

团 体 标 准

T/CESA 1159—2021

软件能力成熟度模型

Software capability maturity model

2021-05-27 发布

2021-06-08 实施

中国电子工业标准化技术协会 发布



版权保护文件

版权所有归属于该标准的发布机构，除非有其他规定，否则未经许可，此发行物及其章节不得以其他形式或任何手段进行复制、再版或使用，包括电子版，影印件，或发布在互联网及内部网络等。使用许可可于发布机构获取。

目 次

前言.....	III
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 软件能力成熟度模型.....	2
4.1 概述.....	2
4.2 模型框架.....	3
4.3 成熟度等级的总体特征.....	4
4.4 软件能力成熟度模型的实践构成.....	5
4.5 实践域（实践）的等级要求.....	6
4.6 软件能力成熟度模型的应用.....	6
4.7 能力成熟度评级.....	6
5 组织治理.....	6
5.1 战略与治理.....	7
5.2 组织管理.....	8
5.3 组织绩效管理.....	9
6 软件开发.....	11
6.1 需求开发与管理.....	11
6.2 设计.....	12
6.3 实现与集成.....	13
6.4 测试.....	15
6.5 发布与部署.....	16
7 项目管理.....	17
7.1 项目准备与启动.....	17
7.2 项目策划.....	18
7.3 监视与控制.....	21
7.4 项目结项.....	22
7.5 风险管理.....	23
8 支持保障.....	24
8.1 配置管理.....	24
8.2 质量保证.....	26
8.3 同行评审.....	26
8.4 供方管理.....	27
参考文献.....	29

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。

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。

本文件由中国电子技术标准化研究院提出。

本文件由中国电子技术标准化研究院、中国电子工业标准化技术协会归口。

本文件起草单位：中国电子技术标准化研究院、同济大学、北京航空航天大学、大连华信计算机技术股份有限公司、四川久远银海软件股份有限公司、麦哲思科技（北京）有限公司、浪潮软件科技有限公司、万达信息股份有限公司、北京信城通数码科技有限公司、江苏赛西科技发展有限公司、深圳赛西信息技术有限公司、江苏省软件产品检测中心、广州赛宝认证中心服务有限公司、北京赛迪认证中心有限公司、吉林省电子信息产品检验研究院、重庆市赛晟信息技术服务有限公司、新疆维吾尔自治区软件行业协会、山东省电子信息产品检验院、大连软件行业协会、陕西省电子信息产品监督检验院、武汉市高新技术产业协会、东软集团股份有限公司、江苏中科智联信息技术有限公司、神州数码信息服务股份有限公司、华为技术有限公司、腾讯云计算（北京）有限责任公司、中兴通讯股份有限公司、北京伟仕佳杰信息技术服务有限公司、联想（北京）有限公司、人民日报媒体技术股份有限公司、四川长虹佳华信息产品有限责任公司、中国航天系统科学与工程研究院、荣联科技集团股份有限公司、中国电子科技集团公司第十研究所、北京华宇信息技术有限公司、上海计算机软件技术开发中心、北京航天自动控制研究所、信安软件测评认证中心（深圳）有限公司、北京数字认证股份有限公司、中金金融认证中心有限公司、北京京航计算通讯研究所、北京直真科技股份有限公司、上海微小卫星工程中心、山东创恩信息科技股份有限公司、山东中创软件工程股份有限公司、广东益安人防工程科技有限公司、成都泰一链科技有限公司、大连文思海辉信息技术有限公司、中关村云计算产业联盟、中移（苏州）软件技术有限公司。

本文件主要起草人：何积丰、孙文龙、吕卫锋、林平、商迪、周平、于秀明、崔静、贾璐、朱爱斌、张立、刘宏、孙佩、徐斌、张娜、刘瑞慧、肖筱华、刘玲、张明英、武迪、张星星、吕雪、黄莹、翟宏宝、宿爱霞、华颜涛、张舰、吴鹏、袁林英、尹宏、王振荣、吴剑文、陈锡民、张丽、尹正茹、郝红霞、刘海涛、王芹、王云、任婷、尚思原、陈玉林、赵月、郭栋、余永刚、黎佩南、应志红、沈颖、郭晓慧、杨彪、李向锋、张大健、张志刚、聂俊平、常亮、刘延峰、刘芹、王公韬、汤维、董雷、张明轩、李娜。

软件能力成熟度模型

1 范围

本文件界定了软件能力成熟度的内容框架，主要包括实践域的分类、实践域和成熟度等级的定义，规定了组织治理、软件开发、项目管理和支持保障4个管理域的能力要求。

本文件规定了软件能力成熟度模型在不同等级中的实践活动要求。

本文件适用于：

- a) 寻求软件开发提供商，并要求确保软件开发质量的顾客；
- b) 希望展现其软件开发和交付能力成熟度的组织；
- c) 通过本文件的有效实施与运行来持续改进软件开发和交付绩效的组织；
- d) 依据本文件的要求实施符合性评价的第二方和第三方；
- e) 符合性评价人员培训或建议的提供者。

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

下列术语和定义适用于本文件。

3.1

软件能力 **Software capability**

组织通过软件开发和交付软件，并创造价值的的能力。由组织治理、软件工程、项目管理和支持保障四部分管理能力构成。

3.2

利益相关方 **stakeholder**

指受到某件事情的输出影响或对输出负有某种责任的群体或个人。利益相关方可能包括项目经理、供方、顾客以及其他人员。

3.3

组织 **organization**

一个行政管理结构，在此结构内部，其人员作为一个整体共同管理一个或多个项目或工作组，由同一位高层管理人员负责，并且在相同方针下运作。主要从事软件开发或服务的实体。可能是一个独立实体，也可能是更大实体的一个组成部分。

3.4

组织业务目标 **organization's business objectives**

由高层管理人员制定的目标，用以确保组织的持续存在，并强化其盈利能力、市场份额以及其它影响组织成功的因素。

3.5

战略 **strategy**

组织为实现长期目标所使用的方法和行动。

注：通常包括为了实现目标而实施的重点工作和任务、内外部资源配置方式、管理模式、技术路线等。

3.6

组织绩效 organizational performance

组织为实现其战略目标而展现在不同层面上的有效输出。

3.7

成熟度 maturity

组织在提升软件产品开发能力或软件服务能力过程实践中，各个发展阶段的软件能力程度。

3.8

实践域 practice area

相关实践的集合，共同实现该实践域中描述的目标、价值和必要活动的信息。

3.9

生存周期模型 lifecycle model

一个包含过程、活动和任务的框架，这些过程、活动和任务涉及软件产品和服务的开发、运行和维护，跨越从需求定义到终止使用的系统生存周期。

3.10

过程 process

把输入转换为输出的一组彼此相关的活动。

3.11

工作产品 work product

由过程产生的任何制品。这些制品可能包括文卷、文档、产品的组成部分、服务、过程、规范以及清册等。

3.12

组织过程资产 organizational process assets

组织里被认为对定义和实施的过程有用的制品，它们供开发、剪裁、维护以及实施过程时使用。

3.13

过程绩效 process performance

遵循某过程而达成的结果的测量项。过程绩效的表现形式既包括过程测量项，如工作量、周期时间、缺陷消除效率等，也包括产品或服务的测量项，如可靠性、缺陷密度、响应时间等。

