 business process**

为了达到某种期望的最终结果,从而实现组织的既定目标,可执行的部分有序的企业活动的集合。

### 3.1.13

**运营观念 concept of operations**

对于某一组织的一项或一系列行动的设想或意图,以文字和/或图形做出的概略表述。

注1:运营观念通常在长远战略规划和年度运营计划中得到体现。在年度运营计划中,运营观念覆盖同时或相继进行的一系列相关行动。运营观念旨在描述组织运行的整体图景。参见3.1.28运行概念。

注2:运营观念为提供确定运行空间、系统能力、界面和运行环境边界的基础。

[来源: ANSI/AIAA G-043A—2012e,5.2]

### 3.1.14

**关注焦点 concern**

(系统)一个或多个利益相关方对某一系统的利益所在。

注:关注焦点涉及对某一系统在其环境方面的各种影响,包括开发的、技术的、业务的、运行的、组织的、政策的、经济的、法律的、监管的、生态的以及社会的影响。

[来源: ISO/IEC/IEEE 42010:2011,3.7]

### 3.1.15

**配置项 configuration item**

**技术状态项**

为了进行配置管理而指定的,在配置管理的过程中作为单个实体对待的硬件、软件或软硬件综合项

或聚合体。

**示例：**软件、固件、数据、硬件、人员、过程(如，为用户提供服务的过程)、程序(如，操作说明和用户手册)、设施、服务、材料和自然存在的实体。

[来源：ISO/IEC/IEEE 24765:2010,3.563]

### 3.1.16

#### **顾客 customer**

接受某项产品或服务的组织或个人。

**示例：**消费者、客户、用户、需方、买方或购买者。

**注：**顾客可以是组织内部的或外部的。

### 3.1.17

#### **设计(动词) design**

(过程)界定架构、系统元素、接口，以及某一系统或系统元素的其他特性。

[来源：ISO/IEC/IEEE 24765:2010,3.800,有修改]

### 3.1.18

#### **设计(名词) design**

设计(3.1.17)过程的结果。

**注1：**信息，包括系统元素及其相互关系的规范，充分完备足以支持架构兼容实现的信息。

**注2：**设计提供了系统元素的详细的实现级的物理结构、行为、时间关系和的其他属性。

### 3.1.19

#### **设计特性 design characteristic**

属于一个产品或服务的可测量描述的设计属性或特有性质。

### 3.1.20

#### **使能系统 enabling system**

对所关注的系统在其生存周期阶段提供支持，但在其运行期间不必直接发挥功用的系统。

**示例：**在软件开发期间用于控制软件元素的配置管理系统。

**注：**每一个使能系统都有自己的生存周期。当使能系统为了其自身的需要也被作为所关注的系统对待时，本文件同样适用于它们。

### 3.1.21

#### **环境 environment**

(系统)决定对一个系统的所有影响的设置和情势的周境。

[来源：ISO/IEC/IEEE 42010:2011,3.8]

### 3.1.22

#### **设施 facility**

促进行动执行的物理手段或设备，例如厂房、仪器、工具。

### 3.1.23

#### **偶发事件 incident**

项目、产品、服务或系统在其生存周期中任意时刻发生的异常、意外事件或者事件、条件、情况的集合。

### 3.1.24

#### **信息项 information item**

#### **信息产品 information product**

供人们使用而制作、存储和交付的可单独识别的信息体。

[来源：ISO/IEC/IEEE 15289:2015,3.1.12]

### 3.1.25

#### **基础设施 infrastructure**

支持计算机系统与软件的设计、开发和改进的硬件和软件环境。

## 3.1.26

**生存周期 life cycle**

系统、产品、服务、项目或其他人工实体从概念到退役的演变。

## 3.1.27

**生存周期模型 life cycle model**

与生存周期相关的过程和活动的框架，可以组织成多个阶段，也可作为交流和理解的通用参考。

## 3.1.28

**运行概念 operational concept**

对于一个组织的，关于一个系统或一组相关系统的运行或一系列运行的，设想或意图的文字和图形化表述。

注：运行概念使用一个或多个特定系统或与一组相关的系统，从用户和操作人员角度，给出运行的整体描述。参见运营观念(3.1.13)。

[来源：ANSI/AIAA G-043A-2012e,5.2]

## 3.1.29

**操作方 operator****操作员**

执行系统操作的个人或组织。

注1:操作者和用户的角色可被同时或顺序授予同一个人或组织。

注2:与知识、技能和规程为一体的操作者都可以被认为是系统的一个元素。

注3:操作者可根据操作指令是否在系统边界内来对系统进行不同的操作。

## 3.1.30

**组织 organization**

职责、权限和相互关系得到安排的一组人员及设施。

示例：公司、集团、商行、企事业单位、研究机构、慈善机构、代理商、社团或上述组织的部分或组合。

注：指定组织的一部分(小到只有一个人)或者指定的组织团体，只要具有职责、权限和相互关系，都可以被当做组织。为了特定目的而组织起来的一部分人也是组织，如俱乐部、联盟、公司、社团。

## 3.1.31

**当事方 party**

达成协定的组织。

注：在本文件中，协定的当事方分别称为需方和供方。

## 3.1.32

**问题 problem**

需要探究并采取纠正措施的困境、不确定性，或其他已认识的和不期望的事件、事件集、条件或状况。

## 3.1.33

**过程 process**

一组将输入转化为输出的相互关联或相互作用的活动。

## 3.1.34

**过程输出 process outcome**

成功达到过程目的、可观察的结果。

## 3.1.35

**过程目的 process purpose**

执行过程的高层次的目标，过程有效实施的可能输出。

注：执行过程的目的是为利益相关方提供权益。



3.1.36

**产品 product**

过程的结果。

注：有四种被认可的通用产品类别：硬件(如发动机机械零件)、软件(如计算机程序，以及可能有关的文件和资料)、服务(如运输)和流程性材料(如润滑剂)。硬件和流程性材料通常是有形的产品，而软件和服务通常是无形的。

3.1.37

**项目 project**

按照明确的开始和结束准则，依据给定的资源和需求创建产品或服务的努力。

注：项目可看作是包含协作和受控活动的独特过程，可由本文件中定义的来自技术管理过程组和技术过程组的活动组成。

3.1.38

**(项目>特定项目包 portfolio**

专注于组织的战略目标的项目的集合。

注：GB/T 22032—2021中使用“项目群”。

3.1.39

**资质 qualification**

证明实体是否有能力满足规定要求的过程。

3.1.40

**质量保证 quality assurance**

质量保障

质量管理的一部分，致力于提供质量要求会得到满足的置信度(统计上的可信程度)。

[来源：GB/T 19000—2016,3.3.6,有修改]

3.1.41

**质量特性 quality characteristic**

与要求有关的，产品、过程或系统的固有属性。

注：严格意义上的质量特性通常包括与健康、安全性、信息安全保证、可靠性、可用性和支持性相关的内容。

3.1.42

**质量管理 quality management**

在质量方面指挥和控制组织的协调的活动。

3.1.43

**发布 release**

用于特定目的的可用配置项的特定版本。

示例：测试发布。

3.1.44

**需求 requirement**

对需要及其约束和条件的表达。

[来源：ISO/IEC/IEEE 29148:2018,3.1.19,有修改]

3.1.45

**资源 resource**

过程执行期间使用或消耗的资产。

注：资源包括可循环使用、可再生的或消耗品。

示例：不同的实体，如资金、人力、设施、固定设备、工具和公共设施；如电力、水力、燃料和通信基础设施。

## 3.1.46

**退役 retirement**

负责运行和维护的组织撤销了对当前系统的有效支持，当前系统已被新系统或升级改造的系统部分或者完全的替代。

## 3.1.47

**风险 risk**

不确定性对目标的影响。

注1:影响是指偏离预期，可以是正面的和/或负面的。正面的影响也被称为机遇。

注2:目标有不同的方面(例如财务、健康与安全，以及环境的目标)，并可应用在不同层次(例如战略、组织范围、项目、产品和过程)。

注3:通常用潜在事件、结果或者两者的组合来区分风险。

注4:通常用事件后果(包括情形的变化)和事件发生可能性的组合来表述风险。

注5:不确定性指对事件及其后果或可能性的信息缺失或了解片面的状态。

[来源：GB/T 23694—2013,2.1]

## 3.1.48

**安全 safety**

系统在给定条件下不会导致人们生命、健康、财产或环境处于危险状态的期望。

## 3.1.49

**信息安全性 security**

以防故意破坏或强使失效的保护。信息安全一般由保密性、完整性、可用性和可追溯性四个属性组成，有时加上第5个属性——易用性，上述5个方面都有保障这些属性实现的相关问题。

[来源：NATO AEP-67]

## 3.1.50

**服务 service**

活动、工作或职责的履行。

注1:服务是自包含的、固有的、离散的，可包含其他服务。

注2:服务一般是无形的产品。

## 3.1.51

**软件元素 software element**

本身是软件的系统元素。

## 3.1.52

**软件工程 software engineering**

将系统化的、严格约束的、可量化的方法应用于软件的开发、运行和维护。

注：即将工程化应用于软件。

## 3.1.53

**软件项 software item**

源代码、目标代码、控制代码、控制数据或这些项的集合。

注：软件项可被视为本文件和GB/T 22032:2021的一个系统元素。软件项通常是配置项。

## 3.1.54

**软件产品 software product**

一组计算机程序、规程以及可能的相关文档和数据。

注：软件产品被看作是一种由过程产生输出(产品)的软件系统。

## 3.1.55

**软件系统 software system**

软件对于利益相关方来说是最为重要的部分的系统。

注1:最常见的软件系统是由硬件、软件、人员及其操作过程组成。

注2:在软件系统中,软件是满足系统需求的主要驱动。

### 3.1.56

#### 软件系统元素 software system element

构成软件系统的一组元素的成员。

注1:软件系统元素可包括一个或更多的软件单元、软件元素、硬件单元、硬件元素、服务以及其他系统元素和系统。

注2:软件系统元素可以被看作是一个系统元素。

### 3.1.57

#### 软件单元 software unit

软件体系结构中,可接受独立测试的最小组件。

注:一些软件单元是可单独编译的代码段。

[来源:GB/T 34590.1—2017,2.125,有修改]

### 3.1.58

#### 阶段 stage

实体生存周期中的一个区段,与实体的描述或实现的状态有关。

注1:使用本文件,与整个实体生存周期中主要过程和成就转折点有关的阶段。

注2:阶段经常会交迭。

### 3.1.59

#### 利益相关方 stakeholder

在系统或所属其特性中有权利、份额、声明或利益,以满足其需要及期望的个人或组织。

示例:最终用户、最终用户组织、支持方、开发方、培训方、维护方、部署方、需方、供方组织和监管机构。

注:某些利益相关方可具有相互对立或系统对立的利益。

### 3.1.60

#### 供方 supplier

与需方达成关于产品或服务供应协定的组织或个人。

注1:一般用于供方的其他术语有承包人、生产者、卖家或供方。

注2:需方和供方有时是同一个组织的一部分。

### 3.1.61

#### 系统 system

为达到一个或多个明确目的而组织起来的、相互作用的元素的组合体。

注1:系统有时可被认为是一种产品或者一种它所提供的服务。

注2:实际中,对系统含义的解释通常通过使用一个联合名词来阐述,如飞行器系统。有时候“系统”这个词也可简单地由依赖语境的同义词来替代,如飞行器,虽然这可能会使系统的视角不太明显。

注3:完整的系统包括相关装置、设备、原料、软件、固件、技术文档、服务和运行、支持所必需的人力,并能在其预期的环境中使用。

注4:参见比较:使能系统,所关注的系统,系统之系统。

### 3.1.62

#### 系统元素 system element

组成系统的一组元素中的成员。

示例:硬件、软件、数据、人、过程(例如提供给用户服务的过程)、规程(例如操作指南)、设备、原料、自然存在的实体或其任意组合。

注:系统元素是系统中的离散部分,通过实现它可以完成规定的需求。

### 3.1.63

#### 所关注的系统 system-of-interest

正在考虑其生存周期的系统。

**3.1.64****系统之系统 system-of-system**

一组集成的或可互操作的系统，以提供任一组成系统都无法单独完成的独特能力。

注：每个组成系统本身都是一个有用的系统，具有其自身的管理、目标和资源，但在SOS内部协调以提供SOS的独特能力。

**3.1.65****系统工程 systems engineering**

一种跨学科方法，用于控制将一系列利益相关方的要求、期望、约束转化为解决方案，并在整个生存周期中支持此解决方案所需要的全部技术工作和管理工作。

**3.1.66****任务 task**

要求的、推荐的或可允许的活动，目的是有助于一个或多个过程输出的达成。

**3.1.67****技术管理 technical management**

运用技术和行政资源来规划、组织和控制工程职能。

**3.1.68****权衡 trade-off**

基于利益相关方的净利益，从各种需求和备选解决方案中做出选择的决策活动。

**3.1.69****可追溯性 traceability**

在两个或两个以上的逻辑实体之间建立关系的程度，特别是相互之间有前后相继或主从关系的实体，例如需求、系统元素、验证或任务。

示例：软件特性和测试用例通常追溯到软件需求。

**3.1.70****用户 user**

在系统使用过程中，与系统进行交互或从系统中获益的个人或组织。

注：用户和操作者的角色可能同时或依次地归属于同一个人或组织。

[来源：GB/T 25000.10—2016,3.27,有修改]

**3.1.71****确认 validation**

通过提供客观证据，对特定的预期用途或应用的需求已得到满足的认定。

注1：系统能够在预期的运行环境下，完成预期的用途、目标和目的（如满足利益相关方的要求），则建立了正确的系统。

注2：在生存周期周境中，确认涉及一系列以获得系统能够在与操作环境类似的环境中完成其预期用途、目标和目的的信心的活动。

**3.1.72****验证 verification**

通过提供客观证据，对特定需求已得到满足的认定。

注：验证是一系列将系统或系统元素与需求的特性相比较的活动集合，包括但不限于规定的需求、设计描述和系统本身。系统是按照正确的过程建造的。

[来源：GB/T 19000—2016,定义3.8.12,有修改]

**3.2 缩略语**

下列缩略语适用于本文件。

- CCB: 配置控制委员会(Configuration Control Board)
- CM: 配置管理(Configuration Management)
- COTS: 商业现货(Commercial-Off-The-Shelf)
- FCA: 功能配置审核(Functional Configuration Audit)
- FOSS: 免费开源软件(Free and Open Source Software)
- GUI: 图形用户界面(Graphical User Interface)
- NDI: 非开发项(Non-Developmental Items)
- PCA: 物理配置审核(Physical Configuration Audit)
- PESTEL: 大环境分析(Political,Economic,Social,Technological,Environmental,and Legal)
- PMI: 项目管理协会(Project Management Institute)
- PMP: 项目管理计划(Project Management Plan)
- PRM: 过程参考模型(Process Reference Model)
- QA: 质量保证(Quality Assurance)
- SCM: 软件配置管理(Software Configuration Management)
- SDP: 软件开发计划(Software Development Plan)
- SEMP: 系统工程管理规划(Systems Engineering Management Plan)
- SOI: 所关注的系统(System-of-Interest)
- SOS: 系统之系统(System of Systems)
- sWOT: 态势分析法(Strengths,Weaknesses,Opportunities,Threats)
- WBS: 工作分解结构(Work Breakdown Structure)

4 符合性

4.1 预期用法

本文件的要求包含在第6章和附录A中。本文件提供了适合在软件系统或产品的生存周期中使用的一些过程要求。特定的项目或组织或许不需要使用本文件提供的所有过程，因此，本文件的实施通常涉及选择和声明适合组织或项目的过程集合。有两种符合本文件规定的实施可被声明的方法——完全符合和剪裁符合。

声明完全符合存在两个准则。不管达到哪个准则均满足符合性，尽管所选择的准则(或原则)要在声明中给出。声明“任务的完全符合”，即表明达到了所声明的那个过程集的活动和任务的所有要求，与此相对地，声明“输出的完全符合”，即表明实现了所声明的过程集的所有要求的输出。输出的完全符合允许在符合过程的实施中具有更大的自由度，并对实现使用于一个创新生存周期模型环境中的过程是有益处的。

注1:为满足应用本文件所需的灵活性，提供了符合性选项。每个过程都有一组目标(称为“输出”)、活动与任务，它们代表实现这些目标的一种方式。

注2:实现被声明过程集的活动和任务的用户可以确定与所选过程的任务完全符合。然而，有些用户可能具有实现被声明过程集的目标(如输出)而无需执行所有的活动和任务的创新过程变体。这些用户可以确定与被声明过程集的输出完全符合。这两个准则——任务符合与输出符合——并不一定是等同的，因为在某些情况下，活动和任务的具体执行可能比仅仅实现输出需要更高的能力水平。

注3:当本文件用于帮助促进供需双方之间的协定时，无论修改与否，都可以选择将本文件中的条款纳入协定。在这种情况下，对需方和供方来说，要求遵守协定比符合本文件更合适。

注4:将本文件作为交易条件的组织(如，国家、行业协会、公司)可以规定并公开构成供方满足协定条件所需过程、输出、活动和任务的最小集。

注5:本文件通过助动词“应”的使用来标记要求,通过助动词“宜”的使用来标记建议,通过助动词“可”的使用来标记允许。然而,尽管使用了助动词,符合性要求还是如前所述的那样被选择。

## 4.2 完全符合

### 4.2.1 输出的完全符合

完全符合的声明给出了声明符合性的过程集。通过证明已完成所声明过程集的所有输出,来实现输出的完全符合。在这种情况下,有关宣称的过程集所含活动和任务的条款是指导性的,而不是要求,尽管在该条款中使用了动词形式。

使用本文件的目的是促进过程评估和改进。就这一意图而言,每个过程的目的是以“输出”的形式给出的,与ISO/IEC 33002 兼容。ISO/IEC 33002为本文件的过程评估、过程改进提供了基础。期望过程评估和改进的用户,可以使用本文件中给出的过程输出,作为ISO/IEC 33002所需要的过程参考模型。

### 4.2.2 任务的完全符合

完全符合的声明给出了声明符合性的过程集。通过证明所声明过程集的所有活动和任务都已符合要求,来实现任务的完全符合。在这种情况下,有关声明的过程集之输出的条款是指导性的,而不是要求,尽管在该条款中使用了动词形式。

注:在合同情况中,当需方或监管者需要详细了解供方的过程,任务的完全符合的声明会比较合适的。

## 4.3 剪裁符合

当本文件作为建立过程集的基础而使用时,其中该过程集并非限制为是完全符合的,那么就可根据附录A中规定的剪裁过程选择本文件的章条内容,或修改之。剪裁文本是为表明剪裁符合而声明的。通过证明已完成剪裁后的输出、活动和任务,来实现剪裁符合。

## 5 关键概念和应用

### 5.1 导引

本章旨在强调并帮助解释本文件所依据的基本概念。

注:这些概念的进一步阐述可以在关于生存周期管理应用的ISO/IEC/IEEE 24748-1、ISO/IEC/IEEE 24748-2和ISO/IEC/IEEE 24748-3中找到。

### 5.2 软件系统概念

#### 5.2.1 软件系统

本文件中涉及的软件系统是人工制造的,创建并使用这些软件系统是为了用户和其他利益相关方的利益提供特定环境下的产品或服务。这些软件系统可以包括以下系统元素:硬件、软件、数据、人、过程(如提供服务给用户的过程)、规程(如操作说明)、设施、服务、原料和自然存在的实体。对用户而言,它们被认为是产品或服务。

本文件适用于对利益相关方来说软件是其最重要部分的系统。它基于系统工程和软件工程的一般原理。本文件的一个基本前提是软件始终运行于系统的背景下。由于软件的运行离不开硬件,因此执行软件的处理器可以看作是系统的一部分。另外,承载软件系统并处理与其他系统通信的硬件或服务也可以视为运行环境中的使能系统或外部系统。

对于特定软件系统及其结构和元素的理解和定义,取决于利益相关方的利益和职责。一个利益相

关方所关注的系统(SOI) 可以当作另一个利益相关方关注的系统元素，也可以被当作另一利益相关方 SOI 环境的一部分。

以下列举了SOI 的重要特性：

- a) 定义的边界封装了有意义的需要和实际的解决方案；
- b) 系统元素间存在层次关系或其他关系；
- c)SOI 中任何层次中的一个实体都可被当作一个系统；
- d) 系统包含集成的、已定义的从属系统元素的集合；
- e) 人既可以被看作是系统外的用户，也可以认为是系统内的系统元素(如操作者)；
- f) 系统可被孤立地当作一个实体(如产品), 或当作能够与周围环境相互作用的功能集合体(如服务集合)。

本文件的概念是通用的，无论选择什么边界来定义系统，允许使用者将生存周期中的各个实例关联起来或使之适应其系统原理。

5.2.2软件系统结构

本文件中生存周期过程的描述是与软件系统相关的。软件系统是由一组互相作用的系统元素组成的(包括软件元素), 每个系统元素能分别实施以完成各自特定的需求(图1)。因此，任一系统元素的实施职责，可以通过协定委托给另一当事方。

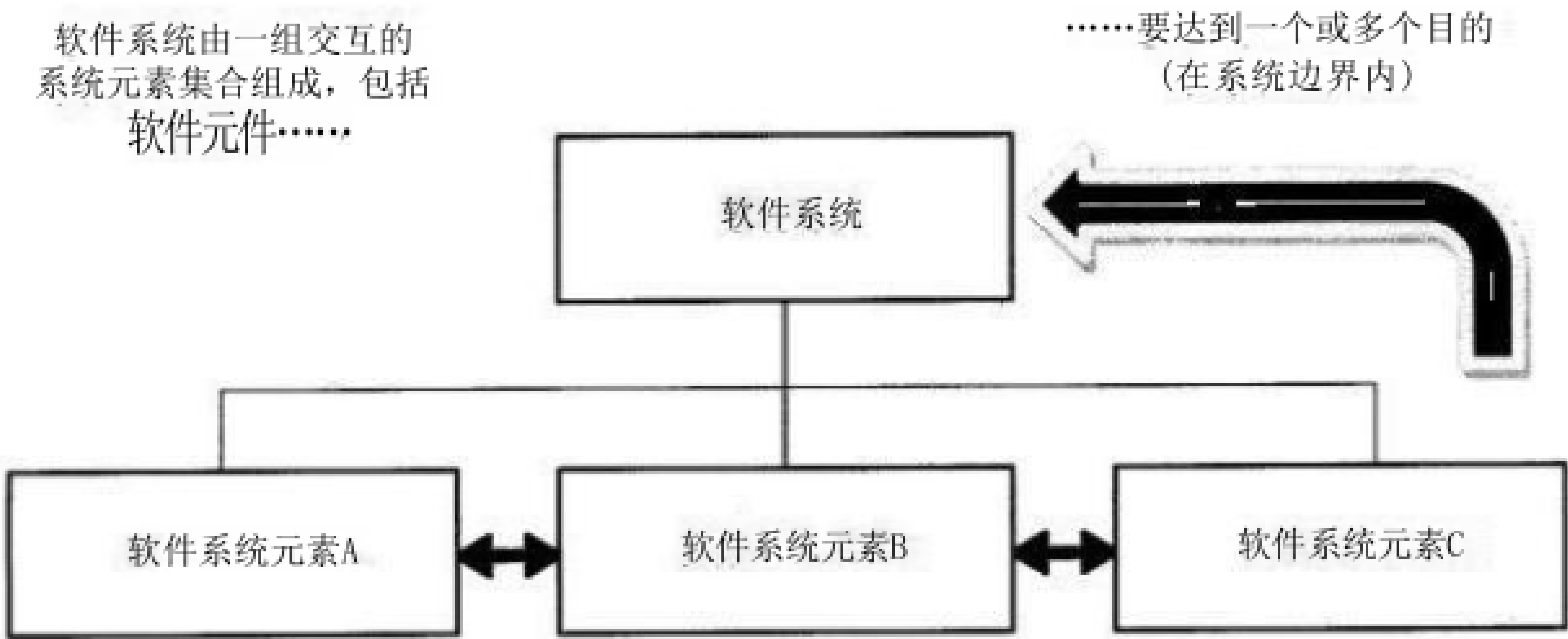


图 1 软件系统和软件系统元素的关系

对于最简单的SOI, 软件系统与其系统元素全集之间的关系通常可以用表示元素间关系的层级结构来描述。分解是某些软件活动中的一种方法，其他方法包括面向对象法，在这种方法中，系统元素以平面(非层次)描述的方式布局，例如在网络图中。对于更复杂的SOI, 在系统元素的全集得到确切定义之前，一个预期的系统元素本身有可能需要被看作一个系统(该系统又进一步由系统元素组成)(图2)。以这种方式，将合适的系统生存周期过程递归地应用于SOI, 用以将其结构分解到可理解的和可管理的系统元素能够实施(创建、调整、获取或重用)的程度。

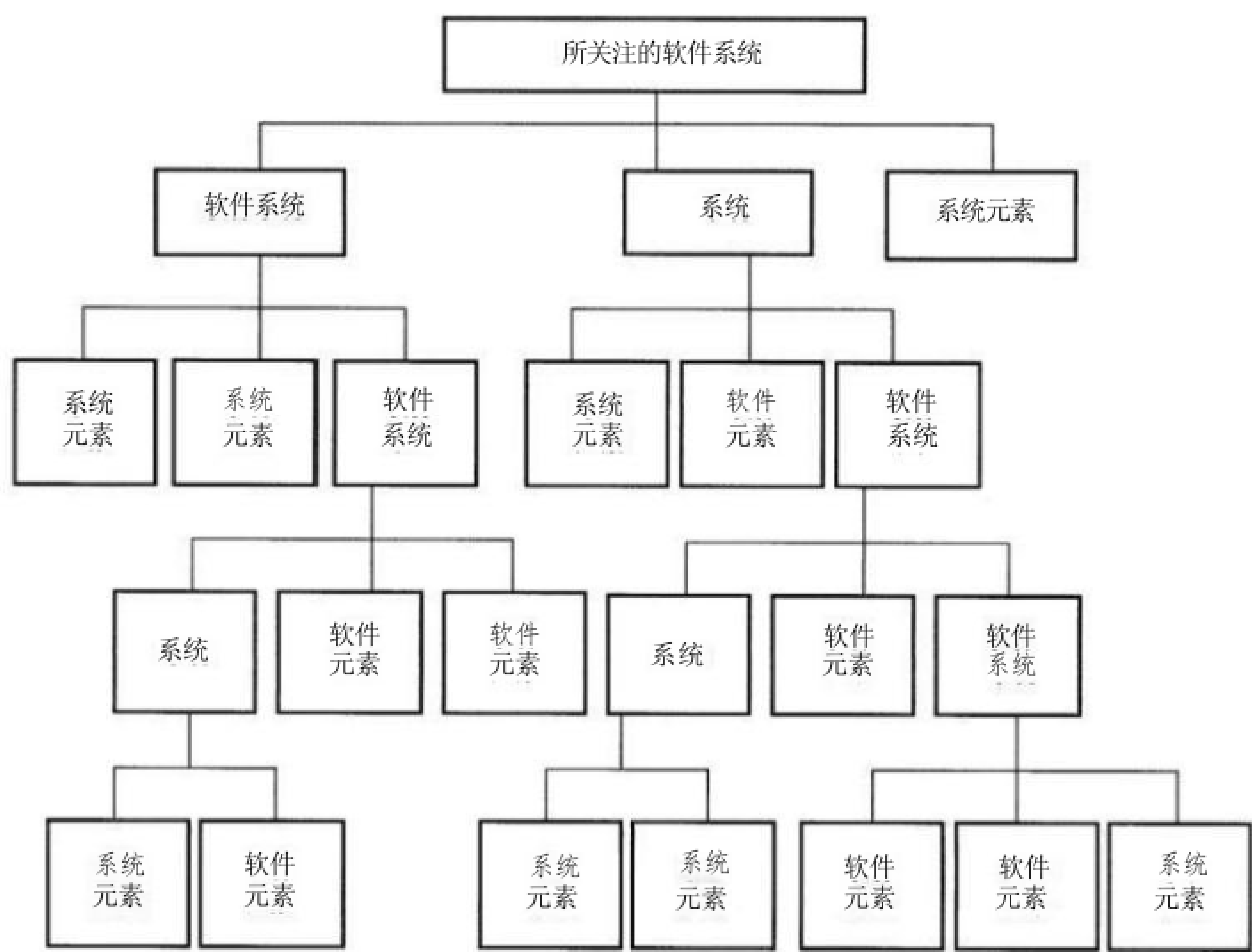


图 2 所关注的软件系统的结构示例

图1、图2隐含了一种层次关系。实际上，从一个或多个方面看，属于非层级结构的系统数量在不断增加，例如网络和其他分布式系统。附录G 讨论了系统之系统(SOS) 的概念。

注：分解对于许多软件活动来说是一种基本活动。并非所有的分解都意味着指定新的软件系统元素以及相应的活动递归应用。只有当适合将不同的需求、设计或实现活动应用于其开发时，才需要将分解的构造指定为元素。适当情况的一个例子是，当元素由不同的组织开发时。另一个例子是，当管理层决定有区别地监视元素的开发或定制状态时。

5.2.3 使能系统

SOI 在其整个生存周期内，需要从那些并不直接属于SOI 运行环境的系统获得必要的服务，例如建模系统、培训系统、维护系统，这些系统每一个都是系统使能的一部分，例如，SOI 的生存周期中要实施的某个阶段。这类系统称为“使能系统”，它们促进了SOI 在其生存周期内的进展。

SOI 为其运行环境提供服务，使能系统为SOI 提供服务，这两种服务的关系见图3。使能系统可视为直接促成了SOI 提供的服务。SOI 和使能系统之间的相互关系可以是双向的，也可是单向的。除了与使能系统交互，SOI 也可能和运行环境中的其他系统交互，如图中的系统A、系统B、系统C 所示。使能系统和运行环境中的其他系统的接口需求，包含在SOI 的需求中。



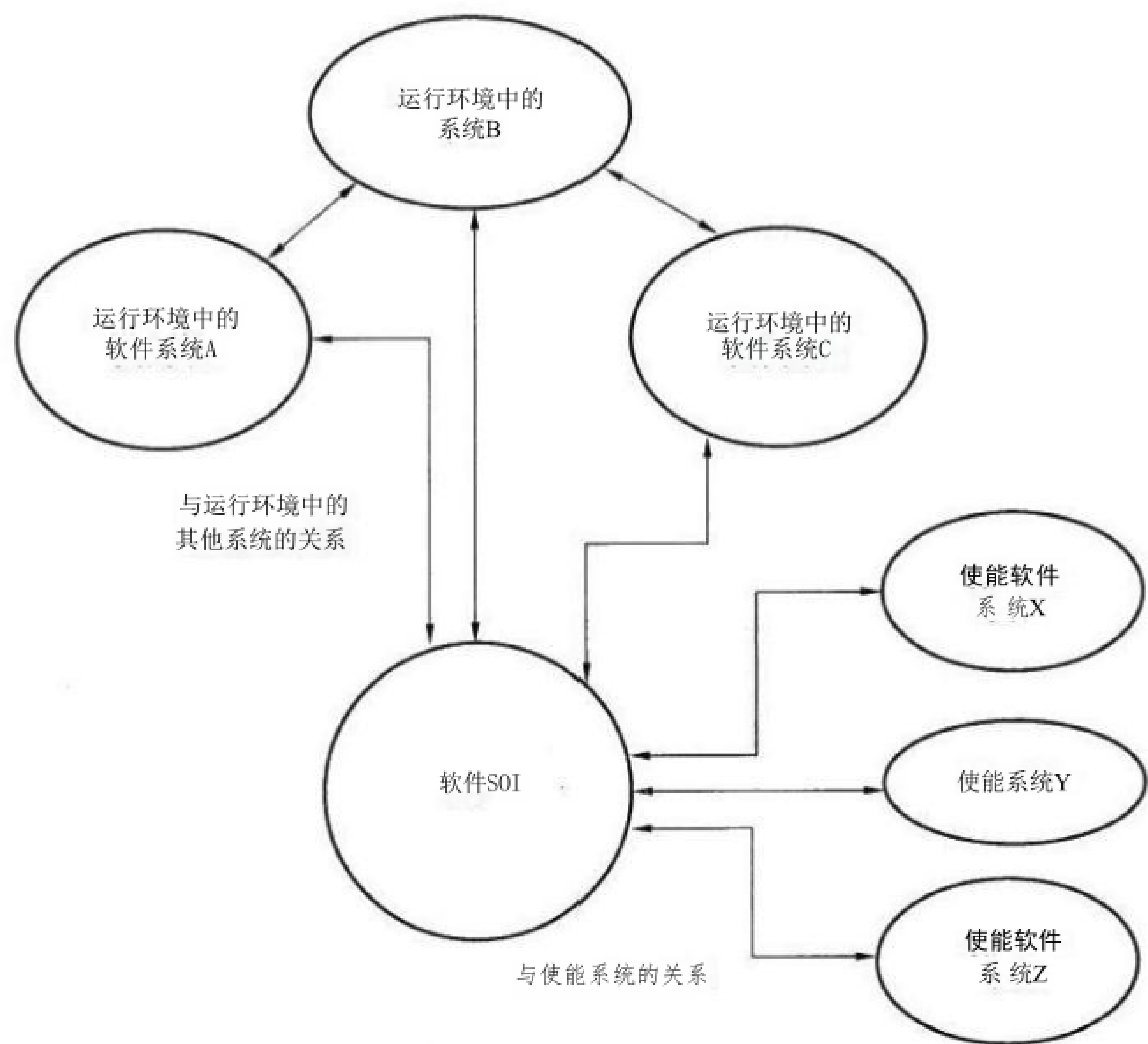


图 3 软件SOI 及其运行环境和使能系统

在软件生存周期的一个阶段中，相关的使能系统和SOI 需统一考虑。因为它们是相互依赖的，它们也可被视为一个系统。当合适的使能系统不存在时，负责SOI 的项目也可直接创建和使用新使能系统。创建使能系统可看成是一个单独的项目，也就是另一个SOI。

这些概念更详细的介绍参见ISO/IEC/IEEE 24748(所有部分)关于生存周期过程的应用。

注：软件开发中的使能系统包括针对目标平台的软件开发和测试环境。

5.2.4 软件系统的生存周期过程

在软件系统中，系统级的需求、架构和设计过程导致将系统需求分配给各种元素。所关注的软件系统主要通过分析软件系统需求、架构和设计来实现，并确定哪些功能将在软件中实现或借助其他元素来实现。SOI 实现软件和其他系统元素，并将这些元素集成为一个软件系统。因此，软件产品或服务可被视为软件系统的一个元素。

在某些情况下，软件系统的架构定义表明，可以将其视为由一组不同的从属元素组成的。反过来，如前面所述，每个软件元素可以被视为一个不同的软件系统。在这些情况下，本文件可以递归地应用于获取或开发从属元素。

本文件与GB/T 22032—2021有很强的相关性，且更适用于软件系统。为说明GB/T 22032—2021和本文件同时适用的情况(如，开发一个包含软件的系统，或者开发一个包含硬件的软件系统)，他们的过程结构是协调一致的。本文件的过程直接对应于专门针对软件产品和服务的GB/T 22032的过程。

若系统中软件元素并不是最重要的，组织可以决定采用GB/T 22032来执行适当的生存周期过程、活动和任务。对于系统中的每个软件元素，组织可以应用本文件来创建、调整、获取或重用软件元素。

## 5.3 组织和项目概念

### 5.3.1 组织

当一个组织作为整体或部分参与到协定中时，就被称作协定方；协定方可以来自同一组织或不同组织。如果个人被授予了组织的相应责任和职权，那么这个组织就与单个个人的地位相当。

在非正式场合，负责执行一个过程的组织有时是以那个过程的名字被提及，例如，执行获取过程的组织有时被称作需方，其他示例包括供方、实现方、维护方和运行方。

本文件跟组织有关的其他术语包括：“用户”是直接参与或从使用产品或服务中获得利益的组织或个人；“顾客”指的是全体用户和需方；“利益相关方”指对系统有利害关系的个人或组织。

过程和组织只是功能上相关。本文件并未指明或暗示组织的结构，也没有规定组织的特定部门执行特定过程。定义合适的组织结构，给执行过程分派合适的角色，是实施本文件的组织职责。

本文件中的过程形成了服务于不同组织的全集。一个组织，无论规模大小，根据其业务目的及获取策略，可以选择合适的过程集(以及相关的活动和任务)来实现该目的。一个组织可以执行一个或多个过程。

本文件既可应用于一个组织的内部，也可用于两个或更多组织的外部。当应用于内部时，达成一致的双方通常根据协定的条款行事，该协定在不同环境下可能会有不同的表现形式。当应用于外部时，达成一致的双方通常根据合同的条款行事。本文件使用的术语“协定”，适用于任何一种情形。

本文件假设任何项目都是在组织的内部环境中进行的。这一点非常重要，因为一个项目取决于组织业务过程产生的各种输出，如，为项目所需职员招聘人员、为项目所需场所提供设施等。为此，本文件提供了一组“组织的项目使能”过程组。这些过程被假定为不足以用于运营业务，而是作为一个集合，用于规定组织对其运行的项目的最基本的支持。

### 5.3.2 组织级和项目级采用

现代业务致力于开发一套健壮的生存周期过程，这些过程被重复应用于业务的项目和服务中。因此，本文件既可在组织级也可在项目级采用并发挥作用。组织可采用本文件，并补充以适当的规程、实践、工具和政策。而组织的某一项目通常遵循组织的过程，而不是直接遵循本文件。

在某些情况下，项目可能会由那些并没有在组织级采用了适当过程集的组织来执行，这样的项目可以直接应用本文件的规定。

## 5.4 生存周期概念

### 5.4.1 软件生存周期阶段

生存周期随着软件系统的性质、目的、用途及实际情况而改变。同时使用不同顺序的阶段，会导致具有明显不同特征的生存周期形式。每个阶段对软件系统的整个生存周期的规划和执行都有不同的目的和贡献。参照ISO/IEC/IEEE 24748-1,典型系统生存周期阶段包括概念、开发、生产、使用，支持和退役。使用这些术语来定义阶段不是规范性要求。软件系统的一组常见阶段是概念设计、开发、维持和退役，以及系统作为一个整体及其元素的阶段之间的转换。

这些阶段代表了与软件系统有关的主要生存时段，这些时段与软件系统描述或软件系统本身的状态相关。这些阶段描述了软件系统整个生存周期的主要进展和成就里程碑。它们带来了生存周期的基本决策点。当创建或使用软件系统时，这些决策点被组织用于理解和管理固有的不确定性和与成本、进度、功能有关的风险。因此，使用“阶段”为组织提供了一个框架，在该框架内组织管理对项目和技术过程有高水平的能见性和可控性。组织以不同的方式定义和使用“阶段”，以满足不同的业务和风险缓解策略。

本文件中定义的生存周期过程并不与软件生存周期中的任何具体阶段一致。所有的生存周期过程都涉及计划、实施和评价活动，宜考虑在每个阶段使用这些活动。

这些概念进一步阐述参见ISO/IEC/IEEE 24748(所有部分)，关于生存周期管理的应用。

#### 5.4.2 软件系统生存周期模型

每个软件系统都有生存周期。可以使用一种抽象的功能模型描述生存周期，这种模型表示了一种对于系统需要及其实现、利用、演变和处置的概念化。

作为行动的结果，软件系统在整个生存周期内逐步发展。这些行动由组织内的人员来执行和管理，人员使用过程来执行这些行动。生存周期模型的细节通过这些过程及其输出、关系和顺序来表达。

本文件并没有规定任何具体的生存周期模型。相反，它定义了过程集合，被称为生存周期过程，这些过程可用于定义系统生存周期。本文件也没有在生存周期模型内规定过程的顺序，过程的顺序由项目目标和所选择的系统生存周期模型来决定。通常，开发阶段以不同的方式被划分得更细。

在前置模型或瀑布模型中一组经常被引用的软件开发阶段是启动、需求、设计、构建和测试。如果这些阶段被认为是连续的，那么每个阶段在进入下一阶段之前都需要产生正确的结果。在实践中，这是非常难以实现的，除非需求是众所周知的，并且初期的成本估计是准确的。在执行瀑布模型的过程中，可能会存在大量返工的风险，而这些返工很可能不在计划的阶段内，因此很可能不在任何预算范围内。