3.14

同行评审 peer review

在工作产品开发或服务期间，为识别和消除缺陷，由具备类似技能的人员对工作产品进行检查。

3.15

组织度量数据库 organizational measurement repository

用于收集有关过程与工作产品的度量结果，并提供使用的专项数据库，特别是收集和提供与组织、标准过程有关的度量数据。这个数据库包含或者引用实际的度量数据以及有关理解和评估该数据的信息。过程和工作产品数据包括：估计的工作产品规模、工作量和成本，实际的工作产品规模、工作量和成本，同行评审效果和覆盖率统计，以及缺陷的数量和严重性。

3.16

质量保证 quality assurance

为了提供足够的信任以表明其能够满足质量要求，而在质量体系中实施并根据需要进行证实的全部有计划和有系统的活动。

4 软件能力成熟度模型

4.1 概述

软件能力成熟度模型旨在通过提升软件组织的软件开发能力帮助顾客提升软件的业务价值。本模型借鉴吸收了软件工程、项目管理、产品管理、组织治理、质量管理、卓越绩效管理、精益软件开发等领域的优秀实践，为软件组织提供改进和评价软件能力的一个成熟度模型。

4.2 模型框架

组织通过软件开发、项目管理、组织治理和支持保障四类活动实现业务价值。软件能力成熟度模型的内容框架见图1：

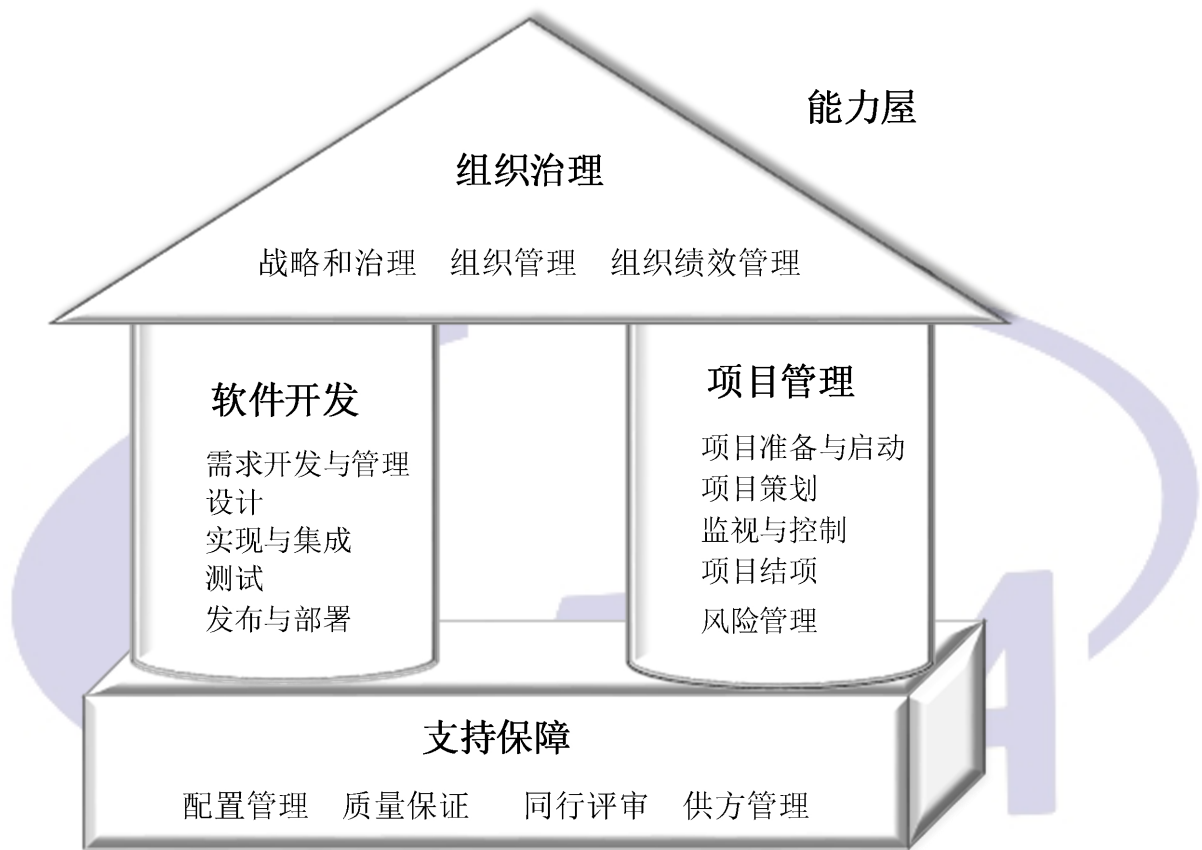


图 1 软件能力成熟度模型的内容框架

组织根据顾客与相关方要求以及对组织外部环境影响因素的分析，确定组织战略与业务目标，分解业务目标，规划产品路线图，配备人力资源，管理合作伙伴，管理组织与团队的绩效，围绕业务目标持续优化组织级的过程。

项目团队和组织通过实施软件开发活动定义、实现、确认与调整业务价值以满足组织级的业务目标，这些活动包括：定义软件的业务价值、目标与范围，实现与集成软件，发布与部署软件等，并在项目全生存周期中进行项目策划和跟踪管理，确保达成项目目标，实现业务价值。

为了更高效的达成业务目标与项目目标，还需积累技术与过程资产，管理环境、工具、安全、保密、问题、数据、风险与机会等，为软件开发项目和业务提供支持保障。

图1中的价值交付包括四类活动：

- a) 组织治理：包括战略和治理、组织管理和组织绩效管理实践，确定组织的战略、产品的方向、组织的业务目标，并对这些目标进行绩效考核，配备并激励员工实现业务目标；