注1:温斯顿·罗伊斯(Winston Royce),是一位公认的生存周期过程模型的早期分析师,描述了对返工阶段的需要,而不是“瀑布”(他没有使用这个术语)。不幸的是,正如人们普遍理解的那样,返工阶段被从“瀑布”模型中移除了。

为了处理不完全已知的需求和不准确的估计,一些其他类型的模型被提了出来:增量式、螺旋式、迭代式和演进式(自适应)。这些生存周期模型包含敏捷技术和方法。这些模型通常会涉及生存周期的过程和阶段在生存周期中的重复实施,例如,对于软件产品的不同增量,对于常见功能异常的更精确处理,或者对于一开始没有完全定义的需求。这些模型可以跨阶段应用,如,开发和利用或部署。这些模型的使用会影响软件服务的软件发布策略和获取策略。

示例:软件原始可以增量地开发,然后在组织的业务周期中适时进行组块操作发布。

“增量开发”模型包括初始规划、初始需求分析、初始架构定义和初始验证,但是将设计、实现、验证(有时是交付)活动分配给一系列的阶段,每个阶段提供预期功能的一部分。该方法提供了一定的灵活性,通过将功能移至后期增量来响应不准确的成本或进度估计。

增量开发的“螺旋”型变化建议对基于风险的功能开发进行排序,并在早期增量中考虑风险最大的问题。这为防止在开发周期后期出现成本意外提供了一定的保护。

“迭代开发”模型执行初始规划,然后包括对需求和解决方案进行原型化、测试、分析和细化的循环过程。“迭代”模型重复执行生存周期过程,以更快地交付优先级高的系统功能,而系统细化的或更复杂的元素将在以后的迭代中出现。

“演进模型”旨在处理不完整的需求知识。它提供了初始规划和初始架构的定义,但是将需求分析、设计、构建、验证、确认和交付分配到一系列的阶段。交付的不满足用户需要的功能可以在演进的后续阶段重新设计。

“敏捷”方法实际上可以应用于各种模型中。虽然敏捷方法在执行演进的生存周期模型中很常见,但它们可用于其他生存周期模型的不同阶段。这些方法的共同之处在于,在一个预期变更(包括需求变更)的环境中,强调在快速生产工作软件过程中的持续检查和协作。附录H提供了敏捷环境下应用本文件的信息。

注2:选择一种模型的名称并不满足定义一个由阶段组成的模型的要求,该模型具有通过本文件的过程实现的明确的目的和结果,

注3:ISO/IEC/IEEE 24748-1、ISO/IEC/IEEE 24748-2、ISO/IEC/IEEE 24748-3以及ISO/IEC/IEEE 24748-4提供了关于生存周期模型和阶段的其他细节。本章中描述的模型不仅适用于软件系统,而且也适用于GB/T 22032—2021中描述的其他系统。

## 5.5 过程概念

### 5.5.1 过程的准则

本文件基于以下三项基本准则确定生存周期过程：

- a) 每个生存周期过程在它的输出、活动和任务之间紧密联系；
- b) 将过程间的依赖性在可行的范围内减少到最低程度；
- c) 在生存周期内，一个过程能由单一组织执行。

### 5.5.2 过程描述

本文件的每个过程均由以下属性的形式描述：

- a) 标题从整体上概括过程的范围；
- b) 目的描述执行过程的目标；
- c) 输出表示通过成功执行过程预期可取得的可见结果；
- d) 活动是一个过程中一组结合紧密的任务；
- e) 任务是需求、建议或许可的用于支持实现输出的行动。

本文件中的过程和过程组在其目的和输出方面与GB/T 22032—2021是一致的，除了一个例外：本文件的系统/软件需求定义过程由GB/T 22032—2021的系统需求定义过程进行了重命名。为了突出系统和软件系统过程的这种一致性，过程目的和输出在第6章中展现。

本文件中，软件特定的活动、任务和工作产品应用于实现过程的输出。附录E 提供了附加的过程视图。

关于过程描述形式的其他细节参见GB/T 30999—2014。

### 5.5.3 过程的通用特性

除5.5.2描述的基本属性外，过程还可被共用于所有过程的其他属性来刻画。ISO/IEC 33020:2015标识了公共的过程属性，在过程能力的测量框架范围内，这些属性将过程能力刻画为6个可达到的级别。附录C 包含了ISO/IEC 33020:2015 中定义的有助于实现较高层级过程能力的过程属性列表。

### 5.5.4 剪裁

附录A 规定了对本文件进行剪裁所需的基本活动。需注意的是，剪裁可能会削弱对本文件符合性要求的感知价值。这是因为其他组织难以理解剪裁可能会删除多少期望的条款。单方面声称符合本文件的组织可能会发现，声明对较小规模过程列表的完全符合，比声明对较大规模过程列表的剪裁符合更具优势。

## 5.6 过程组

### 5.6.1 导引

本文件把可以在软件系统生存周期中执行的活动分成四个过程组。这些组中的每个生存周期过程根据其目的、期望输出和为实现这些输出待执行的一组活动和任务来描述。这四个过程组和每组所包含的过程如图4所示：

- a) 协定过程组；
- b) 组织的项目使能过程组；
- c) 技术管理过程组；

d) 技术过程组。

本文件所描述的过程并不意味着阻止或阻碍组织发现有用的附加过程的使用。本文件中所定义过程的章条的顺序并不决定过程在软件系统生存周期或其任何阶段中执行的顺序。每个过程组的描述见 5.6.2～5.6.5。



图 4 软件生存周期过程

### 5.6.2 协定过程组

组织是软件系统的生产者 and 用户。一个组织(作为需方)可以给另一个组织(作为供方)提出产品或服务的任务,这通过使用协定实现。协定允许需方和供方实现价值,并支持其组织的商业战略。

协定过程组是应用于项目生存周期内外的组织过程。通常,组织同时或依次扮演软件系统的需方和供方。当需方和供方在同一组织中时,使用协定过程的正式程度可以低一些;同样,它们可以在组织内就各自的职责、项目和技术功能达成一致。本过程组包含的过程见图4。

### 5.6.3 组织的项目使能过程组

组织的项目使能过程组关注的是提供所需资源来使能项目,从而满足组织利益相关方的需要和期望。组织的项目使能过程组通常在战略层面关注组织内业务或事业的管理和改进、资源和资产的提供和部署、竞争或不确定情况下的风险管理。组织的项目使能过程组用于项目生存周期内外。

组织的项目使能过程组建立了项目运行的环境。组织建立了用于项目的过程和生存周期模型,建立、重定向或取消项目,提供包括人员和资金在内的资源,对于软件系统以及为内外顾客开发的其他可交付物,确定并监控质量措施。

组织的项目使能过程组既有许多组织创造强大的业务前景,意味着商业和盈利动机,同样也与非营利组织有关,因为它们对利益相关方的资源及其事业中遭遇的风险负责。因此,本文件既可用于非营利组织,也可用于营利组织。本过程组包含的过程见图4。

### 5.6.4 技术管理过程组

技术管理过程组对组织分配的资源 and 资产实施管理,并且利用它们完成组织或组织间达成的协定。技术管理过程组与项目的技术工作相关,特别是成本、时间范围和成果的计划,确保与策划和完成准则一致的行动检查,以及识别和选择纠正措施。它们用于制定和执行项目技术计划、管理跨技术过程组的信息、根据软件系统产品或服务计划评估技术进展、控制技术任务直至完成,为决策过程提供帮助。

注1:技术管理是“技术和行政资源对计划、组织和控制工程功能的运用”。(ISO/IEC/IEEE 24765:2010)

通常,任何一个组织中可以有多个项目共存。技术管理过程可以在组织层次实施,以满足内部需要。本过程组包含的过程见图4。

注2:技术管理过程适用于每个技术过程的实施期间。

### 5.6.5 技术过程组

技术过程组关注整个生存周期中的技术活动。技术过程组把利益相关方的需求转化为产品或服务。通过应用该产品或运营该服务,技术过程组在需要的时间和地点提供可持续的性能,从而满足利益相关方的需求并获得顾客的满意。应用技术过程组是为了创建和使用软件系统,而不论它是模型形式还是可运行的产品。技术过程组应用于软件系统结构层次中的任何层次和生存周期中的任何阶段。本过程组包含的过程见图4。

## 5.7 过程应用

当获取、使用、创建或供应软件系统时,本文件定义的生存周期过程可用于任何组织。它们可应用于系统层次结构中的任何层次和生存周期中的任何阶段。

使用特定的目的、输出和组成过程的一组活动与任务的形式定义过程完成的功能。

根据需要,在生存周期的任何时间点都可引用图4中每个生存周期过程。本文件呈现的过程顺序并不意味着它们的使用有规定的顺序。但是,通过定义生存周期模型引入了顺序关系。存在多种因素影响生存周期中过程的使用目的和使用时机,包括社会的、交易的、组织的和技术的考虑,每个因素都会



在软件系统生存期间发生变化。因此，一个单独的软件系统生存周期，是一个由过程组成的复杂系统，这些过程通常具有并发、迭代、递归和时间依赖的特征。

对过程的并发使用可存在于某个项目内(例如，当创建软件系统的设计行动和预备行动同时执行时)以及项目之间(例如，当系统元素在不同的项目任务下同时被设计时)，

当相同的过程或过程集合被重复地应用于相同的系统，该应用称为迭代。过程的迭代使用对于过程输出结果的逐步细化是至关重要的，例如，连续的验证行动和集成行动之间的交互，可逐步地建立对产品符合性的信心。迭代不仅是合适的，也是所期望的。一个过程或过程集合的应用可产生新的信息。这些信息通常表现为关于需求、风险分析或机遇的问题。应在完成一个过程或过程集合的行动之前解决这些问题。

过程的递归使用，例如，在系统层次结构中相邻层次的系统元素上对相同的过程或过程集合的重复应用，是本文件应用的一个关键方面。在任何层次上的过程输出，无论是信息、制品还是服务，都是在较低等级(例如，自顶向下设计期间)或较高等级(例如，软件系统实现期间)使用的过程的输入。一个应用的输出用于系统结构中下一个更低(或更高)系统的输入，以实现更详细或成熟的输出集合。这种方法为连续系统在系统结构方面增加了价值。

对软件系统的影响不断变化的特性(例如，运行环境的变化、系统元素实现的新机遇、组织中变更的结构和责任)，要求对过程应用的选择和时机不断进行评审。生存周期中的过程应用可以是动态的，以应对软件系统的外部影响。生存周期的方法也允许在下一个阶段中合并更改。面对生存周期中的这种复杂性，通过提供可理解的和可识别的高层次的目的和结构，生存周期阶段帮助生存周期过程的计划、执行和管理。应用生存周期阶段内一组过程，其共同目标是满足该阶段的退出准则或满足该阶段内正式进入评审的准入准则。

本条讨论的软件生存周期过程的迭代和递归使用，并不暗示 SOI、使能系统、组织或项目的任何特定层次的、垂直的或水平的结构。

如果产品质量风险合理，也可以创建特定产品环境中的过程实例的详细描述。过程实例化包含为每个过程实例确定具体的成功准则，从产品需求导出，确定为实现此成功准则所需的活动和任务，从本文件确定的活动和任务导出。建立更详细的过程实例描述，在过程和具体产品需求之间建立联系，可以使产品质量风险得到更好的管理。

这些概念的进一步阐述参见ISO/IEC/IEEE 24748(所有部分)，关于生存周期过程的应用。

## 5.8 过程参考模型

附录C 定义了一个比第6章包含的详细需求更高抽象层级的过程参考模型(PRM)。PRM 适用于评估其过程以便确定这些过程能力的组织。目的和输出是每个过程实施目标的陈述。不同于简单的符合性评估方式，这种目标陈述允许过程有效性的评估。

注：在本文件中，术语“过程参考模型”的用法与ISO/IEC 33001:2015具有相同的含义：“模型包含应用领域中根据过程目的和输出来描述的过程定义，以及描述过程之间关系的体系结构”。

## 6 软件生存周期过程

### 6.1 协定过程组

本条规定了与组织外部和内部的组织实体建立协定的要求。

协定过程组组成如下：

- a) 获取过程——用于组织获取产品或服务；
- b) 供应过程——用于组织提供产品或服务。

这些过程定义了两个组织之间建立协定所必需的活动。若调用获取过程，则提供了与供方开展

业务的方法。这可能包括提供运营软件系统的产品、支持运营活动的服务、系统中的软件，或由供方提供的软件系统。若调用供应过程，则提供了为需方提供产品或服务的协定方法。

注：在系统和软件工程中，信息安全日益受到关注。ISO/IEC 27036(所有部分)为供方和需方提供了如何在供方关系中实现信息安全的要求和指导。供方关系中信息安全的具体内容参见ISO/IEC 27036-3:2013和ISO/IEC 27036-4(开发中)。

### 6.1.1 获取过程

#### 6.1.1.1 目的

获取过程的目的是根据需方的要求获得产品或服务。

注：作为本过程的一部分，当供需双方同意一个影响协定条款的变更请求时，协定即予以修改。

#### 6.1.1.2 输出

获取过程成功实施后，结果如下：

- a) 已准备好供货申请；
- b) 选择了一个或多个供方；
- c) 供需双方之间签订了协定；
- d) 接收了符合协定的产品或服务；
- e) 满足协定中规定的需方义务。

#### 6.1.1.3 活动和任务

需方应当按照与获取程有关的组织方针与规程实施下列活动和任务。

注1:该过程中的活动和由此产生的协定通常适用于供应链中的供方，包括分包供方。

注2:IEEE Std 1062-2015,包含软件获取备选方案的详细活动，包括定制开发、商业现货软件和软件即服务。IEEE Std 1062—2015也提供了软件获取指南，包括其涉及的技术数据权利和知识产权，及其关注的安全保证或信息安全要求，同时为建立软件获取过程的组织提供了相关问题清单。

- a) 做获取准备。该活动由以下任务组成：

##### 1) 定义获取策略；

注1:如果供方在需方组织外部，该策略可描述或参考生存周期模型、风险和问题缓解、里程碑进度表，以及选择准则。它还包括获取的关键驱动因素和特征，如，责任和义务；具体的模型、方法或过程；重要程度；形式；以及相关交易因素的优先权。

注2:决策管理过程和系统分析过程通常用于支持获取策略的权衡。例如包括：在决定自研或购买时，需要评价特定商业现货软件(COTS)适用性或修改现货软件(OTS)的解决方案及供方。

注3:如果该策略要求获取特定的商业现货软件或开源软件，获取可限制在识别供方，接受或协商预设许可、租赁或维护协定的条件，确定软件系统中的知识产权和数据权利，并商定价格。

注4:供方协定中要考虑的一个因素是数据权利和对组成数据和知识产权的横向访问。例如，一个系统组件的供方可能需要与另一个组件的供方合作并共享源代码。协定可以促成这个合作。

##### 2) 应准备所需产品或服务的外包申请，申请应包含所有需求。

注1:如果供方是外部的组织，则该请求包括期望供方遵守的业务惯例和选择供方的准则。

注2:将需求定义提供给一个或多个供方。需求是利益相关方或系统/软件需求，取决于获取途径的类型，并且可以使用相关的需求定义过程。

注3:需方自行开发需求或委托某个供方开发需求。如果需方委托某个供方来开发需求，则需方保留对供方所开发出的需求进行批准的权力。

- b) 发布本次获取并选择供方。该活动由以下任务组成：

##### 1) 向潜在的供方沟通产品或服务的要求；



2) 选择一个或多个供方。

注：为了获得竞争性招标，供方提案将根据甄选准则进行评价和比较，并进行排名。说明对每个提案的评级理由，并告知供方入选或落选的原因。

c) 完成并维护协定。该活动由以下任务组成：

注：通过项目评估与控制过程监控项目成本、进度和执行情况。需要修改协定时都涉及该活动。任何更改系统元素或信息的建议，都通过管理过程的变更管理活动来控制。

1) 与供方达成包括验收准则在内的协定：

注1:协定不拘形式，严如书面合同，宽如口头约定。协定使用恰当的形式规定：需求，开发和交付里程碑，验证、确认和验收条件，异常处理规程，协定变更管理规程和付款进度，以便协定双方理解执行协定的基础。协定的其他规定还包括与技术数据和知识产权相关的权利和限制、验收测试准备和测试环境细节以及供方的参与程度。该协定确定了需要强加给参与分包商的过程要求，如配置管理要求、风险报告测度和测量分析报告。

注2:验收准则，如验收测试，涉及产品或服务在其运行环境中如何满足其预期用途。可以使用确认过程执行验收测试。在协定执行期间或伴随交付的产品或服务产生的意外情况，根据协定建立的规程予以解决。

2) 确定协定的必要变更：

注：在请求协定变更时，需方或供方详细说明变更的具体规格，合理原因和相关背景。

3) 评估协定变更带来的影响：

注：任何变更都要调查其对项目计划、进度、成本、技术能力及质量的影响。变更的方式可以在现有协定内处理，也可以要求修改现有协定，或者签订一份新协定。

4) 与供方协商协定：

注：在供需双方之间对条款进行协商，协商的出现是为了初步的协定，以及为任何变更所需要。变更后的协定以所需的变更和所识别的影响为基础。在协商期间讨论和变更细节，之后需方和供方接受协定条款且协定生效。对于书面合同，在合同签署或按协定中规定进行双方协商。

5) 必要时更新与供方的协定。

注1:协定变更的结果纳入项目计划，并传达给所有受影响方。

**注 2 :**协定可以规定任何一方终止协定的条件，如，策略或可用资金的意外变化，或进展不顺。

d) 监督协定。该活动由以下任务组成：

1) 评估协定的执行情况：

注1:包括确认各方按照协定履行其职责。项目评估和控制过程用于评估预计成本、进度、实施情况，以及非预期结果对组织的影响。此信息与协定条款执行的其他评估相结合。如果本协定的执行没有产生可接受的产品或服务，则需方或供方可以在其条款允许的情况下终止本协定。

注2:可以使用确认过程执行验收测试。根据协定中制定的规程，对在执行本协定期间或伴随着交付的产品或服务出现的意外情况予以解决。

2) 为供方提供所需的数据，及时解决问题。

e) 验收产品或服务。该活动由以下任务组成：

1) 确认交付的产品或服务满足协定：

注：如果满足商定的要求并满足验收准则，则认定供方已履行其义务。未解决的异常情况，例如，关于验收测试的实施或产品适用性的争议，由与争议、仲裁或适用法律法规相关的协定条款来确定。

2) 付款或提供协议规定的其他事项；

3) 按照协定接受供方或其他方的产品或服务；

4) 结束协定。

注：通过特定项目包管理过程终止项目。

## 6.1.2 供应过程

### 6.1.2.1 目的

供应过程的目的是给需方提供满足协定要求的产品或服务。

注：作为本过程的一部分，当供需双方都同意影响协定条款的变更请求时，协定予以修改。

### 6.1.2.2 输出

供应过程成功实施后，结果如下：

- a) 明确了产品或服务的需方；
- b) 响应了需方的请求；
- c) 供需双方之间订立了协定；
- d) 提供了产品或服务；
- e) 供方履行了协定所界定的义务；
- f) 按照协定的规定，转移了所获取的产品或服务的质量责任。

### 6.1.2.3 活动和任务

供方应按照与供应过程有关的组织方针与规程实施下列活动和任务。

- a) 准备供应。该活动由以下任务组成：

- 1) 确定需要产品或服务的需方的存在及身份；

注：这经常通过业务或使命分析过程产生。对于面向消费者开发的产品或服务，某个代理（例如，供方组织内的某个营销职能部门）往往代表需方。

- 2) 定义供应策略。

注：本策略描述或引用生存周期模型、风险和问题缓解，以及里程碑的时间表。它还包括获取的关键驱动因素和特性，如，职责和义务；特定的模型、方法或过程；关键程度；形式；以及相关交易因素的优先权。

- b) 响应产品或服务的供应请求。该活动由以下任务组成：

- 1) 评估产品或服务的供应请求，以确定可行性和如何响应；

- 2) 准备满足请求的响应。

- c) 完成并维护协定。该活动由以下任务组成：

- 1) 与需方订立包括验收标准在内的协定；

注：协定不拘形式，严如书面合同，宽如口头约定。供方确认需求、交付里程碑和验收条件是可实现的，异常处理和协定变更管理规程和付款进度是可接受的，并确保协定的执行并规避不必要的风险。在谈判期间，双方讨论并修改细节，供需双方接受协定条款之后，协定生效。对于书面合同，在合同签字时生效。

- 2) 标识（识别）协定的必要变更；

注：在请求变更协定时，需方或供方应详细说明变更的具体规格、合理原因和相关背景。

- 3) 评估变更对协定的影响；

注：任何变更都要调查其对项目计划、进度、成本、技术能力及质量的影响。变更的方式可以在现有协定内处理，也可以要求修改现有协定或者签订一份新协定。

- 4) 必要时与需方协商协定；

注：协定条款的变更在需方和供方之间进行协商。这包括改变市场环境引起的变化。谈判发生于初始协定以及任何所需的变更时。变更后的协定基于所需的变更和识别的影响。

- 5) 必要时与需方更新协定。

注：协定修改的结果纳入项目计划，并传达给所有受影响方。

- d) 执行协定。该活动由以下任务组成：

- 1) 根据建立的项目计划执行协定；

注：供方有时采用或同意使用需方过程。

2) 评估协定的执行情况。

注：这包括确认各方按照协定履行其职责。项目评估和控制过程用于评估预计成本、进度、实施情况，以及不符期望的输出对组织的影响。配置管理过程的变更管理活动用于控制对系统元素的变更。此信息与协定条款执行的其他评估相结合。如果本协定的执行没有产生可接受的产品或服务，则需方或供方可以在其条款允许的情况下终止本协定。

e) 交付和支持产品或服务。该活动由以下任务组成：

1) 根据协定准则(标准)交付产品或服务；

注：如协定所述，验收标准(如验收测试)涉及产品或服务在其运行环境中如何满足其预期用途。未解决的异常情况，例如，关于验收测试的实施或产品适用性的争议，由与争议、仲裁或适用法律法规相关的协定条款来确定。

2) 按照协定，向需方提供所交付的产品或服务的支持协助；

3) 接受并确认付款或其他议定的报酬；

4) 向需方或协定中指定的其他方交付产品或服务；

5) 结束协定。

注1:通过特定项目包管理过程终止项目。

注2:协定可以规定任何一方终止协定的条件，例如，策略或可用资金的意外变化或进展不顺。

6.2 组织的项目使能过程组

通过项目的发起、支持和控制，组织的项目使能过程组有助于确保组织获取和供应产品或服务的能力。这些过程提供了支持项目所需的资源和基础设施，并有助于确保组织的目标和建立的协定得到满足。组织的项目使能过程组并非一套能够对组织的全面业务进行战略管理的业务过程。

组织的项目使能过程组包括以下内容：

- a) 生存周期模型管理过程；
- b) 基础设施管理过程；
- c) 特定项目包管理过程；
- d) 人力资源管理过程；
- e) 质量管理过程；
- f) 知识管理过程。

6.2.1 生存周期模型管理过程

6.2.1.1 目的

生存周期模型管理过程的目的是定义、维护并确保供组织在本文件范围内使用的方针、生存周期过程、生存周期模型和规程的可用性。

该过程提供了与组织目标一致的生存周期方针、过程、模型和规程，这些方针、过程、模型和规程经过定义、调整、改进和维护，以支持组织内各个项目需要，并且能够通过有效、经过证明的方法和工具进行应用。

6.2.1.2 输出

生存周期模型管理过程成功实施后，结果如下：

- a) 建立了管理和部署生存周期模型和过程的组织方针与规程；
- b) 定义了生存周期方针、过程、模型和规程中的责任、义务和权限；
- c) 评估了组织使用的生存周期模型和过程；

d) 对过程、模型和规程改进定义了优先级。

### 6.2.1.3 活动和任务

组织应根据与生存周期模型管理过程有关的组织方针与规程实施下列活动和任务。

a) 建立过程。该活动由以下任务组成：

注：项目中生存周期实施的细节取决于工作的复杂性、所使用的方法，以及参与执行工作的人员的技能和受到的培训。项目根据其需求和需要，对方针、过程、模型和规程进行剪裁，同时保持与法规和组织方针的一致性。附录A包含了关于剪裁的信息。

- 1) 为过程管理和部署建立与组织策略一致的方针和规程；
- 2) 建立实施本文件要求并符合组织策略的过程；
- 3) 定义角色、责任、义务和权力，以促进过程的实施和生存周期的策略管理；
- 4) 定义控制整个生存周期进展的业务准则；

注：建立关于进入和退出每个生存周期阶段和关键里程碑的判定准则。这些有时是以业务完成情况来表示的。

- 5) 为组织建立包含各阶段的标准生存周期模型，并定义每个阶段的目的和输出。

注：生存周期模型根据需要包含一个或多个阶段模型。根据SOI的范围、规模、复杂性、不断变化的需要和机会，它被组合成一系列可以重叠或迭代的阶段。ISO/IEC/IEEE 24748-1中使用了一个常见的生存周期阶段示例，对阶段进行了阐述。ISO/IEC/IEEE 24748-2和ISO/IEC/IEEE 24748-3提供了系统和软件的具体示例。在一个阶段中，对生存周期过程和活动进行选择、适当剪裁和使用，来达成该阶段的目的和输出。

b) 评估过程。该活动由以下任务组成：

注：ISO/IEC 33002:2015提供了一组更详细的过程评估活动和任务，这些活动和任务可与下面所示的任务相匹配。

- 1) 监控整个组织的过程执行情况；

注：这包括监控性能、分析过程测量，并评审与法规、组织政策、业务标准的遵从性有关的趋势，以及来自项目的有关过程有效性和效率的反馈。

- 2) 对项目使用的生存周期模型进行定期评审；

注：这包括确认项目使用的生存周期模型的持续适用性、充分性和有效性，并酌情进行改进。还包括控制整个生存周期进展的阶段、过程和完成准则。

- 3) 从评估结果中识别改进机会。

注：改进可以影响控制整个生存周期进展的阶段、过程和完成准则。

c) 改进过程。该活动由以下任务组成：

- 1) 对改进机会进行优先级排序和策划；
- 2) 实施改进，并告知利益相关方。

注：过程改进包括对组织中的任何过程的改进。汲取的经验教训是可以捕捉和获得的。

## 6.2.2 基础设施管理过程

### 6.2.2.1 目的

基础设施管理过程的目的是为项目提供基础设施和服务，以支持贯穿整个生存周期的组织和项目目标。

这一过程在本文件范围内为组织的业务定义、提供并维护了所需的设施、工具以及通信和信息技术资产。

### 6.2.2.2 输出

基础设施管理过程成功实施后，结果如下：

- a) 定义了基础设施需求；

- b) 识别并详述了基础设施元素；
- c) 开发或获取了基础设施元素；
- d) 形成了可供使用的基础设施。

### 6.2.2.3 活动和任务

组织应根据与基础设施管理过程有关的组织方针与规程实施下列活动和任务。

#### a) 建立基础设施。该活动由以下任务组成：

- 1) 定义项目基础设施需求；

注1:基础设施元素示例包括设施、工具、硬件、软件、服务和标准。除了组织通用的用以支持其业务过程的通用基础设施资源外，组织还可以为项目提供独特或共享的使能系统，以支持项目的技术过程组。

注2:项目的基础设施资源需要结合组织内的其他项目和资源以及组织的政策和战略规划考虑。影响和控制项目基础设施资源和服务获取的业务约束和时间线也要进行评估。项目计划和未来业务要有助于理解所需的资源基础结构。此外还需考虑物理因素(例如设施)、保障需要和人为因素(包括健康和安全方面)。

注3:ISO/IEC 27036为解决外包基础设施的信息安全问题提供了指导。

- 2) 识别、获取并提供实施和支持项目所需的基础设施资源和服务。

注：通常建立库存资产登记，以跟踪基础设施元素并支持重用。

#### b) 维护基础设施。该活动由以下任务组成：

- 1) 评估交付的基础设施资源满足项目需要的程度；
- 2) 当项目需求变更时，识别并改进或变更基础设施资源。

## 6.2.3 特定项目包管理过程

### 6.2.3.1 目的

特定项目包管理过程的目的是启动并维持必要、充分、适当的项目，以实现组织的战略目标。

该过程确保投入足够的组织经费和资源，批准建立选定项目所需的授权，并实施项目的持续评估，以直接或间接支撑持续投资这些项目的合理性。

对于软件系统，特定项目包管理通常也指为满足组织或顾客的需要和目标、支持技术的革新所进行的产品线管理(产品线指资产、产品及使能系统的项目组合，也可指服务目录)。资产管理是通过项目管理来实现的。

### 6.2.3.2 输出

特定项目包管理过程成功实施后，结果如下：

- a) 对风险机遇、投资或必需品进行确认及排序；
- b) 识别了项目；
- c) 分配了每个项目的资源和预算；
- d) 定义了项目的管理责任、义务和权限；
- e) 维持了满足协定和利益相关方需求的项目；
- f) 更改或终止了不满足协定或利益相关方需求的项目；
- g) 关闭了已完成协定并满足利益相关方需求的项目。

### 6.2.3.3 活动和任务

组织应基于特定项目包管理过程有关的适当的组织方针和过程，实施下列活动和任务。

#### a) 定义和批准项目。该活动由以下任务组成：

1) 识别潜在的新/改良后的能力或使命；

注：对组织业务策略、运营观念、差距分析或机会分析进行评审，以了解当前的差距、问题或机会。新的能力或企业需要通常在业务或使命分析过程中确定，在利益相关方需要和需求定义过程中进一步定义，并通过本过程进行管理。

2) 优先排序、选择和建立新的业务机会、风险投资或经营项目；

注：这些业务机遇、风险投资或经营项目通常与组织的业务战略和行动计划相一致。对潜在的项目进行优先排序，并设立判定门限，以确定将要执行哪些项目。已识别的项目的特征往往是确定的，包括利益相关方的价值、成功的风险和障碍、依赖关系和相互关系、约束、资源需要和资源的相互竞争。然后根据成功可能性和成本效益对每个潜在项目进行评估。决策管理过程和系统分析过程提供了备选方案分析的详细信息。

3) 定义项目、责任和权限；

4) 确定每个项目的预期目标、目的和输出；

5) 确定并分配资源以实现项目的目的和目标；

6) 确定每个项目要管理或支持的多项目接口和依赖项；

注1:这包括多个项目所使用的使能系统的使用或重用，以及多个项目所使用的公用系统元素(包括软件元素)的使用或重用。

注2:理解企业架构背景中的每一项目有助于确保接口和约束得以识别。

7) 明确项目报告要求，并对控制每个项目执行的里程碑进行评审；

8) 授权每个项目开始执行项目计划。

注：有关开发项目计划的其他信息，参见项目规划过程。项目计划在项目生存周期的早期开发和批准时最为有用。

b) 评价项目的特定项目包。该活动由以下任务组成：

1) 评价(评估)项目以确定持续的可行性；

注：可行性包括以下准则：

i) 该项目正在向既定目标推进；

ii) 该项目遵循项目指令；

iii) 该项目根据批准的项目生存周期政策、过程和规程进行；

iv) 该项目保持其可行性，正如对服务的持续需求、切实可行的产品实施和可接受的投资效益等所表明的。

2) 采取行动继续或调整正在取得令人满意进展的或可预期通过适当调整取得令人满意进展的项目。

c) 终止项目。该活动由以下任务组成：

1) 在协定允许的情况下，取消或暂停那些对组织不利或风险大于持续投资收益的项目；

注：从被取消或失败项目中吸取的经验教训对于组织改进或用于其他项目可能是特别有用的。

2) 对产品和服务的协定执行完毕后，关闭项目。

注：关闭是按照组织政策、规程和协定完成的。

## 6.2.4 人力资源管理过程

### 6.2.4.1 目的

人力资源管理过程的目的是为组织实现其业务需要提供必要的人力资源，并保持其能力。

这一过程为执行生存周期过程以实现组织、项目和利益相关方的目标提供具备必要技能和经历的合格人员供应。

### 6.2.4.2 输出

人力资源管理过程成功实施后，结果如下：

- a) 确定了项目所需的技能;
- b) 为项目提供了必要的人力资源;
- c) 开发、保持或提升了人员的技能;
- d) 解决了多项目资源要求冲突。

#### 6.2.4.3 活动和任务

组织应根据与人力资源管理过程适用的组织方针与规程实施下列活动和任务。

- a) 确定技能。该活动由以下任务组成:

- 1) 根据当前和预期的项目确定技能需要;
- 2) 确定并记录人员技能。

- b) 培育技能。该活动由以下任务组成:

- 1) 建立技能培育策略;

注: 该计划包括培训类型和层次、人员类别、计划表、人事资源需求和培训需要。

- 2) 获得或开发培训、教育或辅导资源;

注: 这些资源包括由组织或外部合作方开发的培训材料, 从外部供方可获取的培训课程, 以及基于计算机的教学。

- 3) 提供有计划的技术培育;

- 4) 维护技能培育的记录。

- c) 获得并提供技能。该活动由以下任务组成:

注: 这包括为项目足量配备具有必备经验水平和技能的人员而进行的人员招募、保留, 以及对员工进行评估和评审, 例如, 他们的熟练程度、积极性、团队工作能力, 以及是否有再培训、重新指派或重新分配的需要。

- 1) 当识别到技能缺失时, 获得合格人员;

注: 包括使用外包资源。

- 2) 维护和管理为正在进行的项目配备人员所需的技能人员池;

- 3) 基于项目和员工发展的需要项目分配;

- 4) 激励员工, 例如, 通过职业发展和奖励机制;

- 5) 控制多项目管理接口以解决人员冲突。

注: 这包括正在进行的项目之间在组织基础设施和支持服务及人事资源方面的能力冲突; 或源自项目人员的过度投入带来的冲突。

#### 6.2.5 质量管理过程

##### 6.2.5.1 目的

质量管理过程的目的是, 确保产品、服务和质量管理过程的实施符合组织和项目的质量目标, 并且令顾客满意。

##### 6.2.5.2 输出

质量管理过程成功实施后, 结果如下:

- a) 定义和实现了组织质量管理方针、目标和规程;
- b) 建立了质量评价准则和方法;
- c) 为项目提供了资源和信息, 以支持项目质量保证活动的运行和监控;
- d) 收集和分析了质量保证评价结果;
- e) 基于项目和组织成果改进了质量管理方针和规程。

### 6.2.5.3 活动和任务

组织应按照质量管理过程适用的组织方针与规程实施下列活动和任务。

注：有关建立质量管理体系的信息和需求，参见GB/T 19001—2016。

a) 质量管理策划。该活动由以下任务组成：

1) 建立质量管理方针、目标和规程；

注1:GB/T 19004—2020包含了绩效改进指导方针。

注2:政策、目标和规程基于业务战略，并充分考虑顾客满意和风险管理的要求。

2) 定义实施质量管理的职责和权限；

注：质量管理所需的资源通常由不同的组织分配，以独立于项目管理。

3) 定义质量评价准则和方法；

4) 为质量管理提供资源和信息。

b) 评价质量管理。该活动由以下任务组成：

1) 根据定义的准则，收集和分析质量保证评价结果；

2) 评估顾客满意度；

注：GB/T 19014—2019包含监控和测量顾客满意度的指导方针。软件系统的质量也体现在用户满意度上。

3) 定期评审项目质量保证活动是否符合质量管理方针、目标和规程；

注：建立了质量评价准则和方法。质量评价涉及项目规程的符合性和产品在质量特性方面的一致性。

4) 监控过程、产品和服务的质量改进状态。

c) 执行纠正和预防措施。该活动由以下任务组成：

1) 当质量管理目标未实现时，规划纠正措施；

2) 在有足够的风险使得质量管理目标将无法实现时，规划预防措施；

3) 监控纠正和预防措施的完成，并告知利益相关方。

注1:纠正和预防措施的实施在其他相关过程中执行，例如生存周期模型管理过程或项目评估与控制过程。

注2:GB/T 19001—2016的0.3.3和附录A.4描述了消除潜在不合格的预防措施，作为风险思维的一部分。

### 6.2.6 知识管理过程

#### 6.2.6.1 目的

知识管理过程的目的是，建立使组织能够利用机会重新应用现有知识的能力和资产。这涵盖知识、技能和知识资产，包括系统元素。

注：现有知识的重新应用被称为知识重用，并且包括关于或来自软件元素的知识的重用。

#### 6.2.6.2 输出

知识管理过程成功实施后，结果如下：

a) 确定了知识资产应用的分类法；

b) 开发或获得了组织的知识、技能和知识资产；

c) 组织的知识、技能和知识资产可用；

d) 收集和分析了知识管理使用数据。

#### 6.2.6.3 活动和任务

组织应根据知识管理过程适用的组织方针与规程实施下列活动和任务：

a) 知识管理策划。该活动由以下任务组成：

1) 定义知识管理策略；



注1:知识管理策略一般包括:

- i) 确定知识管理领域并识别知识再应用的潜力;
- ii) 针对知识、技能和知识资产在其有用的生存周期内的获取和维护进行策划;
- iii) 对于要收集和维持的知识、技能和知识资产的类型进行特性描述;
- iv) 制定关于知识、技能和知识资产的接受、合格判定和淘汰的准则;
- v) 制定关于知识、技能和知识资产变更的控制规程;
- vi) 制定关于分类或敏感数据和信息的保护、控制和获取的计划、机制和规程;
- vii) 建立关于存储和检索的机制。

注2:知识管理包括在组织内共享的知识和组织外部与指定的利益相关方、需方和业务伙伴共享的知识,需遵守知识产权和保密协议。

2) 确定要管理的知识、技能和知识资产;

3) 确定可以从知识、技能和知识资产的应用中获益的项目。

b) 在全组织范围内共享知识和技能。该活动由以下任务组成:

1) 为在整个组织内获取和共享知识和技能,建立并维护一个分类;

注:分类包括专门的、公共的、领域的知识和技能,以及从其他任务中获得的经验教训。

2) 获取或获得知识和技能;

3) 在整个组织内共享知识和技能。

c) 在全组织范围内共享知识资产。该活动由以下任务组成:

1) 建立组织知识资产的分类法;

注1:分类法包括以下内容:

- i) 领域边界以及其与其他领域关系的定义;
- ii) 获取基本的共同的/不同的特征、能力、概念和功能的领域模型;
- iii) 领域内系统族架构,包括所具有的共同/不同的特征。

注2:有关产品线模型的更多信息参见ISO/IEC 26550。有关体系架构框架、视角、模型种类、视图和模型的要求,参见ISO/IEC/IEEE 42010:2011。

2) 开发或获取知识资产;

注:知识资产包括系统元素或它们的表示形式(例如,可重用的代码库、参考体系架构)的架构或设计元素(例如,架构或设计模式)、过程、准则或其他与领域知识及经验教训相关的技术信息(例如,培训材料)。

3) 在整个组织中共享知识资产。

注:自动搜索能力改善了对知识资产的访问。

d) 管理知识、技能和知识资产。该活动由以下任务组成:

1) 维护知识、技能和知识资产;

2) 监控并记录知识、技能和知识资产的重用;

3) 定期重新评估技术的流通性和知识资产的市场需求。

注:评估组织通过知识管理实践所获得的业务收益。

### 6.3 技术管理过程组

技术管理过程组用于建立和逐步发展计划,执行计划,根据计划评估实际成果和进展,并控制执行直至目标实现。根据计划或未预见的事件的需要,可在生存周期的任何时间,在项目层次结构的任何层次,调用单个技术管理过程组中的过程。应用这些过程时的严格与正式程度,取决于项目的风险和复杂性。

某个技术管理过程的范围是一个项目或其产品,到包括软件产品或所关注的系统的技术管理。

注:实施技术管理过程集合,是为了能够有效地执行软件系统特定的技术过程。技术管理过程组不包括用于项目管理的管理系统或完整全面的过程集,因为这超出了本文件的范围。

技术管理过程组包括:

- a) 项目规划过程；
- b) 项目评估与控制过程；
- c) 决策管理过程；
- d) 风险管理过程；
- e) 配置管理过程；
- f) 信息管理过程；
- g) 测量过程；
- h) 质量保证过程。