- b) 软件开发：包括需求开发与管理、设计、实现与集成、测试、发布与部署实践，这些实践在软件生存周期中，确保通过软件工程过程交付满足需求的软件，为顾客与相关方增加价值；
- c) 项目管理：包括项目准备与启动、项目策划、监视与控制、项目结项和风险管理实践，这些实践覆盖了软件开发项目的全过程，以确保软件项目能够按照既定的成本、进度和质量交付，能够满足顾客与相关方的要求；
- d) 支持保障：包括配置管理、质量保证、同行评审和供方管理实践，为软件开发提供各种支持和全面管理的保障。

4.3 成熟度等级的总体特征

本模型按照软件能力的成熟度水平，按照由低到高演进发展的形势，定义了五个等级，高等级是在低等级充分实施的基础之上进行，见图2。



图2 成熟度等级模型

达到每个等级后的总体特征见表1。

表 1 成熟度等级的总体特征

等级	结果特征	行为特征
1级：无序管理	有部分的软件交付活动，工作完成结果无法确定	软件项目交付成果可能成功
2级：初步管理	软件交付具有不确定性，顾客满意度低	a) 在项目级或组织级实施了软件开发、管理、和维护的基础过程； b) 没有在组织级形成标准过程
3级：改进管理	可以按时、保质交付顾客价值，顾客满意度得到提升	a) 在2级充分实施的基础之上进行持续改进； b) 软件开发活动能结合组织的业务战略、经营管理者需求以及外部监管需求； c) 建立并持续改进组织级的标准软件过程、技术资产和过程资产； d) 项目根据自身特性，选择和剪裁组织级标准软件过程，利用组织级技术资产和过程资产，实现项目目标，并回馈组织级资产

表1 成熟度等级的总体特征（续）

等级	结果特征	行为特征
4级：量化管理	基于稳定的交付质量与效率，可以快速帮顾客实现业务价值，顾客认可当前的软件能力	a) 在3级充分实施的基础上进行量化管理； b) 组织层面认识到能力改进的重要性，了解软件能力在业务目标实现、绩效提升等方面的重要作用，在制定业务战略时可获得项目数据的支持； c) 组织和项目建立了量化的质量与过程绩效目标，支持组织业务目标的实现； d) 建立了定量的过程绩效基线与过程绩效模型； e) 采用有效的数据分析技术，分析关键软件过程的能力，预测结果，识别和解决目标实现的问题以达成目标
5级：优化革新	可以帮助顾客实现业务创新，组织级的业务目标得到持续优化，顾客、股东、管理者、员工均认可组织的软件能力	a) 在4级充分实施的基础上进行优化革新； b) 组织将软件能力作为核心竞争力，利用软件开发创造更多的价值和提升改善组织的效率； c) 采用有效的数据分析技术，识别组织软件过程中影响业务目标的实现问题，通过针对性的软件过程改进，实现组织业务目标； d) 能将组织自身软件能力建设的经验作为行业最佳案例进行推广

4.4 软件能力成熟度模型的实践构成

4.4.1 模型的层次结构

本模型由4个管理域、17个实践域、132个实践组成，模型的层次结构见图3。

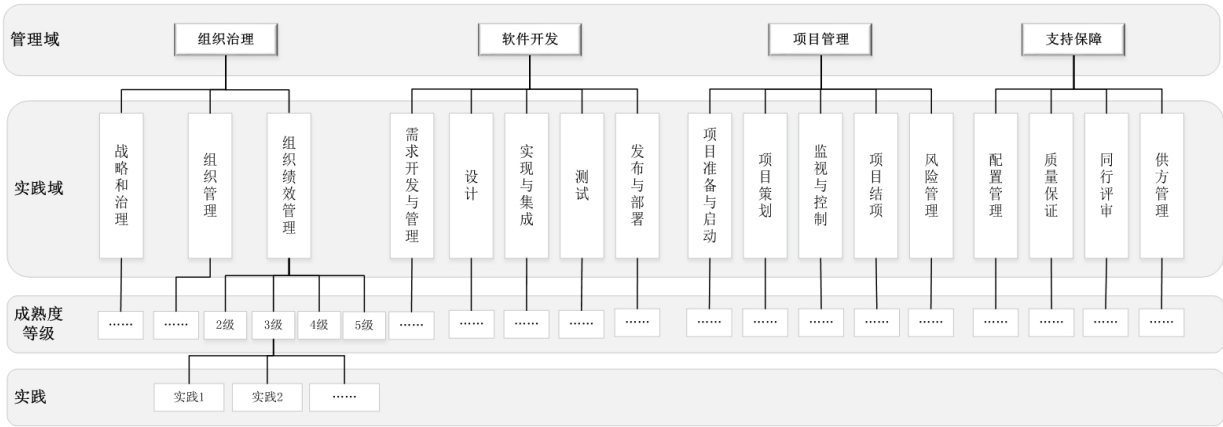


图3 软件能力成熟度层次结构

4.4.2 实践域的结构

实践域的结构如下：

- a) 概述：归纳说明本实践域的内容；
- b) 目标：说明本实践域要解决的问题；
- c) 相关实践域：与本实践域有关的其它实践域；

d) 实践列表：列出实践的等级编号、内容与解释。

4.4.3 实践的结构

实践的结构如下：

- a) 实践编号；
- b) 实践内容：实践的描述；
- c) 实践解释：对实践的详细说明。

4.5 实践域（实践）的等级要求

实践域（实践）的等级要求见图4。

组织治理			软件开发					项目管理					支持保障			
5		5				★									★	
4		4				★			4	4				★	★	★
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

注：★对应实践域在高成熟度等级可存在量化管理和优化革新级实践。

图4 实践域（实践）与成熟度等级的对应关系

4.6 软件能力成熟度模型的应用

软件能力成熟度模型可以应用于多个场景：

- a) 软件开发组织的自我改进。组织可以根据本组织的瓶颈问题与弱项，参考本模型选择实践域、选择等级进行改进，也可以参考本模型的所有实践域进行全面改进；
- b) 软件能力的自我评价。组织可以参考本模型对自己的软件能力进行评价，以识别改进点；
- c) 软件能力的外部评价；
- d) 顾客对供方的评价；
- e) 全面评价能力或局部评价能力。可以对照适用的全部实践域进行评价，也可以仅对部分实践域评价能力等级；
- f) 本模型适用的组织包含但不限于：
 - 1) 作为供方的开发组织；
 - 2) 存在大量分包的顾客的软件组织；
 - 3) 离岸外包的组织；
 - 4) 软件开发的研究机构；
 - 5) 应用软件开发、嵌入式系统开发、实时系统开发、科学计算软件开发、基础设施软件开发等。