项目规划和项目评估与控制过程是所有管理实践的关键。这两个过程建立了管理某一项目或过程的通用方法。本过程组中的其他过程为实现某个专门的管理目标提供了具体的集中任务集。从一个完整的组织到单个生存周期过程及其任务，技术管理过程组的这些过程在任何一项任务的管理中都是显而易见的。本文件中，选取项目作为描述过程的环境。同样的过程也适用于服务的实施。

### 6.3.1 项目规划过程

#### 6.3.1.1 目的

项目规划过程的目的是产生和整合有效且可行的计划。

这个过程确定项目管理和技术活动的范围，识别过程输出、任务和可交付成果，建立任务执行的进度，包括实现准则和完成任务所需的资源。这是一个贯穿整个项目的持续过程，并定期对计划进行修订。

所有其他软件生存周期过程中定义的策略，为项目规划过程提供输入，并在本过程中集成。项目评估与控制过程，用于评估计划是否集成、一致和可行。

#### 6.3.1.2 输出

项目规划过程成功实施后，结果如下：

- a) 定义了目标和计划；
- b) 定义了角色、责任、职责和权限；
- c) 正式请求并承诺了为达成目标所需的资源和服务；
- d) 启动了项目执行的计划。

#### 6.3.1.3 活动和任务

项目应根据与项目规划过程有关的适用的组织方针与规程实施下列活动和任务。

- a) 定义项目。该活动由以下任务组成：

- 1) 识别项目目标和约束条件；

注1:目标与约束包括性能和其他质量因素、成本、时间、顾客和用户满意度。每个目标的识别，详细到允许适当过程和活动的选择、剪裁和实施。

注2:ISO/IEC 15026(所有部分)系统与软件保证、ISO/IEC 27001信息安全管理系统和ISO/IEC 27036供方关系的信息安全，提供了关于与保证和信息安全有关的目标和限制的进一步指导。

- 2) 根据已确定的协议定义项目范围；

注：包括为了满足业务判定准则和成功完成项目所需的相关活动。一个项目可以对完整软件系统生存周期中的一个或多个阶段负责。项目规划包括为维护项目计划、执行评估和控制项目而定义适当的行动。

- 3) 使用组织中已定义的若干生存周期模型，定义和维护一个由若干阶段构成的生存周期模型；

注：ISO/IEC/IEEE 24748-1提供了关于生存周期阶段的详细信息，以及一个恰当的生存周期模型的定义。

它定义了一个典型系统生存周期阶段的通用集合，包括概念、开发、生产、使用、支持和退役。它还确定了一个通用的软件生存周期阶段的典型集合，包括需求确定、概念探索与定义、论证与评价、工程/开发、生产/制造、部署/销售、运营、维护与支持，以及退役。

4) 基于可交付产品或软件系统不断演进的架构，建立一个工作分解结构(WBS);

注1:软件系统架构的每个元素，以及相应过程和活动，对其描述的详细程度与已识别的风险相一致。工作分解结构中的相关任务按性能分组。项目任务识别正在开发或产生的工作项。项目管理协会(PMI)关于工作分解结构(WBS)的实践管理标准包含了WBS的更多细节。

注2:对于使用敏捷或迭代方法的项目，从用户的角度看，一个WBS元素可对应于在迭代期间产生的主要特征。

5) 定义和维护将应用于项目的过程。

注1:这些过程基于组织已定义的过程(参见生存周期模型管理过程)。附录A包含了可用于针对项目具体需要的剪裁信息。过程的定义，包括进出准则、输入、过程的顺序约束(前驱/后继关系)、过程的并发性需求(哪些过程和任务要与其他过程区域的任务或活动同时工作)、有效性度量/性能属性度量以及范围和成本参数(用于至关重要的成本估算)。

注2:确定与其他项目或组织单元的接口，通过特定项目包管理过程处理。

b) 规划项目和技术管理。该活动由以下任务组成：

1) 根据管理和技术目标，以及工作估计，明确并维护项目进度计划；

注：这包括下列事项定义——活动的持续时间、关系、依赖项和顺序，成果里程碑，使用的资源以及为了实现项目的及时完成所必要的评审和为风险管理保留的进度计划。

2) 定义生存周期阶段决策门的完成准则、交付日期和对外部输入或输出的主要依赖关系；

注：内部评审之间的时间间隔，是根据组织关于诸如业务与系统关键性、进度和技术风险等问题的方针来确定的。

3) 确定成本并制定预算；

注：预算成本基于进度计划、软件规模和复杂性估计、人力估计、基础设施成本、采购项、获得的服务和使能系统估计，以及风险管理预算准备金。

4) 定义角色、责任、职责和权限；

注：包括定义项目组织、人员获得和员工技能的开发。权限包括(视情况而定)负有法律责任的角色和个人，例如，设计授权、安全授权以及适用的认证或认可责任。

5) 定义所需的基础设施和服务；

注：这包括定义所需的容量、可用性和对项目任务的分配。基础设施包括设施、服务、工具、通信和信息技术资产。还制定了针对每个生存周期阶段对使能系统和服务的需求。

6) 规划从项目外部供应的物料、使能系统和服务的获取；

注1:必要时，这包括招标、供方选择、验收、合同管理和合同终止。协定过程用于已制定计划的获取。

注2:ISO/IEC 27036(所有部分)供方关系的信息安全，为基础设施和服务的获取提供指导。

7) 为项目和技术的管理和执行，包括评审，生成并沟通一项计划。

注1:软件系统的技术规划通常体现在系统管理计划(SEMP)或软件工程管理计划或软件开发计划(SDP)中。ISO/IEC/IEEE 24748-5 提供了关于软件工程技术管理规划的更多细节，并包括了一个SDP的注释大纲。项目的规划通常体现在项目管理计划中。ISO/IEC/IEEE 16326 提供了关于项目规划的更多细节。

注2:其他过程的每项策略性活动和任务为项目规划过程提供输入，并在项目规划过程中集成。项目评估与控制过程用于确保计划的完整、一致和可行。

c) 激活项目。该活动由以下任务组成：

1) 获得开始项目的许可；

注：开始的许可(授权继续进行)通过特定项目包管理过程获得。

2) 针对执行项目所需的资源，提交请求并获得承诺；

注：这包括对使能系统或服务的访问。

3) 执行项目计划。

## 6.3.2 项目评估与控制过程

### 6.3.2.1 目的

项目评估与控制过程的目的是评估计划是否一致和可行；确定项目的状态、技术与过程绩效；指导计划的执行，以确保项目绩效符合计划和进度要求，在计划的预算之内，满足技术目标。

这个过程，定期以及在重大事件节点，依据需求、计划和总体业务目标，评价进展和完成情况。当发现显著差异时，为管理行动提供信息。适当时，这个过程也包括对项目的活动和任务进行调整，以纠正来自其他技术管理或技术过程已经识别的偏差和变化。必要时调整可包括重新规划。

### 6.3.2.2 输出

项目评估与控制过程成功实施后，结果如下：

- a) 形成了可用的项目绩效测度或评估结果；
- b) 评估了角色、责任、职责和权限的充分性；
- c) 评估了资源的充分性；
- d) 执行了技术进展评审；
- e) 研究与分析了项目绩效与计划的偏差；
- f) 将项目状态告知了受影响的利益相关方；
- g) 当项目成果不满足目标时，定义和指导了纠正措施；
- h) 必要时，启动了项目重新规划；
- i) 授权了从计划的一个里程碑或事件节点到下一个要进行(或不进行)的项目活动；
- j) 实现了项目目标。

### 6.3.2.3 活动和任务

项目应根据与项目评估与控制过程有关的适用的组织方针与规程实施下列活动和任务。

- a) 为项目评估与控制制定计划。该活动由以下任务组成：

- 1) 定义项目评估与控制策略。

注：本策略确定了预期的项目评估与控制活动，包括计划的评估方法和时间表、必要的管理和技术评审。

- b) 评估项目。该活动由以下任务组成：

- 1) 评估项目目标和计划与项目环境的一致性；

- 2) 按照目标，对管理和技术计划进行评估，以确定其充分性和可行性；

- 3) 按照合适的计划对项目和管理进行评估，以确定实际的和预计的成本、进度和绩效方面的差异；

- 4) 评估角色、责任、职责和权限的充分性；

注：这包括评估人员能力是否足够胜任项目角色并完成项目任务。尽可能使用目标测度，例如，资源使用效率和项目成果。

- 5) 评估资源的充分性和可用性；

注：资源包括基础设施、人员、资金、时间或其他相关项。这个任务包括评价现有过程和基础设施资源的重用，以及确认组织内部的承诺是否得到了满足。

- 6) 使用测量的成果和里程碑完成情况来评估进展；

注：包括收集和评估劳动力、材料、服务成本和技术性能的数据，以及关于目标的其他技术数据，如负载能力。这些都是与成果测度进行对比。这包括进行有效性评估，以确定演变中的软件系统对需求的满足程度。当需要时，它还包括使能系统交付服务的就绪度。

7) 进行必要的管理和技术评审、审核和检查；

注：这些评审、审核和检查可以是正式的或非正式的，执行的目的是确定转入生存周期的下一阶段或项目里程碑是否已经就绪，以帮助确保项目和技术目标正在得到满足，或者从利益相关方获得反馈。

8) 监控关键过程和新技术；

注：包括识别和评估技术成熟度与新技术引入的可行性。技术成熟度是一项运行使用技术的就绪程度，并且通常按照从低(仅作为概念存在)到高(在运行使用中得到验证)的等级进行测量。

9) 分析测量结果并提出建议；

注：对测量结果进行分析，以便从包括潜在关注在内的计划值中识别偏差、变化或不良趋势，并对纠正或预防措施提出适当建议。这包括，在适当情况下，对表明趋势的测度进行统计分析，例如，表示输出质量的故障密度，表示过程可重复性的测量参数分布。

10) 记录与提供来自评估任务的状态和发现；

注：这些通常在协定、方针和规程中指定。

11) 监控项目内的过程执行。

注：包括对过程测度的分析和关于项目目标趋势的评审。识别出的任何改进措施，都可通过质量保证过程质量管理过程或生存周期模型管理过程来处理。

c) 控制项目。该活动由以下任务组成：

1) 发起必要的行动以解决已识别的问题；

注1:这项任务在项目或技术成果没有达到计划目标时发生。这包括为预防、纠正和解决问题而采取的行动。当发现人员、工具和基础设施资产不足或不可用时，或当项目或技术成果超出目标或计划时，行动通常需要重新规划或重新分配人员、工具和基础设施资产。他们通常影响成本、进度、技术范围或定义。行动有时需要对生存周期过程的实施和执行进行的变更。

注2:记录和评审活动，以确认其充分性和及时性。

2) 启动必要的项目重新规划；

注1:项目重新规划是在项目目标或约束已经改变，或者计划假设被证明无效时发起的。

注2:需要改变供需双方之间协定的任何变更都会调用获取过程和供应过程。

3) 当因某个需方或供方要求的影响而发生的关于成本、时间或质量的合同变更时，启动变更行动；

注：这包括考虑修改供应的条款和条件，或启动新的供方选择，这将调用获取过程和供应过程。

4) 若合理，则授权项目进入下一个里程碑或事件。

注：项目评估与控制过程用于达成完成里程碑的协定。

### 6.3.3 决策管理过程

#### 6.3.3.1 目的

决策管理过程的目的是提供一个结构化的分析框架，为生存周期中任意时刻的决策，客观地识别、描述和评估一组备选方案，并从中选择最有益的行动方案。

这个过程用于解决技术或项目问题，并响应在软件生存周期中遇到的决策请求，以确定为该情况提供首选结果的备选方案。最常用于决策管理的方法是权衡研究和工程分析。每个备选方案都根据判定准则(例如，成本影响、进度影响、计划约束、监管影响、技术性能特性、关键质量特性和风险)进行评估。通过适当的选择模型，对比较结果进行排序，以此确定最优方案。关键研究数据(例如，假设和决策依据)，通常进行维护，为决策者提供信息并支持未来的决策制定。

注：当有必要对某一准则的参数进行详细的评估时，可采用系统分析过程执行评估。

#### 6.3.3.2 输出

决策管理过程成功实施后，结果如下：

- a) 识别了需要进行替代分析的决策项；
- b) 识别并评估了备选行动方案；
- c) 选择了首选的行动方案；
- d) 形成了决议、决策依据和假设。

### 6.3.3.3 活动和任务

项目应根据与决策管理过程有关的适用的组织方针与规程实施下列活动和任务。

- a) 为决策做好准备。该活动由以下任务组成：

- 1) 定义决策管理策略；

注：决策管理策略包括对角色、责任、职责和权限的识别。该策略考虑获取信息输入和及时返回决策的需要，包括确定决策类别和优先级方案。决策通常是由有效性评估、技术权衡、需要解决的问题、作为对超出可接受阈值的风险的响应所需的行动，或项目进展到下一个生存周期阶段的新的机会或批准引起的。组织或项目指南决定了应用于决策分析的严格程度和正式程度。

- 2) 确定决策的环境和需要；

注：对问题或机会，以及对其结果作出决议的备选行动方案，进行记录、分类和报告。

- 3) 让相关的利益相关方参与决策，以吸取经验和知识。

注：确定分析和决策所需的专业知识技能是很好的实践。

- b) 分析决策信息。该活动由以下任务组成：

- 1) 选择并声明每个决策的决策管理策略；

注：确定解决这些问题或机会所需的严格程度，评价备选方案所需的数据和系统分析。确定达成决策的时间框架。

- 2) 确定期望的结果和可测量的选取准则；

注：所有可量化标准的期望值和超出则属性不符合要求的阈值，以及标准的权重因子都是确定的。

- 3) 确定权衡空间和备选方案；

注：如果存在大量的备选方案，则进行定性筛选，以减少备选方案至可控的数量，以便进行进一步的详细系统分析。这种筛选通常基于对风险、成本、进度和监管影响等因素的定性评估。

- 4) 依据准则评价每一种备选方案。

注：必要时，系统分析过程用于量化每一个待评价权衡备选方案的具体准则。这包括新的设计参数、不同的架构特征以及关键质量特性的范围。系统分析过程评估参数变化的范围，以获得每一项已评价的交易备选方案的敏感性分析。这些结果用于评估各种交易备选方案的可行性。

- c) 做出和管理决策。该活动由以下任务组成：

- 1) 为每个决策确定首选方案；

注：用选择标准来定量评价备选方案。选定的备选方案通常提供一个确定的决策的优化或改进。

- 2) 记录解决方案、决策依据和假设；

- 3) 记录、跟踪、评估和报告决策。

注1:这包括根据协定或组织规程的规定，以允许审计和从经验中学习的方式，对问题和机会及其处置进行的记录。

注2:这使组织能够确认问题已得到有效解决，不利的趋势已被逆转，而且优势机会已被抓住。

### 6.3.4 风险管理过程

#### 6.3.4.1 目的

风险管理过程的目的在于持续识别、分析、处理和监控风险。

风险管理过程是一个持续的过程，系统地解决系统产品或服务生存周期中的风险。它可以应用于与系统的获取、开发、维护或运行相关的风险。

注：GB/T 23694-2013中，风险被定义为“对目标不确定性的影响”。这有一个相关的注释，“影响是偏离预期的积极的和/或消极的”。积极的风险也被认为是一个机会，并可在风险管理过程中处理。

#### 6.3.4.2 输出

风险管理过程成功实施后，结果如下：

- a) 识别了风险；
- b) 分析了风险；
- c) 确定了风险处理的选项，按优先级进行排序，并做出选择；
- d) 实施了恰当的处理；
- e) 评价了风险，以评估状态的变化和处理的进展。

#### 6.3.4.3 活动和任务

项目应根据与风险管理过程有关的适用组织方针与规程实施下列活动和任务。

注：ISO/IEC/IEEE 16085提供了一组更详细的风险管理活动和任务。这个风险管理过程与ISO 31000:2009、GB/T 23694-2013保持一致。在GB/T 19001—2016的6.1中描述了关于风险和机遇的规划。

- a) 规划风险管理。该活动由以下任务组成：

- 1) 定义风险管理策略；

注：包括了所有供应链供方的风险管理过程，并说明来自所有供方风险将如何传递到下一个级别，以便纳入项目风险过程中。

- 2) 定义并记录风险管理过程的周境。

注1:包括对利益相关方的视角、风险类别的描述，以及技术和管理方面的目标、假设和约束的描述(或许是参考)。风险类别包括软件系统的相关技术领域，并有助于产品生存周期中的风险识别。参见ISO 31000，这一步骤的目标是根据可能产生、加强、预防、降低、加速或延迟实现目标的那些事件来生成一份全面的风险清单。

注2:机会也是一种风险，它为软件系统或项目提供了潜在的益处。所追寻的每一个机会都有与预期收益相关的风险。这包括与不追寻机会的相关风险，以及未实现机会效应的风险。

- b) 管理风险概况。该活动由以下任务组成：

- 1) 定义并记录风险阈值，以及在什么条件下风险级别是可接受的；

- 2) 建立并维护风险概况；

注：风险概况记录：风险管理周境；每个风险状态的记录，包括其发生的可能性、后果和风险阈值；根据利益相关方提供的风险准则确定的每个风险的优先级；以及风险措施请求及其处理状态。当单个风险的状态发生改变时，要更新风险概况。风险概况中的优先级用于确定处理风险所需的资源应用。

- 3) 基于利益相关方的需要，定期向其提供相关的风险概况。

- c) 分析风险。该活动由以下任务组成：

- 1) 识别在风险管理周境中所描述的类别的风险；

注：风险通常通过各种分析来识别，如，安全、可靠、信息安全和性能分析；技术、架构和成熟度评估以及权衡研究。这些风险通常在生存周期的早期确定，并持续到软件系统的利用、支持和退役。此外，风险还常常通过持续演进的软件系统的测量分析来确定。

- 2) 评估每个已识别风险发生的可能性和后果；

注：某个风险的后果通常涉及技术、进度、成本或质量的影响。

- 3) 根据风险阈值评估每个风险；

- 4) 对于每个超过其风险阈值的风险，定义并记录推荐的处理策略和措施。

注：风险处理策略包括但不限于消除风险、降低风险发生的可能性或后果的严重程度，或接受风险。风险处理还包括承担或增加风险，以寻求机会。测度提供了关于处理方案有效性的信息。

- d) 处理风险。该活动由以下任务组成：



- 1) 识别推荐的风险处理备选方案；
  - 2) 为了让利益相关方决定采取行动使风险可接受，执行风险处理方案；
  - 3) 当利益相关方接受超出其风险阈值的风险时，将该风险视为高优先级，并持续监控它，以确定是否有必要在未来采取风险处理措施，或其优先级是否已发生变化；
  - 4) 一旦选择了风险处理，便要和管理行动进行协调。
- 注：参考项目评估与控制过程。
- e) 监控风险。该活动由以下任务组成：
- 1) 持续监控风险和风险管理环境的变化，并在风险状态已经发生改变时评价风险；
  - 2) 执行和监控以评价风险处理有效性的措施；
  - 3) 在整个生存周期中持续监控新的风险和来源的出现。

### 6.3.5 配置管理过程

#### 6.3.5.1 目的

配置管理的目的是在生存周期中管理和控制系统元素和配置。配置管理(CM) 还管理产品及其相关配置定义之间的一致性。

软件配置管理(SCM) 应用于软件系统及其接口。接口管理的目的是与接口合作方就通过软件系统和服务之间的通信进行数据的交换达成一致。附录E(见E.5) 提供了接口管理过程视图的示例。

软件配置通过新版本的受控发布进行更改。发布的目的是授权并实现特定目的的软件特征、功能或系统的可用性，而对有或没有用户子集不作限制。

#### 6.3.5.2 输出

配置管理过程成功实施后，结果如下：

- a) 识别和管理了需要配置管理项；
- b) 建立了配置基线；
- c) 对配置管理项的变更进行了控制；
- d) 配置状态信息可用；
- e) 完成了所需的配置审核；
- f) 控制和批准了系统的发布和交付。

#### 6.3.5.3 活动和任务

项目应按照与配置管理过程有关的适用的组织方针与规程实施下列活动和任务。

注：ISO/IEC/IEEE 19770提供了IT 资产管理系统的规程和需求。

- a) 规划配置管理。该活动由以下任务组成。

- 1) 定义配置管理策略，包括以下方法：

- i) CM 的治理，包括角色、责任、职责和权限，以及配置控制(变更控制)委员会的使用；
- ii) 在批准配置基线以及定期和紧急变更请求时考虑风险和影响等级；

注：使用已批准的规程或正在开发的经单元测试的软件元素的签入和签出的定期计划的变更通常是自动执行的，或作为例行程序每天进行评审和批准。相比之下，对项目成本和进度有较大影响的软件系统设计的重大变更可能涉及广泛的分析、与供方的协商、利益相关方的评审和组织最高级别的批准。

- iii) 在软件系统的生存周期，或者在协定或项目的范围内(视情况而定)，跨需方、供方和供应链组织的集合协调CM；
- iv) 控制对配置项的访问、更改和处置；



- v) 需要建立的必要基线，包括启动配置控制和维护演进配置基线的准则或事件；
- vi) 控制软件许可、数据权限和其他知识产权资产；
- vii) 软件版本和发布的频率、优先级和内容；
- viii) 用于确认配置定义信息的连续完整性和信息安全的审核策略与职责；
- ix) 变更管理，包括为利益相关方特别是用户做好运行的软件系统和服务的变更准备。

注1:关于配置管理活动的附加指导可参见ISO 10007、IEEE Std 828和 SAE ANSI/EIA-649-B。

注2:SWEBOK(软件工程知识体系的指南)提供了关于SCM的详细讨论。这个知识领域涉及系统周境中的SCM、SCM项目和过程规划、SCM计划和大纲、工具选择、分包商控制、监视和其他审计、软件配置项和关系、软件库以及配置管理过程活动。

注3:SCM 策略通常记录在计划中，例如，配置管理计划，或有时是在项目的 SEMP、SDP 或项目管理计划(PMP)中。配置管理的策略规划是通过项目规划过程进行协调的。在建立点，以建立基线并进行审计时，CM规划与软件生存周期保持一致。重复的SCM活动的频率与技术过程和阶段的迭代保持一致。SCM规划通常包括决定何时评审配置管理规划，什么条件需要更新CM 计划，以及谁有权更改配置控制中的CM 计划和项目。

## 2) 定义配置项、CM 工作产品和记录的存储、归档和检索规程。

注：软件系统配置项(如源代码和可执行软件)的存储位置和条件是按照指定的完整性、信息安全和安全性级别建立的。

## b) 执行配置标识。该活动由以下任务组成：

### 1) 选择要唯一标识为受配置控制的配置项的软件系统元素；

示例：软件系统中受配置控制的配置项通常包括系统/软件需求规格说明、接口规范；产品和系统元素(例如，软件对象、硬件和服务)在开发过程中，为每两个阶段之间的过渡而建立的以及为运行使用而发布的基线配置或软件版本；不同平台或版本的源代码或可执行软件的主(黄金)副本；运行使用中的特定配置；信息项，如协议、体系架构模型、服务描述和操作过程；以及使能系统中的项目。

注1:唯一标识可应用于软件组件、版本或单独授权的副本。标识符符合相关标准和产品部门惯例，使得配置控制下的项目可明确地追溯到其供方及其规格说明或同等的记录描述。信息项通常与其他配置项分开进行标识和管理。

注2:当一个以上的开发人员或维护程序员在同一软件功能上工作时，配置标识符有助于实现可追溯性，从而可以成功地重新组装和测试各种代码分支。

注3:ISO/IEC 19770(多部分)标准提供了一个用于跟踪软件许可的IT 资产管理系统。

### 2) 识别配置项的属性；

注1:属性指项目状态，或对管理或维护软件系统有用的物理或逻辑特征。硬件和软件配置项的适当属性可能不同。

注2:配置属性和标识符可以反映软件系统的分解，以便在需要控制变更的级别上跟踪配置项。

示例：来自外部供方的软件可以通过其许可和维护协定进行跟踪，这可能涉及跟踪到使用它的系统的位置、数量或大小，或者允许的并发用户数量。软件版本可以追溯到他们实现的利益相关方需求。

### 3) 定义全生存周期的基线；

注1:在制定时间或在确定环境下，基线捕获软件系统元素不断演进的配置状态。基线的内容是通过技术过程开发的，但通过配置管理过程在某个时间点及时正式化。基线构成了后续变更的基础。选定的基线通常在需方和供方之间正式化，这取决于行业惯例以及需方在配置管理过程中的参与。通常有三种主要类型的系统级基线：功能基线、分配基线和产品基线。这些因领域或本地策略而不同。

注2:软件系统开发通常需要建立多个开发基线，以在生存周期的关键点上满足不断演进的软件配置需要，例如，允许在软件设计、原型、集成和测试发布期间同时控制软件版本。这可能涉及分布式配置管理职责和对档案的访问限制，例如，软件开发或测试库以及主配置支持库。

### 4) 获取需方和供方协定以建立基线。

注：项目评估和控制过程用于达成协定。当软件被正在开发用于商业或内部使用时，需方或项目发起人可授权批准基线。

## c) 执行配置变更管理。该活动由以下任务组成：

注：配置变更管理建立了规程和方法，用于在基线建立后管理基线变更。这有时被称为配置控制。术语“变更管理”也用于管理组织规程和业务工作流的变更。

## 1) 识别并记录变更请求和差异请求；

注：对差异的请求通常被称为偏差、放弃或让步。

## 2) 协调、评价和处理变更请求和差异请求；

注1:评价通常包括对软件和互操作系统影响的基本原理和需要分析，并考虑风险和机会、质量、用户、计划进度和成本。做出是否执行或拒绝变更请求的决定。

注2:变更请求和差异请求通常在配置控制委员会(CCB)的正式控制之下。

## 3) 跟踪和管理已批准的基线变更、变更请求和差异请求。

注1:这项任务涉及优先级排序、跟踪、计划和关闭变更。然后通过技术过程组进行变更。这些变更通过验证或确认过程进行验证或确认，以帮助确保已批准的变更已得到正确应用。

注2:变更和理由通常在批准和完成时进行记录。

## d) 执行发布控制。该活动由以下任务组成：

## 1) 标识并记录发布请求，识别发布中的软件系统元素；

注：生存周期模型有助于确定迭代或增量软件发布的频率。集成过程用于选择和配置发布包、软件版本、更新或补丁。这些变更通过验证或确认过程进行验证或确认。变更是通过技术过程组，特别是交付(过程)实现的。

**示例：**对于软件测试、软件或系统鉴定或其他正式测试、或试用(beta)或运行使用的发布。

## 2) 批准软件系统的发布和交付；

注1:发布通常涉及优先级排序、跟踪、计划和关闭变更。对运行使用发布的批准可包括接受经过验证和确认的变更。发布的批准准则通常包括假若发布不成功时的回滚计划或应急计划。

注2:对于软件系统，自动化版本控制工具可以帮助确保只有正确的源代码版本被访问、更新、测试、记录，以供适当的人员批准变更，并发布。

## 3) 跟踪和管理软件系统发布到指定环境或软件交付的分发。

注：在受控环境中，可以在系统或项目的生存周期中维护已发布软件版本的主副本或增量变更副本。软件供方通常跟踪交付给需方的许可软件副本，以便提供约定的软件维护。软件系统的发布按照协定以及涉及的组织方针进行存储和分发。

## e) 执行配置状态纪实。该活动由以下任务组成：

## 1) 开发并维护软件系统元素、基线和版本的CM 状态信息；

注1:配置状态核算提供了在整个产品生存周期中关于系统元素做出决策所需的受控产品的状态数据。如，软件状态可以包括贯穿软件功能生存周期各个阶段的过去、当前和计划的进展，以及软件元素的验证和确认活动的完成。配置状态信息允许对其他配置状态向前或向后的跟踪。配置状态记录通过软件或项目生存周期进行维护，然后根据协定、相关法规或组织惯例进行存档。

注2:管理当前配置状态和先前配置状态的记录、检索和合并，以确认信息的正确、及时、完整和信息安全。执行审计以验证基线是否符合架构视图、接口控制文档、软件许可协议和其他协定要求。

## 2) 捕获、存储和报告配置管理数据。

## f) 执行配置评价。该活动由以下任务组成：

## 1) 确定CM 审核的需要，并安排事件的时间表；

2) 通过将需求、约束、弃权(差异)与正式验证活动的结果进行比较，验证产品配置是否满足配置要求，这可能涉及抽样方法；

## 3) 监控已批准配置变更的合并；

## 4) 评估软件系统是否满足为基线确定的功能和性能能力；

注：这有时称为功能配置审核(FCA),它确保产品配置满足规定的要求。

## 5) 评估运行软件系统元素是否符合已批准的配置信息；

注：这有时称为物理配置审核(PCA)。对于软件项，PCA的准则可包括是否根据软件许可或协定在指定的系统上安装了规定的配置项。

6 ) 记录CM 审核结果和处置措施项。

6.3.6 信息管理过程

6.3.6.1 目的

信息管理过程的目的是向指定的利益相关方生成、获取、确认、转换、保留、检索、传播和处置信息。

信息管理计划、执行并控制向指定的利益相关方提供明确、完整、可验证、一致、可修改、可追溯和可呈现的信息。信息包括技术、项目、组织、协定和用户信息。信息通常来自组织、系统、过程或项目的数据记录。

注：管理信息具有这些质量特性：明确、完整、可验证、一致、可修改、可追溯和可呈现。

6.3.6.2 输出

信息管理过程成功实施后，结果如下：

- a) 识别了要管理的信息；
- b) 定义了信息的表达形式；
- c) 获取、开发、转换、存储、确认、呈现和处理了信息；
- d) 确定了信息的状态；
- e) 信息对指定的利益相关方是可用的。

6.3.6.3 活动和任务

项目应按照与信息管理过程有关的适用的组织方针与规程实施下列活动和任务。

注：ISO/IEC/IEEE 15289总结了对生存周期过程信息项(文档)内容的要求，并为其开发提供了指导。

a) 为信息管理做准备。该活动由以下任务组成：

1) 确定信息管理策略；

注：关于同一主题的信息可以在生存周期的不同点和针对不同的受众以不同的方式进行开发。

2) 定义要管理的信息项；

注：这包括将在软件生存周期中管理的信息，并可能在以后的一段时间内进行维护。这是根据组织政策、协定或法规来完成的。

3) 指定信息管理的权限和职责；

注：注意信息和数据法规、信息安全和隐私，如，所有权、协定限制、数据访问权和数据所有权、知识产权和专利。在应用限制或约束的情况下，相应地标识信息。了解这些信息项的工作人员要知晓其义务和责任。

4) 定义信息项的内容、格式和结构；

注：信息源于并终止于多种形式(如，视听的、文本的、图形的、数字的)和媒介(如，电子的、印刷的、磁性的、光学的)。要考虑组织约束，如，基础设施、组织间通信和分布式项目运作。根据政策、协议和法规约束来使用相关信息项目的标准和约定。

5) 定义信息维护动作。

注：信息维护包括对存储信息的完整性、有效性和可用性的状态评审。它还包括在必要时复制或转换到替代媒介的任何需要，在技术变化时保留基础设施以便读取存档介质，或将存档介质迁移到较新的技术。

b) 执行信息管理。该活动由以下任务组成：

1) 获取、开发或转换所识别的信息项；

注：这包括从适当的来源(产生于任何生存周期过程)收集数据、信息或信息项，以及编写、说明或将其转换为利益相关方的可用信息。它包括根据信息标准评审、确认和编辑信息。

2) 维护信息项及其存储记录，记录信息的状态；

注1:信息项根据其完整性、信息安全和隐私要求进行维护。维护信息项的状态(如,版本描述、发行日期或有效日期、分发记录、安全分类)。可读信息以易于检索的方式存储和保留。

注2:用于转换信息的源数据和工具以及生成的文档,按照配置管理过程进行配置控制。ISO/IEC/IEEE 26531提供了对生存周期信息和文档有用的内容管理系统的要求。

3) 向指定的利益相关方发布、分发或提供对信息和信息项的访问;

注:按商定的时间表或确定的情况所要求的,以适当的形式向指定的利益相关方提供信息。根据需要,信息项包括用于认证、认可、许可或评估等级的文件。

4) 存档指定信息;

注:归档是根据审核、知识保留和项目关闭目的进行的。根据规定的存储和检索期,以及组织方针、协定和法规来选择信息的媒介、位置和保护。要安排到位,以在项目结束后保留必要的信息项。

5) 处理不需要的、无效的或未验证的信息。

注:这是根据组织方针、信息安全和隐私要求完成的。

### 6.3.7 测量过程

#### 6.3.7.1 目的

测量过程的目的是收集、分析和报告客观的数据和信息,以支持有效的管理,并论证产品、服务和过程的质量。

注:测量具有这些质量特点:可验证的、有意义的、可操作的、及时的和具有成本效益的。

#### 6.3.7.2 输出

测量过程成功实施后,结果如下:

- a) 确定了信息需要;
- b) 基于信息的需要,确定或制定了一组合适的测度;
- c) 收集、验证和存储了所需的数据;
- d) 对数据进行了分析,并对结果进行了解释;
- e) 信息项提供了支持决策的客观信息。

#### 6.3.7.3 活动和任务

项目应按照与测量过程有关的适用的组织方针与规程,实施下列活动和任务。

注1:ISO/IEC 15939提供了一组更为详细的测量活动和任务,与本文件中的活动和任务相一致。

注2:GB/T 19001—2016规定了测量和监控的质量管理体系要求。

a) 准备测量。该活动由以下任务组成:

- 1) 定义测量策略;
- 2) 描述与测量相关的组织特性,如,业务和技术目标;
- 3) 确定信息需要并确定其优先级;

注:信息需要是基于组织的业务目标、项目目标、已识别的风险以及与项目决策相关的其他项目。测量可以与项目、过程、产品或决策相关。

4) 选择并明确满足信息需要的测度;

注:所定义的测度是可验证的且具有成本效益的。

5) 定义数据收集、分析、访问和报告规程;

6) 定义评价信息项和测量过程的准则;

7) 确定并规划将要使用的必要的使能系统和服务。

b) 执行测量。该活动由以下任务组成:

- 1) 将数据生成、收集、分析和报告的手动或自动化规程集成到相关过程中;

注：这项任务可能涉及为了完成规程集成而对其他生存周期过程产生的变更影响。

- 2) 收集、存储和验证数据；
- 3) 分析数据并开发信息项；
- 4) 记录结果并通知测量用户。

注：测量分析结果以及时、可用的方式报告给相关的利益相关方，以支持决策制定，并协助纠正措施、风险管理 and 改进。结果报告给决策过程参与者、技术和管理评审参与者，以及产品和过程改进过程所有者。

6.3.8 质量保证过程

6.3.8.1 目的

质量保证过程的目的是帮助确保组织的质量管理过程有效应用于项目。

质量保证侧重于提供满足质量要求的信心。对项目生存周期过程和产出进行主动分析，以确保正在生产的产品达到预期的质量，并遵循组织和项目的方针与规程。

6.3.8.2 输出

质量保证过程成功实施后，结果如下：

- a) 确定并实施了项目质量保证规程；
- b) 确定了质量保证评价的准则和方法；
- c) 根据质量管理方针、规程和要求，对项目的产品、服务和过程执行了评价；
- d) 向利益相关方提供了评价结果；
- e) 偶发事件得到解决；
- f) 处理了优先考虑的问题。

注：输出a)~d)与质量管理过程活动和任务的输出保持一致。

6.3.8.3 活动和任务

项目应按照与质量保证过程有关的适用的组织方针与规程实施下列活动和任务。

注：IEEE Std 730—2014(软件质量保证过程)提供了额外的细节。

- a) 为质量保证做准备。该活动由以下任务组成：
  - 1) 定义质量保证策略。该策略与组织的质量管理方针与目标相一致，并包括：
    - i) 将质量保证资源应用于对交付产品和服务的质量影响最大的过程和任务的优先事项；
    - ii) 定义角色、职责、责任和权限；
    - iii) 过程、产品和服务的评价准则和方法，包括产品或服务验收的准则；
    - iv) 适用于每个供方(包括分包商)的活动；
    - v) 针对产品或服务所需的验证、确认、监控、测量、评审、检查、审核和测试活动；
    - vi) 问题的解决以及过程和产品的改进活动。

注：在软件项目中，对产品质量有重大影响的活动和任务包括就新的和变更的需求、同行评审和单元测试的实施、问题报告和用户反馈的分析达成一致；就项目里程碑评审中分配的纠正措施的完成情况，以及缺陷的根本原因分析进行验证。

- 2) 建立独立于其他生存周期过程的质量保证。

注：质量保证的资源通常由不同的组织分配，以独立于项目管理。

- b) 执行产品或服务评价。该活动由以下任务组成：
  - 1) 评价产品和服务是否符合既定的准则、合同、标准和规章；

注：这项任务包括验证产品或服务验收准则是否反映在验证和确认活动中。派生的系统/软件质量要求通常

在需求定义过程中与质量特性相关联。GB/T 25000.10和GB/T 25000.30提供了关于系统/软件质量特性的附加信息。

2) 监视生存周期过程的产出是否得到验证和确认，以确定是否符合规定的要求。

c) 执行过程评价。该活动由以下任务组成：

1) 评价项目生存周期过程的符合性；

2) 评价支持或自动化实施过程符合性的工具和环境；

3) 评价供方过程对于过程要求的符合性。

注：考虑诸如协同软件开发环境，供方所要求提供的过程测度，或供方所要求的使用的风险过程等项目。这包括对整个供应链的过程实施的监督评审。

d) 管理QA 记录和报告。该活动由以下任务组成：

1) 创建与质量保证活动相关的记录和报告；

注：使用信息管理过程，根据组织、规章和项目需求创建记录和报告。

2) 维护、存储和分发记录和报告；

3) 识别与产品、服务和过程评价相关的偶发事件和问题。

注：这包括获取经验教训。确定了解决问题的责任。

e) 处理偶发事件和问题。该活动由以下任务组成：

注1:在质量管理术语中，问题通常被描述为“不符合项”，如果不加以处理，它们可能导致项目无法满足其需求。

注2:有关问题类别和优先级分类的更多信息和示例，参见 ISO/IEC/IEEE 24748-1。

1) 对事件进行记录、分析并分类；

2) 识别与已知错误或问题相关的选定事件；

3) 对问题进行记录、分析并分类；

注：分析结果包括潜在的处理方案。

4) 在可行的情况下，确定问题的根本原因和解决方法；

5) 确定问题处理（问题解决）的优先级并跟踪纠正措施；

注：项目评估和控制过程启动后，在技术过程中完成实施。问题升级的组织规程有助于将资源集中在滞后的问题解决上。

6) 分析偶发事件和问题的趋势；

7) 确定过程和产品的改进，这可防止未来的偶发事件和问题；

注：风险管理过程用于处理风险和机会。生存周期模型管理过程用于改进组织的过程。

8) 向指定的利益相关方通报偶发事件和问题的状态；

9) 跟踪偶发事件和问题直至关闭。

## 6.4 技术过程组

技术过程组用于定义软件系统需求，将需求转换为有效产品，必要时可保证产品再生产的一致性，使用产品来提供所需服务，保证提供服务的持续性，在产品从服务中退役时进行处置。

技术过程组中定义了能够支持组织和项目职能的活动，以优化利益并减少技术决策和行动带来的风险。这些活动使得软件系统和服务具有及时性、可用性、成本有效性、功能性、可靠性、维护性、生产率、易用性，以及供需组织所需的其他质量特性。它们同时也可以使产品和服务符合社会的预期或法定要求，包括健康、安全、信息安全和环境因素。

技术过程组包括：

a) 业务或使命分析过程；

b) 利益相关方需要和需求定义过程；

c) 系统/软件需求定义过程；

d) 架构定义过程；

- e) 设计定义过程；
- f) 系统分析过程；
- g) 实现过程；
- h) 集成过程；
- i) 验证过程；
- j) 移交过程；
- k) 确认过程；
- l) 运行过程；
- m) 维护过程；
- n) 处置过程。

注1:对于软件系统,这些过程被递归地应用在更广泛或更详细的层级上以完成软件系统的定义和实现。

注2:对于软件系统,这些过程经常同时进行、相互迭代,以建立一个在需求、关键性能测度和关键质量特性方面具有令人满意的权衡的解决方案。在任何抽象层级上,通过适用的技术过程迭代,使需求和模型保持一致。当不能直接实现需求和模型时,可以在某个更详细的层级上或从不同的系统视图中递归应用这些技术过程。

注3:ISO/IEC/IEEE 24748-1中详细描述了生存周期阶段概念和这些过程在任一阶段中的应用,它具有一个完整的示例阶段和阶段输出的集合,可用于软件生存周期内技术过程的制定。

注4:接口管理是一组贯穿软件工程过程的活动。它们是技术过程组和技术管理过程组的交叉活动,作为过程和软件系统的一个具体视图进行应用和跟踪。接口管理过程视图的示例参见附录E(E.5)。

注5:ISO/IEC 27002和ISO/IEC 27034为在软件系统的技术过程组中应用信息安全问题提供了指导。有关软件保证过程视图的一个示例,请参见附录E(E.6)。

## 6.4.1 业务或使命分析过程

### 6.4.1.1 目的

业务或使命分析过程的目的是定义业务或使命的问题或机会,刻画出解决方案空间,并确定可能解决问题或利用机会的潜在解决方案类别。

注1:业务和使命分析与组织有关,包含与软件系统生存周期活动有关的利益相关方。该过程与组织的战略相互影响,组织战略一般不属于本标准的范围。组织战略的分析结果包括组织级的运营观念、战略目标和计划、新的市场或使命元素,以及已识别的问题和机会。组织战略决定了业务和使命分析的周境。组织级的运营观念与领导者运行组织的方式有关。它描述了组织的假设,以及组织如何使用、获取、供应待开发系统、现有系统和可能的未来系统,以支持业务的总体运行和一系列运行活动。如果组织是SOL,那么组织战略是系统定义的一部分。

注2:本过程贯穿软件系统解决方案的生存周期,如果环境、要求或者其他驱动因素变化时,需要重新启动本过程。

注3:在某些领域内,业务或使命分析与识别和分析组织所需或所期望的能力的概念有关。这个过程关注于必要的能力,并与群管理过程相互作用,以确定能够处理该能力的交易空间。已识别的问题或机会经常被转化为目标能力。在应用于给定的领域时,问题或机会空间包含目标能力。

### 6.4.1.2 输出

建立了业务或使命的问题和机会、首选备选方案类别的可追溯性。

业务或使命分析过程成功实施后,结果如下:

- a) 定义了问题或机会空间;
- b) 刻画了解决方案的空间;
- c) 定义了初步的运营观念和其他生存周期阶段的概念;
- d) 定义和分析了备选解决方案类别;
- e) 选择了(一个或多个)首选的备选解决方案类别;



f) 业务或使命分析所需的任何使能系统或服务均可用。

#### 6.4.1.3 活动和任务

项目应根据业务或使命分析过程相关的组织方针与规程，实施下列活动和任务。

a) 准备业务或使命分析。该活动由以下任务组成：

1) 评审在组织战略中已识别的、与预期的组织目的或目标相关的问题和机会；

注：包括与组织业务或使命、愿景、运营观念以及组织的其他战略目的和目标有关的问题或机会；包括已识别的现有能力、系统、产品或服务的缺陷或差距。

2) 定义业务或使命分析战略；

注：包括用于识别和定义问题空间、刻画解决方案空间和选择方案类别的方法。

3) 识别和计划所需要的使能系统或服务，以支持业务或使命分析；

注：包括识别使能系统的需求和接口。业务或使命分析的使能系统包括组织或其他可访问实体的业务系统和知识库。

4) 获得或获取将使用的使能系统或服务的访问权限。

注：利用确认过程，客观地确认使能系统能够满足其使能功能的预期用途。

b) 定义问题或机会空间。该活动由以下任务组成：

1) 在相关权衡空间因素的周境中分析客户投诉、问题和机遇；

注1：与权衡研究所需的系统分析和决策管理关注的综合性不同，此分析主要关注理解问题或机会的范围、基础或驱动因素。此处的焦点包括使命需求、业务机会、能力的变化、性能改进、现有系统的缺失、信息安全和安全改进，诸如成本和有效性、法规变更、用户的不满度、PESTEL(政治、经济、社会、技术、环境和法律)等因素。相关因素可以通过外部、内部或SWOT(优势、劣势、机遇、威胁)分析来识别。

注2：分析的输出可以看作是特定项目包管理决策的一部分。

2) 定义使命、业务或运行的问题或机会。

注：该定义包括周境和所有关键参数，不涉及具体解决方案，因为解决方案可以是运行的变更、现有产品或服务的变更或一个新系统。

c) 描述方案空间。该活动由以下任务组成：

1) 定义初步的运营观念和其他生存周期阶段概念；

注1：包括利益相关方需要和需求定义过程中定义的主要利益相关方的识别，例如顾客、用户、管理者、监管者和系统所有者。

注2：初步的生存周期概念包括初步的获取概念、初步的部署概念、初步的运营观念、初步的支持概念和初步的退役概念。运营观念包括高层级的运行模式和状态、运行场景、潜在用例或者所提出的业务战略内的使用。这些概念能够备选方案进行可行性分析和评价，并在利益相关方需要和需求定义过程中进一步细化。

注3：运行环境可能存在某些与特定的信息安全威胁和安全危害相关的漏洞。需要理解这些漏洞与正在开发的产品关系。系统人机接口是系统保证周境的一个元素，相关漏洞在任务关键性威胁周境中进行检查。

2) 识别覆盖潜在方案空间的备选方案类别。

注：这些类别的范围可以从简单的运行变更到各种软件系统的开发或修改。这个解决方案空间可以包括对适合重用的现有资产、系统、软件产品和能够满足运行或功能修改需要的服务变更的识别。这包括推断需要哪些潜在的预期服务。解决方案空间特征化，通常从用户架构视角，援引ISO/IEC/IEEE 42010中提出的架构定义过程，生成架构视图(例如能力视图、计划视图和运行视图)。

d) 评价备选方案类别。该活动由以下任务组成：

1) 评估每个备选方案类别；

注1：根据已建立的基于组织战略所定义的准则对每个备选方案类别进行评估。方案类别的可行性是一个关键的判定准则。特定项目包管理过程提供了一些待考虑的准则。



**注2:**系统分析过程用于评估每个备选方案类别的每个准则的价值。宜使用结构化的可承受性权衡方法。将成本作为一个准则会辅助可承受性决策。备选方案的评估可包括建模、仿真、解析技术或者专家意见,以理解备选方案类别的风险、可行性和价值。

2) 选择(一个或多个)首选备选方案类别。

**注:**决策管理过程用于评价备选方案和指导选择。在组织战略的周境下确认所选择的备选方案。提供了风险、可行性、市场因素和备选方案的反馈,用于更新组织战略。

e) 管理业务或使命分析过程。该活动由以下任务组成:

1) 维护业务和使命分析的可追溯性;

**注:**在整个生存周期中,维护业务和使命的问题和机会、首选备选方案类别与组织战略、利益相关方需要和需求、支持决策的系统分析结果之间的双向可追溯性。

2) 提供为基线选择的关键制品和信息项。

**注:**配置管理过程用于建立和维护配置项和基线。业务或使命分析过程识别构成基线的备选内容,信息管理过程控制信息项。

## 6.4.2 利益相关方需要和需求定义过程

### 6.4.2.1 目的

利益相关方需要和需求定义过程的目的,是定义利益相关方对系统的需求,该系统能够在已定义的环境下,提供用户及其他利益相关方需要的功能。

该过程定义了在整个系统生存周期内涉及到的利益相关方(或利益相关方类别)及其需要。它分析并把这些需要转化为一个利益相关方需求通用集合。该集合表示系统与其运行环境之间的预期交互,同时为了证实系统满足需要,该集合也是确认每一个作为结果的运行服务的参考。定义利益相关方需求时应考虑具有交互系统和使能系统的SOI的周境。

**注:**软件工程知识体系指南(SWEBOK)中的软件需求知识领域讨论了软件需求的基础知识(如,定义、类型、属性、质量特性)和其他主题,如利益相关方、需求获取、分析和管理,这为软件系统提供额外的指导。

### 6.4.2.2 输出

利益相关方需要和需求定义过程成功实施后,结果如下:

- a) 定义了系统利益相关方;
- b) 定义了生存周期各阶段能力和概念(包括运营观念)所需的特性和环境;
- c) 定义了系统约束;
- d) 定义了利益相关方需要;
- e) 对利益相关方的需要进行了优先级排序,并将其转换成定义清晰的利益相关方需求;
- f) 定义了关键性能测度;
- g) 实现了利益相关方关于其在需求中的需要和期望得到了充分的表达的一致同意;
- h) 利益相关方需要和需求所需要的任何使能系统和服务均可用;
- i) 建立了从利益相关方需求到利益相关方及其需要的可追溯性。

### 6.4.2.3 活动和任务

项目应根据与利益相关方需要和需求定义过程有关的组织方针与规程实施下列活动和任务。

a) 准备利益相关方需要和需求定义。该活动由以下任务组成:

1) 识别整个生存周期中对软件系统有利益的利益相关方;

**注:**对系统有利益的利益相关方包括个人或某类利益相关方,可以是用户、操作者、支持者、开发者、生产者、培训者、维护者、处置者、需方和供方组织、负责外部接口实体的相关方、监管机构以及享有合法权益的其他各方。当无法与利益相关方直接交流时(如消费类产品或服务),选择与其代表或指定代理的利益相关

方进行交流。

2) 定义利益相关方需要和需求定义策略；

注：某些利益相关方的利益与需方的利益相抵触(如，市场竞争对手、黑客、恐怖分子)或彼此对立。当利益相关方的利益彼此冲突，但与软件系统并不冲突时，本过程的目标是在各类利益相关方之间达成共识，建立一套共同可接受的需求。通过风险管理过程、系统分析过程中的威胁分析，或者系统/软件对信息安全性、适应性、恢复性的要求，来处理那些与需方对立或削弱系统的意图或期望。在这种情况下，尽管利益相关方的要求并没有得到满足，但如果遭遇损害者的行动，会以有助于确保系统的保证性和完整性的方式来处理。

3) 识别并计划支持利益相关方需要和需求定义所需的必要的使能系统或服务；

注：包括对于使能系统的需求和接口的识别。利益相关方需要和需求定义的使能系统包括促进工具和需求管理。

4) 获得或者获取使能系统或者服务的使用权。

注：确认过程用于客观地确认使能系统完成它的使能功能的预期用途。

b) 定义利益相关方的需要。该活动由以下任务组成：

1) 在运营观念和初步生存周期概念内定义使用周境；

注：使用周境参见ISO/IEC 25063的描述。初步生存周期概念在业务或使命分析过程中建立。

2) 识别利益相关方的需要；

注1:利益相关方的需要的识别包括：直接从利益相关方的需要中提取，基于对领域知识和周境理解的隐含的利益相关方需要的识别，以及在此前活动中记录的差距。需要通常包含对有效性的测量。功能分析通常用于辅助需要的提取。同时，GB/T 25000.10中的质量模型的质量特性与GB/T 25000.30中对于需求分析的质量模型的应用，可用于提取和识别通常是隐含的利益相关方需要的非功能需求中的质量需求。

注2:软件工程知识体系指南(SWEBOK)中的软件需求知识领域，讨论了一些用于提取和阐明软件需求的附加技术，例如，原型、观察、用户故事以确定所需的功能、数据挖掘，以及分析竞争对手的产品。

注3:利益相关方的需要描述了已识别的利益相关方的需要、希望、愿望、期望以及感知的约束。理解利益相关方对运营环境必需的最低的信息安全和隐私要求，可以最大程度地减少对计划、进度和实施形成的潜在破坏。如果可能出现与用户和其他利益相关方以及他们所参与的或与之交互的软件系统的有关重大问题，识别和处理和人类系统问题的建议参见ISO TS 18152,

3) 对需要进行优先级排序并向下选择；

注：决策管理过程通常用于支持优先级的排序。系统分析过程用于分析可行性需要或其他因素。

4) 定义利益相关方的需要和理论依据。

注：需要专注于系统目的和行为，并在运行环境和条件周境下进行描述。这对追溯需要的来源和理论依据很有用。

c) 开发运营观念和其他生存周期概念。该活动由以下任务组成：

注：其他生存周期概念包括获得、部署、支持、信息安全和退役的概念。在该活动中，随着相关场景和交互的定义，业务或任务分析过程中定义的初步生存周期概念将在特定利益相关方的周境中得到进一步开发。关于运营观念的更多信息参见ISO/IEC/IEEE 29148:2018中第5章和第6章，关于系统运营观念的注释大纲参见ISO/IEC/IEEE 29148:2018的附录A。

1) 定义一组有代表性的场景，以识别与预期的运行和其他生存周期概念相符的所需的能力；

注1:场景用于分析系统在其预定环境下的运行，以便于识别那些还没有被任何利益相关方所明确识别的需要或需求，如，法律的、规章的和社会的义务。定义并分析系统的使用周境，包括用户为达到系统目标而进行的活动、用户的相关特性(如，期望的培训和知识、系统的使用频率、责任、可访问性问题等)、物理环境(如，可用的光照、温度)和任何将要使用的设备(如，保护或者通信设备)。在适当的时候，应对社会和组织对于用户的影响中那些可能影响系统使用或约束系统设计的因素进行分析。关于攻击者的场景，他们的环境、工具、技术和能力是对运营观念开发的关键考虑因素。场景需要进行优先级排序，以便于体现各种运行需要的加权重要性。

注2:这些场景通常会激发对运营观念或其他生存周期概念的更新。非正常或失败场景强调了对于附加功能需求(或更具体的派生需求)的需要,以降低在非正常或失败场景中识别的风险。

2) 识别影响用户和系统之间交互的因素:

- i) 用户预期的身体的、心理的和学习的能力;
- ii) 工作地点、环境、设施,包括使用周境中的其他设备;
- iii) 普通的、不常见的、紧急的情况;
- iv) 操作者和用户的招募、培训和文化。

注1:可用性要求考虑人的能力和技能的局限性。在可能的情况下,使用适当的标准,例如 ISO 9241,及可接受的专业实践。

注2:如果可用性是重要的,应该在整个生存周期过程中,对可用性需求进行计划、规范和实施。关于人-系统问题的信息,参见ISO TS 18152,关于可用性的信息,参见ISO/IEC 25060:2010。

d) 把利益相关方需要转换为利益相关方需求。该活动由以下任务组成:

1) 识别关于系统解决方案的约束;

注:这些约束来自1)利益相关方定义解决方案的实例或者区域;2)在系统层次结构的较高层级做出的实施决策;3)已经定义的使能、遗留或接口的系统、系统元素、资源以及人员的所需要的使用;4)利益相关方定义的可承受性目标。包括那些已有协定、管理决策和技术决策所不可避免的结果。

2) 识别与关键质量特性相关的利益相关方需求和功能,例如保证、安全、信息安全、环境或健康等;

注1:关于系统和软件保证的附加信息参见ISO/IEC/IEEE 15026。

注2:识别安全性风险有助于安全性需求和功能的识别。安全性风险包括那些与运行和支持的方法、健康和安全性、对财产的威胁以及环境影响有关的内容。使用适当的标准与已被接受的专业实践。例如,IEC 61508:2010提供了详细的要求。

注3:识别信息安全风险有助于识别附加的信息安全需求和功能。如果有正当理由,适用的系统信息安全范围可以包括物理、规程、通信、计算机、程序、数据和传播。这包括访问和破坏受保护的个人信息、财产和信息,损害敏感信息,并拒绝已批准的对财产和信息的访问。当有强制的或者相关的要求时,这还包括所需的信息安全功能,例如缓解与遏制,参考适用的标准和已被接受的专业实践。有关生存周期的软件保证视图,参见附录E(E.6)。

注4:从使用质量的角度获取有关质量特性的更多信息,参见GB/T 25000,30。

3) 定义与生存周期概念、场景、交互、约束、关键质量特性相符的利益相关方需求。

注1:利益相关方需求信息参见ISO/IEC/IEEE 29148:2018第5章和第6章,利益相关方需求规格说明注释大纲的描述参见第8章和第9章。

注2:在生存周期关键决策节点,应对利益相关方的需求进行评审,以帮助确保已经考虑到需要的任何变化。

注3:利益相关方需求以一种适合在整个软件生存周期中进行需求管理的方式进行记录。这些记录确定利益相关方需求基线,并保存整个软件生存周期内需要的改变和来源。这些记录是在业务或使命分析过程中做出的决策,以及利益相关方需要、系统需求,后续的软件系统元素的可追溯性的基础。

注4:利益相关方需求是软件系统及其元素的确认准则的基础。

e) 分析利益相关方需求。该活动由以下任务组成:

1) 分析利益相关方需求的完备集;

注1:对利益相关方需求进行分享,以了解各个需求的特性以及需求集合的特性。潜在的分析特性包括需求是必需的、无实施要求、无歧义的、一致的、完整的、唯一的、可行的、可追溯的、可验证的、经济上可承受的、有界限的。ISO/IEC/IEEE 29148提供了关于需求特性的附加信息。

注2:系统分析过程用于评估可行性和经济可承受能力。验证和确认过程用在利益相关方需求的评审中。

2) 定义关键性能测度,以实现技术成果的评估;

注:关键性能测度包括定义利益相关方的需求中识别的与每个有效性测度有关的技术和质量测度,及其关键性能参数。定义、分析和评审关键性能测度(例如,有效性测度和适用性测度),以确保利益相关方的要求得到了满足,并有助于确保任何与不合规有关的项目成本、进度或绩效风险的识别。ISO/IEC 15939提

供了识别、定义和使用适当的测度的过程。INCOSE TP—2003-020-01,技术测量,提供了关于关键性能测度的选择、定义和实施的信息。GB/T 25000.23 提供了相关的质量测度。

3) 向适当的利益相关方反馈经过分析的需求,以确认他们的需要和期望得到充分地理解和表达;

4) 解决利益相关方需求问题。

注:这包括违反ISO/IEC/IEEE 29148定义的每个需求或者需求集合的特性的需求。

f) 管理利益相关方需要和需求定义。该活动由以下任务组成:

1) 获得与指定的利益相关方就关于利益相关方需求达成明确的协定;

注:包括确认利益相关方需求表达是正确的、对于发起人是可理解的,并确认对于需求中的冲突的解决方案没有破坏或违背利益相关方的意图。

2) 维护利益相关方需要和需求间的可追溯性;

注:在整个生存周期中,都需要维护利益相关方的需要和需求、组织策略,以及业务和使命的问题和机会之间的双向可追溯性。利益相关方需求在系统/软件需求定义过程中被追溯到系统/软件需求。可追溯性通常通过使用合适的数据知识库进行维护。

3) 提供已为基线所选择的关键制品和信息项。

注:配置管理过程用于建立和维持配置项和基线。利益相关方需要和需求定义过程确定基线的备选内容,信息管理过程控制信息项。对于本过程,利益相关方需要、利益相关方需求和运营观念都是基线化的典型信息项。

### 6.4.3 系统/软件需求定义过程

#### 6.4.3.1 目的

系统/软件需求定义过程的目的是,把期望的能力从面向利益相关方或用户的视图,转化为满足用户运行需要的解决方案的技术视图。

为了满足利益相关方的需求,本过程创建了一组特定的可测量的系统需求,从供方的角度指定了系统应该具备哪些特性、属性以及功能和性能需求。只要条件允许,需求不应隐含任何特定实现方式。

注1:从软件系统的高级视图来看,此过程可用于定义系统的总体需求。当软件系统被分解成元素时,每个元素依次被视为一个系统、功能或一组功能,这个过程可以用来进一步指定需求。需求分析和工具支持软件系统及其元素之间需求的可追溯性。

注2:软件工程知识体系指南(SWEBOK) 的软件需求知识领域讨论了软件需求定义、分析、建模、规格说明、确认、管理和其他为软件系统提供额外指导的主题。

注3:系统/软件需求定义过程输出的措辞与GB/T 22032—2021的系统需求定义过程输出略有不同。“系统/软件需求”措辞的使用强调了本文件对软件系统的适用性,包括软件需求和系统需求。这是为了帮助从结构层次或在不同阶段定义系统需求和软件需求的用户。

#### 6.4.3.2 输出

系统/软件需求定义过程成功实施后,结果如下:

a) 定义了对某个系统解决方案的系统或元素的描述,包括接口、功能和边界;

b) 定义了系统/软件需求(功能需求、性能需求、过程需求、非功能性需求和接口需求)和设计约束;

c) 定义了关键的性能指标;

d) 分析了系统/软件需求;

e) 系统/软件需求定义过程所需的任何使能系统或服务均可用;

f) 建立了从系统/软件需求到利益相关方需求的可追溯性。

### 6.4.3.3 活动和任务

项目应根据与系统/软件需求定义过程有关的组织方针与规程实施下列活动和任务。

a) 准备系统/软件需求定义。该活动由以下任务组成：

1) 根据提供的行为和属性定义软件系统或元素的功能边界；

注：功能边界定义部分以利益相关方需求和需求定义过程框架中所定义的使用周境和运行场景为基础。这包括软件系统的激励(输入)及其对用户和外部系统的响应，以及软件系统与其运行环境之间根据接口属性和约束所需的交互分析与描述，如规程流程、调用顺序、数据格式和数据流、吞吐量，以及时机等。这在其边界处建立了预期的、以量的方式表达的软件系统行为。对于软件，边界通常表示为应用程序接口(API)和图形用户界面(GUI)或接口文件或接口服务(包括数据格式)。附录E(E.5)提供了生存周期过程的接口管理视图。

2) 定义系统/软件需求定义策略；

注：这包括使用选定的生存周期模型(例如，演化、增量或迭代)来识别、定义和管理系统/软件需求的方法。许多因素会影响策略，例如，软件系统的复杂性以及要管理的信息和功能；需要多个团队成员随时访问并达成共识；需方或用户代表在整个开发阶段的合作参与程度；该项目是否涉及新的开发、修改、重用或现有系统的集成；以及过程文件要求，包括保存期限。生存周期模型将影响系统/软件需求定义的完成时间和频率。附录H描述了在项目中使敏捷方法逐步开发需求的过程。

3) 识别和规划为支持系统需求定义所必要的使能系统或服务；

注：这包括识别使能系统需求和接口。用于系统需求定义的使能系统包括促进和需求管理工具。与软件开发、测试和配置管理集成的软件需求管理工具可以简化跟踪并加快软件构建。支持系统/软件需求定义过程的使能系统计划和建模技术的描述可以合并到SDP中。

4) 获得或获取所要使用的使能系统或者服务的访问权限。

注：确认过程用于客观地确认使能系统实现了其使能功能的预期用途。

b) 定义系统/软件需求。该活动由以下任务组成：

1) 定义软件系统或元素需要执行的每个功能；

注1:软件功能可以在用例、用户故事或场景中描述，并涉及数据和信息的转换以实现用户需要(利益相关方需求)。在某些情况下，功能来自对关键质量特性的分析，如，性能、信息安全或可用性(如，系统诊断功能或针对可靠性的高频数据备份功能)

注2:同时识别和定义了支持SOI实现其功能所需的使能功能和SOI的功能。这有助于确保系统环境中的使能功能得到识别和说明。

2) 识别软件系统所需的运行状态或模式；

注1:运行状态或模式可以在多种建模技术和视角中进行建模和表示，以给出所期望的系统或元素需求的充分完整的描述。

注2:功能的执行条件通常涉及跨功能或元素的互操作性。如，一些软件需求(例如，性能时序限制)可以跨多个软件系统元素进行分配，从而影响测试用例或回归测试中对需求的处理。

3) 定义必要的实施约束；

注：对于软件元素，这包括从软件系统的更高层级架构定义中分配，并由利益相关方需求或解决方案限制所引入的实施决策。实施约束包括系统能够执行该功能的条件、系统开始执行该功能的条件(输入)以及系统停止执行该功能的条件(输出)。

4) 识别与风险、软件系统的关键性或者关键质量特性相关的需求；

注1:软件系统中的非功能性要求和关键质量特性通常包括那些与健康、安全、信息安全性、可靠性、可用性和可支持性(维护性)，以及吞吐量和性能的时间约束相关的需求。系统分析过程可用于确定性能需求的适当值，同时考虑达到这些值的预期成本及其对系统运行和使用的影响。

注2:安全性的分析和定义考虑因素包括那些与运行和维护方法、环境影响，以及人员伤害风险相关的要求。就必要的风险降低和制定安全相关系统的分配而言，它还包括表达每个安全相关功能及其相关完整性。使用涉及功能安全性的适用标准，例如IEC 61508,以及环境保护的适用标准，例如ISO 14001。分

析包括安全性考虑因素，例如，那些与敏感信息、数据和材料的泄露和保护有关的考虑因素。酌情使用适用的信息安全标准，定义了信息安全相关的风险，包括：管理的、人员的、物理的、计算机、通讯、网络、传播和环境因素。关于系统和软件保障的指南，参见ISO/IEC 15026-4、ISO/IEC 27036提供了关于产品和服务外包的信息安全要求的指导。GB/T 25000.30提供了关于外部系统质量因素和特性的指导。附录E(E.6)提供了生存周期过程的软件保证视图。

注3:对于用于人机交互的软件系统，需考虑人机工程规范(人机工程学)。对于具有可用性要求的系统，可以在ISO/FDIS 9241-220中找到获得期望的可用性级别的建议。

5) 定义系统/软件需求和需求属性，包括以下内容：

- i) 数据元素、数据结构和格式，以及数据库或数据保存要求；
- ii) 用户界面、用户文档(用户信息)和用户培训；
- iii) 与其他系统和服务的接口；
- iv) 功能和非功能特性，包括关键质量特性和成本目标；
- v) 来自现有的自动和手动系统运行过程和数据的转换、迁移方法和时间表、软件安装和产品验收；
- vi) 需求属性，如基本原理；优先级；对软件系统元素、测试用例和信息项的可追溯性；验证的方法；列入核准的基线以及经过评价的风险。

注1:需求定义涉及与其他生存周期过程并行的迭代和递归步骤。根据所使用的生存周期模型，将用于确保需求的初始正确性所要花费的资源与基于验证和确认结果进行需求演进所花费的资源进行比较，是有用的。

注2:具有一定详细程度的，并以一种适合于贯穿整个生存周期的需求管理的形式记录了系统/软件需求和属性。关于系统需求的更多信息参见 ISO/IEC/IEEE 29148:2018的第5章和第6章，关于系统需求规格说明和软件需求规格说明的描述和注释大纲参见其第8章和第9章。

c) 分析系统/软件需求。该活动由以下任务组成：

1) 分析系统/软件需求的完整集；

注1:分析需求以了解各个需求的特征以及需求集的特征。潜在的分析特性包括需求是必要的、无需实现的、明确的、一致的、完整的、单一的、可行的、可追溯的、可验证的、可负担的和有边界的。验证过程用于确定需求是否满足符合标准的要求的属性和特性。在某些情况下，对需求确认和验证备选公式的技术和经济可行性进行评价。ISO/IEC/IEEE 29148提供了关于需求特性的更多信息。

注2:考虑到软件系统的估计成本、进度和技术性能，系统分析过程可用于评估可行性、可承受性、平衡性以及其它需求特性。系统分析过程用于确定需求参数的适当值。

注3:预期某些需求可以逐步实现，甚至可以推迟或放弃，可以对需求优先级进行排序。

2) 定义能够评估技术成果的关键性能指标；

注：包括定义技术和质量测度以及与软件系统元素需求中确定的每个有效性测度相关的关键性能参数。分析和评审关键性能测度(如，性能测度和技术性能测度)，以帮助确保系统/软件需求得到了满足，并帮助确认识别到任何与不合规有关的项目成本、进度或相关性能风险。ISO/IEC 15939 提供了识别、定义和使用合适措施的过程。INCOSE TP-2003-020-01,技术测量，提供了关于关键性能测度的选择、定义和实施的信息。GB/T 25000.23提供了相关的质量测度。

3) 将分析后的需求反馈给合适的利益相关方进行审核；

注：反馈有助于确认指定的需求已被充分获取和表达。确认它们是对利益相关方需求的必要和充分的响应，以及对其他过程(特别是软件架构定义过程、设计过程和验证过程)的必要和充分的输入。验证过程用于确定系统/软件需求是否满足用户的需要。

4) 识别并解决整套需求中的问题、不足、冲突和缺点。

注：这包括那些不能验证的、含糊不清的、违反个别需求特性的，或者与需求集中的其他需求不一致的需求。在某些特定的生存周期模型中，需求的问题可以迭代解决。

d) 管理系统/软件需求。该活动由以下任务组成：

注：维护系统/软件需求包括通常在正式的配置管理下定义、记录和控制基线，以及管理因其他生存周期过程



(如, 架构过程和设计过程)的应用而产生的任何变更。

1) 就系统/软件需求达成明确的一致意见;

注: 包括确认系统/软件需求得到了正确地表达, 易于发起人和实施者理解, 并且需求冲突的解决方案与利益相关方的决策一致。

2) 维护系统/软件需求的可追溯性;

注: 在整个生存周期中, 系统/软件需求和利益相关方需求、架构实体、接口定义、分析结果、验证方法或者技术以及分配、分解和派生的需求间保持双向可追溯性。可追溯性允许验证可实现的利益相关方需求能够通过一个或多个系统或元素需求来满足, 并且这些需求满足或有助于满足至少一个利益相关方需求。追溯通常由适当的数据知识库或集成的开发和测试基础设施来促进。

3) 提供为基线所选择的关键制品和信息项。

注: 配置管理过程用于建立和维护配置项和基线。系统/软件需求定义过程识别基线的备选内容, 并为配置管理过程提供信息项。对于本过程, 系统/软件需求是基线化的典型制品。

#### 6.4.4 架构定义过程

##### 6.4.4.1 目的

架构定义过程的目的是产生系统架构备选方案, 选择构建利益相关方关注且满足系统需求的一个或多个备选方案, 并用一组一致的视图进行表达。

通常使用架构定义过程与业务或使命分析过程、系统/软件需求定义过程、设计定义过程以及利益相关方需要和需求定义过程的迭代, 以便对需要解决的问题达成协商一致的理解并确定出满意的解决方案。架构定义过程的结果被广泛使用于整个生存周期过程。架构定义可以应用于多个抽象层次, 突出在该层次决策所必需的相关细节。

注1: 系统架构涉及基本原理、概念、属性和特性以及它们与SOf的结合。架构定义具有更多用途, 而不仅仅是设计的一个驱动因素或一部分。关于架构的描述、使用和基本特征的更多信息, 参见ISO/IEC/IEEE 42010:2011。

注2: 架构定义过程支持识别利益相关方及其关注点。随着过程的展开, 可以深入了解为软件系统指定的需求与由于系统元素之间的交互和关系而产生的系统的紧急属性和行为之间的关系。有效的架构尽可能地与设计无关, 以便在设计权衡空间中实现最大的灵活性。即使对于单一产品软件系统, 产品的设计也可能随着时间的推移变化, 而架构保持不变。有效的架构还突出并支持设计定义过程以及其他可能过程(如, 特定项目包管理过程、项目规划过程、系统/软件需求定义过程和验证过程)的权衡。

注3: 架构定义可以应用于产品线, 而不是单个的软件系统。产品线架构描述了构建了一组具有公共组件和相互关系的相关系统的结构属性。在产品线架构中, 该架构必然跨越多个设计。该架构有助于产品线的统一, 并有助于确保产品线之间的兼容性和互操作性。ISO/IEC 26550:2015描述了如何为产品线建立领域架构。

注4: 软件工程知识体系指南(SWEBOK) 的软件需求、软件设计和软件工程模型和方法知识领域讨论了软件架构与系统之间的关系的方面, 以及关于设计的迭代。

##### 6.4.4.2 输出

架构定义过程成功实施后, 结果如下:

- a) 通过架构处理了已识别的利益相关方的关注;
- b) 开发了架构视角;
- c) 定义了系统的周境、边界和外部接口;
- d) 开发了系统的架构视图和模型;
- e) 对系统的架构决策有重要意义的概念、属性、特性、行为、功能或约束被分配给架构实体;
- f) 识别了系统元素及其接口;
- g) 评估了架构备选方案;
- h) 实现了整个生存周期过程的架构基础;

- i) 实现了架构与需求、设计特性的一致性；
- j) 架构定义过程所需的任何使能系统或服务均可用；
- k) 开发了架构元素对利益相关方和系统/软件需求的可追溯性。

#### 6.4.4.3 活动和任务

项目应根据与架构定义过程有关的组织方针与规程实施下列活动和任务：

##### a) 准备架构定义。该活动由以下任务组成：

###### 1) 评审相关信息并识别架构的关键驱动因素；

注1:通过评审确定关键驱动因素：(a)市场研究、行业预测、竞争者产品计划和科学发现；(b)组织战略、组织层面的运营观念、组织的政策与指令、监管与法律约束和利益相关方需求；(c) 业务或使命运营观念、SOI与相关系统的运营观念、运行环境、技术路线图和系统/软件需求；以及(d)影响软件系统整个生存周期适用性的其他因素。这种对关键驱动因素的分析通常从业务或使命分析过程、利益相关方需要和需求定义过程以及系统/软件需求定义过程中构建。

注2:架构的关键驱动因素可包括架构样式和模式、元素、原则(如可替换组件、实现和集成的可行性)；COTS和开源组件的可用性；数据密集型系统的数据源；以及性能影响。如果对软件系统进行适当的架构设计，则可以减少选择各种设计元素的影响。

###### 2) 识别利益相关方的关注点；

注1:在利益相关方需要和需求定义过程中初步识别了利益相关方。在架构定义过程中识别其他利益相关方。与架构相关的利益相关方关注点包括对系统完整性的关注，即软件系统将通过威胁代理有意或无意地受到危害，或作为安全隐患导致事故。利益相关方的期望或约束通常与系统生存周期阶段有关，例如利用(例如，可用性、信息安全性、有效性、易用性、与现有系统的互操作、系统中数据的可用或风险)、支持(例如，系统在其预计寿命期内的可支持、报废管理)、系统及其环境的演进(例如，适应性、可扩展性、可生存性)、产品(例如，分配、可测试性)，以及退役(例如，敏感数据清除或保留)等。

注2:影响软件系统架构的因素包括：数据密集型系统的数据源和性能影响，以及对外包的、现有的、新研的专用的、商用的或开源软件元素(包括软件许可)的使用限制。虽然软件架构设计理想情况下是不可知的，但对于大多数系统，如何在一个经济可承受的软件系统中实现架构的可行性，是一个重大制约因素。

###### 3) 明确架构定义的路线图、方法和策略；

注：包括与指定的利益相关方沟通机会的确定、架构评审活动的定义、评价方法与准则、测量方法和测量手段(参见测量过程)。路线图表明了架构将如何演变为预期的最终状态，并且通常涵盖了比当前 SOI 更长的时间范围。方法是指工作完成的方式，例如：如何与利益相关方接触、如何检查结果或在哪里开展工作。该策略涉及执行与路线图一致的方法的系统行动计划。

###### 4) 根据利益相关方关注点和关键需求确定评价准则；

###### 5) 确定并规划支持架构定义过程所需的必要使能系统或者服务；

注：包括确定使能系统与服务的需求和接口。用于架构定义的使能系统包括用于协作和架构开发的工具，以及关于制品(如架构模式、架构模型、参考架构等)的架构重用知识库。

###### 6) 获得或获取使能系统或者服务的访问权限。

注：确认过程用于客观地确认使能系统是否达成它的使能功能的预期用途。基础设施管理过程支持使能系统的重用。

##### b) 开发架构视角。该活动由以下任务组成：

###### 1) 根据利益相关方关注点选择、调整或开发视角和模型种类；

###### 2) 建立或确定用于开发模型和视图的潜在架构框架；

注：某些架构框架确定了利益相关方及其关注点，以及处理这些关注点的相关视角，而其他架构框架在其指导下更为通用。视角指定了要使用的模型种类以及结果模型如何用于生成架构视图。参见 ISO/IEC/IEEE 42010, 获得更多架构框架和架构描述实践的信息。

###### 3) 获取选择框架、视角和模型类型的基本原理；



4) 选择或开发支持建模技术和工具。

注：SWEBOK 和 ISO/IEC/IEEE 24748-3 都描述了支持软件元素的架构定义和设计定义的建模技术。

c) 开发备选架构的模型和视图。该活动由以下任务组成：

1) 根据接口和与外部实体交互来定义软件系统周境和边界；

注：此任务主要是基于业务或使命分析过程的输出，并与利益相关方需要和需求定义过程同时进行。它包括识别软件系统外部的实体（即构成系统周境的现有与规划的系统、产品和服务）和定义软件系统边界（即通过接口跨越系统边界与这些外部实体交互）。外部实体应包括必要的使能系统。架构定义过程定义了基本架构决策和理解所需程度的接口。然后通过设计定义过程来细化这些接口定义。

2) 识别能够解决关键的利益相关方关注点和关键的软件系统需求的架构实体以及实体之间的关系；

注：架构不一定涉及所有需求，而是只涉及了驱动架构的那些系统/软件需求。另一方面，设计定义过程应处理和考虑所有需求。有时，通过架构定义过程，会有一些被认为不适当、不可负担或者不适用的需求。这些是通过系统/软件需求定义过程的迭代来解决的需求问题。同样重要的是，该架构解决了关键利益相关方所关心的问题，因为并非所有此类问题都会在需求中得到体现。

3) 将对软件系统的架构决策具有重要意义的概念、属性、特性、行为、功能或约束分配给架构实体；

注：分配的条目可以是物理的、逻辑的或概念的。

4) 选择、调整或开发软件系统备选架构的模型；

注：在架构定义中使用模型是很常见的。所使用的模型是那些最能解决关键利益相关方关注问题的模型。有关如何完成该任务，参见 ISO/IEC/IEEE 42010。历史上，在架构定义中使用逻辑和物理模型是很常见的。附录 F 提供了有关逻辑模型和其他模型的信息。

5) 根据确定的视角从模型中构建视图，以表达架构如何解决利益相关方关注的问题并满足利益相关方和系统/软件需求；

6) 相互协调架构模型和视图。

注：框架中的对应规则是建立视图之间协调的一种方式，参见 ISO/IEC/IEEE 42010。

d) 将架构与设计相关联。该活动由以下任务组成：

1) 识别与架构实体相关的软件系统元素以及这些关系的本质；

注：有时软件系统元素最初只是概念性的，直到出现“设计定义”为止，因为这取决于要完成的实际设计。有时使用这些概念系统元素创建“参考架构”来作为表达架构意图和检查设计可行性的手段。

2) 定义软件系统元素之间以及外部实体之间的接口和交互；

注：这是在传达架构意图所必需的细节层次上定义的，并且可以在设计定义过程中对其进一步细化。

3) 将需求划分、调整和分配至架构实体和系统元素；

4) 将软件系统元素和架构实体映射到设计特性；

5) 定义软件系统设计和演变的原则。

示例：原则可以包括互操作性、选定的设计模式的使用、易于替换和升级系统元素或信息安全级别。

e) 评估备选架构。该活动由以下任务组成：

1) 根据约束和要求评估每个备选架构；

2) 使用评价准则，根据利益相关方的关注点评估每个备选架构；

注：系统分析过程和风险管理过程可以用于支持该任务。

3) 选择首选架构并获取关键决策和理由；

注：决策管理过程可以用于支持该任务。

4) 建立所选架构的架构基线。

注：架构基线由模型、视图和其他相关架构描述组成。

f) 管理选定的架构。该活动由以下任务组成：

1) 正式制定架构的治理方法并指定与治理相关的角色与责任、职责，以及与设计、质量、信息

安全、安全等相关的权限；

2) 获得利益相关方对此架构的明确接受；

注：确认过程用于确认架构模型和视图是否反映了利益相关方需求，解决了利益相关方关注的问题，并帮助确保未来的软件系统架构迭代能更好地解决利益相关方关注的问题。

3) 保持架构实体及其架构特性的一致性和完整性；

注：所要检查的实体不仅是技术方面的，也可能是，例如，法律、经济、组织和运营的实体，通常是利益相关方的需求和关注的一部分。

4) 组织、评估和控制架构模型和视图的演进，以帮助确保架构意图得到满足，期望架构和关键概念得到正确实施；

5) 维护架构定义和评价策略；

注：包括根据技术(例如，过时)、实施、运行经验以及在此系统分解层级上定义的内部和外部接口管理来更新架构。

6) 维护架构的可追溯性；

注：在整个生存周期中，架构实体或元素(模型、视图和视角)、需求(包括分配、分解和派生)、利益相关方关注点、软件系统设计、接口定义、分析结果以及验证方法或技术之间维持可追溯性。