4.7 能力成熟度评级

本文件是针对组织从第二级开始评价，对第一级不作评价。

5 组织治理

5.1 战略与治理

5.1.1 概述

组织基于使命、愿景和业务目标，确定软件开发过程工作的战略、方向和期望，确保软件开发过程和组织业务目标保持一致。最高管理层履行其领导的作用和承诺，为软件开发过程提供充足的资源，监督软件开发过程绩效，并基于量化分析，做出组织治理决策，确保软件过程绩效和业务目标的实现。

5.1.2 目标

战略与治理的目标如下：

- a) 指导最高管理层履行其在过程活动中的职责；
- b) 发挥最高管理层对过程活动的治理作用；
- c) 确保过程得到实施并促成组织业务目标的达成。

5.1.3 相关实践域

战略与治理的相关实践域包括组织绩效管理、组织管理。

5.1.4 成熟度等级

5.1.4.1 二级

实践 1 确定软件开发过程能力建设的范围和边界

组织可通过分析影响其软件开发过程达成预期结果的内外环境，理解相关方的需求和期望，识别组织软件开发过程的强项和弱项，确定软件开发能力建设的范围和边界。

5.1.4.2 三级

实践包括如下：

- a) 实践 1 定义、维护和沟通软件开发过程的组织方针

基于组织软件业务战略，最高管理层应设定、更新和传达软件开发过程的组织方针，为设定软件开发能力目标提供框架，指导软件开发能力的建设。

- b) 实践 2 确定组织软件开发能力目标，确保与组织业务目标保持一致

最高管理层制定软件开发能力目标，与软件开发能力的组织方针保持一致，目标得到监视和沟通，适时进行更新。

- c) 实践 3 确保资源的提供

最高管理层应提供足够的资源，确保实现目标所需的软件开发过程得到执行，以达成期望的结果。

- d) 实践 4 识别、采集和使用相关的信息，支持最高管理层进行能力绩效的监督和治理

组织识别软件开发过程能力绩效的相关信息，排定优先级，将过程实施和设定目标进行比较，确定过程的效率和有效性，最高管理层依据数据分析，做出决策并采取治理行动。

- e) 实践 5 分配职责，明确责任和权利

最高管理层确保软件开发过程的职责和权限得到分配，确保员工理解并接受被分配的职责和权限。最高管理层督促员工遵守组织的方针并为实施软件开发过程和达成预期结果而承担责任。

5.1.4.3 四级

实践 1 基于度量数据的统计分析，进行组织治理决策

组织对软件开发过程能力的绩效进行数据分析，最高管理层根据分析结果对软件开发能力建设改进做出决策，确定过程改进策略，适时更新软件开发能力的方针、目标。

5.1.4.4 五级

实践包括如下：

a) 实践 1 基于量化分析结果，做出软件能力建设优化的革新决策

组织利用数据和分析技术，使最高管理层可视化掌握软件开过程绩效和目标实现的可能性、存在的差距和问题，结合组织的业务目标，做出软件能力建设优化的革新决策。

b) 实践 2 实施优化革新，改善组织的价值创造能力

组织基于组织治理能力、软件开发能力、项目管理能力、支持保障能力的量化分析，结合组织能力绩效和业务目标，识别优化革新的机会，改善组织的价值创造能力。

5.2 组织管理

5.2.1 概述

组织按本文件的要求，建立、实施、维护组织治理、软件开发、项目管理、支持保障的标准过程，包括需要的过程及过程之间的接口，运行过程所需的资源配置，并持续改进，基于效率，建立组织资产库，以实现组织绩效目标。

5.2.2 目标

组织管理的目标如下：

定义和改进过程、配置资源，以支持组织实现其绩效目标。

5.2.3 相关实践域

组织管理的相关实践域包括战略和治理、组织绩效管理、需求开发与管理、设计、实现与集成、测试、发布与部署、项目策划、监视与控制、风险管理、项目结项、配置管理、质量保证、同行评审、供方管理。

5.2.4 成熟度等级

5.2.4.1 二级

实践包括如下：

a) 实践 1 建立组织治理、软件开发、项目管理和支持保障的所需的关键过程

基于组织软件业务战略，建立、实施、维护组织治理、软件开发、项目管理、支持保障的标准过程，包括需要的过程及过程之间的接口，最高管理层应确保将标准过程整合到组织的整个过程中，以实现组织绩效目标。

b) 实践 2 提供并维护软件开发所需的软件和硬件资源等

组织应基于软件开发运行所需的条件，提供相应的物理环境和相关设施、软件和硬件设备、信息和通讯技术等基础设施。

c) 实践3 提供并管理软件开发所需的人力方面的资源

组织应基于软件开发运行所需的人员技能要求，提供相应的团队组建规则，包括项目经理的选择、技术核心人员的选择。组织应对人员能力进行评价，适当时提供相应的培训。

d) 实践4 建立并维护组织过程资产库

组织应基于软件开发质量和效率，将可以复用的组件、可以参考的范例、可以操作的指南、可以进行决策和分析的模型、组织的标准过程定义等纳入组织过程资产库，应用并维护过程资产库。

5.2.4.2 三级

实践包括如下：

a) 实践1 识别强项、弱项和改进机会

组织基于绩效目标，评估过程运行的符合性和有效性、资源配置的合理性，找出强项和弱项，识别过程改进机会。这些弱项包括来自相关方的改进建议和分析结果。

b) 实践2 制定改进计划并实施

组织应将识别出的弱项进行记录、分类，排定优先级，根据预期价值和影响来选择改进项。改进实施需制定改进计划，确定改进目标并分配职责，保障改进效果。

c) 实践3 评价改进效果

组织基于改进目标，针对改进结果，做出改进效果评价。

d) 实践4 管理改进成果

组织应管理改进成果。

5.3 组织绩效管理

5.3.1 概述

组织基于业务目标，建立组织绩效管理，对软件开发、项目管理、组织治理和支持保障4大类价值交付活动进行测量、评价，利用绩效分析结果，聚焦组织业务目标实现，识别能力提升机会，以提高组织整体的绩效和能力。

5.3.2 目标

组织绩效管理的目标如下：

a) 通过组织绩效管理，确保组织能力水平达到即定标准；

b) 通过组织绩效管理，识别改进机会，监控改进成果，确认软件能力提升效果，保障组织业务目标实现。

5.3.3 相关实践域

组织绩效管理的相关实践域包括战略和治理、测试、项目策划、监视与控制、同行评审、质量保证、供方管理。

5.3.4 成熟度等级

5.3.4.1 二级

实践包括如下：

a) 实践1 明确绩效指标

组织应根据业务目标的需要，明确建立绩效指标，管理所需的软件能力。这些绩效指标在组织内充分沟通，指导和驱动组织成员的项目开发与过程改进活动，达到所需的绩效。

b) 实践 2 跟踪绩效指标达成情况

组织应根据建立的绩效指标，持续收集所需的度量数据，确定绩效指标的实际结果，评价其达成情况。对于存在显著差异的绩效指标，组织应进行原因分析，制定必要的改进计划并实施。