7) 提供为基线所选择的关键制品与信息项。

注：配置管理过程用于建立和维持配置项和基线。架构定义过程识别基线的备选内容，并为配置管理过程提供信息项，例如架构描述(架构模型、架构视图、评价，以及可追溯性)。

## 6.4.5 设计定义过程

### 6.4.5.1 目的

设计定义过程的目的是提供有关系统及其元素的足够详细的数据和信息，以便使实施与系统架构模型和视图所定义的架构实体相一致。

对于软件系统，设计活动通常与系统/软件需求定义过程和架构定义过程中的活动进行迭代。设计定义通常迭代地和增量地应用于开发详细的设计，包括软件元素、接口、数据库和用户文档。软件设计通常与软件实施、集成、验证和确认同时进行。附录H 讨论了使用敏捷方法的软件设计。在设计和实施过程中，进一步的过程应用细化了软件元素之间演进需求的分配。

注1:设计定义过程是由通过架构评审的需求和更详细的可行性分析所驱动的。架构关注适用性、可行性和可取性，而设计侧重于技术和其他设计元素的兼容性以及实施和集成的可行性。一个有效的架构尽可能地与设计无关，以允许在设计权衡空间中拥有最大的灵活性。

注2:本过程向系统架构提供反馈，以固化或确认架构实体的分配、划分和排列。

### 6.4.5.2 输出

设计定义过程成功实施后，结果如下：

- a) 定义了每个系统元素的设计特性；
- b) 系统/软件需求分配给了系统元素；
- c) 选择或定义了设计定义所必需的设计使能者；
- d) 定义或改进了系统组成元素之间的接口；
- e) 评估了系统元素的设计备选方案；
- f) 开发了设计工作制品；
- g) 设计定义过程所需要的任何使能系统或服务均可用；
- h) 建立了设计特性对系统架构实体的可追溯性。

注：设计定义考虑适用的技术及其对系统解决方案的贡献。设计提供了“实现到”这一级别的定义，例如图纸、状态图、故事和详细设计说明。对于软件元素，这个过程可以产生一个可根据需求和软件架构进行验证的详细设计

说明。即使软件设计在正式的描述中没有被完全地指定，它也足够详细以允许软件实施(构建)和测试规划。

### 6.4.5.3 活动和任务

项目应根据与设计定义过程有关的组织方针与规程实施下列活动和任务。

注：软件工程知识体系指南(SWEBOK)提供了关于软件设计的详细讨论。该知识领域涉及基本原理、关键问题、设计策略和方法、设计符号等。

a) 准备软件系统的设计定义。该活动由以下任务组成：

1) 定义设计定义的策略，使其与所选择的生存周期模型和预期的设计制品保持一致；

注：软件设计策略可包括初始的或增量分解成系统元素；自动化程序、数据结构和控制系统的各种视图的创建；设计模式的选择，或对象及其关系的逐步地更详细的定义。

2) 选择设计原则和设计特征，并确定其优先级；

注：设计原则包括控制思想，如，抽象化、模块化和封装、接口和实施的分离、并发性，以及数据的持久性。信息安全考虑因素包括最小特权原则、分层防御、对系统服务的受限访问，以及最小化和防御系统受攻击面的其他考虑因素。设计特征包括，如，可用性、容错性和恢复能力、可伸缩性、可用性、容量和性能、可测试性、可移植性和可负担性。

3) 确定并规划支持设计定义所需的必要的使能系统或服务；

注：包括使能系统的需求和接口的确定。设计定义的使能系统包括软件和系统平台的选择、编程语言、用于协作与设计开发的设计符号和工具、设计重用知识库(用于产品线、设计模式、设计制品)和设计标准。

4) 获得或获取对所要使用的使能系统或服务的访问权限。

注：确认过程用于客观地确认使能系统是否实现其使能功能的预期用途。

b) 建立与每个软件系统元素相关的设计。该活动由以下任务组成：

1) 将架构和设计特性转换为软件系统元素的设计；

注：特性适用于物理和逻辑系统元素，如数据库结构、内存和存储的规定、软件过程和控制、外部接口(如用户界面)或服务。ISO 9241-210提供了以人为本的设计/人机工程学设计指南。

2) 定义并准备或获得必要的设计使能因素；

注：设计使能因素包括模型、方程、算法、计算、形式化表达和参数值、模式，以及启发等。这些与使用适当表达形式(如图纸、逻辑图、流程图、编码约定、逻辑模式、信息模型、业务规则、用户配置文件、场景、用例或用户故事，指标表及其值等)的设计特性相关，例如，功能点或用户故事点。

3) 检查设计的备选方案和实施的可行性；

注1:对于软件系统和软件元素，通常检查了重用、改编、外包服务或新的开发。

注2:评估实现设计特性的可行性。如果评估结果有保证，当设计特性无法实现时，检查其他备选设计方案，或在架构或需求中进行权衡。

4) 细化或明确软件系统元素之间及其与外部实体之间的接口；

注：在架构定义过程(见6.4.4)中识别和定义了接口，以达到架构意图和理解所需的级别或范围。在设计定义过程中基于软件元素与构成软件系统的其他元素以及与外部实体的设计特性、接口和交互，对这些进行了细化。有时会识别和定义架构定义中未提及的其他接口。

5) 建立了设计工作制品。

注：此任务根据实现技术，通过专用的工作制品对软件系统元素的设计特征进行正式化处理。工作制品的示例包括原型、数据模型、伪代码、实体关系图、用例、用户角色和特权矩阵、接口规范、服务描述，以及规程。为选定的备选方案开发、获得或修改设计工作制品。数据与实现的详细可接受裕度相关联(如果与此过程或任务迭代相关)。

c) 评估获得软件系统元素的备选方案。该活动由以下任务组成：

1) 确定构成软件系统的每个元素所需的技术；

注：某些技术有时用于给定的软件系统元素，例如，互联网的存在、嵌入式系统，开源软件的改编、人工操作员角色。

2) 确定软件系统元素候选的备选方案；

注：备选方案包括新设计和建造的项目；现有产品线、组件、对象或服务的改造；非开发项目(NDI)的获取或重用。NDI包括COTS(商业现货软件)或FOSS(自由开源软件)程序包或元素、先前设计的重用或现有资产(包括需方提供的项目)。

3) 根据从预期设计特性和系统元素要求所制订的准则，评估每个候选方案，以确定预期应用的适用性；

注：一个“自制或购买”的决定以及由此产生的实施和集成的方法通常涉及设计准则的权衡，包括成本。设计选择通常考虑用所需的使能系统来测试备选方案(测试驱动的设计和开发)和系统生存周期的可持续性，包括维护成本。维护过程可用于确定设计是否适合长期维护和可持续性。

4) 为软件系统元素在候选设计方案中选择首选方案。

注：系统分析过程可用于在执行选择中支持决策管理过程的分析和评估。使用确认过程对设计进行评审。

d) 管理设计。该活动由以下任务组成：

1) 捕获设计和基本原理；

注：通常捕获的信息包括软件系统元素和相关的需求及设计数据，例如，用于软件元素、内部和外部接口、数据结构、实现和测试需求、用于集成的单元聚合数据以及测试用例。基本原理通常包括关于主要实施方案和使能技术的信息。根据该策略对最终的设计进行控制。

2) 建立详细设计元素、系统/软件需求和软件系统架构实体之间的可追溯性；

注1:该任务有助于向架构定义过程提供反馈以进行潜在修改，例如，修改软件系统元素的分配，以获得预期的架构特性；或者可能由于设计过程中发现的因素而修改预期的架构特性；或者使利益相关方意识到其潜在的影响。

注2:在整个生存周期中，设计与验证方法或技术以及软件系统元素需求之间保持双向可追溯性。对软件元素、软件单元和附属工作制品进行分配和设计属性的委派，其详细程度足以允许软件测试和实施，包括构建。

3) 确定软件系统和元素设计的状态；

注1:测量过程用于在设计过程中建立完整性和质量的测度。援引验证和确认过程来验证和确认详细设计和实现。

注2:这包括在软件系统及其架构的不断演进中定期评估设计特性，以及预测组件和技术的潜在过时，在软件系统的生存周期中随着时间的推移被其他组件和技术代替的情况，设计定义的后果。风险管理过程通常用于评价设计策略、初始设计和不断演进的设计中的风险。

4) 提供为基线选择的关键工作制品和信息项。

注：配置管理过程用于建立和维持工作制品(如设计模型)的配置项和基线。设计定义过程确定基线的候选内容，信息管理过程控制信息项，例如设计描述和规范。

## 6.4.6 系统分析过程

### 6.4.6.1 目的

系统分析过程的目的是为技术理解提供严格的数据和信息基础，以帮助跨生存周期的决策。

系统分析过程适用于开发任何技术评估所需的输入。它可为系统需求、架构和设计的效用和完整性提供信心。系统分析广泛涵盖了各种不同的分析功能、复杂性等级和严苛性等级。它包括用于分析技术性能、系统行为、可行性、经济可承受性、关键质量特性、技术风险、生存周期成本，以及在所有生存周期阶段中对参数的潜在取值范围进行敏感性分析的数学分析、建模、仿真、实验和其他技术。它用于涉及运营观念的各种分析需要、需求值的确定、需求冲突的解决、备选架构或系统元素的评估，以及工程策略的评价(集成、验证、确认和维护)。分析的正式性和严密性将取决于信息需要或所支持的工作产品的重要性、可用的信息或数据量、项目的规模，以及结果的进度计划。

注：系统分析过程可用于整个软件系统或任何元素。此过程通常与决策管理过程结合使用。

### 6.4.6.2 输出

系统分析过程成功实施后，其结果如下：

- a) 识别了所需要的系统分析。
- b) 确认了系统分析的假设和结果。
- c) 为决策提供了系统分析结果。
- d) 系统分析所需的任何使能系统或服务均可用。
- e) 建立了系统分析结果的可追溯性。

6.4.6.3 活动和任务

项目应根据与系统分析过程有关的组织策略和规程实施下列活动和任务。

- a) 确定系统分析策略，并为系统分析做准备。该活动由以下任务组成：
  - 1) 识别需要分析的问题或疑问；  
注：包括分析的技术、功能性和非功能性目标。非功能性目标包括关键质量特性、各种属性、技术成熟度和技术风险。问题陈述或通过分析待解答的疑问对建立分析目标以及结果的预期和效用至关重要。
  - 2) 识别分析的利益相关方；
  - 3) 定义分析的范围、目标和保真度等级；  
注：必要的保真度(准确度或精确度)等级是决定适当精确等级的一个因素。
  - 4) 选择支持分析的方法；  
注：基于时间、成本、精度、技术驱动因素和分析的重要性来选择分析方法。分析方法涵盖广泛的严格性等级，包括专家判断、电子工作表计算、参数估计和计算、历史数据和趋势分析、工程模型、仿真、可视化和原型。由于成本和进度的限制，大多数项目仅对关键特性进行系统分析。
  - 5) 识别并规划支持分析的必要的使能系统或服务；  
注：包括识别使能系统的需求和接口。系统分析使能系统包括支持分析所需的工具、相关模型和潜在的数据仓库。所选分析方法是决定哪些工具适合支持分析的主要因素。这还包括确定可重用的或其他相关模型和数据或资源的可用性。
  - 6) 获得待使用的使能系统或服务的访问权限；  
注：基础设施管理过程支持提供系统分析服务。确认过程用于客观地确认使能系统实现其使能功能的预期用途。
  - 7) 收集分析所需的数据和输入。
- b) 执行系统分析。该活动由以下任务组成：
  - 1) 识别与确认周境和假设；
  - 2) 应用所选择的分析方法执行所需的分析；
  - 3) 评审分析结果的质量和有效性；  
注：将结果与之前已完成的相关分析进行协调一致。
  - 4) 得出结论和建议；  
注：确定合适的主题专家和利益相关方参与到该任务中。
  - 5) 记录系统分析的结果。
- c) 管理系统分析。该活动由以下任务组成：
  - 1) 维护系统分析结果的可追溯性；  
注：在整个生存周期中，在分析结果与所有由此分析支持决策或提供基本原理的软件系统项(例如，系统/软件需求值、备选架构)之间保持双向可追溯性。该任务通常需要合适的数据仓库支持。
  - 2) 提供为基线所选择的关键工作制品和信息项。  
注：配置管理过程用于建立和维护配置项和基线。系统分析过程识别基线的候选内容，信息管理过程控制信息项。对于本过程，分析结果或报告是被管理的典型信息项。

## 6.4.7 实现过程

### 6.4.7.1 目的

实施过程的目的是完成特定的系统元素。

本过程根据所选实现技术的实践，使用适当的技术专业或学科，将需求、架构和设计(包括接口)转换为创建系统元素的行动。本过程产生满足指定系统需求(包括分配和派生的需求)、架构和设计的系统元素。

对于软件系统，实施过程的目的是实现软件系统元素。

软件系统元素可包括硬件、软件和服务。对于软件实现，本过程将指定的设计、行为、接口和实施约束转换为行动，从而将已实现的软件系统元素创建为软件产品或服务，也称为“软件项”。软件实现产生通过验证满足特定需求和通过确认满足利益相关方需求的软件元素。软件实现包括各种构件的组合(对新构件的软件元素进行编码)、新软件包的获取(如来自开源、商业或组织来源)或现有元素的重用(有或没有修改)。

软件实现通常涉及协定过程的使用来获得非开发项(NDI)，如硬件和操作系统(平台)或使能系统和服务。软件实现通常与软件集成同时执行。实施通常伴随着所有技术管理过程和很多技术过程，特别是：

- a) 验证过程，它提供了客观证据证明软件实现满足其特定需求，并识别实现相关的信息项(如系统/软件需求、架构、设计或其他描述)、过程、软件元素、项、单元中的异常(包括错误、缺陷、故障)；
- b) 确认过程，它确认实施满足某个软件工作产品某一具体特定预期用途的要求。

### 6.4.7.2 输出

实施过程成功实施后，其输出结果如下：

- a) 识别了影响需求、架构或设计的实现约束；
- b) 实现了系统元素；
- c) 打包或存储了系统元素；
- d) 实施所需的任何使能系统或服务均可用；
- e) 建立了可追溯性。

### 6.4.7.3 活动和任务

项目应按照与实施过程有关的适用组织政策和规程，实施下列活动和任务。

- a) 准备实施。该活动由以下任务组成。
  - 1) 制定实施策略，考虑以下几点：
    - i) 开发政策和标准，包括规范适用的安全、信息安全、隐私及环境实践的标准；编程或编码标准；单元测试原则；以及实现信息安全特性的语言特定标准；
    - ii) 对于重用或改编的软件，确定所重用系统元素的层次、来源和适用性以及供应链信息安全的方法；
    - iii) 软件开发(构建)和单元测试开发的规程和方法；以及在实施过程中使用同行评审、单元测试和代码走查；
    - iv) 在软件构建过程中配置管理控制的使用；
    - v) 用于手动过程的变更管理考虑因素；
    - vi) 支持数据和软件迁移和过渡的实施优先事项，以及遗留系统的退役；



vii) 在软件单元(测试驱动开发)创建之前,创建手动或自动测试规程,以验证软件单元满足其需求;

viii) 用于实现和管理需求、模型和原型、可交付的系统或软件元素,以及测试规范和测试用例的综合或专门的生存周期开发和支持环境。

注:实施策略通常记录在项目的SDP或SEMP中,或有时也记录在PMP中。

2) 识别来自实施策略和实现技术对系统/软件需求、架构特性、设计特性或实现技术的约束。

注1:约束包括所选择实现技术(例如,软件、操作系统、数据库管理系统、web服务)当前或预期的限制、需方提供的用于改编的材料或系统元素,以及所要求的实施使能系统的使用所带来的限制。

注2:软件的实施策略通常确定并分配“实施到”准则,例如,软件架构和设计特性、系统/软件需求,包括软件保证、使用性考虑因素、配置管理,可追溯性,或其他需要满足的条件。这些标准则可以阐明适当的单元聚合层次、规范和约束条件。

3) 识别和规划必要的和独特的软件环境,包括支持开发和测试所需的使能系统或服务。

注:软件的实现通常使用在配置控制下与运行(生产)环境相分离的独特环境。常见的实施过程、使能系统和服务包括用于实现和管理需求、模型和原型、可交付的元素以及测试环境、规范和测试用例的综合或专门的生存周期开发和支持环境;用于外部系统、培训系统的模拟器,以及用于用户文档的内容管理系统。

4) 获得或获取对软件环境和其他使能系统或服务的访问权限。

注:确认过程用于客观地确认集成使能系统实现了其使能功能的预期用途。

b) 执行实施。该活动由以下任务组成:

注:在整个实施过程中,验证过程用于客观地确认系统元素是否符合要求。确认过程用于根据利益相关方需求,客观地确认元素是否适合在其预期的运行环境中使用。

1) 根据策略、约束和定义的实施规程,实现或适配软件元素;

注1:获取了、识别了从组织资产中用于重用的,或者开发了(构造)软件元素。获取的软件元素可以从根据组织或项目采购规则进行的简单产品采购,到涉及获取和供应过程的软件系统的复杂获取。适配包括对所重用或修改的软件元素的配置。构建可涉及软件编码、现有单元的适配重用与集成、代码重构、数据库开发,以及为每个单元构建手动或自动测试规程。

注2:对于所开发的软件元素,在实施的最低层次,可执行软件单元进行了构建(通常带有相关的数据结构、应用程序编程接口、服务描述、用户文档、测试用例或者其他元素)、控制、使授权角色可用,并根据开发工作制品的CM规程进行了存储。

注3:SWEBOK, 软件工程知识体系指南提供了关于软件构建的详细讨论。这一知识领域涉及基本原则、管理、测量、实践考虑因素(如,构建设计、语言、测试、重用和集成)、构建技术(如,面向对象、错误和异常处理,可执行模型、分布式软件),以及工具和环境。

2) 实现或适配软件系统的硬件元素;

注:对硬件元素进行获取或者使用与所选物理实现技术和材料相关的适用技术进行制造。适当时,对硬件元素与特定系统需求和关键质量特征的符合性进行验证。对于重复的系统元素实施(如,批量生产、替换系统元素)的情况下,对实施规程和制造过程进行定义,并可以自动地实现一致的和可重复的可生产性。软件系统中某些常见的硬件元素包括对所获取COTS系统的集成、特殊的修改(如,对测试或运行环境),以及带有嵌入式软件的硬件控制。

3) 实现或适配软件系统的服务元素;

注:服务元素包括一系列待提供的服务。ISO/IEC 20000(IEEE Std 20000)应用于在服务中所实现的系统元素的管理,包括策略、设计和移交。适当时,对服务元素与系统需求和服务准则的符合性进行验证。如,对运行资源元素与系统需求和运营观念的符合性进行验证。服务元素可包括网络通信、培训、软件打包和分发服务、针对客户特定需要的软件定制服务、运行和信息安全性监控,以及用户帮助。

4) 根据实施策略和准则,评价软件单元及附属数据或其他信息;

注1:评价的准则通常包括单元需求和测试准则的满足、单元测试覆盖率、可追溯性需求,与软件元素需求或设计的一致性、内部单元需求的一致性,以及进一步过程活动的可行性,如:集成、验证、确认、运行和维护。

注2:使用管理实施的结果活动来记录构建和处理异常。

5) 打包并存储软件系统元素;

注: 包含软件系统元素, 以实现其特性的连续性。运输和储存及其持续时间可以影响具体的控制。对于软件, 已实现软件的主副本(电子的或物理媒介上的)存储在受控的位置, 并使其对授权角色来说是可用的(如, 在集成和移交过程中使用)。当存储元素时, 配置和产品信息由配置管理和信息管理过程捕获。

6) 记录软件系统元素满足需求的客观证据。

注: 根据供货协定、法规和组织方针提供证据。证据包括由于处理变更或在验证和确认过程中发现的不符合项而进行的元素修改。客观证据是通过配置管理过程建立的元素已实施的配置基线的一部分, 并包括单元测试、分析、检查、走查事件、论证、产品或技术评审, 或者其他验证活动的结果。

c) 管理实施结果。该活动由以下任务组成:

1) 记录实施结果和遇到的异常;

注: 包括由于实施策略、实施使能系统或不正确的软件系统定义导致的异常。项目评估与控制和质量保证过程用于分析数据, 以识别根本原因, 采取纠正或改进措施, 并记录经验教训。

2) 维护所实施软件系统元素的可追溯性;

注1:为在运行和维护期间支持整个生存周期的可追溯性, 对供应链中软件许可和其他系统资产的来源进行记录。信息管理和配置管理过程用于维护软件应用程序及其所需基础设施(主机系统)的许可和维护支持条款。ISO/IEC 19770标准提供了对于IT资产管理系统的要求。

注2:维护已实施的元素与软件系统架构之间的双向可追溯性; 设计和相关需求, 包括接口需求和必要的实施定义; 确认和验证计划、规程和结果。

3) 提供为基线所选择的关键工作制品和信息项。

注: 配置管理过程用于建立和维护配置项和基线。本过程识别基线的候选内容, 并由信息管理过程控制信息项。对于本过程, 软件系统元素(如源代码)、软件包和单元测试结果是基线化的典型工作制品。

## 6.4.8 集成过程

### 6.4.8.1 目的

集成过程的目的是将一组系统元素合成为满足系统/软件需求、架构和设计的已实现的系统(产品或服务)。

本过程装配已实现的系统元素。识别并激活接口, 以实现系统元素预期的互操作性。本过程将使能系统和所关注的系统进行集成, 以促进互操作性。

软件系统集成迭代地组合已实现的软件系统元素, 以形成完整的或部分的系统配置, 从而构建产品或服务。软件集成通常每天执行, 或在开发和维护阶段使用自动化工具持续不断地执行。持续集成包括在CM控制下, 对软件库中的项进行频繁的纳入或者替换和归档。

注: 通过架构定义和设计定义过程定义接口。本过程与其他过程协调, 以检查已实施和集成的接口定义是否充分, 以及它们是否考虑了集成需要。

### 6.4.8.2 输出

集成过程成功实施后, 结果如下:

- a) 识别了影响系统需求、架构或设计(包括接口)的集成约束;
- b) 定义了装配接口和系统功能正确操作的方法和检查点;
- c) 集成所需的任何使能系统或服务均可用;
- d) 集成了由已实现系统元素组成的系统;
- e) 检查了组成系统的已实现系统元素之间的接口;
- f) 检查了系统和外部环境之间的接口;
- g) 识别了集成结果和异常;



h) 建立了集成系统元素的可追溯性。

#### 6.4.8.3 活动和任务

项目应根据与集成过程有关的组织方针与规程实施下列活动和任务。

a) 准备集成。该活动由以下任务组成。

1) 定义集成策略；

注1:集成构建了逐步完善的软件系统元素或软件项配置的序列。它依赖于适用的软件系统元素的可用性,并与故障隔离和诊断策略相一致。为系统结构中的元素进行集成过程、验证过程,以及确认过程(当适当时)的重复应用,直到所关注的系统得到实现。模拟器或原型通常用于尚未实现的系统元素,例如,从接口系统接收数据。集成已实现的软件系统元素是基于相关需求和架构定义的优先级,通常专注于接口之上,同时最小化集成时间、成本和风险。软件系统集成通常通过配置管理过程来维护版本控制,以选择待集成的配置项。

注2:对于软件集成,集成策略通常与回归策略一致,回归策略应用于当相关软件单元(以及潜在相关需求、设计和用户文档)发生变更时重新验证软件元素。

注3:为软件单元和元素集成定义策略通常伴随着为其他并发过程定义策略,例如:

- i) 实现过程有助于确保实现和集成过程任务和使能系统的及时协调,例如,组合软件开发和测试环境,以支持软件单元和元素的自动化的或连续的实施和集成;
- ii) 验证过程提供客观证据证明已集成的软件是否满足其特定需求,并识别了集成相关信息项(如,系统/软件需求、架构、设计、测试或其他描述)、过程、软件元素、项、单元中的异常(错误、缺陷、故障);
- iii) 确认过程确认工作产品是否满足集成软件功能的特定预期用途的要求;
- iv) 质量保证过程支持集成过程和工作产品的审核和审查,并解决问题、不符合项,或事件报告和处理。

注4:集成策略通常记录在一个计划中,例如,一个集成计划,或者一个项目的SDP或SEMP。

2) 确定和定义集成准则以及将要验证接口和所选择的软件系统功能正确操作和完整性的点;

注1:使用验证过程执行接口的详细验证。软件集成通常涉及组合软件元素,从而产生一组与软件设计一致的集成软件元素,并在与运行环境相当条件下,满足功能性和非功能性软件/系统需求。

注2:对于涉及多个供方或开发团队的项目,在项目评估与控制过程下,用于集成的软件系统元素的可用性通常具有里程碑项目进度计划的一部分。随着软件在功能、性能和对特定场所或特定平台环境的适用性中的验证,集成工作继续进行。在主要的集成点,例如,某个阶段、元素或版本的完成,通常保持用于与利益相关方进行评审和确认的检查点。这些评审的频率与所选择的生存周期模型和开发方法有关。

3) 识别和规划支持集成所需的使能系统或服务;

注:包括识别使能系统需求和接口。集成使能系统通常包括集成设施、专用设备、培训系统、差异报告系统、模拟器、测量设备和环境的信息安全。对于软件,这可以涉及用于软件系统集成测试的回归测试套件和CM系统、事件和问题报告系统、代表外部系统或尚未开发元素的模拟器,以及用于开发作业的软件库管理系统。使能系统所需的变更或特殊化以支持待确定和定义的集成任务。通常,由于软件系统和支持的环境会演进到运行状态,在开发阶段用于集成的使能系统或服务也可以帮助支持系统元素集成。这种“DevOps”方法支持迭代的软件系统实现、集成、验证、移交、确认、运行和维护过程。

4) 获得或获取用于支持集成的使能系统或服务的访问权限;

注:确认过程用于客观地确认集成使能系统实现了其使能功能的预期用途。

5) 识别整合到系统/软件需求、架构或设计中的集成约束。

注:这包括例如集成商的可访问性、供应链信息安全、安全的需求,已实现的软件系统元素集和使能者所需的互联,以及接口约束。

b) 执行集成。依次集成软件系统元素配置,直到合成完整的系统。该活动由以下任务组成:

1) 根据协定的进度计划,获取已实现的软件系统元素;

注：已实现的软件系统元素由开发人员提供，或者从供方、需方或其他资源处接收，并且通常置于CM 控制之下。根据相关的健康、安全、信息安全和隐私等考虑因素对元素进行处理。

#### 2) 集成已实现的元素：

注1:根据集成策略的规定，使用已定义的规程和接口控制描述以及相关的集成使能系统，来执行本任务，以实现软件系统元素配置(完整的或部分的)连接已实现的元素。

注2:就软件而言，集成已实现的元素可能涉及到将目标代码片段链接在一起，或者只是简单地将作为软件配置一部分的已实现元素有条理的逐段方法组合在一起。软件元素通常被编译成一个“构件”，以便分支单元在组装的系统元素中被适当地连接或合并。固件元素通常作为原型制造，并安装在硬件元素中。如果软件功能尚不可用于集成，则可以使用模拟功能来临时地支持软件元素的集成或表示来自外部接口的输入。成功的聚合会产生一个集成的软件元素，该元素可以被存储并可用于进一步的处理，即额外的软件系统元素集成、验证或确认。

注3:在执行集成、识别和定义检查点时，可能会出现防伪造、防篡改、系统和软件保证以及互操作性问题。出于信息安全与隐私考虑，集成和验证过程通常使用虚拟数据。ISO/IEC/IEEE 15026和 ISO/IEC 27000 系列包括了关于影响集成的保证、完整和信息安全考虑因素的信息。

#### 3) 检查集成软件接口或功能是否在预期的数据值范围内从启动到预期终止都在运行。

注：作为已实现的软件系统元素验收活动的一部分，对所选择的元素进行检查，以帮助确保它们符合集成策略和适用协定中规定的验收准则。检查可以包括与约定配置的符合性、接口的兼容性和强制性信息项的存在。项目评估与控制过程可以根据集成策略来规划和进行已集成软件系统元素的技术评审，例如，测试准备就绪评审，以帮助确保用于鉴定测试的集成元素或系统及其附属数据和信息项准备就绪。

### c) 管理集成结果。该活动由以下任务组成：

#### 1) 记录集成结果和遇到的异常：

注：这包括由于集成策略、集成使能系统、集成执行或不正确的系统或元素定义而导致的异常。当系统、其特定的运行环境和开启使用阶段的系统之间存在接口不一致时，则偏差会形成纠正措施。异常的解决方案通常涉及技术过程组，通常是实现过程的重复应用。质量保证、项目评估与控制过程用于分析数据以识别根本原因，采取纠正或改进措施，并记录经验教训。

#### 2) 维护已集成软件系统元素的可追溯性：

注：保持已集成系统元素与软件系统架构、设计以及系统或元素需求(例如用例，以及包括集成时所必要的接口需求和定义)之间的双向可追溯性，已集成软件元素及其组件通过版本标识。已集成软件元素的版本通常可以追溯到已实现的单元、测试规程和测试用例。

#### 3) 提供为基线所选择的关键工作制品和信息项。

注：配置管理过程用于建立和维护配置项和基线。集成过程识别基线的备选内容，信息管理过程控制信息项。对于本过程，测试用例、回归测试和自动化测试脚本是典型的基线化制品。集成策略是一个典型的基线化信息项。

## 6.4.9 验证过程

### 6.4.9.1 目的

验证过程的目的是提供客观证据来证明系统或系统要素满足其特定需求和特性。

验证过程使用适当的方法、技术、标准或规则识别信息项(如系统/软件需求或架构描述)、已实现的系统元素或生存周期过程中的异常(如错误、缺陷或故障)。本过程提供了对已识别的异常制定解决方案的必要信息。

验证可以贯穿所有技术过程。验证过程通常用于软件系统生存周期的关键点，以证明需求(包括功能性和非功能性需求)已经得到满足，或者过程结果已获得，或者过程活动已执行。不同的专业和工程或开发领域可以确定不同的里程碑、验证策略和准则。

对于软件系统，验证过程通常是为实现以下目标而实例化的过程：

#### a) 确认软件工作产品或服务适度反映了规定的需求(通常称为软件验证)；

- b) 确认集成的软件产品符合其预定的需求(通常称为软件合格测试);
- c) 确认对每个系统/软件需求的实现进行了符合性测试,并已准备交付(通常称为系统合格测试)。

注1:验证过程确定“产品被正确地构建”。确认过程确定“构建了正确的产品”。

注2:GB/T 38634.1(在多部分内容中)提供了通过测试进行验证的详细过程和技术。IEEE Std 1012—2012 提供了关于正在开发、维护或重用的系统、软件、硬件和接口的详细附加信息。

注3:SWEBOK,《软件工程知识体系指南》,提供了关于软件测试的详细讨论。这一知识领域涉及基础理论、术语、问题、技术、应用、过程规划、度量方法、工具,以及实践考虑,并提供了参考文献。该指南还讨论了软件质量管理过程中的软件验证和确认问题,并确定了支持验证和确认的方法和技术。SWEBOK还致力于解决诸如用于验证的软件构建和软件工程模型和方法支持。

#### 6.4.9.2 输出

验证过程成功实施后,其结果如下:

- a) 识别了影响需求、架构或设计的验证约束;
- b) 验证过程所需的所有使能系统或服务均可用;
- c) 验证了系统或系统元素;
- d) 报告了提供纠正措施信息的数据;
- e) 提供了证明所实现的系统满足需求、架构和设计的客观证据;
- f) 识别了验证结果和异常;
- g) 建立了验证系统元素的可追溯性。

#### 6.4.9.3 活动和任务

项目应根据与验证过程有关的适用组织策略和规程实施下列活动和任务。

- a) 准备验证。该活动由以下任务组成。

- 1) 确定验证策略,包括:

注1:验证策略通常关注最小化成本、进度或风险,提供一种平衡的方法来确认软件系统或元素已被“正确构建”。

注2:当异常结果(事件、事故或问题)发生时,验证策略和进度要考虑动态变更。根据项目进展,当非预期事件或系统演进发生时,重新定义或重新安排已计划的验证活动。

注3:验证策略可以记录在项目计划中,如验证计划或项目SDP或SEMP中。

- i) 识别验证范围,包括要验证的软件系统、元素或构件、属性和预期结果;

注:总体的验证范围包括所关注的软件系统或系统元素,包括接口。对于每个验证活动,范围标识了要验证的软件系统、元素或构件(如,实际系统、模型、原型、代码、程序、计划或其他文档)和其他预期结果,如符合性、性能、容错和服务中断后的恢复。需要验证的属性包括需求、架构和设计特性、集成和文档的准确性。设计特性可以包括在计划运行环境中设计的安全含义,以及需求说明书中所描述的关键质量特性的实现。

- ii) 识别可能限制验证活动可行性的约束条件;

注:约束条件包括技术可行性、成本、时间、验证使能项或具有资质的人员的可用性、合同约定以及任务的重要性等特性。这些约束条件经常对验证策略的确定产生影响,例如组织独立验证工作是否必要或合理。

- iii) 确定验证的优先次序。

注:在软件系统中,验证每个可能的场景(100%的代码覆盖率)通常是不可行的。验证策略通常根据约束或限制条件对将要验证的内容(范围)进行权衡,并推断要执行什么验证活动,以及验证活动需要多少次迭代和返工以降低风险。基于模型的测试方法可以生成和管理多个应用场景。对于拟删减的验证活动,需要评估删减带来的风险。

2) 从系统/软件需求、体系架构或设计的验证策略来识别约束条件。

注：识别内容包括精度、不确定性、受验证支持方影响的重复性等实践性限制、相关的度量方法、软件系统集成的需求，以及验证支持方的可用性、可达性和关联性。

3) 定义每个验证动作的目的、条件和符合性准则。

4) 为验证活动选择适当的验证方法或技术以及相关的标准，如检查、分析、演示或测试。

注1：验证方法或技术的选择是根据系统类型、验证目的、项目目标和可接受的风险水平进行的。验证方法或技术包括检查(包括代码走查和同行评审)、分析(包括建模和仿真，以及类比/类推)、演示，以及动态和静态测试。

注2：所选择的验证方式、方法和技术可以与相关利益相关方协商，以确保验证方法是可接受的。

5) 识别并规划必要的使能系统或服务以支持验证。

**注：**验证使能系统包括验证设施、合格的人员、设备、模拟器、测试自动化工具、事件和问题管理系统。软件系统验证通常在不干扰运行软件或与正在进行的开发工作的受控环境中执行。如果用于验证的使能系统的能力与计划的系统运行环境不同，则可以使用测量过程来校准验证使能系统的性能和验证操作的适用性。

6) 获得或获取用于支持验证的使能系统或服务的访问权限。

注：使能系统的获取可以通过各种方式实现，如，租赁、采购、开发、组织资产重用或分包。通常，可以采用上述方法的组合获得完整的使能项套件。确认过程用于客观地确认验证使能系统实现了其使能功能的预定用途。

b) 执行验证。该活动由以下任务组成：

1) 定义验证规程，每个验证规程支持一个或一组验证动作；

注：验证规程可由自动化脚本执行，包括需要验证的需求、需要验证的软件系统元素或构件的类型(如，实际系统、模型、模型的原型、模型、代码、过程、计划或其他信息项)，以及预期结果(成功准则)，如一个功能或能力在响应时间或吞吐量方面的符合性或性能。规程用成功准则(预期结果)识别验证的目的，识别所应用的验证技术、必要的使能系统(设施、设备)以及执行每个验证规程的环境条件(资源、合格的验证人员、特定的程序安装设置或工作指导)。验证规程包括如何记录、分析、存储和报告验证规程结果。

2) 执行验证过程。

注：根据验证策略，验证发生在进度计划的适当时刻，在预定义环境中，使用已定义的使能系统和资源。验证行动的实施包括从验证规程的执行中获得结果；将得到的记录结果与预期结果进行比较；并推断所提交元素的正确性(成功/失败)程度。

c) 管理验证结果。该活动由以下任务组成：

1) 核查验证结果和验证过程中发生的异常，并确定后续行动；

注1：包括由于验证策略、验证使能系统、验证执行或不正确的系统定义而导致的异常。利用项目评估、控制和质量保证过程分析数据识别根本原因，采取纠正或改进措施，并记录经验教训。

注2：根据验证目的的不同，验证结果评价和后续纠正措施可能会有很大差别。对于软件元素包括诸如需求的修改或放弃，需要针对失败的软件元素进行简单的缺陷修复后的重新验证，对重点项目中未能达到关键里程碑的失败(如，软件系统合格测试失败)需要重新定位项目的方向。对于验证过程中发现的异常通常采用简单或推荐的解决方案，并记录解决方案和验证结果，以方便分析并采用可能的纠正措施。

2) 记录验证过程中的偶发事件和问题并跟踪其解决途径；

**注：**通过质量保证、项目评估与控制过程来实施问题解决方案。在软件验证期间，问题发生的条件需要文档记录，以便在可能的情况下复现问题，并识别软件缺陷产生的根本原因。使用其他技术过程组中的过程来实现需求、架构、设计或系统元素的变更。

3) 取得利益相关方同意，确认软件系统或元素满足指定的需求；

4) 保持已验证软件系统元素的可追溯性；

**注：**在已验证系统元素与验证活动、系统架构、设计或系统/软件需求记录之间保持双向可追溯性。

5) 提供纳入基线的关键构件和信息项。

注：配置管理过程用于建立和维护配置项和基线。此过程识别基线的候选内容，并由信息管理过程控制信息项。对于这一过程，验证策略和验证规程是典型的信息项。

#### 6.4.10 移交过程

##### 6.4.10.1 目的

移交过程的目的在于为系统建立能够在运行环境中提供由利益相关方需求所规定服务的能力。

此过程以有序、有计划的方式将系统推进到运行状态，使系统具有功能性、可操作性并与其他运行系统兼容。此过程将安装已验证系统以及相关使能系统，如在协定中定义的规划系统、支持系统、操作员培训系统和用户培训系统。本过程用于系统结构的各个层次和阶段，完成为退出此阶段而建立的准则，包括准备适用的存储、处理和装运使能系统。

对于软件系统，移交过程的目的是在不同环境中建立系统以提供服务能力。

移交过程通常涉及将软件反复部署到不同的环境中。例如从开发环境到测试或维护环境，或在不同的测试环境之间切换，或从一个运行环境迁移到另一个运行环境(如，重新移植或使用云服务)。转移到备份或应急站点通常是为了业务连续性和灾难恢复而计划和演练的。软件系统的移交可能还涉及硬件的物理重新配置、不同位置物理或虚拟基础设施或使能系统的安装、激活或停用，或对物理基础设施不做变更。移交可包括对数据源和数据结构的变更或功能软件的更新或升级，也包括为了安全和其他方面考虑而进行的重新计划或应急补丁和系统修复。移交可以包括组织之间的切换，还包括向现有软件系统或服务添加大量的新用户。向一个新系统的转换通常与现有系统退役和处置过程同时进行，并需要对数据进行迁移。

注：移交可以包括使用知识管理过程实现知识转移。

##### 6.4.10.2 输出

移交过程成功实施后，其结果如下：

- a) 识别了影响系统/软件需求、架构或设计的移交约束；
- b) 移交所需的任何使能系统或服务均可用；
- c) 移交场所准备就绪；
- d) 系统安装在其运行位置的系统能够交付指定的功能；
- e) 对操作人员、用户和其他利益相关方进行了系统使用和支持的培训；
- f) 识别了移交结果和异常；
- g) 已安装的系统被激活并准备运行；
- h) 建立移交元素的可追溯性。

##### 6.4.10.3 活动和任务

项目应根据与移交过程相关的组织方针与规程实施下列活动和任务。

a) 准备移交。该活动由以下任务组成。

1) 定义一个管理软件发布和其他软件系统移交的策略，包括以下考虑：

- i) 建立移交类型和移交成功标准；
- ii) 确定重复移交的频率，如，对开发、测试和运行的软件系统的更新和升级；
- iii) 最小化移交期间的安保风险、中断和宕机时间；
- iv) 将旧系统转换到新系统时数据的存档、销毁、迁移和确定，包括从外部接口接收的数据；
- v) 问题解决、备份、返回到最近工作系统版本的应急计划；
- vi) 制定与正在进行的业务处理过程相一致的移交计划，采用分阶段或同步系统移交；

- vii) 利益相关方的变更管理，包括接口合作伙伴、人工操作员、系统管理员、软件系统或服务用户；

注：变更管理活动通常用于设计与新系统相关的业务流程变更、业务过程中的移交计划，以及获得用户对新系统积极使用的承诺。

- vii) 用于确认移交系统或元素的相关策略；

- ix) 初始化用户支持和维护活动，并移交和更新系统设计文档、用户文档和测试程序；

- x) 新系统启用和旧系统退役时，移交、运行和处置过程需同时执行。

注：策略包括角色和职责、权限批准、使用准备就绪的评审和培训。

- 2) 识别和定义软件系统安装或移交所需的设施、场地、通信网络或目标环境的变更。

注：对于每个移交过程，识别并定义基础设施或使能系统所需的任何更改。现场调查确定安装或使用软件系统所需的物理环境变更，如为维护系统物理和信息安全所做的变更。

- 3) 识别信息需求，安排操作人员、用户和其他利益相关方在系统使用和安全保障方面所必需的用户文档和培训。

注：移交包括迁移或激活用户对软件系统的访问。建立用户角色，实现用户账户和访问控制。

- 4) 准备详细的移交信息，如计划、进度计划和程序。

注1:移交策略通常记录在计划中，如一个移交计划、项目的 SDP或SEMP中。移交进度计划有助于验证是否有足够的资源和基础设施支持移交，使移交活动能在合理的时间范围内执行并将干扰降到最小。进度计划可包括对复杂移交的演练，这在当中，如，数据库、系统备份和恢复以及软件安装等规程，需要被测试以验证其持续时间和结果的正确性。

注2:在具体的转换或并发运行期间，需要对服务的转移进行管理，保证其持续符合利益相关方的需求或达到协定的服务水平。如果需要旧系统和新系统并行运行一段时间，则需要确定和开发具体的流程以保证从接口合作方接收和利用数据。

- 5) 识别软件系统需求、架构或设计中涉及移交的系统约束。

- 6) 识别和规划保障移交所需的必要的使能系统或服务。

注：包括对使能系统的需求和接口的识别。移交通常涉及使用高度自动化的基础设施来交付、安装和激活或停用软件。对于电子方式的软件分发，通常需要对软件和数据迁移以及持续保障进行临时或持续的连接变更。使能系统可以包括在移交期间使用的备份或备用系统。

- 7) 获得或获取待使用的使能系统或服务的访问权限。

注：确认过程用于客观地确认移交使能系统达到了其使能功能的预期用途。

- b) 实施移交。该活动由以下任务组成：

- 1) 根据安装需求准备运行场所或虚拟环境；

注：根据适用的健康、安全、信息安全和环境法规准备场所。对虚拟环境和新的通信资源进行初始化和验证。安排了物理系统元素和使能系统的传输和接收。

- 2) 在正确的时间和地点移交用于安装的软件系统或元素；

注1:软件通常以电子方式交付。对于物理介质、硬件和嵌入式软件系统，有时需要在交付或安装之前考虑临时存储。

注2:以电子或实物形式交付协定的信息项，如培训材料、后勤支持包或用户文档。

- 3) 将产品安装在系统物理或虚拟运行位置和环境接口处；

注：产品安装包括使用所需的运行数据、环境变更或业务过程变更来配置软件系统。实例化数据库并进行适当的数据迁移。根据协定转移系统元素和其他知识产权的许可和维护协议。

- 4) 向操作者、用户和其他利益相关方提供系统使用和安全保障必要的用户文档和培训；

- 5) 根据协定执行系统激活和检查，包括以下内容：

注1:此任务根据运行规程、组织方针和规范，采取必要步骤将产品激活到运行状态，包括初始化、环境条件评估和其他准备情况评估。如果运行的确切位置或环境不可用，或者软件将从多个位置或移动位置访问，则选择一个具有代表性的示例。



注2:有时在协定中会预先定义验收测试证明其满足安装需求。此任务与确认过程相互作用,客观地确认系统能够在运行环境中满足利益相关方需求。协定所规定的验收测试可以定义一些准则来证明软件系统实体在运行环境中安装和维护时,具有交付所需功能和服务的能力,特别要关注系统关键功能和逻辑接口。

注3:作为配置管理过程的一部分,通常在系统激活时进行物理配置审计(PCA)和已开发文档的更新,以及防伪条款的确认。

i) 证明软件系统的安装正确;

注:此任务可以包括数据和操作的完整性检查。如确保软件代码和数据表示正确地初始化、执行和终止。

ii) 证明所安装或移交的产品能够交付所需的功能;

注:这是一项运行就绪任务,此任务检查运行状态下功能能力的准备情况。特别需要注意数据接口和安全问题:执行信息安全保障和交互操作功能。

iii) 证明此系统所提供的功能在使能系统中是可持续的;

注:这是一项运行就绪任务,此任务检查运行状态下使能系统的准备情况。如,演示如何激活监视、问题报告、访问控制、备份和恢复以及用户帮助(客户支持)。

iv) 评审软件系统的运行就绪状态;

注:这包括功能演示、确认活动和持续演示的结果。可以进行准备情况评审。影响移交过程成功的缺陷、风险和问题需要得到解决、同意豁免或关闭。

v) 软件系统试运行。

注:包括在系统启动(试运行)期间为用户、管理员和操作人员提供支持。

c) 管理移交结果。该活动由以下任务组成:

1) 记录移交结果和遇到的所有异常;

注:包括由于移交策略、移交使能系统、移交执行或不正确的软件系统或数据库系统定义而导致的异常。当系统与运行环境和使能系统之间存在不一致时,通过纠正措施(包括需求变更)来解决偏差。项目评估与控制过程和质量保证过程用于分析数据以识别其根本原因,采取纠正或改善措施,并记录经验教训。

2) 记录移交偶发事件和问题并跟踪解决;

注:通过质量保证过程和项目评估与控制过程来执行问题解决。在移交过程中,记录问题发生的条件,以便在可能情况下复制问题,识别其根本原因。使用其他技术过程实现对需求、架构、设计或软件系统元素的变更。

3) 维护移交软件系统元素的可追溯性;

注:在被移交和部署的系统和元素与软件系统和使能系统经批准和控制的版本之间保持双向可追溯性。

4) 提供为基线所选择的关键制品和信息项。

注:配置管理过程用于建立和维护配置项和基线,包括被移交的软件系统元素。此过程识别基线的候选内容,并由信息管理过程控制信息项。本过程中的移交策略、培训材料、安装、移交和数据迁移过程以及用户文档是典型的基线化的信息项。

## 6.4.11 确认过程

### 6.4.11.1 目的

确认过程的目的是提供客观的证据,证明系统在使用时满足其业务或任务目标,以及利益相关方需求,并在预期的运行环境中实现预期的使用。

确认一个系统或系统元素的目的是在特定的运行条件下获得对其实现预期任务或使用能力的信心。确认是由利益相关方批准的。此过程提供了必要的信息,以便通过创建适当技术过程来解决已识别的异常。

确认过程通常在产品生存周期的关键点上使用,以证明产品满足了利益相关方预期的运行使用需求。确认也适用于软件工程制品(视为软件系统元素)。不同的领域和工程或开发社区可以以不同的方

式识别里程碑、确认策略和准则。

对于软件系统，高度迭代的生存周期模型通常有一个特点，那就是需方、用户代表或其他利益相关方频繁地参与进来，以确认：如，迭代中包含需求的优先级，通过原型软件接口的可用性，以及执行业务任务和满足运行概念的软件适用性。

对于软件系统，确认过程的目的如下：

- a) 确认软件工作产品的特定预期用途的要求得到满足（通常称为软件确认）；
- b) 实现交付的产品满足利益相关方需求并适合使用的信心（特别是与需方或顾客）（通常称为软件验收测试）。

注1:验证过程确定“构建了正确的产品”。确认过程确定“产品是正确构建的”。

注2:验收准则用于验收测试，包括确定交付产品是否适合使用的准则。确认准则可由双方指定并商定，即一个需方和一个供方，并包含在利益相关方需求中。

注3:IEEE Std 1012—2012 提供了详细的要求。SWEBOK,《软件工程知识体系指南》，从软件质量管理过程的角度讨论了软件验证和确认，并包含了支持验证和确认的方法和技术。SWEBOK还处理需求和模型验证等主题。

#### 6.4.11.2 输出

确认过程成功实施后，结果如下：

- a) 定义了利益相关方需求的确认准则；
- b) 确认了利益相关方所需服务的可用性；
- c) 确定了影响需求、体系架构或设计的确认约束；
- d) 确认了系统或系统元素；
- e) 确认了所需的任何使能系统或服务均可用；
- f) 识别了确认结果和异常；
- g) 提供了已实现的系统或系统要素满足利益相关方需要的客观证据；
- h) 建立了所确认系统元素的可追溯性。

#### 6.4.11.3 活动和任务

项目应根据与确认过程有关的适用组织方针和规程实施下列活动和任务。

- a) 确认准备。该活动由以下任务组成。

- 1) 定义确认策略，确认策略包括：

注1:确认策略通常关注于通过逐步建立对利益相关方的软件系统的质量和适用性的信心来最小化成本、进度或风险。

注2:确认策略反映了生存周期模型，通常涉及对迭代、增量或演进生存周期的重复验证。

注3:确认策略可以记录在计划中，如，一个确认计划、项目的SDP或SEMP

- i) 确定确认范围，包括待确认的软件系统、元素或制品的特性，以及确认的预期结果；

注：软件系统验证通常在不影响运行软件或正在进行的开发的不同的受控环境中执行，以及在运行环境中执行，通常在完全运行使用之前(如，beta测试或验收测试在规定的时间内，按照约定的标准进行)。范围包括利益相关方需求，其包含系统的相关视图(如，脚本或运行概念)，以供评估。范围取决于系统生存周期阶段的适当内容：系统参数或系统元素或工程制品，如概念描述或文档、运行场景、模型、模拟或原型。范围还包括评价软件产品或服务在其主要或关键功能的预期环境中是否可用。需要确认的其他特性包括文档的可用性；软件的容错、恢复和复原功能。

- ii) 确定可能限制确认操作可行性的约束条件；

注：约束包括验证使能项、相关度量方法、可用性、可访问性和与使能项的互连所施加的准确性、不确定性和可重复性的实际限制。确认策略受项目进度的限制；特别是，当意外事件或系统演进发生时，重新定义或重新调度计划的确认操作。确认可以扩展到包括用户满意度和顾客投诉的持续测量。



iii) 确定确认优先级。

注1:为了有效利用利益相关方的时间和专业知识,确认通常关注利益相关方的优先级,而验证用于非功能性需求。潜在的确认行动是删除候选对象,评价它们的删除带来的风险。

注2:供方、需方或需方代理人参与或执行确认。责任通常在协定中指定。

2) 从确认策略中识别系统约束,并将其纳入利益相关方需求中。

3) 定义每个确认动作的目的、条件和一致性标准。

4) 为每个确认操作选择适当的确认方法或技术以及相关的标准。

注1:软件系统验证方法或技术包括检查、分析、类比/相似、演示、模拟,同行评审和测试。软件确认技术通常包括演示、检查、评审和代码演练、可用性测试和软件的试用(如,软件测试、beta测试、操作测试、用户测试或具有约定标准的验收测试)。确认方法或技术的选择是根据系统类型、确认目的、项目目标、法律和法规要求以及确认活动的可接受风险来进行的。对于具有人类交互的软件系统,可用性测试通常用于确认代表性用户能够在特定的使用环境中有效、高效和满意地实现指定的目标。关于可用性测试的更多细节见ISO/IEC TR 25060:2010。

注2:在适当的情况下,定义了确认步骤或状态(如,包括内部确认、现场确认、运行确认),逐步建立对不断发展的软件系统符合性的信心,并协助诊断任何遇到的差异。

注3:利益相关方接受服务性能的准则通常以服务级别表示,并记录在服务级别协议(SLA)中、服务级别通常度量服务的容量、可用性、可靠性和及时响应,并产生支持系统的性能需求。

5) 识别和计划支持确认所需的必要使能系统或服务。

注:根据所选择的确认方法或技术,使能系统可以包括模拟器、可用性实验室或测试设施、合格的人员、利益相关方和用户代表。这包括为使能系统确定需求和接口。

6) 获得或获取用于支持确认的使能系统或服务的访问权限。

注:可以调用基础设施管理过程或获取过程来获得对使能系统的访问权限,如,租用、采购、开发、重用或分包。通常,访问完整的使能项是这些方法的混合。确认过程还用于客观地确定确认使能系统是否实现了其使能功能的预期用途。

b) 执行确认。该活动由以下任务组成:

1) 定义确认过程,每个确认过程支持一个或一组确认动作;

注:确认规程确定了待确认的利益相关方需求,相关的软件系统制品(例如:实际的系统、模型、模拟、原型、代码、一组指令或其他信息项),以及预期的结果(成功准则),如功能的完成和及时执行。规程通过成功准则(预期结果)、所应用的确认技术、必要的使能系统(设施、设备)和执行每个验证规程的环境条件(资源、合格人员、参与的利益相关方、专门的程序设置或工作指示)来确定确认的目的。确认策略包括如何记录、分析、存储和报告验证过程的结果。

2) 在定义的环境中执行确认过程。

注:根据确认策略,确认发生在进度计划的适当时间,在定义的环境(如操作环境、类似的测试环境或其他代表性环境)中,使用定义的使能项和资源。确认行动的绩效通常包括捕获执行结果,将获得的结果与成功准则进行比较,并推断软件系统、元素、服务或工程制品的依从性或利益相关方满意度的程度。

c) 确认结果管理。该活动包括以下任务:

1) 评审确认结果和遇到的异常,并确定后续行动;

注1:确认利益相关方需要的系统服务是可用的。异常可能来自确认策略、启用确认的系统、确认的执行、不正确的系统定义,或者低效或无效的系统设计、实现和集成。

注2:确认结果的评估和后续行动包括将异常视为低风险事件的接受。根据确认结果的影响,纠正措施可能会有很大的不同。对于软件元素,例子包括对失败的软件元素进行简单的缺陷修复,对用户进行额外的培训,在文档中进行更正和澄清,或者基于未能达到关键里程碑的重大项目指示,例如,软件系统验收测试失败。

2) 记录确认过程中的事件和问题,并跟踪其解决情况;

注:项目评估与控制过程和质量保证过程用于分析数据,以确定问题的根本原因,采取纠正或改进措施,并记录所吸取的教训。在软件确认过程中,利益相关方的期望和系统性能之间的差距被记录下来,以便在可

能的情况下，可以确定差异的根本原因。问题解决通常涉及确定问题的严重性和影响，以及是否或何时将软件差异作为已知错误进行纠正或接受一段时间。对于确认过程中发现的异常，通常使用确认记录简单或推荐的解决方案结果有助于分析和潜在的纠正措施。利益相关方和系统/软件需求、架构、设计或系统元素的实际更改是在其他技术过程中完成的。

3) 取得利益相关方同意，软件系统或元素满足利益相关方的需求；

4) 保持已确认系统元素的可追溯性；

注：在被确认的系统元素和利益相关方需求之间保持双向的可追溯性，并记录确认结果。

5) 提供已经为基线选择的关键工件和信息项。

注：配置管理过程用于建立和维护配置项和基线。此过程识别基线的候选对象(如，经过确认的软件系统或元素)，而信息管理过程控制信息项。对于这个过程，确认策略和确认结果是典型的基线化信息项。

## 6.4.12 运行过程

### 6.4.12.1 目的

运行过程的目的是使用系统以交付其服务。

此过程建立需求，为运行系统分配人员，并监视服务和运营商系统性能。为了维持服务，它识别和分析关于协定、利益相关方需求和组织约束的运行异常。

运行过程通常旨在控制或降低运行成本，同时保持可接受的或改进的服务水平。

软件系统可以有专用的基础设施，但通常在其他软件系统和服务(如互联网)是激活的分布式环境中运行。因此，所关注的软件系统的信息安全、可用性和运行性能是一个更大的系统之系统中的所关注的焦点。它可以包括与预先存在的、由提供相同或类似服务的其他系统提供的并发或持续服务的协调。

注：ISO/IEC 20000-1:2001(IEEE Std 20000-1)是一种服务管理系统标准，规定了管理运营服务的设计、转换、交付和改进的要求，并支持运营过程以实现其目的。

### 6.4.12.2 输出

运行过程成功实施后，结果如下：

- a) 确定了影响系统/软件需求、架构或设计的运行约束；
- b) 运行所需的任何使能系统、服务和材料均可用；
- c) 有经过训练的、合格的操作者；
- d) 交付了满足利益相关方需求的系统产品服务；
- e) 运行过程中监控了系统产品的性能；
- f) 为客户提供支持。

### 6.4.12.3 活动和任务

项目应根据与运行过程有关的适用组织方针和规程实施下列活动和任务。

a) 准备运行。该活动由以下任务组成。

1) 制定运行策略，包括以下几点考虑：

- i) 服务的预期或商定的容量、可用性、响应时间和信息安全，包括服务的引入、常规操作和退出；
- ii) 人力资源策略，根据需要定义培训和资格要求，培训或获取人员以控制和监控软件系统运行，管理系统访问，支持客户服务请求和用户协助；
- iii) 软件系统的发布准则和进度计划，以允许对现有或增强的服务进行修改；
- iv) 在运作概念中实施运行模式的方法，包括正常运行以及预想的应急行动类型的准备和测试；

- v) 能够洞察绩效水平的运行措施；
- vi) 操作者和其他在运行过程中使用或接触软件系统的人员的操作和职业安全策略，包括安全规定；
- vii) 软件系统运行的环境保护和可持续发展策略。

注：ISO/IEC 16350信息技术-系统和软件工程-应用管理操作指南。

2) 从系统/软件需求、架构、设计、实现或转换的变更中识别系统约束。

3) 识别和规划支持运行所需的必要的使能系统或服务。

注：这包括对使能系统的运行需求和接口的识别。运行软件系统的特殊模式，如有时一种训练模式可以与一种完整的运行模式一起使用，也可以代替后者。使能系统包括监视软件系统受到的威胁的变化。

4) 获得或获取对所使用的使能系统或服务的访问权限。

注：确认过程用于客观地确定运行使能系统实现其使能功能的预期用途。

5) 确定或定义软件系统运行所需人员的培训和资格要求。

注：培训和资格认证包括对软件系统在其运行环境中的认识，以及明确的熟悉程序、适当的故障检测和隔离指令。操作者的知识、技能和经验要求指导人员的选择准则，并在相关情况下确认其操作授权。资格的范围取决于所关注的系统及其环境。例如，在某些环境中，法规要求包括操作者的认证，而在其他环境中没有认证要求。

6) 根据运行过程中人为干预和控制的需要，安排经过培训的合格人员作为操作者。

注：由于考虑职责分离，如行政系统访问控制和信息安全问题的调查，许多现代软件产品最大限度地减少了操作者与最终用户的不同需求。操作者通常支持使能系统，如云服务、数据库和系统软件、信息安全监控、数据存储、和帮助台。

b) 执行操作。该活动包括以下任务。

1) 在预定的运行环境中使用软件系统。

注：在双方同意的情况下，当软件系统取代正在退役的现有系统或元素时，保持了持续的服务能力和质量。

2) 根据需要，将材料和其他资源应用于运行软件系统并维持其服务。

注：这包括用于硬件的能量来源、用于软件的连接以及人工或自动操作者。

3) 监控软件系统的运行，包括以下考虑：

- i) 管理对运营策略的遵守(如操作规程)；
- ii) 记录和报告重大事件，如可能违反软件和数据机密性和完整性；
- iii) 以安全的方式运行软件系统，并符合法规指引，例如那些有关职业安全与环境保护的；
- iv) 当软件系统或服务性能不符合可接受的参数时进行记录。

注：这包括由于操作策略、操作使能系统、操作执行或不正确的软件系统定义而导致的异常。当在硬件中实现的系统元素降低或超过了它们的使用寿命，或者系统的操作环境影响软件操作时，系统有时会表现出不可接受的性能。超过容量阈值的工作负载、竞争应用程序的利用率、安全漏洞或软件缺陷。

4) 与运行策略保持一致，开发并在可行的情况下自动化运行规程，以最大限度降低运行异常的风险。

注：这包括处理日常(预先批准的)变更请求和服务请求、故障排除和事件报告的程序，特别是信息安全事件。

5) 根据运行策略，分析测量数据，确认：

- i) 服务表现在议定工作量的可接受参数或议定服务水平内；
- ii) 系统和服务的可用性和响应时间是可接受的；
- iii) 运行成本符合目标和约束条件；
- iv) 识别潜在的改进并按优先级排列。

注：操作员的反馈和建议通常是改善软件系统运行性能的有效输入。可以采用质量保证和测量过程。

6) 必要时进行应急操作。

注：这包括以降级模式运行软件系统、执行退出和恢复操作、系统关闭、执行恢复操作的应变规程或其他特殊情况下的模式。必要时，操作者执行必要的步骤以进入应急操作并可能关闭系统。应急操作是按照预先制定的此类事件的规程进行的。这些规程通常伴随着一个连续性计划。

c) 运营结果管理。该活动由以下任务组成：

1) 记录运行结果和遇到的异常；

注：项目评估与控制和质量保证过程用于分析偶发事件和问题数据，以确定根本原因，采取纠正或改进措施，并记录所吸取的教训。

2) 记录运行偶发事件和问题，并跟踪解决；

注1:执行偶发事件和问题解决是通过质量保证过程和项目评估与控制过程来处理的。需求、架构、设计或软件系统元素的更改通过使用其他技术过程组中的过程来完成的。

注2:如果在运行过程中发生事故，操作者记录事故(或收到自动通知),并执行经验证操作规程中规定的行动以恢复正常运行。有些规程允许提供临时的变通解决方案，直到可以执行根本原因分析为止。

注3:在软件运行期间，问题发生的条件通常被记录下来，这与维护或恢复运行可用性是一致的，这样，如果可能的话，问题可以在测试环境中复现，并确定根本原因。问题解决通常涉及确定问题的严重性和影响，以及是否或何时将问题作为已知错误进行纠正或接受一段时间。

3) 保持运行服务和配置项的可追溯性；

注：在运行服务和业务或任务需求、运行概念、运营观念和利益相关方需求之间保持双向可追溯性。运行配置项可以追溯到发布的版本，并通过PCA或FCA进行确认。

4) 提供为基线所选择的关键制品和信息项。

注：这个过程识别基线的候选对象，而信息管理过程控制信息项，例如关于运行服务性能的报告。用于运行的关键制品(信息项)列在附录B中。

d) 提供顾客支持。该活动由以下任务组成：

1) 协助客户和用户解决投诉、事件、问题和服务请求；

注1:协助和咨询包括提供或推荐培训、文档、漏洞解决、防伪活动，以及支持有效使用软件系统的其他服务来源。

注2:客户支持包括与服务顾客、用户和其他利益相关方的沟通，以接收服务请求和变更请求，解决投诉，并提供解决偶发事件和问题的信息。

2) 记录并监控用于支持的请求和后续行动；

3) 确定所交付的软件系统或服务满足顾客和用户需求的程度。

注：分析结果并确定恢复或修改软件系统或服务以提供持续的顾客满意度和软件系统可用性所需的行动。在可能的情况下，与利益相关方或其代表就此类行动的利益达成一致。顾客满意度数据也作为质量管理过程的输入。

## 6.4.13 维护过程

### 6.4.13.1 目的

维护过程的目的是维持系统提供服务的能力。

该过程监控系统提供服务的能力，记录偶发事件以供分析，采取纠正、适应、完善和预防措施，并确认恢复的能力。

对于软件系统，维护过程对已部署的软件系统和元素进行修正、更改和改进。软件系统维护方法根据免费提供的系统、广泛的商业分布系统或在少数受控环境中运行的系统的不同而有所区分。

除了潜在的系统缺陷之外，软件系统维护的需要还可以由多种原因引起，例如对接口系统或基础设施的更改、不断发展的信息安全威胁、系统元素的技术过时以及跨系统生存周期的使能系统。通常，功能的扩展、中期的升级或遗留系统演进成为一个新的软件系统开发项目，该项目将应用集合在适当的生存周期内的过程。如果是这样的话，项目群管理过程就是开始工作的起点。在其他情况下，软件系统维

护是作为一系列连续的优先工作项来执行的，可能是在工作的基础上。软件系统元素的维护可以包括硬件、软件和服务，例如通信或web 服务。维护与配置管理过程和软件资产管理紧密相连，并与其他技术过程组中的过程同时执行。

注：ISO/IEC/IEEE 14764:2006和ISO/IEC 16350提供了补充的详细内容。SWEBOK,软件工程知识体系指南，软件维护知识领域，讨论了软件维护的基础、关键问题、测量、技术、维护过程和支持活动以及工具。指南还讨论了支持软件可靠性的模型、技术和措施。

#### 6.4.13.2 输出

维护过程成功实施后，结果如下：

- a) 确定了影响系统需求、架构或设计的维护约束；
- b) 维护所需的任何使能系统或服务均可用；
- c) 系统元素的备件、维修件和升级件可用；
- d) 报告了纠正性、完善性或适应性维护的变更需要；
- e) 确定故障和生存期数据，包括相关成本。

#### 6.4.13.3 活动和任务

项目应根据与维护过程有关的适用组织方针和规程实施下列活动和任务。

- a) 准备维护。该活动由以下任务组成。

- 1) 确定维护策略，包括考虑以下几点：

- i) 根据运行可用性需求，建立执行、验证、分发和安装软件维护变更的优先级、典型的进度计划和规程；
- ii) 建立技术和方法，以了解纠正性、适应性和完善性维护的需要；
- iii) 在软件系统及其体系架构发展的情况下，定期评估设计特性；
- iv) 利用有关系统的技术变化资料，预测部件和技术可能过时的情况；
- v) 建立优先级和资源以获得正确版本的产品和执行维护所需的产品信息(如定期或分阶段安装、维护补丁或软件升级)；
- vi) 维护措施，提供对性能级别、有效性和效率的洞察，包括对历史故障和事故的访问；
- vii) 在问题解决和维护期间，对数据的约定权利和对系统数据的影响；
- viii) 确保系统中不引入伪造或未经授权的系统元素的方法；
- ix) 维护变更对其他软件系统元素的影响，与保留报告的软件正处于异常的风险相对比；
- x) 考虑到健康、安全、信息安全和环境方面的法律法规要求，实施系统或软件修补或替换、修复、补丁、更新和升级所需的技能和人员水平。

- 2) 对于非软件元素，定义一个贯穿整个生存周期的后勤策略，包括获取和运行方面的考虑：要存储的替换元素的数量和类型，以及它们的存储位置和条件，它们的预期替换率，以及它们的存储寿命和更新频率。

注：可支持性影响在概念探索或开发阶段就被考虑到了。在整个部署和维持阶段，后勤有助于确保在适当的地点和时间提供适当数量和质量的必要材料和资源。

- 3) 确定系统/软件需求、架构或设计中包含的维护约束。

注：这些通常是由于需要1)重用现有的维护和验证使能系统；2)重用现有的可替换系统元素，并适应再供应的限制；3)在特定地点或环境进行维护。例如，强调封装、模块化和可伸缩性的软件架构和设计可更易维护。当需要维护时，记录系统设计和构建的需求可以减少对系统和元素进行逆向工程所需的工作。系统架构和设计反映了在问题解决期间回滚、备份和恢复数据的需要。使系统可用于远程诊断和维护的功能可以合并到架构和设计中。

4) 确定交易，以使系统和相关的维护和后勤行动产生一个可负担的、可操作的、可支持的和可持续的解决方案。

注：系统分析和决策管理过程用于执行评估和交易决策，

5) 识别和规划支持维护所需的必要使能系统或服务。

注：这包括对使能系统的需求和接口的标识。选择支持维护的系统通常反映了在初始系统实现期间重新使用现有的或等效的设计、开发和配置管理基础设施的需要。

6) 获得或获取对所使用的使能系统或服务的访问权限。

注：确认过程用于客观地确定维护使能系统实现了其使能功能的预期用途。

b) 执行维护。该活动由以下任务组成：

1) 评审利益相关方需求、投诉、事件、偶发事件和问题报告，以确定正确性、适应性、完善性和预防性的维护需求；

注：对于具有迭代生存周期的软件系统，变更需求可以被看作是适应性和完善性的维护活动的来源。对于软件维护，此过程对已部署的软件进行修正、更改和改进，并进行补丁和更新以维护系统信息安全。

2) 分析维护变更对数据结构、数据和相关软件功能、用户文档和接口的影响；

注：评审和分析往往包括维修行动的类别等因素；修改的尺寸、成本、时间修改，以及对性能、安全或信息安全的影响。

3) 遇到导致软件系统故障的意外故障时，将系统恢复到运行状态；

注：恢复到完全或降级的运行状态通常可以通过回滚、工作区或识别和纠正故障原因来完成。如果完全恢复延迟或不可能，系统将恢复到降级模式，这与应急计划一致。如果可能，使用类似于运行环境的不同环境复现错误并识别故障的根源。配置管理过程，特别是发布管理活动，被调用来控制系统的计划和紧急变更。

4) 执行缺陷（缺点）和错误纠正或系统元元素更换或升级的规程；

注1：缺陷和错误的纠正使用问题解决方案，可以通过质量保证过程和项目评估与控制过程来处理。

注2：通常，回归测试被执行来验证维护变更没有引入其他问题，即在不影响原始的、未修改需求性能的情况下，完整、正确地实施新的和修改后的需求。移交过程可应用于部署主要的维护变更；次要的修复通常作为维护过程的一部分处理。记录行动，以便于将来的维护和问题解决，并对可降级系统元素进行后勤分析。

注3：系统和数据恢复程序以及维护信息通常在介质上可用，这在执行维护的时点是适用的。

5) 通过更换、打补丁、扩充或升级软件系统元素来进行预防性维护，以提高预计将达到不可接受服务水平的软件系统的性能（如由于需求或存储数据的增加导致的能力不足），或为了避免不可接受的运行条件（如使用过时的信息安全软件）；

6) 确定何时需要进行适应性或完善性维护；

注：适应性和完善性维护行动通常涉及系统/软件需求、架构和设计的变更。可以启动一个新项目来修改现有的软件系统。

c) 提供物流支持。该活动由以下任务组成：

注：后勤行动使软件系统保持准备就绪。这些行动包括人员配备、供应支持、支持设备、技术数据需求（用户文件）和商定的数据权利、培训支持、通信、设备/计算资源支持，以及设施。

1) 获取资源，通过软件系统的生存周期或项目的生存周期（获取物流）来支持软件系统；

注：获取后勤方面的考虑包含在协定过程组产生的协议中。这包括执行分析，以确定系统初始设计的成本效益变化，以支持和易于维护，以及在使用/部署期间分配软件修复和升级的安排。这些决策常常受到可用性需求的约束，并影响供应链管理。

2) 监控替换元件和使能系统的质量和可用性，它们的交付机制和在存储期间的持续完整性；

注：运行后勤涉及在整个运行周期内所关注的系统和使能系统的同时调整，以帮助确保有效和高效地交付软件功能。它还包括技术资源的可用性。如，可靠的使能系统可以读取存储在以前媒介格式中的软件，或者将备份文件迁移到当前媒介和当前维护的使能系统。



3) 实施软件系统或元素分配机制，包括生存周期过程中项目所需的包装、处理、存储、通信或运输；

注1:软件分发和安装通常是自动化的。软件包通常包括软件许可条款(包括数据权利)和软件资产管理元素。为了支持集成和移交过程的目标，常常需要对其他系统元素进行后勤规划。

注2:考虑需要将软件的备用元素或备份副本存储在现场或其他位置，以维护所需的软件系统功能(对于应急操作可能需要降低级别)。

4) 确认后勤活动以满足软件系统或元素的支持要求或运行就绪准备实现了规划和实施。

注: 这些后勤行动包括人员配备、供应支持、支持设备、技术数据需求(用户文档、说明、清单),培训支持、通信、设备/计算资源支持,以及设施。

d) 维护和后勤结果管理。该活动由以下任务组成：

1) 记录偶发事件和问题，包括它们的解决方案，以及重要的维护和后勤结果；

注：这包括由于维护策略、维护使能系统、维护和后勤的执行或不正确的系统定义而导致的异常。项目评估与控制和质量保证过程用于执行维护问题的识别和解决，例如，分析数据以确定根本原因，采取纠正或改进措施，并记录所吸取的教训，这个活动可以包括对后勤或软件分发过程的变更。软件系统需求、架构或设计的变更是在其他技术过程组过程中完成的。

2) 识别和记录偶发事件、问题、维护和后勤行动的趋势；

注1:趋势数据和问题解决报告用于通知那些正在创建或使用类似系统实体项目的运行和维护人员、顾客，以及其他利益相关方

注2:偶发事件和问题报告，包括所采取的行动，通过质量保证过程的偶发事件和过程管理活动进行跟踪。

3) 保持正在被维护的系统元素的可追溯性；

注: 在记录的维护行动与软件系统元素和生存周期制品之间保持双向的可追溯性。软件资产管理的变化，例如软件许可证分配给替换系统，都应被记录下来。

4) 提供为基线所选择的关键制品和信息项；

注: 配置管理过程用于建立和维护配置项和基线，以及跟踪许可和数据权限。这个过程识别基线的候选对象，而信息管理过程控制信息项，例如维护过程。

5) 监控和测量顾客对系统和维护支持的满意度。

注: GB/T 19014-2019包含顾客满意度监控和度量的指南。当收集到顾客满意度数据后，将其用于质量管理过程。

## 6.4.14 处置过程

### 6.4.14.1 目的

处置过程的目的是终止一个或多个系统元素的存在，以满足指定的预期用途，适当处理替换或退役的元素，并妥善处理已确定的关键处置需求(如，根据协议、根据组织方针，或关于环境、法律、安全、信息安全方面)。

此过程使系统或其任何元素失效、反汇编，并从特定用途中移除。它处理废止项，把它们置于最终状态，并将环境恢复到原来的或可接受的状态。在生存周期的任何阶段，废弃产品都可能在生产过程中产生，例如：制造过程中的废料。按照法规、协议、组织约束和利益相关方的要求，以无害环境的方式销毁、储存或回收系统元素和废弃产品。处置包括防止过期、不可重复使用或不充分的元素重新进入供应链。如有需要，该处置会备存记录，以监控操作者和使用者的健康，以及环境的安全。当系统的一部分将以修改后的形式继续使用时，处置过程有助于确保正确处理了已退役的部分。

软件系统的处理包括服务的终止和软件元素、存储数据、介质和固件、信息项和相关硬件元素的处理，这些硬件元素不会被重用或移交到另一个系统。处置过程旨在适用于软件系统生存周期的任何阶段。对于软件，处置过程适用于整个生存周期的源代码或可执行文件软件的拷贝，在软件系统中使用的个人可识别的或受控制的数据，以及在集中配置控制下保留或分发使用的相关信息项，如，在早期生存



周期阶段处置原型，以及在使用/部署和支持阶段替换来自修改的退役元素。当系统参数因技术或能力升级而被修改时，只有受影响的元素被停用和移除。

注：业务或使命分析过程和决策管理过程通常用于处理系统和新系统功能对利益相关方潜在的影响。

#### 6.4.14.2 输出

处置过程成功实施后，结果如下：

- a) 提供了作为需求、架构、设计和实施输入的处置约束；
- b) 处置所需的任何使能系统或服务均可用；
- c) 根据需求(安全 and 信息安全需求)对系统元素或废弃产品进行了销毁、储存、回收或循环利用；
- d) 环境恢复到原始或约定的状态；
- e) 具有处理行动和分析的记录。

#### 6.4.14.3 活动和任务

项目应根据与处置过程有关的适用组织方针和规程实施下列活动和任务。

- a) 准备处置。该活动由以下任务组成。
  - 1) 定义软件系统的处置策略，以包括每个系统要素，以及识别和处理关键的处置需求，包括以下考虑：
    - i) 系统功能与服务交付的永久终止，如，数据存储设备的物理破坏，或软件系统元素的移交，以便将来在修改或改编的形式中重用；
    - ii) 识别软件系统中保留或销毁数据和知识产权的所有权和责任；
    - iii) 将产品转变为或保持在社会和物理上可接受的状态，从而避免对利益相关方、社会和环境后续的不利影响；
    - iv) 处置行动所涉及的健康、安全、信息安全和隐私问题，以及由此产生的物理材料和信息的长期状况；
    - v) 向利益相关方通报重大处置活动，如，系统、软件产品或服务的退役或更换、退役时间表或替代选项；
    - vi) 确定处置活动的进度计划、行动、职责和资源。
  - 2) 确定系统/软件需求、架构和设计特性或实现技术的处置约束。
 

注：这包括访问和可用的档案或长期存储位置和可用的技术资源，用于系统停用及利益相关方与接口合作伙伴的沟通。
  - 3) 确定和规划必要的使能系统或服务，以支持处置。
 

注：这包括对使能系统的需求和接口的标识。
  - 4) 获取或获取对所使用的使能系统或服务的访问权限。
 

注：确认过程用于客观地确定处置使能系统实现了其使能功能的预定用途。
  - 5) 如果要存储软件系统或数据，应在符合信息安全和环境考虑的前提下，规定封隔设施、存储位置、检验准则和存储期限。
  - 6) 定义预防方法，以防止不应被重新利用、回收或重用的已处置元素和材料重新进入供应链。
- b) 执行处置。该活动由以下任务组成：
  - 1) 将软件系统或元素停用，以备移除；
 

注：根据特殊规程或说明，以及相关的健康、安全、信息安全和隐私约束，考虑与其他系统的接口。
  - 2) 从使用或生产中移除软件系统及其元素、数据和不可重复使用的材料，以便进行适当的配置和操作；

注：处置包括再使用、回收、翻新、大修或销毁。处置和随后的行动是根据相关的安全、信息安全、隐私和环境标准、指令和法律进行的。软件系统中剩余的有使用寿命的元素，无论是在当前状态下还是在修改之后，都会被转移到其他所关注的系统或组织中。在适当的情况下，考虑翻新系统元素以延长其使用寿命。

3) 将受影响的操作人员从软件系统或系统元素中撤出，并记录相关操作知识；

注：重新分配、重新部署或退役操作者。这是根据相关的安全、信息安全、隐私和环境标准、指令和法律进行的。运用知识管理过程，保护和保障操作者的知识和技能。

4) 重用、回收、修复、检修、存档或销毁指定的软件系统元素；

注：处理系统元素及其部件，这些元素和部件不以确保它们不会返回到供应链的方式进行重用为意图。

5) 对系统元素进行必要的破坏，以减少废物处理量或使废物更容易处理。

注：当元素不可维护或不可回收时，有必要防止元素重新进入供应链，如，从所有系统存储介质中彻底清除所有软件，并删除许可证密钥、数据和接口。这一活动包括获得必要的熔化、粉碎、焚化、拆除或根除系统或其要素的销毁服务。

c) 确定处理。该活动由以下任务组成：

1) 确认处置后的有害健康、安全、信息安全和环境条件已得到识别和处理；

2) 将环境恢复到原始状态或协议规定的状态；

3) 对产品生存周期内收集的信息进行存档，以便在对健康、安全、信息安全和环境造成长期危害时进行审计和评审，并允许未来的软件系统创建者和用户根据经验建立知识库。

附录 A  
(规范性)  
剪裁过程

A.1 导引

本附录规定了剪裁本文件的要求。

注1:对于本文件的符合性来说,剪裁并非强制性要求。若声明“完全符合”,即不准许剪裁。只有声明“经剪裁的符合”,才应按本过程执行剪裁。

注2:剪裁的补充指南参见ISO/IEC/IEEE 24748(所有部分)指南中关于生存周期过程应用的相关内容。

A.2 剪裁过程

A.2.1 目的

剪裁过程的目的是满足以下特殊的情况或者因素而对本文件中的过程进行调整:

- a) 在协议中围绕采用本文件的组织;
- b) 影响某个项目,该项目要求满足参考本文件的某个协定;
- c) 体现组织的需要,以便供应产品或服务。

A.2.2 输出

剪裁过程成功实施后,结果如下:

为实现一个生存周期模型的目的和输出,定义了修改过的或全新的生存周期过程。

A.2.3 活动和任务

如果本文件被剪裁,那么组织或项目应依照与剪裁过程有关的组织方针与规程实施下列任务。

- a) 识别和记录影响剪裁的条件,包括但不限于:
  - 1) 运行环境的稳定性和多样性;
  - 2) 各利益相关方所关心的商业或性能风险;
  - 3) 新颖性、规模和复杂性;
  - 4) 开始使用日期和使用期限;
  - 5) 诸如安全、信息安全、隐私、易用性、可用性等的完整性问题;
  - 6) 新兴的技术机遇;
  - 7) 预算概况和组织内的可用资源;
  - 8) 使能系统的服务的可用性;
  - 9) 在系统的全生存周期中的角色、职责、责任和权限;
  - 10) 符合其他标准的要求。
- b) 对系统的关键属性,要根据其关键程度来考虑相关的标准或建议/强制要求的生存周期结构。
- c) 从受剪裁决策影响的当事各方获得输入,包括但不限于:
  - 1) 系统的利益相关方;
  - 2) 与组织达成的协定有关的各利益相关方;
  - 3) 有贡献的组织内部职能部门。
- d) 应用决策管理过程做出剪裁决策,获得所选定的生存周期模型的目的和输出。

注1:组织建立的标准生存周期模型是生存周期模型管理过程的一部分。为了实现建立的生存周期模型的目的和

输出，组织有时可剪裁所采纳的本文件的过程。

注2:项目根据组织选定所建立的生存周期模型是项目规划过程的一部分。为了实现所选定的生存周期模型的阶段目的和输出，有时可以剪裁组织所采纳的过程。

注3:如果项目直接应用本文件，为了实现适当生存周期模型的目的和输出，有时可以剪裁本文件所采纳的过程。

e) 选择需要剪裁的生存周期过程，删除选定的输出、活动或任务。

注1:不论剪裁与否，如何剪裁，始终允许组织和项目所实施的相关过程达到超出本文件要求的更多其他输出，或者实施超出本文件要求的更多其他活动和任务。

注2:某些组织或项目有时会遇到需要修改执行本文件条款的情况。考虑到可能对其他过程、输出、活动或任务带来不可预期的后果，应当尽可能避免条款的修改。如果确有必要，可以通过以下方式进行修改：删除此条款（针对剪裁符合性做出适当声明），并仔细考虑后果，实施的过程达到超出剪裁后要求的更多其他输出，或者实施超出剪裁后要求的更多其他活动和任务。

附录 B  
(资料性)  
过程信息项示例

表 B.1 提供了一组可能的工作产品，包括与每个过程相关联的制品、记录、信息项和数据存储。这个表并非包含全部：对于每个过程，组织可能决定制定一个策略、计划、规程、报告和记录，以便呈现执行活动和任务的输出。如果认为不太密集的文档已经足够，则可以合并信息项。组织的策略和过程也可以应用到每个过程和项目中。列出典型项的标题，并在括号中用常见例子加以解释。

注：关于信息项的内容和管理指南，请参阅ISO/IEC/IEEE 15289。

制品、记录、记录存储和信息项通常在一个过程中启动，并在其他过程中修改、增强或完成。为了方便起见，通常在刚开始出现的那个过程中进行初始化，并列入表格。当软件系统制品和信息在另一个过程转换或阐述时，可追溯性就得到了维护，并且可以产生一个可追溯性映射。例如，可以维护组织和项目过程之间，以及需求、架构和设计元素之间，以及验证用例之间的可追溯性。

表 B.1 过程信息项示例

过程组	过程	典型项的标题	项的类型
协定过程组	获取过程	获取计划	信息
		供应请求(如提案请求、投标请求等)	信息
		协定变更请求	信息
		协定(如合同)	信息
		协定变更管理规程	信息
		交付验收报告	信息
	供应过程	供应响应(如提案、投标等)	信息
		协定变更请求	信息
		协定变更管理规程	信息
		供应交付记录(用于系统、软件、产品或服务)	记录
组织的项目使能过程组	生存周期模型管理过程	组织方针	信息
		生存周期过程	制品
		生存周期过程描述	信息
		生存周期模型	制品
		组织的规程(过程管理规程)	信息
		过程评估报告	信息
		过程改进计划	信息
		过程改进报告	信息
	基础设施管理过程	基础设施要求	制品
		基础设施元素	制品
		基础设施描述	信息
		基础设施变更请求	信息

表B.1 过程信息项示例(续)

过程组	过程	典型项的标题	项的类型
组织的项目使能过程组	特定项目包管理过程	特定项目包	存储
		项目预算	制品
		项目授权	信息
	人力资源管理过程	技能需求报告	信息
		技能开发资产(培训材料)	制品
		技能开发记录(技能目录, 培训记录)	记录
		合格人员	制品
		员工分配记录	记录
	质量管理过程	质量管理计划(方针、目标)	信息
		质量管理规程	信息
		质量保证评估结果	记录
		纠正/预防措施报告(问题管理报告)	信息
	知识管理过程	知识管理计划	信息
		知识管理规程	信息
		知识资产记录	记录
		知识资产	制品
技术管理过程组	项目规划过程	项目计划(如项目技术管理计划, 系统或软件工程管理计划, 软件开发计划, 移交计划)	信息
		工作分解结构	制品
		资源请求	信息
		项目进度	制品
		项目基础设施及服务要求	制品
	项目评估与控制过程	测量分析结果与建议	信息
		项目评估报告	信息
		评审会议记录	信息
		前往下一个里程碑的授权	信息
	决策管理过程	决策请求	信息
		决策记录	记录
	风险管理过程	风险管理计划	信息
		风险概述	记录
		风险措施请求	信息
	配置管理过程	配置管理计划	信息
		配置管理规程	信息
		配置管理记录	记录

表 B.1 过程信息项示例(续)

过程组	过程	典型项的标题	项的类型
技术管理 过程组	配置管理 过程	配置管理基线	制品
		配置管理变更/偏差请求	信息
		配置状态报告	信息
		配置评价报告	信息
		系统/软件发布报告	信息
	信息管理 过程	信息项档案	存储
		信息管理规程	信息
		信息管理报告	信息
	测量过程	测量记录	记录
		测量规程	信息
		测量信息需要报告	信息
		测量报告	信息
	质量保证 过程	质量保证规程	信息
		质量保证评价报告	信息
		质量保证记录	记录
		偶发事件记录	记录
		问题记录	记录
技术过程组	业务或使命 分析过程	初始生存周期概念	制品
		解决方案备选类别评估报告	信息
	利益相关 方需要和 需求定义 过程	运行概念	制品
		利益相关方需要评估	信息
		利益相关方需求	制品
		利益相关方需求规格说明	信息
		利益相关方需求报告	信息
		关键性能测度	制品
	系统/软件 需求定义 过程	系统或元素描述	信息
		系统/软件需求	制品
		系统/软件需求规范说明	信息
		需求变更请求	信息
	架构定义 过程	架构视角	制品
		架构视图和模型(架构描述)	制品
		(初始)接口定义	制品
	设计定义 过程	设计制品	制品
		设计制品报告(设计描述)	信息
		接口规格说明	信息



表 B.1 过程信息项示例(续)

过程组	过程	典型项的标题	项的类型
技术过程组	系统分析过程	系统分析报告	信息
	实现过程	软件系统元素	制品
		实施规程	信息
		实施记录(单元测试结果)	记录
	集成过程	接口控制描述	信息
		集成和测试规程	信息
		集成的软件系统元素(软件库)	制品
		集成记录	记录
	验证过程	已验证的系统	制品
		验证规程	信息
		验证记录	记录
		验证报告	信息
	移交过程	用于运行的已准备的场地	制品
		移交的系统/软件	制品
		移交报告	记录
	确认过程	已确认的系统	制品
		确认规程	信息
		确认记录	记录
		确认报告	信息
	运行过程	连续性计划	信息
		运行规程(用户文档)	信息
		运行记录	记录
		问题报告	信息
		顾客支持请求	信息
		顾客支持记录	记录
		运行报告	信息
	维护过程	系统元素更换	制品
		维护规程(后勤规程)	信息
		维护(后勤)记录	记录
		维护请求	记录
		维护(后勤)报告	信息
	处置过程	处置记录	记录
		存档报告	信息

## 附录 C

### (资料性)

### 用于评估目的的过程参考模型

#### C.1 导引

本文件的使用者如果希望按照ISO/IEC 33000系列评估标准对实施过程进行评估，本附录提供了一个与这些标准共同使用的过程参考模型(PRM)。

PRM 由本文件正文中的过程组成，包括每个过程的名称、目的和输出。C.3 确定了过程参考模型中的过程和定义这些过程的条款。

#### C.2 与 ISO/IEC 33004的符合性

##### C.2.1 概述

ISO/IEC 33004的5.3说明了使用此标准对其进行评估的过程参考模型的要求。以下各条引用了ISO/IEC 33004的过程参考模型的要求，并描述了本文件如何满足这些要求。在以下各条中，斜体文本引用了来自ISO/IEC 33004的要求；非斜体(正常)文本描述了在本文件中该要求被满足的方式。

##### C.2.2 过程参考模型的要求

*过程参考模型应包含：[ISO/IEC 33020:2015,5.3.1]*

- a) 过程参考模型域的声明。第1章提供了此声明；
- b) 过程参考模型与其预期的应用之间关系描述。第5章提供了此描述；
- c) 在过程参考模型范围内满足参考模型在ISO/IEC 33004的5.4中要求的过程描述，此描述满足标准。第6章提供了每个过程的描述；
- d) 过程参考模型中定义的各过程之间关系的描述。5.6提供了此描述。

*过程参考模型应记录模型的利益共同体以及在利益共同体内为了达成共识所采取的行动：[ISO/IEC 33020,5.3.2]*

- a) *应定性或指定相关的利益共同体。相关的利益共同体指GB/T 22032—2021和本文件的用户；*
- b) *对达成共识的程度应记录在案。GB/T 22032—2021和本文件都是符合ISO/IEC 和IEEE共识需求的国家标准(不适用)；*
- c) 如果没有采取任何行动来达成共识，对此事的陈述应形成文件(不适用)。

在过程参考模型中定义的过程应具有唯一的过程描述和标识。[ISO/IEC 33020,5.3.3]过程描述是唯一的。此标识由唯一的名称和本附录的条款编号提供。

##### C.2.3 过程描述

*过程参考模型的基本元素是模型范围内的过程描述。*

*过程参考模型中的过程描述包含了过程目的的声明，用于在较高的层级上描述执行此过程的总体目标，以及一组能够表明此过程目标已成功实现的输出。*

*这些过程描述应满足下列需求：*

- a) 应针对过程的目的和输出进行描述；
- b) 过程的输出对达到此过程的目的而言是必要且充分的；

- c) 过程描述不应包含或隐含超出ISO/IEC 33003的任何相关过程测量框架的基本级别以上的内容。

*过程输出描述为下列内容之一[ISO/IEC 30004:2015,5.4]*

- a) 人工制品的生产；
- b) 重大状态变更；
- c) 满足特定的约束，如要求、目标等。

第6章的过程描述满足相关要求。在一些相关的过程测量框架中，相关过程输出有助于1级(基本的)以上能力等级。然而，不要求达到这些更高的等级能力来实现相关过程。

### C.3 过程参考模型

过程参考模型(PRM) 由本文件第6章中的每个过程的目的的声明和输出的声明组成。软件生存周期的PRM 由图4中的过程集合组成，如5.6.1中所述。

## 附 录 D

### (资料性)

### 过程集成和过程构建

#### D.1 导引

ISO/IEC JTC1/SC7正在进行一项协调项目\_\_\_\_并行细致的协同修订 ISO/IEC/IEEE 15288 和 ISO/IEC/IEEE 12207,并制定ISO/IEC/IEEE 24748(所有部分)指南,用来为这两个国际标准提供指导——此项目是朝着实现一套描述系统和软件生存周期的标准集合这样一个目标所迈进的第一大步。持续的过程改进和能力评估的概念现已得到很好的建立和认可,并在ISO/IEC 33000系列(取代 ISO/IEC 15504系列)标准中进行了规范。GB/T 22032—2021与本文件的附录C 中的过程参考模型与 ISO/IEC 33020:2015标准结合使用,以评估生存周期过程的能力。对过程的能力确定要求的描述中,应包含对过程目的的明确陈述和对预期结果输出的说明。定义活动、任务和实施说明有助于过程的一致性实施。因此两个生存周期标准的生存周期过程均采用了附录D.2 中定义的通用过程结构,并与 ISO/IEC/IEEE TR 24774中的过程定义指南保持一致。

#### D.2 过程结构及其用法

本文件中的过程描述遵循明确的定义规则。首先,这些过程按照一定的逻辑来分类。分类的依据是:

- a) 过程之间的逻辑关系;
- b) 过程执行的责任。

本文件将系统生存周期中可能执行的活动分为四个过程组。这些过程组的顶层描述可参见5.6。这些过程组中的每个生存周期过程均用目的、输出以及实现这些输出所需的活动与任务列表来描述。

- a) 协定过程组: 2个过程(6.1);
- b) 组织的项目使能过程组: 6个过程(6.2);
- c) 技术管理过程: 8个过程(6.3);
- d) 技术过程组: 14个过程(6.4)。

过程描述规则的一致性实施允许对条款进行规范化编号。在本文件中,编号为6.x 的子条款表示一个过程组,6.x.y 表示过程组中的一个过程;6.x.y.1 描述过程的目的,6.x.y.2 描述过程的输出,6.x.y.3 描述过程的活动和任务。

图 D.1 是本文件和GB/T 22032—2021中使用的过程结构的表示。这些是过程、活动、任务和注释。

过程需要一个名称、目的和至少一个输出。每个过程至少有一个活动。过程及其目的和输出的陈述共同构成了一个过程参考模型(PRM)。

活动是为了特定输出而定义的一组任务。活动通过将过程中的相关任务组织在一起,来改进对过程的理解和交流。如果活动具有足够的内聚性,可以通过刻画其目的和输出结果集的方式将其转化为较低层次上的一个过程。

任务是实施过程的详细描述,它可能是一个要求(“应”),建议(“宜”),或允许(“可”)。

注释用于解释过程、活动或任务的意图或机制。注释提供了更进一步的潜在实现或应用范围,如列表、示例和其他考虑。

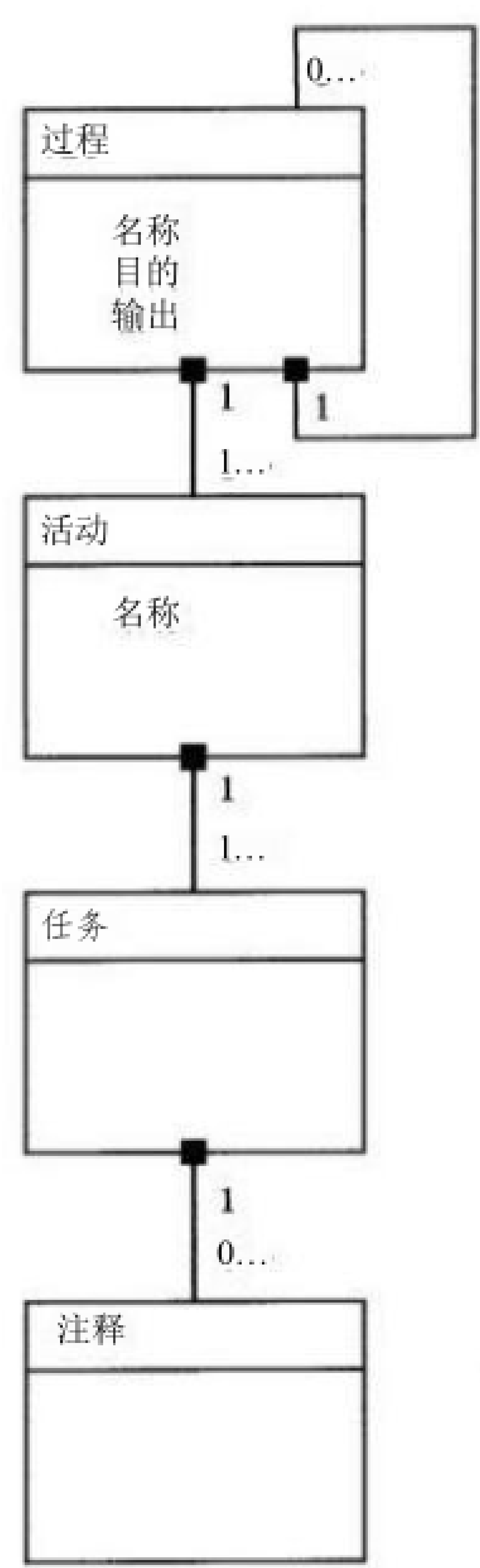


图 D.1 本文件和GB/T 22032—2021 过程构建

附录 E  
(资料性)  
过程视图

E.1 导引

某些情况下，负责某部分工程事务的人员需要能在一个地方一目了然地查阅到全部过程活动的集合。为此，可以把从 GB/T 22032—2021 或本文件中挑选出的各项过程、活动和任务整理成为过程视图，从而能直接关注跨域的整个或部分生存周期的行为方式。本附录提供了一个过程的视角，可以用于在这些实例中下定义过程视图。

E.2 过程视图的概念

某些情况下，需要把不同过程中的活动和任务收集起来，用统一的视图清晰地表达出他们贯穿整个生存周期的概念或线索。建议本文件的用户根据需要来识别和定义这些活动，即使他们无法找到相关的单个过程。

为此，本文件说明了过程视图的概念。就像过程一样，过程视图也包括了对目的和输出的描述。所不同的是，过程视图的描述不包含活动和任务，而是在描述中包含了怎样使用本文件和 GB/T 22032—2021 中各种过程的活动和任务来获得这个输出。过程视图可以使用 E.3 中描述的过程方法模板来构建。

E.3 过程方法

过程视图符合过程方法，这里提供的过程方法可用于创建过程视图。

- a) 过程方法由以下几个方面进行定义：
  - 1) 利益相关方：本文件的用户；
  - 2) 关注过程方法框架的人员：过程需要反映某一特定工程领域的利益。
- b) 完成的过程视图应包括：
  - 1) 过程视图的名称；
  - 2) 过程视图的目的；
  - 3) 过程视图的输出；
  - 4) 实施过程视图所需的过程、活动和任务的识别和描述，以及这些过程、活动和任务在其他标准中的引用来源。

注：过程方法的文档要求参见 ISO/IEC/IEEE 42010:2011 的 5.4，与此处描述是一致的。

E.4 专项工程的过程视图

本条提供了采用过程方法实现专项工程过程视图的示例，旨在演示如何在项目中通过组合本文件的过程、活动和任务，对选定特别关注的产品特性的实施过程进行集中展示。

本示例所处理的过程视图通常称为“专项工程”，包括并不限于可用、维护、可靠、安全、信息安全、人为因素和易用性等领域。本文件中这些特性需求称为“关键质量特性”。这些特性决定了产品在特定领域多大程度上满足了特定需求。与(软件系统的信息安全保证)有关的过程视图参见 E.6。

注1:这是一个过程视图的通用示例，包含了众多与专项工程相关的功能性和非功能性特性，提供了一种比较宽泛的过程视图。如果某一关键质量特性相对其他特性有更高的优先级，可以专门为这个特性创建过程视图，包括更详细的信息和需求。

注2:GB/T25000.30《系统与软件工程 系统与软件产品质量要求和评价(SQuaRE)第30部分:质量需求框架》,可在规定软件产品质量要求时参考。

注3:INCOSE系统工程手册,包含对许多专项工程领域和相关关键质量特性的描述和阐述。

#### **名称: 专项工程过程视图**

**目的:** 专项工程过程视图的目的是提供客观证据证明系统中所特别关注的某些关键质量特性已得到了满足。

#### **输出:**

专项工程过程视图的输出包括以下内容:

- a) 挑选出了需要特别关注的产品关键质量特性;
- b) 定义了实现关键质量特性的需求;
- c) 选定了这些需求的测量方法,并与期望的关键质量特性建立关联;
- d) 定义并实现了这些期望的关键质量特性的方法;
- e) 持续监控这些需求实现的程度;
- f) 明确指出这些关键质量特性得到满足的程度。

上述输出中,期望的关键质量特性有可能无法直接测量,但是可以通过其他可测量的产品或过程特性推导或者分析出来。测量可用于显示与已建立的标准的一致性。需方和供方就本次合规性验证所使用的特定标准达成一致。

#### **专项工程过程、活动和任务**

过程视图可以使用本文件中下列过程、活动和任务来完成。

- a) 项目评估与控制过程(6.3.2)对需求和关键质量特性实现的程度进行监控,并把结果通报给利益相关方和管理人员。相关的活动和任务包括6.3.2.3中的b)6)、b)7)、b)9)和b)10)。
- b) 决策管理过程(6.3.3)对照设计规范,对备选需求、架构特性和设计特性进行评估,包括关键质量特性。这些比对结果通过合适的选择模型进行排序,用于确定出一个优化方案。相关的活动和任务包括6.3.3.3中b)的全部任务,以及c)1)。
- c) 风险管理过程(6.3.4)用于识别、评估和处理系统风险,包括和满足关键质量特性相关的风险。
- d) 信息管理过程(6.3.6)用于为记录和沟通系统实现程度的信息项提供规范、开发和维护方法。应该注意的是,用于关键质量特性的信息条目有时候是专用的。这些信息条目描述的来源包括工业协会、监管机构和专业标准。
- e) 测量过程(6.3.7)用于定义所需关键质量特性对应的测量方法。
- f) 质量保证过程(6.3.8)用于识别关键质量特性的异常状况(偶发事件和问题)。
- g) 业务或使命分析过程(6.4.1)帮助确定问题空间的定义,描述求解空间的特征,包括相关的权衡空间因素和基本的生存周期概念。这个过程包括对背景和很多关键参数进行理解,如关键质量特性中的安全威胁、人身危害、人机界面、操作特性,以及系统保证等。相关的活动和任务包括6.4.1.3中的b)1)、b)2)、c)1)和d)1)。
- h) 利益相关方需要和需求定义过程(6.4.2)提供了相关特性的选择和定义,包括关键质量特性和相关的信息条目。对应的活动和文档用于对关键质量特性进行识别、优先排列、定义以及记录需求。相关的活动和任务包括6.4.2.3中的a)1)、a)2)、b)2)、b)3)、b)4)、c)1)、c)2)、d)的全部任务,以及e)2)。
- i) 系统/软件需求定义过程(6.4.3)为将要开发的特定系统提供了关键质量特性的参数规格,同时为跟踪需求实现的测量提供了选择方法。相关的活动和任务包括6.4.3.3中a)1)、b)的全部任务,以及c)2)。
- j) 架构定义过程(6.4.4)从架构的角度识别利益相关方的关注点,这些关注点通常转化为与关键质量特性相关的全生存周期阶段的期望或者约束,比如效用(例如可用性、安全性、有效性、



适用性),支持(如可修复性、报废管理),系统与环境的演进(例如适应性、伸缩性、耐受性),生产(例如可制造性、易测性),退役(如环境影响、转运能力)等。这个过程进一步解决了驱动架构决策的这些关键质量特性需求,包括针对关注点和相关特性的架构评估。相关的活动和任务包括6.4.4.3中的a)2)、a)4)、b)1)、c)2)、c)3)、c)4)、c)5)、d)1)和e)2)。

- k) 设计定义过程(6.4.5)用于确定出必要的设计特性,包括关键质量特性,比如专项特性设计准则的保证措施,以及使用这些准则进行修改设计的评估措施。相关的活动和任务包括6.4.5.3中的a)2)、b)1)、b)2)、b)3)、b)4)、b)6)和c)2)。
- l) 系统分析过程(6.4.6)通过数学分析、建模、仿真、实验,以及其他技术进行相应级别的分析,用于理解关键质量特性相关的权衡空间。这些分析结果作为输入,在决策管理过程中用于支持其他技术过程。相关的活动和任务包括6.4.6.3中a)的全部任务,以及b)的全部任务。
- m) 实现过程(6.4.7)用于记录关键质量需求得到满足的依据。相关的活动和任务包括6.4.7.3中的b)3)。
- n) 集成过程(6.4.8)用于考虑关键质量特性的集成计划,并且确保这些特性的实现得到确认和记录。相关的活动和任务包括6.4.8.3中的a)1)、b)3)和c)1)。
- o) 验证过程(6.4.9)用于实施验证策略的计划和执行,包括关键质量特性。选择的验证策略可能会增加设计约束,从而影响这些特性的实现。相关的活动和任务包括6.4.9.3中的a)1)、a)3)、b)1)、b)2)、c)1)和c)2)。
- p) 移交过程(6.4.10)用于系统在工作环境中的安装。因为一些专项属性涉及设计约束和操作约束之间的权衡,系统安装通常变得很重要,需要引起注意。相关的活动和任务包括6.4.10.3中的a)4)、b)4)、b)6)和b)7)。
- q) 确认过程(6.4.11)用于证明系统提供的服务满足了利益相关方需求,包括关键质量特性。相关的活动和任务包括6.4.11.3中的a)1)、a)3)、b)1)、b)2)、c)1)和c)2)。
- r) 运行过程(6.4.12)描述系统使用方法。确保关键质量特性得到正确实现,包括对系统运行进行监控。相关活动和任务包括6.4.12.3中的b)3)、b)4)、c)1)、c)2)、d)1)和d)2)。
- s) 维护过程(6.4.13)用于维持系统的能力,尤其是确保其持续的可用性来发挥功用,包括关键质量特性。这包括失效分析、维护任务,以及后勤任务等,用于确保系统连续运行。相关的活动和任务包括6.4.13.3中的b)的全部任务、c)的全部任务,以及d)1)和d)2)。
- t) 处置过程(6.4.14)使系统的存在终止。预期处置的内在需要可能会对系统发展带来约束。事实上,这些约束本身可能就是关键的质量特性。相关的活动和任务包括6.4.14.3中的a)2)、b)1)、b)2)和c)3)。

## E.5 接口管理的过程视图

本条提供了采用过程方法实现接口管理过程视图的示例,旨在演示如何在项目中通过组合本文件的过程、活动和任务,对选定特别关注的产品特性的实施过程进行集中展示。

这个例子是过程视图的一个特例,称为接口管理,包括但不限于界面定义、设计,以及变更管理。在本文件中,构成接口管理的任务完全包含在现有的过程中。

**注:**对于软件系统而言,与其他系统的接口是接收输入数据或返回输出数据(报告)的典型方式。与外部系统和服务的接口允许软件系统在其运行环境和使能系统中运行。GUI是用于与软件系统进行交互的专用界面。

**名称:** 接口管理过程视图

**目的:** 接口管理过程视图的目的是帮助软件系统接口进行识别、定义、设计和管理。

**输出:**

接口管理过程视图的输出包括下列内容:

- a) 识别出接口相关的业务或使命需要;

- b) 识别出接口相关的利益相关方需要；
- c) 定义了接口需求；
- d) 识别并定义了软件系统要素之间的接口；
- e) 识别并定义了软件系统与外部系统之间的接口；
- f) 持续监控这些接口需求所要实现的程度。

### **接口管理过程、活动和任务**

过程视图可以使用本文件中下列过程、活动和任务来完成。

注：INCOSE系统工程手册对接口管理有许多详细描述。

- a) 项目评估与控制过程(6.3.2)对需求包括接口实现的程度进行监控，并把结果通报给利益相关方和决策人员。相关的活动和任务包括6.3.2.3中的b)6)、b)7)、b)9)和b)10)。
- b) 决策管理过程(6.3.3)对照设计规范，对备选需求、架构特性和设计特性进行评估，包括接口。这些比对结果通过合适的选择模型进行排序，用于确定出一个优化方案。相关的活动和任务包括6.3.3.3中b)的全部任务，以及c)1)。
- c) 风险管理过程(6.3.4)总体上用于识别、评估并处理系统风险，包括那些接口相关的风险。
- d) 配置管理过程(6.3.5)提供了接口的标识和控制，包括接口规范和接口控制文档的管理。内部和外部的接口需求和变更是根据项目的配置管理策略(通常在配置管理计划中表示)编制的。相关活动包括6.3.5中的b)1)和d)1)。
- e) 信息管理过程(6.3.6)总体上用于为记录接口运行性能的信息项提供规范、开发和维护方法。
- f) 测量过程(6.3.7)总体上用于定义所需接口信息需要对应的测量方法，随后制定出并使用这些方法来满足接口信息需要。
- g) 质量保证过程(6.3.8)总体上用于识别接口需求实现对应的异常状况(事件和问题)。
- h) 业务和使命分析过程(6.4.1)提供了问题空间的定义和求解空间的特性，包括环境和背景的描述，以及基本操作概念，包括软件系统接口和使能系统接口。通常会识别出与SOI有接口交互的外部系统。相关的活动和任务包括6.4.1.3中的b)1)、b)2)和c)1)。
- i) 利益相关方需要和需求定义过程(6.4.2)提供了与接口相关的利益相关方的识别、运营观念的定义，系统和用户、要转换的现有接口、给定环境(包括其他系统)之间的交互。通常会识别出与SOI有接口交互的外部系统。相关的活动和任务包括6.4.2.3中的c)1)、c)2)、d)1)和d)3)。
- j) 系统/软件需求定义过程(6.4.3)提供了系统边界和接口需求的定义。相关的活动和任务包括6.4.3.3中的a)1)、b)3)、b)4)、c)的全部任务，以及d)1)和d)3)。
- k) 架构定义过程(6.4.4)参考架构模型演进，从架构的角度识别系统接口。这个过程按照架构描述需要，进一步描述和定义系统接口。相关的活动和任务包括6.4.4.3中的a)2)、c)1)~c)4)、d)2)、f)2)~f)6)。
- D) 设计定义过程(6.4.5)提供了接口的细化和完整定义，以及用于规定接口特性和协议的必要的信息项的创建。相关的活动和任务包括6.4.5.3中的b)2)、b)5)、b)6)、c)3)~d)2)。
- m) 系统分析过程(6.4.6)通过建模、仿真、实验，以及其他技术进行相应级别的分析，用于在接口架构、设计限制、接口性能要求以及运行性能测量之间进行权衡。这些分析结果作为输入，在决策管理过程用于支持其他技术过程。相关的活动和任务包括6.4.6.3中a)的全部任务，以及b)的全部任务。
- n) 实现过程(6.4.7)用于接口开发，以及对某个已完成的系统要素，记录其接口需求得到满足的依据。相关的活动和任务包括6.4.7.3中的b)1)和b)3)。
- o) 集成过程(6.4.8)用于考虑集成计划，包括软件系统元素和外部系统接口的考虑。也包括系统或系统要素和接口的集成，以及确定集成软件系统中的接口。相关的活动和任务包括6.4.8.3中的a)1)、a)2)、a)5)、b)的全部任务，以及c)1)。

- p) 验证过程(6.4.9)用于证明系统提供的服务满足了系统需求,包括接口需求。这个过程提供了实施验证策略的计划和执行,包括接口需求。选择的验证策略可能会引入接口约束,从而影响他们的实现。相关的活动和任务包括6.4.9.3中的a)1)、a)3)、b)1)、b)2)和c)1)。
- q) 移交过程(6.4.10)规定了使用接口将软件系统或元素和数据集规划和迁移到不同的环境中,并在新系统中建立接口的使用。其中包括识别约束,检查接口的安装、激活和运行状态。相关活动和任务包括6.4.10.3中的a)1)、a)3)、b)3)、b)4)、b)6)和b)7)。
- r) 确认过程(6.4.11)用于证明系统提供的服务满足了利益相关方需求,包括接口需求。选择的确认策略可能引入接口约束,从而影响他们的实现。确认包括与接口系统和服务的利益相关方代表进行通信。相关活动和任务包括6.4.11.3中的a)4)、b)1)、b)2)、c)1)和c)2)。
- s) 运行过程(6.4.12)描述软件系统使用方法。可能出现对接口操作带来约束的情况。确认接口需求恰当地实现了,包括监视系统的运行,识别并执行接口不能正常工作时的纠正措施,以及支持与接口合作伙伴(客户)的联系。相关活动和任务包括6.4.12.3中的a)1)、a)2)、b)1)、b)3)、b)4)、c)1)、c)2)、d)的全部任务。
- t) 维护过程(6.4.13)用于维持软件系统的能力,包括系统接口,帮助确保其持续的可用性来发挥功用。这包括问题分析和维护任务,以及确保持续和及时运行所需的后勤任务。相关活动和任务包括6.4.13.3中a)2)、b)的所有任务、d)1)、d)2)和d)3)。
- u) 处置过程(6.4.14)使系统或接口终止存在。它涉及分离和中断接口的活动。相关活动和任务包括6.4.14.3中的a)2)、a)3)、b)1)、b)2)和b)6)。

## E.6 软件保证(信息安全)的过程视图

本条提供了采用过程方法实现软件保证过程视图的示例,旨在演示如何在项目中通过组合本文件的过程、活动和任务,对选定特别关注的软件保证特性的实施过程进行集中展示。如 ISO/IEC/IEEE 15026所述,软件保证特性、实现程度和相关信息可能支持软件保证声明。

此示例主要关注防止由于软件架构、设计或实现(特别是代码的构造)而导致的故意破坏或强制失效。

### 名称: 软件保证过程视图

**目的:** 软件保证过程视图的目的是提供客观的证据,证明软件达到了令人满意的确定性水平,从而对由于软件架构、设计或代码的构造而导致的故意破坏或强制失效提供了足够的保护。

注: 以符合ISO/IEC/IEEE 15026标准的方式,对于所选择需特别注意的软件特性的保证声明已经实现,并且提供了显示这些声明实现的信息。

### 输出:

软件保证过程视图的输出包括下列内容:

- a) 挑选出了需要特别关注的产品软件保证特性;
- b) 定义了实现关键软件保证特性的需求;
- c) 选定了这些需求的评估方法,并与期望的软件保证特性建立关联;
- d) 定义并实现了这些期望的软件保证特性的方法;
- e) 持续监控这些需求实现的程度;
- f) 规定并实现了关键软件保证特性的令人满意的水平。

上述输出中,期望的软件保证特性或声明有可能无法直接测量,但是可以通过其他可测量的产品或过程特性推导出来。测量可用于显示与已建立的标准的一致性。需方和供方就本次合规性验证所使用的特定标准达成一致。

### 软件保证过程、活动和任务

过程视图可以使用本文件中下列过程、活动和任务来完成。

- a) 协定过程(6.1)提供了软件保证相关的期望和责任保证,包括法律协议和许可要求、保护供方可访问的组织资产、在移交过程中妥协的可能性、异常检测、采用的应用元素仿冒品的检测、对缺陷解决的期望、补丁管理、服务水平的协议,以及防范供应链威胁的措施。相关活动和任务包括获取过程的6.1.1.3中的a)1)、a)2)、c)1),以及供应过程6.1.2.3中的c)1)、e)1)和e)2)。
  - b) 生存周期模型管理过程(6.2.1)规定了软件保证策略和程序的建立和维护,包括软件安全开发生存周期和对外包代码的受控使用。相关活动和任务包括6.2.1.3中a)2)、a)3)、b)1)和c)的全部任务。
  - c) 基础设施管理过程(6.2.2)从整体上提供了安全的开发和运行环境、软件保证工具、及时的补丁管理,以及根据从项目、组织和行业中获得的经验教训进行了适当维护和改进的代码库。
  - d) 人力资源管理过程(6.2.4)从整体上提供了对能够使用软件或软件开发环境的员工和供方的相关筛选,并基于整个生存周期中的角色和职责定义了与软件保证相关的具体培训。
  - e) 知识管理过程(6.2.6)用于收集和维护知识,并向项目提供有关威胁环境的变化、软件保证实践的演进、软件漏洞的缓解、从事件和事故响应中吸取的教训以及软件保证测试工具的演进信息。相关活动和任务包括6.2.6.3中b)的所有任务,以及d)3)
- 6决策管理过程(6.3.3)从整体上根据决策标准评估备选需求、架构特性和设计特性,也包括软件保证特性。将这些比较的结果与选择模型一起进行排序和记录,然后用于确定最优解决方案。利益相关方可以根据提供的结论做出决策。
- g) 风险管理过程(6.3.4)从整体上提供了对无法实现造成用户(包括接口合作伙伴)风险或导致软件未按预期使用的必要应用程序信息安全的可能性进行评价。软件信息安全是每个风险分析中的一个风险类别。
  - h) 配置管理过程(6.3.5)规定了建立和维护项目或过程的所有已识别输出的完整性,包括指定存储系统的资产和信息安全的监控,并使其对授权方可用。这包括对软件和软件版本的更改记录,以及对处理信息安全功能的软件项的所有访问控制和审计。相关的活动和任务包括6.3.5.3中的a)1)、d)1)和f)2)。
  - i) 信息管理过程(6.3.6)从整体上向相关的利益相关方(包括监管或批准机构)提供有关软件保证实现的信息。收集相关的软件可量化信息以支持一组论据,这些论据证明了关于系统的软件保证的声明是合理的。由于数据的敏感性,需要特别注意为各种保证措施确定适当的受众。信息管理包括关于软件保证的敏感信息项的保护。
  - j) 测量过程(6.3.7)从整体上为收集软件保证声明、策略和证据(有时称为保证案例)的信息提供了一个公共平台。
  - k) 质量保证过程(6.3.8)评估项目和供方过程是否符合软件保证要求和程序。它处理与软件保证特性的实现相关的已识别的异常(偶发事件和问题)。相关活动和任务包括6.3.8.3中c)的全部任务和e)的全部任务。
- 1)业务或使命分析过程(6.4.1)提供了对软件系统在开发和与软件保证相关的法律、政策、风险和约束下的运行环境的理解。相关的活动和任务包括6.4.1.3中的b)1)和c)1)。
- m) 利益相关方需要和需求定义过程(6.4.2)规定了任务或信息的风险和威胁的选择和定义。它在定义与软件保证(信息安全)相关的需求时,结合了这些知识,包括软件为了发挥作用时的保密性、可用性和完整性,同时还将考虑到误用和滥用的情况。利益相关方需要就软件保证的哪些方面是足够的达成一致。相关活动和任务包括6.4.2.3中a)2)、b)1)、b)2)和c)的全部任务。
  - n) 系统/软件需求定义过程(6.4.3)规定了软件保证(信息安全)相关需求的选择和定义,包括软件为了发挥作用时的保密性、可用性和完整性,同时还将考虑在发生误用和滥用时对软件完整性的要求。相关活动和任务包括6.4.3.3中的b)3)、b)4)和c)3)。