5.3.4.2 三级

实践包括如下：

a) 实践 1 根据业务目标确定组织的绩效目标

组织应根据业务目标的需要，分析组织过程体系中的过程与子过程对业务目标的影响，识别关键绩效，确定关键绩效的目标或目标范围。

b) 实践 2 度量组织能力与绩效

组织应根据确定的绩效目标，明确定义需要的度量规格，并按照度量规格持续收集度量数据，确定组织能力与绩效的实际结果，评估绩效目标的达成情况。

c) 实践 3 评价并改进组织能力与绩效

对于存在显著差异的组织能力与绩效目标，组织应进行基于度量数据的原因分析（原始度量数据和衍生度量数据），识别引起组织能力与绩效问题的过程体系中的原因和其它根本原因。针对这些原因，组织需要制定必要的改进计划并实施，包括具体的过程改进措施和其它改进措施，从根本上解决组织能力与绩效问题，提高实现业务目标的可能性（参考组织管理实践域）。

5.3.4.3 四级

实践包括如下：

a) 实践 1 建立组织绩效目标

组织应根据业务目标的需要，使用统计和其它量化技术及方法，分析组织过程体系中的过程与子过程对业务目标的影响，识别关键绩效，确定关键绩效的目标范围。

b) 实践 2 建立组织绩效基线

组织应使用统计和其它量化技术及方法，分析显著影响组织绩效目标的过程或子过程绩效数据，建立组织绩效基线。

c) 实践 3 建立组织绩效模型

组织应基于组织标准过程集与组织绩效模型基线建立组织绩效模型。组织使用组织绩效模型评估软件能力改进活动的改进效果。

d) 实践 4 采用统计分析技术和方法，预测组织绩效目标达成的可能性

组织应收集显著影响组织绩效目标的过程或子过程绩效数据，使用统计和其它量化技术及方法，持续分析和预测绩效目标达成的可能性。

5.3.4.4 五级

实践包括如下：

a) 实践 1 识别潜在优化革新机遇

当组织绩效不能满足业务目标的需要时，组织应使用统计和其它量化技术及方法，分析过程绩效度量数据，识别引起组织绩效不足的过程体系方面的原因或其它根本原因。针对这些原因，识别并确定具体的可以进行优化和革新的机遇。

b) 实践 2 优化革新，提升组织绩效

针对已识别的优化革新机遇，组织应使用统计和其它量化技术及方法，预测优化革新带来的组织绩效提升。根据预测结果，制定必要的绩效改进计划，分配所需的资源和预算。在实施绩效改进计划的过程中，持续收集所需的绩效度量数据，使用统计和其它量化技术及方法，评估改进效果。如果没有达到

期望的改进效果，应进一步分析原因，调整改进计划，直到实现期望的改进效果。达到期望的改进效果后，组织应形成新的过程资产，并推广到需要的其它领域，驱动组织业务发展。

6 软件开发

6.1 需求开发与管理

6.1.1 概述

需求开发和管理之间是同步实施、互为依赖的，两者共同作用使需求得到有效的管理、控制和优化。需求开发的目的是引出、分析并建立顾客、服务、软件的需求；需求管理的目的是确保需求与基于需求的计划、服务和工作软件之间的一致性。

6.1.2 目标

需求开发与管理的目标如下：

- a) 开发需求以满足顾客和相关方的需要；
- b) 建立软件开发的基础；
- c) 管理需求的变化，跟踪需求。

6.1.3 相关实践域

需求开发与管理的相关实践域包括设计、实现与集成、测试、配置管理、监视与控制。

6.1.4 成熟度等级

6.1.4.1 二级

实践1 收集需求、记录需求、理解需求

需求应形成文档，包括顾客明示的需求、潜在的期望，以确保可以追溯、交流、审查和书面确认。

6.1.4.2 三级

实践包括如下：

a) 实践1 挖掘需求

获取需求相关方明示的、潜在的和外部接口等方面的需求，以及必须遵守的限制与约束条件。

在理解业务场景基础上，通过需求调查，获取需求。调查范围覆盖顾客要求的业务范围以及软件生存周期的所有阶段。需求相关方应涵盖任何可能影响需求的关键人员或事物。与需求提供者达成对需求的一致理解。

b) 实践2 分析需求

通过对需求进行分析，识别软件需求。

通过对需求进行分析，识别软件的功能性和非功能性需求；确定软件需求的优先级别。识别具有共用特性的软件需求；解决软件需求中存在的矛盾或不一致等方面的问题。

分析需求结果应进行适当文档化，确保相关方能够有效沟通与理解。宜采用直观和易于理解方式描述，如：图形、表格、文字、页面原型等方式。

c) 实践3 确认需求

通过对需求的确认，与相关方达成对软件需求一致的理解。

采用有效的软件需求理解方法，使相关方能够充分地理解软件需求以及需求与软件需求之间的关系。解决相关方的疑问以及发现的问题。通过仿真评估、虚拟测试、模拟试验等手段确认软件需求，使得最终软件能在要求的环境中按预期运行。

d) 实践 4 管理需求变更

管理所有的需求变更，并识别需求与项目管理计划以及与软件的工作产品之间的不一致性。

理解需求变更的内容与范围，确保相关方对需求变更理解一致。分析需求变更影响的范围和开发团队人员，开发团队人员包括项目管理人员和相关开发人员等。开发团队人员承诺处理需求变更，并执行需求变更。维护需求与其它软件工作产品的可追溯性关系。

确定需求变更的优先级别、顺序和紧急程度，对于紧急变更应得到相关方的同意。对需求变更进行统计分析，识别需求变更产生的原因，并采取纠正措施。

e) 实践 5 双向可追溯、保持需求一致性

通过维护需求与软件生存周期中的工作产品的关系，确保双向可追溯。

根据商业、管理或技术等方面的需要，确定建立双向可追溯的范围。及时更新追溯关系，保持需求一致性。

对双向可追溯关系进行管理，宜采用适当的支持管理工具。

6.2 设计

6.2.1 概述

软件设计是对实现软件的结构、系统的数据、系统组件之间的接口以及所用算法的描述。软件设计又可细分为软件概要设计和软件详细设计。

6.2.2 目标

设计的目标如下：

设计满足需求的软件。

6.2.3 相关实践域

设计的相关实践域包括需求开发与管理、实现与集成、测试、发布与部署、配置管理、质量保证、同行评审、供方管理。

6.2.4 成熟度等级

6.2.4.1 二级

实践 1 制定满足需求的设计方案

组织应能够提供满足软件需求的设计方案，支持编码实现。将软件需求转化为软件体系结构，定义软件层次结构，确定软件组件间的接口等内容，将软件设计文档化。

6.2.4.2 三级

实践包括如下：

a) 实践 1 开展软件开发、采购、重用等分析，确定解决方案

组织应根据组织业务发展需要，能够识别设计出适用于整个软件生存周期中成本、进度、绩效和风险等方面取得平衡的备选解决方案集合，且应制定和维护与其业务目标一致的选择准则，以缩小备选解决方案集合的范围。

组织应充分理解软件需求，开展软件开发、采购、重用等分析，基于选择准则，评价和选择最佳的解决方案集合。确定解决方案，并用于软件开发过程中，解决方案应不断演进，保持与软件需求的一致性。

b) 实践 2 开展软件架构设计

组织应在设计阶段，结合架构需求，开展架构设计，包括：

- 1) 建立分块（子系统或模块）的结构关系，有关分块内的元素之间的规则，以及有关分块之间接口的规则；
- 2) 选择支持功能需求和非功能需求的架构模式，并将模式实例化，以建立软件架构；
- 3) 识别主要的内部和所有的外部接口；
- 4) 定义软件组件行为与交互作用；
- 5) 开发软件组件模板，或者类与框架；
- 6) 建立设计规则；
- 7) 识别主要可复用的方法和资源；
- 8) 确保可追溯的需求等。

c) 实践 3 开展软件接口设计

组织应结合在需求分析期间，识别的接口需求，进行接口设计，包括：

- 1) 定义接口准则，接口准则可以是组织过程资产的一部分；
- 2) 识别与其它软件组件相关联的接口；
- 3) 识别与外部项相关联的接口；
- 4) 依据准则，选择接口设计方案；
- 5) 将选定的接口设计与选择准则文档化等。

d) 实践 4 开展概要设计

组织应根据架构设计内容，开展概要设计，包括：

- 1) 定义架构风格和模式，如安全和性能要求、软件运行机制、多业务间的交互调用等；
- 2) 识别软件组件，如核心的业务逻辑设计；
- 3) 确定系统状态和模式，如数据库物理模型设计；
- 4) 定义主要的组件接口或连接，如：定义接口的输入输出和使用方法；
- 5) 开发待采用的算法；
- 6) 数据定义等。

e) 实践 5 开展详细设计

组织应根据概要设计内容，开展详细设计，包括：

- 1) 对确定的架构进行详细设计；
- 2) 对每一个软件组件进行详细设计，宜细化到可以被编码、编译、测试的软件单元。确保所有软件组件需求都被分配到了软件单元，并形成文档；
- 3) 对已识别的接口进行详细设计，并形成文档；
- 4) 规定要测试的软件单元的测试需求和进度安排，并形成文档；
- 5) 验证和确认需求。

f) 实践 6 开展设计评审

开展设计评审，识别并记录解决方案和设计文件的缺陷和问题，通过更新设计来解决识别的缺陷和问题。

6.3 实现与集成

6.3.1 概述

实现与集成分为实现、集成两个过程，两个过程可以重叠或相互作用，并且可以重复或循环地进行。实现是根据软件设计阶段的要求，实现设计，例如软件编程、调试、开展静态分析、代码审查，形成软件组件，并验证软件组件与设计说明的一致性。集成是将就绪的软件组件集成至更复杂的软件子系统或完整的软件系统，从而满足顾客的业务需求。

6.3.2 目标

实现与集成的目标如下：

- a) 开发和实现满足设计要求的软件组件；
- b) 将软件组件组装成软件，验证所集成的软件满足解决方案和软件设计的要求。

6.3.3 相关实践域

实现与集成的相关实践域包括需求开发和管理、设计、测试、风险管理、配置管理、供方管理。

6.3.4 成熟度等级

6.3.4.1 二级

实践包括如下：

a) 实践 1 建立编码规范

组织应根据业务发展需要，支持软件实现，建立编码方法规范，支持软件组件的实现。编码规范应包括命名规范、语法规则、书写规范、代码编译、代码测试和质量保证等要求。

b) 实践 2 编写代码，实现设计

组织应依据详细设计要求，对软件进行编程、调试，确保代码编写符合标准规范，通过静态分析等方法对代码进行检查，满足软件需求和软件设计。

c) 实践 3 单元测试

组织应对单个软件单元或相关单元组合进行测试。根据测试结果，及时修正软件相关单元，确保编码与设计文档一致。单元测试方法包括语句覆盖测试、分支覆盖测试、路径覆盖测试、边界值测试和特殊值测试等。

d) 实践 4 制定集成方案

组织应建立并维护支持软件组件集成所需的环境，集成的环境可以通过采购或开发获得。建立和维护集成软件和软件组件的规程和准则。根据集成规程和相关准则制定集成方案，以便将软件单元和软件组件进行集成。集成方案包括软件组件信息、集成环境、测试工具和设备、集成规程和准则、集成策略和方法、受影响的相关方、参与集成的人员安排等。

e) 实践 5 根据集成方案集成软件

组织按照软件集成方案、规程和准则集成软件组件。集成前，应确认每个软件组件符合其需求和设计要求及接口需求。集成过程中，按照集成方案进行集成，在此过程中，若出现问题，应记录并处理问题。

f) 实践 6 测试已集成的软件，确保其符合需求和设计

在软件集成方案确定的不同阶段评价已集成组件。使用软件集成规程、准则和环境，检查、集成和测试组件的功能、性能、适用性和就绪性。集成测试可以是集成方案和规程的一部分。

g) 实践 7 开发支持文档

组织应根据顾客和相关要求编制操作文档和用户文档。如安装手册、操作手册和维护文档等。

6.3.4.2 三级

实践包括如下：

a) 实践 1 执行代码评审

组织应制定并维护代码评审策略和规程，明确代码评审的类型、方法、工具和过程等。组织应采用适当的评审方式开展代码评审，如代码走查或审查等，识别并消除代码缺陷。

b) 实践 2 管理接口，保证完整性和一致性

组织应在整个软件生存周期中，评审并管理接口描述的覆盖范围、完整性和一致性。适用时，管理供方提供的软件组件及与其它相关接口之间的完整性和一致性。

c) 实践 3 验证接口满足要求

在集成之前，组织应确认软件组件的接口符合接口或连接描述。集成后，组织应评价和测试已集成的软件组件，确保组件内的接口满足要求。记录和使用评价结果，解决接口或连接不合规问题，并与受影响的相关方一起沟通结果。