- o) 架构定义过程(6.4.4)通过建立威胁模型及评估产品架构及设计面对潜在攻击时的漏洞,从架构的角度识别利益相关方所关注的事项,从而了解威胁情况及相关的架构元素。相关活动和任务包括6.4.4.3中a)2)、a)4)、b)1)、c)的全部任务、d)5)、f)1)、0)2)和05)。
- p) 设计定义过程(6.4.5)规定了确定必要的设计特性(包括攻击面减少、组件的位置),软件保证设计模式以及避免反模式。相关活动和任务包括6.4.5.3中的a)2)、a)3)、b)2)、b)3)、b)6)、c)2)和d)1)。
- q) 实现过程(6.4.7)提供了利用安全编码实践来避免常见的编码错误所导致的可利用的产品漏洞,并使用包括检查真实性、模糊测试、静态分析测试和动态测试在内的各种测试技术来识别和解决软件缺陷和漏洞。相关活动和任务包括6.4.7.3中的a)1)、和b)1)、b)2)、b)3)和b)4)。
- r) 集成过程(6.4.8)规定了集成的计划,包括软件保证特性的考虑,以及确定和记录特性实现的保证。接口标准的实施促进了系统和元素的可持续性和元素的可重用性。相关活动和任务包括6.4.8.3中的a)2)、a)5)、b)3)、c)1)和c)2)。
- s) 验证过程(6.4.9)规定了验证策略的规划和执行,以验证软件保证需求(包括软件保证特性)已经实现。所选择的验证策略可以在开发过程或维护期间引入对代码缺陷的测试。威胁分析为测试计划和案例的创建提供输入。这些结果包括了纠正软件中不符合项或对其起作用的过程纠正措施所需要的信息,以及验证活动中的不确定性,如测试工具的可靠性和结果的不确定性水平(即,假阳性和假阴性的比率)。其他需要考虑的因素包括软件运行的弹性。相关活动和任务包括6.4.9.3中a)1)、a)4)、a)5)、a)6)、b)的全部任务、c)1)和c)2)。
- t) 移交过程(6.4.10)规定了在不同的环境中安装软件系统或元素。由于一些软件保证特性涉及到设计约束和操作约束之间的权衡,所以对安装和用户文档的关注通常是很重要的。相关活动和任务包括6.4.10.3中的a)2)、b)1)、b)4)、b)5)、b)6)、c)1)和c)2)。
- u) 确认过程(6.4.11)提供证据,证明软件系统提供的服务满足利益相关方的需求,包括软件保证特性。确认方法包括漏洞扫描、使用各种工具和技术代码评估和确认,如静态代码分析、动态代码分析、二进制代码分析、代码覆盖工具、压力测试,以及使用工具收集远程维护活动导致的更改的证据。其他需要考虑的因素包括软件在使用和滥用情况下的弹性。相关活动和任务包括6.4.11.3中a)1)、a)3)、a)4)、b)的全部任务、c)2)和c)3)。
- v) 运行过程(6.4.12)规定了软件系统的使用。要确保适当地实现软件保证特性,需要监视系统的运行,以便在预期的环境中交付服务,并为软件产品的客户提供支持。此过程的计划将考虑在整个系统生存周期中实现的应用安全性,诸如访问控制之类的运行限制,以及在早期阶段所做的关于应用安全性与运行环境的一致性的假设。该过程包括建立报告系统和程序,用于调查和处理与应用安全性相关的事件。相关活动和任务包括6.4.12.3中a)1)、b)的全部任务、c)1)、c)2)、c)3)和d)1)。
- w) 维护过程(6.4.13)维持软件系统的能力,特别是其持续的可用性,以提供其软件保证特性。这包括失效分析、维护任务和确保系统持续运行所需的后勤任务。维护过程用于评价在维护期间更改软件对软件保证的影响,并为软件保证维护适当的证据。它包括一个文档化的过程,用于修补和纠正软件,检测、移除或停用未经授权的和恶意的软件,以及需方或接口合作伙伴的通知和补救机制的过程。漏洞修复的优先级基于各种因素,包括风险。在实施软件补救时,遵循文档化的开发和维护实践。相关活动和任务包括6.4.13.3中a)1)、b)的全部任务、c)的全部任务、d)1)和d)2)。
- x) 处置过程(6.4.14)使软件系统终止存在。预期处置的内在需求可能会限制软件系统中包含的数据的开发和管理。实际上,这些约束可以是软件保证特性。相关活动和任务包括6.4.14.3中 a)1)、a)2)、a)5)、b)的全部任务、c)1)和c)3)。

附 录 F  
(资料性)  
软件系统架构建模

**F.1 导 引**

在本文件中，架构和设计活动被描述为独立的过程。当构建一个软件元素的架构，比如决定一个实体或功能是通过集成现有软件、适应性重用还是构建新软件来实现时，可能涉及架构定义、设计定义和实现等过程的几次迭代。在处理复杂系统的系统和软件工程社区中，体系架构可以伴随不同产品线中不同系统的不同设计。这种情况下，分别执行这两个过程是很重要的。此外，定义架构经常并非只是为了作为设计的基础，而是一些其他原因，例如驱动技术投资、实现一致性或降低组织的产品线或特定项目包的复杂性，或指导需方—供方决策。

软件系统的体系架构可以理解为一组结构化的体系架构实体及其关系，选择这些实体是为了实现诸如互操作性、可伸缩性、环境弹性、封装性、可用性、可负担性、健壮性、执行效率或任务有效性(适用性)等特征。软件系统架构处理各种实体之间的关系，例如场景、功能、功能流、接口、资源流项、信息或数据元素、对象、物理组件和环境、容器、节点、链接、通信资源、约束、公式和参数模型。

本附录描述了用于创建和评估软件系统架构的一些模型(模型种类)。

注：GB/T 22032-2021的附录F描述了模型和视图是如何应用于一般系统架构的，并提供了与物理产品架构的视图和建模相关的附加信息，如质量模型和布局模型。

**F.2 软件系统架构中使用的视图、模型和模型种类**

体系架构定义的过程使用各种各样的软件系统模型，包括F.2.1~F.2.7中列出的示例模型。模型种类指定了用于这类模型的语言、符号、约定、建模技术、分析方法或其他操作。(传统的系统工程实践将其中一些模型划分为“逻辑模型”或“物理模型”，但是在本文件的应用中没有必要进行分类区分。)各种视图用于表示系统体系架构如何处理利益相关方的关注点。视图由模型组成。例如，软件的逻辑视图可以在其功能中表示业务过程；过程视图可以表示在软件的不同状态中发生的事件和转换，还可以包括并发性和时间问题；结构视图表示不同的系统组件，这些组件可以与物理或虚拟系统元素相关联，信息视图表示软件中包含和转换的数据元素之间的关系。

有关体系架构术语的定义和体系架构概念和模型的更多详细信息，请参阅 ISO/IEC/IEEE 42010。

**F.2.1 功能模型**

系统的功能模型是一组功能的表示，这些功能定义了将输入转换为系统执行的输出以实现其任务或目的。这些功能取决于系统在按预期使用时的行为方式。因此，每个系统功能都与系统及其环境之间的交互有关，一般通过分析功能性需求、性能需求、非功能性需求和约束需求以确定功能和输入输出流。当功能与系统元素相关联时，设计定义过程需要确定每个系统/软件元素是否有关联，以被充分指定来建造或购买。如果为了达到这个充分性而进一步解析系统元素，那么与系统元素相关的功能也会进一步解析，并适当地与子元素相关。通常，有多种方法来分解有助于定义多个候选体系架构的功能。

**F.2.2 静态模型**

静态模型描述软件系统的结构。在面向对象编程中，它是通过一组对象(类)及其关系(继承、关联和依赖)来表示的，这些关系被描述为节点和链接。



### F.2.3 数据模型

数据模型(语义或信息模型)表示软件系统将处理的数据元素及其关系和属性(特性)。逻辑数据模型使用模式来反映可以在数据库中实现的数据实体之间的结构关系。数据模型反映了不同类型的数据(文本、图形、地理数据、图像、一般对象)及其在系统功能中的使用(更改频率、数据量、搜索使用),以及数据元素之间的逻辑关系。数据模型应用于接口和软件服务的开发、数据分析和数据报告。物理数据模型反映了存储和检索数据记录的模式。

### F.2.4 行为模型

行为模型(动态模型)是功能和接口(内部和外部)的一种安排,它定义了系统或其元素在维持操作场景的条件下如何工作,包括执行顺序、同步和并发、行为更改的条件和性能。行为模型适用于软件控制系统。行为模型可以用一组相互关联的场景来描述。这包括识别行为元素(例如,模式/状态、转换、触发事件和操作场景)。

### F.2.5 时间模型

系统的时间模型是一种表示,它表示在系统或其元素的行为中如何考虑时间,从而表示函数的执行频率级别(例如,战略级别、战术级别、运行监控级别、监管级别)对应的决策级别,使人员和程序逻辑能够监视和控制系统操作。这包括从运行概念和系统需求中确定时间元素(例如,持续时间、频率、响应时间、触发器、超时、停止条件)。

### F.2.6 结构模型

系统的结构模型是一种表示,它显示元素相对于其他元素的排列,并在必要时显示元素与外部实体之间的接口。该模型支持合并或标识系统层次结构级别中的系统元素之间和系统层次结构级别之间的物理接口,以及与相关系统的外部实体相关的物理接口(在其环境/上下文中)。结构模型可以是分层分解的,也可以是面向对象的。

### F.2.7 网络模型

网络模型定义了节点和链接的安排,以帮助理解资源(例如,信息和人)从一个节点遍历到另一个节点。网络模型可用于确定诸如吞吐量、延迟和拥塞点等约束。网络模型有时与协议栈一起建模,用于理解网络中不同层次之间如何在堆栈中向上和向下垂直交互。

## F.3 其他模型注意事项

利益相关方的生存周期关注点,如维护、演进、处置、环境的潜在变化、过时管理和其他非功能性需求,通过定义架构特征如模块化、相对独立性、可伸缩性、可升级性、适应多种需求、环境、有效性水平、可靠性、健壮性和弹性来确定的。其他必要的模型可以包括这些特征中的一些或其他关键的质量特征。例如,一个软件保证案例,作为一个模型,可以帮助推断出潜在的架构缓解措施,以最小化与关键关注点和功能相关的运行风险(由于被利用的安全漏洞导致的任务损失)。

确定在系统定义中使用哪些模型可以基于对利益相关方关注点的检查。模型和结果视图可以用来表达系统架构和设计如何处理它们的关注点,并更好地理解它们的实际需要、需求和期望。

此外,除了体系架构和设计定义之外,模型还可以用于其他生存周期过程。基于模型的系统工程(MBSE)是建模的形式化应用,以支持整个生存周期的系统需求、体系架构、设计、分析、验证和确认活动。

注:验证和确认模型定义了测试信息的表示形式,它可以支持体系架构的验证。验证和确认模型可以生成测试分析、数据、案例和其他信息。



附 录 G

(资料性)

将软件生存周期过程应用于系统之系统

G.1 导引

系统之系统(SOS) 是一个其元素本身就是系统的所关注焦点的系统(SOI)。系统之系统集合一组系统来完成一项任务，而这些任务是任何系统都无法单独完成的。每个组成系统保持自己的管理、目标和资源，同时在SOS 内部协调和适应以满足 SOS 的目标。在5.2.3讨论的术语中(如图3所示), 包含原始SOI、使能系统和交互系统的复合系统集成构成一个SOS。当存在影响组成集的问题时，系统之系统就成为了SOI,SOI 被认为是用来满足单个组成系统不能满足的业务或任务目标，或者是理解组合的紧急行为。

本附录阐述了系统生存周期过程对于此类 SOS 的应用。它描述了一般特征、SOS 的常见类型，以及贯穿生存周期的含义。

G.2 sOS特征和类型

系统之系统的特点是组成系统管理和运行的独立性，在许多情况下，这些系统被开发并同时继续支持最初识别的用户和系统之系统的用户。在其他情况下，每个组成系统本身是一个所关注的系统；它的存在往往早于SOS，而它的特性最初是为了满足用户的需求而设计的。作为SOS 的成员，他们的考虑范围扩展到 SOS 的更大需求。这意味着增加了复杂性，特别是当系统还需独立于SOS 继续演进时。组成系统通常还保留其原始的利益相关方和治理机制，这限制了解决SOS 需求的备选方案。

根据组成系统和SOS 之间的治理关系，SOS 被描述为四种类型(表G.1)。最强的治理关系适用于系统的定向式系统，其中SOS 组织对组成系统具有权威，尽管组成系统最初并没有被设计来支持SOS。对公认式的SOS 的控制要少一些，因为在组成系统和系统之系统之间分配的权限会影响一些系统工程过程的应用。在缺乏系统权威的协同式 SOS 中，系统工程的应用依赖于各组成系统之间的协作。虚拟式 SOS 在很大程度上是自组织的，并限制了SOS 的系统工程机会。

表 G.1 SOS类型

类型	特征
虚拟式	<ul style="list-style-type: none"><li>●去中心化的管理授权</li><li>●没有明确的、集中商定的目的</li><li>●依赖于相对不可见持续机制的交互行为</li></ul>
协同式	<ul style="list-style-type: none"><li>●组成系统自愿交互以实现商定目的</li><li>●共同决定如何合作、执行和维护标准</li></ul>
公认式	<ul style="list-style-type: none"><li>●针对SOS的公认目标、指定的管理者和资源</li><li>●组成系统保留其独立的所有权、管理和资源</li></ul>
定向式	<ul style="list-style-type: none"><li>●建立和管理整合的SOS以实现特定的目的</li><li>●集中管理的和演进的</li><li>●组成系统保持独立运行的能力</li><li>●正常的运行模式，服从于中心目的</li></ul>

交互是SOS 的一个关键特性——在SOS 层面的不可预知效应归因于组成系统的复杂交互动态。然而，在SOS 中，有意将组成系统组合在一起考虑，以便获得和分析一些从单一系统无法获得的结果。组成系统的复杂性，以及它们可能在设计时未考虑其在SOS 中作用的事实，可能会导致新的、意想不到的行为。识别和处理无法预知的交互结果是工程化SOS 面临的一个特殊挑战。

注：一些最大的软件系统之系统，如Internet,是虚拟式 SOS,其中的组成系统被设计成遵循通用的建议和通信协议。虚拟式SOS可表现出有益的交互特点，如冗余、动态重构、协作和弹性

### G.3 应用于SOS的系统工程过程组

#### G.3.1 概述

SOS 的上述特性对四类系统生存周期过程组中的每一个应用都有影响。

#### G.3.2 协定过程组

协议过程组对SOS 至关重要，因为它们在负责SOS 和通常独立的组成系统的组织之间建立了开发和运行控制的模式。由不同组织获取和管理的组成系统，有时具有与SOS 不一致的初始目标。除了在定向式SOS 的情况下，SOS 组织不能对没有合作的组成系统下达任务。在共识式或协同式 SOS 中，这些任务与组成系统本身作为SOI 的任务相平衡。对于虚拟式SOS, 协定过程组中的过程可能是非正式的，或者仅出于分析目的而考虑。

即使在组成系统的所有者之间的协议中，仍然有一个需方和一个供方。系统所有者可以同时是另一个组成系统的需方和供方。

#### G.3.3 组织的项目使能过程组

在一个典型的SOI 中，组织的项目使能过程组中的过程建立了实施项目的环境。组织建立项目使用的过程和生存周期模型；建立、重定向或取消项目；提供所需资源，包括人力和财政资源；并设置、跟踪为内、外部客户所开发的系统和其他可交付物的质量测度(6.2)。

在 SOS 中，组成系统的所有者通常保留对其系统进行工程设计的责任，他们每个人都有自己的组织项目授权流程。根据SOS 的类型，SOS 还将这些组织的项目使能过程组中的过程应用到SOS 的特殊考虑：计划、分析、组织和将现有系统和新系统的能力整合到SOS 的能力中。

因此，在SOS 中，这些组织的项目使能过程组中的过程在两个层面上实施。负责组成系统的组织为其SOI 独立于SOS 实现这些过程。SOS 组织(或通过SOS 协议系统的协作系统)为SOS 实施这些过程，以实现适用于整个SOS 的那些考虑。例如，每个组成系统组织为其系统工程解决人力资源管理问题。SOS 组织只针对在整个组成系统中应用到SOS 的系统工程活动解决此问题。

SOS 工程中一个特别的挑战是，组成系统和SOS 的组织的项目使能过程组中的过程之间缺乏一致性。组成系统的过程是为了满足它们自己的结果而设计的，有时与SOS 的结果不一致。例如，当组成系统的组织完全控制其特定项目包中的组成系统和其他系统和项目时，与SOS 组织的特定项目包管理不同，特定项目包管理可以是某个组成系统职责。

#### G.3.4 技术管理过程组

在一个典型的SOI 中，技术管理过程组中的过程与管理组织管理层分配的资源 and 资产有关，并应用这些资源和资产来履行组织或组织签订的协议。它们涉及项目的管理，特别是在成本、时间尺度和成就方面的规划，对符合计划和业绩标准的行动进行检查，以及查明和选择纠正行动以弥补进展和成就方面的不足。它们用于建立和执行项目的技术计划，管理整个技术团队的信息，根据系统产品或服务的计划评估技术进展，控制技术任务直至完成，并在决策过程中提供帮助(6.3)。

技术管理过程组中的过程也在SOS 和组成系统的层面上实施。技术管理过程组中的过程应用于

SOS 工程的特殊考虑——规划、分析、组织和将现有系统和新系统进行混合的能力集成到SOS 能力。与此同时，组成系统的组织保留对其系统的工程设计和其自身技术管理过程组中的过程的责任。

SOS 组织处理技术管理过程组中的过程，因为它们适用于整个SOS，而这些过程也在组成系统组织中独立实现。如，对于配置管理，组成系统管理它们自己的配置，而SOS 处理配置管理，因为它适用于SOS 中的系统组合。组成系统的风险管理是基于对系统输出的风险评估，而 SOS 的风险管理着眼于该SOS 的风险。

规划、评估和控制(6.3)是所有管理实践的关键；SOS 工程中的一个关键挑战是SOS 组织对组成系统的过程缺乏控制(特别是对于公认式和协同式SOS)，在其自身组织需求的驱动下，每个组成系统都可以处于不同于其他组成系统的开发或升级计划中。SOS 组织计划了一个集成的生存周期，除了SOs 启动的生存周期中的变化(将SOS 视为SOI) 之外，该生存周期还识别组成系统中的独立变化。这通常涉及稳定的中间形式定义，这些形式通过在组成系统之间添加增量功能来强调SOS 的发展。

### G.3.5 技术过程组

技术过程组涉及整个生存周期中的技术操作。它们首先将利益相关方的需求转化为产品，然后通过应用该产品，在需要的时候和地方提供可持续的服务，以实现客户满意度。无论它是以模型还是成品的形式，应用技术过程组是为了创建和使用某个系统，并且它们适用于系统结构层次中的任何层次(6.4)。

与应用于SOS 的其他程序一样，SOS 和组成系统的技术程序均已执行；在某些情况下，SOS 的实施是通过执行组成系统的过程，而不是针对整个SOS。

针对某个SOS 的业务或使命分析观察整个 sOS 的业务和使命环境。在一定程度上，当组成系统开发至可在此空间中运行时，系统和组成系统的业务或使命分析将在很大程度上被共享。目标是确定提供所需能力的最佳方法。

利益相关方需要和需求定义关注于顶层SOS，但也考虑利益相关方对单个系统的不同需求如何导致对SOS 的约束。

SOS的系统/软件需求定义往往是在满足利益相关方需求和任务目标所需的层次上定义的，并将其转换为构成系统的需求，而SOS 则作为提出组成系统新需求的“利益相关方”。

SOS 的架构是一个框架，用于将现有系统和新系统进行混合的能力组织和集成到SOS 能力中，将组成系统的架构留给它们的组织。由于SOS 中的组成系统通常早于SOS，所以sOS 架构的定义通常从SOS 的实际架构开始。然后检查架构备选方案，以构建利益相关方的关注并满足顶层SOS 的需求，同时认知新需求对于组成系统的影响并适应组成系统架构的约束。

设计定义过程提供了足够详细的数据和信息来支持SOS 的实现。这涉及到与组成系统组织的协作，这些组织将进行他们自己的设计交易，以确定当SOS 需求应用于他们的系统时进行处理的方法。这些是组成系统组织的责任。实施工作由组成系统完成，而SOS 组织则担当监视的角色。

集成、验证、移交和确认由组成系统完成，这些组成系统为支持 SOS 生成的需求而进行实现的变更。当升级后的组成系统集成到 SOS 中并验证和确认性能时，这些过程也适用于SOS。SOS 中组成系统的独立性和异步性对在传统SOI 中实现这些过程的有效实现提出了挑战。在某些情况下，SOS 级别的评价只能在运行环境中进行，在这种情况下，应考虑预防措施以避免出现不良的SOS 行为。

最后，鉴于组成系统的管理和运行独立性，它们的运行、维护和处置过程往往由各自执行。SOS 级别的交互可以促进这些过程的互操作性，并减少它们的重复工作。

## 附录 H

## (资料性)

## 敏捷的应用

本文件旨在应用于使用敏捷方式和方法的组织和软件项目，以及使用其他正式工程方法的组织和软件项目。敏捷开发是软件开发(包括软件维护)中使用最广泛的方法之一，因为人们认为敏捷开发更方便，并且可以更快地交付可用的产品。在大型软件开发工作和小型项目中，许多敏捷方法可以与各种生存周期模型一起使用，并且可以在生存周期的不同节点使用不同的方法。本附录指出了本文件中对过程需求的解释，这些解释适用于常用的敏捷技术。

如5.4.2所述，敏捷项目中使用的生存周期模型通常是高度增量和进化的。然而，使用敏捷方法的组织确实应用了本文件中定义的生存周期过程，包括组织，技术管理和技术过程(并且可以在协议过程下操作)。如5.4.1所述，本文件没有规定生存周期模型中任何特定的过程序列。过程的顺序由项目目标和生存周期模型的选择决定。敏捷项目，因为它在创建或改进工作软件时转换或组合活动，可能会发现声明完全符合结果(4.2.1)比声明完全符合活动和任务(4.2.2)更合适。

敏捷开发之所以成功，部分原因在于软件的性质，即在构建软件的过程中在设计上保持灵活性。在敏捷实践中，软件设计、实现(构建)和持续集成通常是同时执行的。这种做法与一种正式的自上而下的可追溯性方法形成对比，在这种方法中，只有在设计获得批准，从而将已构建的软件追溯到先前批准的详细设计之前，施工才能开始。因此，敏捷项目充分利用了本文件中的方法，与顺序阶段(理想化的瀑布模型)项目相比，过程是并发发生的。

在敏捷项目中，方案探索、开发、构建、测试、转换和先前软件的退役都可并行实施以连续迭代。敏捷项目经常与上面提到的活动同时执行重新规划。在这种方法中，将阶段的末尾用作管理检查点或控制并不是很有用。其他敏捷方法在指定迭代之间的节点上执行重新规划(例如，短期冲刺或预定义时间限制节奏)，因此，每个迭代都可以被视为一个阶段。

敏捷项目可以有紧密联系的开发和软件发布周期。为了方便客户，或者根据组织策略，可以将开发迭代的完成与计划软件发布的管理分开。

除了应用高度迭代和进化的生存周期模型外，敏捷组织对项目规划、项目评估和控制过程有特定的实践。敏捷项目通常不会在阶段或过程之间建立主要的控制点，而是在一个时间周期的末尾设立不太正式的检查点或回顾性的评审，以便就下一个周期的改进达成一致。每个迭代包括设计、开发和测试活动(测试驱动开发)。在大约1周~4周或更长时间的短期冲刺之后，新的工作软件元素被接受为“完成”——完全开发、验证(测试)和确认。总结经验并确定过程改进，然后开始另一个短期冲刺的工作。持续的学习、风险管理和过程改进可以通过规划每个迭代的启动会议和在每个迭代结束时举行的回顾会议来促进。

敏捷方法强调利益相关方的需要和需求定义过程，通过高度的利益相关方参与度促进变更。在敏捷项目中，关键的利益相关方，例如需方或用户代表，不仅仅是信息、度量和评估报告的审批者。持续的利益相关方参与与本文件是一致的，本文件确定了利益相关方在每个技术过程以及剪裁过程中的参与(附录A)。在每次迭代中他们密切参与需求管理[6.4.3.3d]]，通过引入新的需求和优先级的变化，并在从未开发的故事或功能的积压中选择优先级需求并进一步细化以供开发时参与。在项目的一般范围内，迭代方法鼓励增加、调整优先级或延迟的灵活性。同样，利益相关方在每次迭代中参与测试软件的批准意味着在整个项目中验证是持续的。

随着需求演进的增量定义：项目范围的概念在敏捷项目中不同于由指定需求的预定基线定义范围的项目。当一个敏捷项目需要某个定义好的产品时，它的范围最初是与高层次或基本需求相联系的。随着在构建过程中获得更多的知识，预计将出现更详细的产品定义级别。没有预先定义的产品敏捷

工作(例如,努力维护的等级)。通过有时间限制的计划进度表或资源有限的团队来控制范围。这种方法特别适用于软件维护工作,在软件维护工作中,纠正或适应性工作的范围或内容在最初没有完全指定。

与更传统的开发工作相比,敏捷开发项目的基线规范在程度和时间上也有所不同。需求的基线最初可能包括高级用户描述(“epics”)和关键性能测量,包括可用性标准。敏捷项目充分利用“在生存周期中定义基线”的任务(6.3.5.3b)]。在开发过程中,在配置控制下,至少每天都要建立新的基线。一个软件元素通常可以追溯到一个高级的功能需求,并且可以紧密地追溯到它实现的用例和用来验证它的功能和性能的测试用例。一个新的软件元素可以简单地追踪到一个设计元素或者对象,而不是追踪到一个以前被批准的、基线化的设计文档,事实上,这个设计元素或者对象是在软件构建过程中创建的,然后才置于配置控制之下。

在敏捷项目中,规范、设计工件、信息项或文档的准备工作通常是有限的,而软件开发人员将他们的时间和技能用于将功能的场景或叙述(“用户故事”)转换为可工作的、可测试的软件元素。与准备详细的评审包以在不常见的主要里程碑评审中进行简报不同,团队经常与利益相关方会面,以提供完成一组功能的非正式证据,并就下一次迭代的内容达成一致。文档化的信息项关注过渡、操作和维护所需的内容,例如操作员和终端用户文档,以及带有测试计划和测试用例的软件测试和发布版本的基线。项目重用组织过程进行配置和发布管理、验证以及偶发事件和问题管理。在可能的情况下,双向可追溯性由集成的自动化系统和过程来支持和实施,这些系统和过程包括需求管理、体系架构和设计、配置管理、度量和信息管理。

敏捷开发的增量和迭代特性可以促进有效的技术和管理过程和实践,从而减少与变更相关的成本。相比之下,连续体的瀑布式末端管理的项目,通过最小化变更的数量、限制控制点的数量以及在整个项目中评审和跟踪详细规范的基线化,来寻求减少总的返工成本。

针对复杂系统的敏捷项目试图通过对早期实现的最重要功能进行优先排序来管理成本。如果组织将其整个软件系统特定项目包的开发和维护作为一个系统来管理,关注花费率而不是总开支,那么原则上,组织可以作为一个持续的敏捷开发来管理特定项目包,类似于高度迭代的“Kanban”管理维护工作。

附录 NA  
(资料性)

本文件与GB/T 8566—2007的差异

与 GB/T 8566—2007相比，本文件的结构差异见表NA.1。

表 NA.1 结构差异对应表

GB/T 8566—2022	GB/T 8566—2007	备注
1 范围	1 范围	
1.1概述		增加了与GB/T 22032—2021《系统与软件工程 系统生存周期过程》使用环境的说明
1.2目的	1.1目的	
1.3应用领域	1.2应用范围	
1.4限制	1.5限制	增加了质量管理体系兼容性说明
2规范性引用文件	2规范性引用文件	本文件没有规范性引用文件
3术语、定义和缩略语	3术语、定义	增加了20个缩略语
3.1术语和定义		
3.2缩略语		
4 符合性	—	新增章节
4.1预期用途	—	新增章节
4.2完全符合性	1.4符合性	增加了完全符合概念，分为输出的完全符合和任务的完全符合
4.3剪裁符合	1.3本标准的裁剪	
5关键概念和应用		新增章节
5.1导引	—	新增章节
5.2软件系统概念	—	新增章节
5.3组织和项目概念		新增章节
5.4生存周期概念		新增章节
5.5过程概念	—	新增章节
5.6过程组	—	新增章节
5.7过程应用	—	新增章节
5.8过程参考模型	—	新增章节
6软件生存周期过程	4.1.1生存周期过程	新增过程组概念，过程数从21个增至30个
6.1协定过程组	5.1获取过程+5.2供应过程	
6.2组织的项目使能过程组	7.1管理过程+7.2基础设施过程+7.4人力资源过程	新增生存周期模型管理过程、特定项目包管理过程和知识管理过程

表 NA.1 结构差异对应表(续)

GB/T 8566—2022	GB/T 8566—2007	备注
6.3技术管理过程组	6.2配置管理过程+6.3质量保证过程	新增项目规划过程、项目评估与控制过程、决策管理过程、风险管理过程、信息管理过程和测量过程
6.4技术过程组	5.3开发过程+5.4运作过程+5.5维护过程	新增业务或使命分析过程、利益相关方需要和需求定义过程、系统/软件需求定义过程、架构定义过程、设计过程、系统分析过程、集成过程、移交过程、确认过程和处置过程
附录A(规范性)剪裁过程	附录A(规范性附录)剪裁过程	
附录B(资料性)过程信息项示例		新增章节
附录C(资料性)用于评估目的的过程参考模型		新增章节
附录D(资料性)过程集成和过程构造		新增章节
附录E(资料性)过程视图		新增章节
附录F(资料性)软件系统架构建模		新增章节
附录G(资料性)将软件生存周期过程应用于系统之系统		新增章节
附录H(资料性)敏捷的应用		新增章节
附录NA(资料性)本文件与GB/T 8566—2007的差异		新增章节

其中，术语、定义和缩略语具体变化情况如下：

- 更改了术语和定义，包括需方(见3.1.1, 2007年版的3.1), 获取(见3.1.2, 2007年版的3.2), 协定(见3.1.5, 2007年版的3.3), 审核(见3.1.10, 2007年版的3.4), 基线(见3.1.11, 2007年版的3.5), 配置项(见3.1.15, 2007年版的3.6), 生存周期模型(见3.1.27, 2007年版的3.12), 操作方(见3.1.29, 2007年版的3.17), 过程(见3.1.33, 2007年版的3.18), 质量保证(见3.1.40, 2007年版的3.22), 发布(见3.1.43, 2007年版的3.23), 信息安全性(见3.1.49, 2007年版的3.26), 软件单元(见3.1.57, 2007年版的3.29), 供方(见3.1.60, 2007年版的3.31), 系统(见3.1.61, 2007年版的3.32), 用户(见3.1.70, 2007年版的3.35), 确认(见3.1.71, 2007年版的3.36), 验证(见3.1.72, 2007年版的3.37)。
- 增加了术语和定义，包括活动(见3.1.3), 敏捷开发(见3.1.4), 架构(见3.1.6), 架构框架(见3.1.7), 架构视图(见3.1.8), 架构视角(见3.1.9), 业务过程(见3.1.12), 运营观念(见3.1.13), 关注焦点(见3.1.14), 顾客(见3.1.16), 设计(动词)(见3.1.17), 设计(名词)(见3.1.18), 设计特性(见3.1.19), 使能系统(见3.1.20), 环境(见3.1.21), 设施(见3.1.22), 偶发事件(见



3.1.23), 信息项(见3.1.24), 基础设施(见3.1.25), 生存周期(见3.1.26), 运行概念(见3.2.28), 组织(见3.1.30), 当事方(见3.1.31), 问题(见3.1.32), 过程输出(见3.2.34), 过程目的(见3.2.35), 产品(见3.2.36), 项目(见3.1.37), (项目)特定项目包(见3.1.38), 资质(见3.1.39), 质量特性(见3.1.41), 质量管理(见3.1.42), 需求(见3.1.44), 资源(见3.1.45), 风险(见3.1.47), 安全(见3.1.48), 服务(见3.1.50), 软件元素(见3.1.51), 软件工程(见3.1.52), 软件项(见3.1.53), 软件系统(见3.1.55), 软件系统元素(见3.1.56), 阶段(见3.1.58), 利益相关方(见3.1.59), 系统元素(见3.1.62), 所关注的系统(见3.1.63), 系统之系统(见3.1.64), 系统工程(见3.1.65), 任务(见3.1.66), 技术管理(见3.1.67), 权衡(见3.1.68), 可追溯性(见3.1.69)。

——删除了术语和定义, 使用周境(见2007年版的3.7), 合同(见2007年版的3.8), 开发方(见2007年版的3.9), 评价(见2007年版的3.10), 固件(见2007年版的3.11), 维护方(见2007年版的3.13), 监督(见2007年版的3.14), 非交付项(见2007年版的3.15), 现货产品(见2007年版的3.16), 合格性认定(见2007年版的3.19), 合格性需求(见2007年版的3.20), 合格性测试(见2007年版的3.21), 招标(标书)(见2007年版的3.24), 软件服务(见2007年版的3.28), 工作说明(见2007年版的3.30), 测试覆盖率(见2007年版的3.33), 可测试性(见2007年版的3.34), 版本(见2007年版的3.38), 过程目的(见2007年版的3.39)。

——增加了缩略语20条(见3.2)

与GB/T 8566—2007中的软件生存周期过程模型对比, 本文件中图4定义的软件生存周期过程模型变化情况见表NA.2。

表 NA.2 软件生存周期过程模型变化情况表

GB/T8566—2022	GB/T 8566—2007
6.1.1获取过程	5.1获取过程
6.1.2供应过程	5.2供应过程
6.2.1生存周期模型管理过程	7.7领域工程过程
6.2.2基础设施管理过程	7.2基础设施过程
6.2.3特定项目包管理过程	
6.2.4人力资源管理过程	7.4人力资源过程
6.2.5质量管理过程	
6.2.6知识管理过程	—
6.3.1项目规划过程	
6.3.2项目评估与控制过程	
6.3.3决策管理过程	
6.3.4风险管理过程	—
6.3.5配置管理过程	6.2配置管理过程
6.3.6信息管理过程	
6.3.7测量过程	6.3质量保证过程
6.3.8质量保证过程	

表 NA.2 软件生存周期过程模型变化情况表(续)

GB/T 8566—2022	GB/T 8566—2007
6.4.1业务或使命分析过程	5.3开发过程
6.4.2利益相关方需要和需求定义过程	
6.4.3系统/软件需求定义过程	
6.4.4架构定义过程	
6.4.5设计定义过程	
6.4.6系统分析过程	
6.4.7实现过程	
6.4.8集成过程	
6.4.9验证过程	6.4验证过程
6.4.10移交过程	6.5确认过程
6.4.11确认过程	
6.4.12运行过程	5.4运转过程
6.4.13维护过程	5.5维护过程
6.4.14处置过程	
—	6.1文档编制过程
—	6.6联合评审过程
—	6.7审核过程
—	6.8问题解决过程
—	6.9易用性过程
	7.1管理过程
	7.3改进过程
—	7.5资产管理过程
	7.6重用大纲管理过程

## 参 考 文 献

- [1]GB/T 11457—2006 信息技术软件工程术语
- [2]GB/T 18757—2008 工业自动化系统 企业参考体系结构和方法的需求
- [3]GB/T 19000—2016 质量管理体系基础知识和词汇
- [4]GB/T 19001—2016 质量管理体系要求
- [5]GB/T 19004—2020 质量管理组织的质量实现持续成功指南
- [6]GB/T 19014—2019 质量管理 客户满意度 监控和测量准则
- [7]GB/T 22302—2021 系统与软件工程 系统生存周期过程(ISO/IEC/IEEE 15288:2015, IDT)
- [8]GB/T 23694—2013 风险管理管理术语
- [9]GB/T 25000.10—2016 系统与软件工程系统与软件质量要求和评价(SQuaRE) 第10部分: 系统与软件质量模型
- [10]GB/T 25000.30—2021 系统与软件工程系统与软件质量要求和评价(SQuaRE) 第30部分: 质量需求框架
- [11]GB/T 29246—2017 信息技术安全技术信息安全管理体系概述和词汇
- [12]GB/T 30999—2014 系统和软件工程生存周期管理过程描述指南
- [13]GB/T 34590.1—2017 道路车辆功能安全第1部分: 术语
- [14]GB/T 38634.1—2020 系统与软件工程 软件测试第1部分: 概念和定义
- [15]ISO 9241-210:2010 Ergonomics of human-system interaction—Human-centred design for interactive
- [16]ISO 9241-220:2019 Ergonomics of human—System interaction—Part 220:Processes for enabling,executing and assessing human-centred design within organizations
- [17]ISO 10007:2003 Quality management systems—Guidelines for configuration management
- [18]ISO/IEC 10746-3:2009 Information technology-open distributed processing—Reference model:Architecture
- [19]ISO 14001:2004 Environmental management systems—Requirements with guidance for use
- [20]ISO/IEC/IEEE 14764:2006 Software engineering—Software life cycle processes—Maintenance
- [21]ISO/IEC 15026-3:2011 System and software engineering—Systems and software assurance—Part 3:System integrity levels
- [22]ISO/IEC 15026-4:2012 System and software engineering—Systems and software assurance—Part 4:Assurance in the life cycle
- [23]ISO/IEC/IEEE 15289:2015 System and software engineering—Content of life-cycle information products(documentation)
- [24]ISO/IEC 15939:2007 System and software engineering—Measurement process
- [25]ISO/IEC 16085:2006 System and software engineering—Life cycle management—Risk Management
- [26]ISO/IEC/IEEE 16326:2009 System and software engineering—Life cycle management—project management

[27]ISO/IEC 16350:2015 Information technology—System and software engineering—Application management

[28] ISO TS 18152:2010 Ergonomics of human-system interaction—Specification for the process assessment of human-system issues

[29]ISO/IEC 19770-1:2012 Information technology—Software asset management—Part 1: Processes and tiered assessment of conformance

[30]ISO/IEC TR 19759:2015 Guide to the Software Engineering Body of Knowledge(SWEBOK) V3,IEEE Computer Society,2014

[31]ISO/IEC 20000-1:2011(IEEE Std 20000-1:2013)Information technology—Service management—Part 1:service management system requirements

[32]ISO/IEC/IEEE 24748-1:2018 System and software engineering—Life cycle management—Part 1: Guide for life cycle management

[33]ISO/IEC/IEEE 24748-2:2018 System and software engineering—Life cycle management—Part 2: Guide to the application of ISO/IEC 15288(System life cycle processes)

[34]ISO/IEC/IEEE24748-3:2020 System and software engineering—Life cycle management—Part 3: Guide to the application of ISO/IEC 12207(Software life cycle processes)

[35]ISO/IEC/IEEE 24748-4:2016 System and software engineering—Life cycle management—Part 4: Application and management of the systems engineering process

[36]ISO/IEC/IEEE 24748-5 System and software engineering—Life cycle management—Part 5: Software development planning

[37]ISO/IEC/IEEE 24765 System and software engineering—Vocabulary

[38]ISO/IEC TR 25060:2010 System and software engineering-System and software product Quality Requirements and Evaluation(SQuaRE)—Common Industry Format(CIF)for usability:General framework for usability-related information

[39]ISO/IEC 25063:2014 System and software engineering—System and software product Quality Requirements and Evaluation(SQuaRE)-Common Industry Format(CIF)for usability:Context of use description

[40]ISO/IEC/IEEE 26515:2011 System and software engineering:Developing user documentation in an agile environment

[41]ISO/IEC/IEEE 26531:2015 System and software engineering—Content management for product life-cycle,user,and service management documentation

[42] ISO/IEC 26550:2015 Software and system—Reference model for product line engineering and management

[43]ISO/IEC 27000:2016 Information technology—Security techniques—Information security management systems—Overview and vocabulary

[44]ISO/IEC 27001:2013 Information security Management System

[45]ISO/IEC 27002:2013 Information security cybersecurity and privacy protection—Information security controls

[46]ISO/IEC 27034:2018 Information Technology—Application security

[47] ISO/IEC 27036(all parts) Information technology—Security techniques—Information security for supplier relationships

- [48]ISO/IEC/IEEE 29148:2018 Systems and software engineering—Life cycle process—Requirements engineering
- [49]ISO 31000:2009 Risk management—Principles and guidelines
- [50]ISO/IEC 33001:2015 Information technology—Process assessment—Concepts and terminology
- [51]ISO/IEC 33002:2015 Information technology—Process assessment—Requirement for performing process assessment
- [52]ISO/IEC 33004:2015 Information technology—Process assessment—Requirement for process reference,process assessment,and maturity models
- [53]ISO/IEC 33020:2015 Information technology—Process assessment—Process measurement framework for assessment of process capability
- [54]ISO/IEC/IEEE 42010:2011 Systems and software engineering—Architecture description
- [55]IEC 61508:2010(all parts)Functional safety of electrical/electronic/programmable electronic safety-related systems
- [56]IEEE Std 730TM—2014 IEEE Standard for Software Quality Assurance Processes
- [57]IEEE Std 828TM—2012 IEEE Standard for Configuration Management in Systems and Software Engineering
- [58]IEEE Std 1012TM—2012 IEEE Standard for System and Software Verification and validation
- [59]IEEE Std 1062IM—2015 IEEE Recommended Practice for Software Acquisition
- [60]ANSI/AIAA G-043A-2012e ANSI/AIAA Guide to the Preparation of Operational Concept Documents
- [61]INCOSE TP-2003-020-01 Technical Measurement
- [62]INCOSE.2015.System Engineering Handbook:A Guide for System Life Cycle Processes and Activities,version 4.0.Hoboken,nj,USA:john Wiley and Sons,Inc,ISBN:978-1-118-99940-0:
- [63]NATO AEP-67 Engineering for System Assurance in NATO Programs
- [64]PMI Practice Standard for Work Breakdown Structures—Second Edition
- [65]SAE ANSI/EIA 649B,configuration Management Standard
-