6.4 测试

6.4.1 概述

测试是证实软件部件或软件系统的需求是否完成且正确，每一阶段工作软件是否实现在上一阶段规定的的需求或条件，以及最后的软件是否满足指定的需求的过程。

6.4.2 目标

测试的目标如下：

- a) 验证软件是否满足指定的需求；
- b) 确认软件在预定环境下是否实现其预期用途。

6.4.3 相关实践域

测试的相关实践域包括需求开发与管理、设计、实现与集成、配置管理。

6.4.4 成熟度等级

6.4.4.1 二级

实践包括如下：

a) 实践 1 建立软件系统测试方案

组织应建立软件系统的测试方案，确定软件系统测试的范围、职责、使用的工具和方法、进度和周期、规程和准则、测试资源、测试报告文档要求等。

b) 实践 2 测试准备

测试人员为完成测试活动，应建立、维护测试用例，并准备测试环境和必要的工具。

c) 实践 3 执行测试，记录结果

依据测试方案执行测试，记录测试执行的结果，对测试过程进行总结形成测试报告。测试结果以及对结果的处理应与相关方沟通并达成共识。

6.4.4.2 三级

实践包括如下：

a) 实践 1 建立和使用组织测试过程资产

组织应具备较完备测试运行管理机制，包括文件化的测试规程和准则、相关职责、测试工具、环境资源等，配备有独立的系统测试团队，各项测试活动按计划协调开展。

为提高测试的效率和质量，建立组织测试过程资产规范，将可供后续工作重用的测试方案、测试用例、仿真模块、驱动、数据等进行提炼，形成组织测试过程资产。

b) 实践 2 分析测试数据

应从组织级和项目级对测试数据进行综合分析，识别改进测试过程的机会。

6.5 发布与部署

6.5.1 概述

当软件经过系统测试证实其满足质量要求之后，应按照合同或相关规定进行发布和部署。发布是指对照交付清单自配置库提取将要交付给顾客的配置项，包括软件包、用户手册、帮助文档、安装手册、培训材料等，确认交付项完备无误，按规定的要求对交付项打包封装，并按照约定的方式提交顾客，完成交付项签收等相关事宜的过程。部署是指按部署计划将软件分发和安装到目标环境中，通过参数配置和调试等活动，实现软件在目标环境中稳定运行的过程。

6.5.2 目标

发布与部署的目标如下：

将软件部署到目标环境中，满足运行要求。

6.5.3 相关实践域

发布与部署的相关实践域包括实现与集成、测试、配置管理。

6.5.4 成熟度等级

6.5.4.1 二级

实践包括如下：

a) 实践 1 准备发布成果，提交给接收方

按相关要求形成交付项清单，自配置库提取要交付的配置项，并将软件成果物进行整理打包，形成发布记录。发布记录应包括发布日期、交付项、版本、接收发布的组织和发布授权等信息。发布成果应按约定的方式提交给接收方。

b) 实践 2 制定部署实施计划

应与相关方协商制定部署实施计划，部署实施计划至少应包括部署实施的对象、组织、相关责任人及其职责、部署实施活动、日程安排以及作业规程等。

c) 实践 3 按计划实施部署

应组织开发和实施人员按计划在目标环境分发和安装软件，并完成配置和调试等实施活动。对于部署不成功的软件实施回退或采取补救措施。

d) 实践 4 提供运行支持

应按相关要求和时间段，如试运行期，对软件运行提供技术支持，包括处理软件运行和使用中出现的各种情况，培训和辅导用户使用软件，获得反馈信息；必要时基于业务需要调整需求，完善软件质量。

在此时间段基于软件验收准则和约定的质量要求，准备提供相关文件和证据，配合相关方开展软件验收活动，及时处理软件运行期间识别出的不满足验收准则或条件的各种问题。对于无法满足的情况要与相关方协商处理意见。

应向相关方移交技术文件和资料，如软件配置文件、运行记录等，完成约定的由开发转运维的交接活动，如参与审核运行维护服务方案、手册等，指导编制软件运维操作规范文件等。

6.5.4.2 三级

实践包括如下：

a) 实践 1 建立和使用组织发布与部署过程资产

建立和使用组织发布与部署过程资产，对软件发布包进行统一收集、检查、存档、更新，对软件间的依赖关系进行记录与更新，对发布和部署相关计划进行审批和记录，以保障发布包版本的准确性。

b) 实践 2 跟踪监控发布和部署过程

组织应对软件的发布和部署过程进行跟踪监控，确保软件按要求完成发布和部署相关计划和流程，对于不合乎规范的发布和部署作业应及时指出并予以纠正。

c) 实践 3 管理变更

软件发布后的变更，应遵从组织的变更管理规程。

7 项目管理

7.1 项目准备与启动

7.1.1 概述

项目正式开始之前，应开展项目准备与启动工作，明确项目目标，确定项目范围，确定项目相关方，设置里程碑，分配必需的项目资源，进行必要的决策。

7.1.2 目标

项目准备与启动的目标如下：

- a) 进行必要的项目分析与决策，例如必要的可行性分析（项目价值等）、立项意向、项目决策等；
- b) 确定项目目标、项目范围、项目相关方里程碑计划以及所需资源等。

7.1.3 相关实践域

项目准备与启动的相关实践域包括项目策划、监视与控制、项目结项、风险管理。

7.1.4 成熟度等级

7.1.4.1 二级

实践包括如下：

a) 实践 1 确定项目目标

在项目准备与启动阶段，应明确项目需要实现的目标。项目目标应支持组织的业务战略和IT战略。项目目标包括但不限于价值目标、技术目标、业务目标等。

b) 实践 2 识别项目相关方

识别项目的相关方及其依赖关系和相关方主要人员或代表。

c) 实践 3 设置项目里程碑

设置项目里程碑，制定里程碑计划。

d) 实践 4 正式启动项目

正式启动项目，例如召开项目启动会，确保相关方在项目目标、项目范围、关键时间节点、资源、预算等方面达成一致。

7.1.4.2 三级

实践包括如下：

a) 实践 1 分析项目价值

根据业务战略、IT战略、市场需求（招投标内容）和行业发展需要，分析项目收益和价值，指导项目立项意向、项目决策。

b) 实践 2 进行立项决策

根据项目价值分析结果和立项准则，进行项目立项决策。

7.2 项目策划

7.2.1 概述

项目策划是开发和维护指导项目实施的各项计划，并获得各相关方对计划的承诺。具体包括：

- a) 根据项目目标、交付物，以及需求、分解任务等，对工作产品和任务进行估算，估算它们的规模、工作量、成本、工期等；
- b) 根据组织的标准过程和项目的约束条件，制定项目的预算、进度计划、资源计划、培训计划和质量管理计划等；
- c) 识别项目的相关方，根据项目对相关方的依赖关系，确定相关方参与项目的具体形式、活动、交付物、频率等，并获得相关方的承诺；
- d) 识别项目风险，并制定必要的缓解措施；
- e) 在项目进展中，根据需要，对项目进行重新估算，更新项目计划等。

7.2.2 目标

项目策划的目标如下：

建立有效的项目计划，为项目执行提供依据，提高实现目标的可能性。

7.2.3 相关实践域

项目策划的相关实践域包括项目准备与启动、监控与控制、风险管理、项目结项、组织绩效管理。

7.2.4 成熟度等级

7.2.4.1 二级

实践包括如下：

a) 实践 1 制定项目管理的策略与方法

根据顾客类型、技术需求、约束条件、项目组资源配置等情况，制定项目管理的策略与方法。包括但不限于项目变更管理模式，进度、质量、范围、成本的平衡策略，软件开发模式或方法，项目生存周期阶段，项目资源需要，风险应对策略、沟通机制、安全和保密措施等。

b) 实践 2 确定项目的范围

根据项目目标、需求、交付物等要求，确定项目的范围，作为项目估算、计划和监视与控制的边界。项目范围包括但不限于需求与目标、工作分解结构、产品或冲刺待办事项列表等。

c) 实践 3 估算项目的规模

常见的软件项目规模度量有：功能点、故事点、代码行、对象数量、模块数量、需求/特性数量和其它自定义的功能点等。规模是一种客观度量，应显著影响项目工作量、成本、工期等。软件规模估算方法通常需要事先定义，避免估算结果过多受到估算者主观经验的影响，导致项目策划过渡乐观（增加延期风险）或悲观（导致成本浪费）。除了项目初期需要估算软件规模，为项目策划提供客观依据，在

整个项目生存周期中，必要时需要进行重新估算，用以评估软件规模变化给项目带来的风险，例如，需求发生明显的变更时。

d) 实践4 估算项目工作量、成本和工期

根据软件规模估算的结果，估算项目的工作量、成本、工期等，为制定项目计划提供客观依据。项目组通常应用一些预先定义的估算模型，历史项目度量数据，专家分析判断等，或者它们相结合。自上而下和自下而上的估算通常一起使用，进行反复验证。为了提高项目估算的可信程度，项目组通常根据项目的假设条件，和相似项目的历史度量数据，反复校准估算数据。与软件规模估算相同，除了项目初期，在整个项目生存周期中，当软件规模或其它估算因子发生变化时，有时需要重新估算剩余的工作量，成本，工期等，例如，项目组人员显著减少或增加时，项目严重延期时。

e) 实践5 制定项目总体计划及相关计划

根据项目管理策略与方法、项目阶段、阶段目标、工作产品、估算结果等。制定项目总体计划，作为执行与跟踪项目的基本依据。当项目规模比较大，工期比较长，或项目组成员比较多时，应根据项目的具体需要，制定必要的相关计划，并确保所有计划之间协调一致。在整个项目生存周期中，有时需要重新修订项目整体计划。有些项目采用迭代的方式制定计划，例如敏捷开发。项目计划有多种载体，例如，文件、项目管理工具及其中记载的信息与数据、看板等。项目总体计划不仅仅是一个任务与日程清单，它应包含所有为了实现项目目标，而计划需要从事的工作和管理的事务。具体包括但不限于：

- 1) 任务，及任务之间的相互关系；
- 2) 预算；
- 3) 进度计划或时间计划；
- 4) 里程碑管理计划；
- 5) 需要使用的资源（人力资源和非人力资源）；
- 6) 项目组成员知识技能要求；
- 7) 项目环境，基础设施，后勤等需要；
- 8) 角色与职责；
- 9) 项目绩效目标；
- 10) 质量管理办法；
- 11) 项目风险和应对措施；
- 12) 项目信息和资料管理办法；
- 13) 项目相关方，及其依赖关系和管理办法；
- 14) 供方管理办法；
- 15) 需要的项目承诺等。

f) 实践6 制定项目进度计划和预算

根据项目的范围、生存周期阶段、估算结果、资源、依赖关系、约束条件等信息，制定项目进度计划和预算，作为执行与跟踪项目日程和费用支出的基本依据。进度计划包括但不限于项目的里程碑时间计划，任务及其时间与相互依赖关系，任务资源等。预算项目对费用支出的计划，包括但不限于总预算，不同阶段的预算，不同类型的预算等（例如，设备采购预算，软件开发工作量成本支出，外包预算等）。负责预算制定和控制的人员，可能是项目经理，也可能是组织的高层管理人员或财务管理人员。

g) 实践7 建立项目组

根据项目特征，项目管理策略与方法，和组织人员部门情况，组建项目团队，定义项目组组织结构，项目组运作与协作模式，对内对外沟通方式，分配角色，计划必要的培训。

h) 实践8 制定项目相关方管理计划

项目应识别项目生存周期各个阶段涉及的主要相关方及依赖关系，明确各相关方的义务和职责，需要参与的活动，提供的交付物和交付时间。项目应识别相关方的信息沟通方式需要，确定信息和沟通的

过程和策略，选用的沟通技术、方法和工具，制定项目沟通计划和信息管理计划。必要时，需要考虑不同地区、语言障碍和交叉文化。

i) 实践 9 制定项目资源管理计划

项目应识别项目开发所需的各种资源（设备、软件、工具、材料等），及其数量和时间。在计划阶段协调资源的可用性，所需的许可和授权，识别相关风险。

j) 实践 10 评审项目计划

项目应组织评审项目计划，并获得相关方的承诺。所有影响项目的计划都应该被评审和批准，并获得所有相关相关方对范围，目标，角色，职责，工期，时间，交付物等的共识和承诺。确保做出承诺的相关方全力兑现承诺。承诺包括对内的和对外的。

7.2.4.2 三级

实践包括如下：

a) 实践 1 剪裁组织级标准过程，建立项目已定义过程

项目应按照组织级过程剪裁指南，基于组织标准过程，根据项目目标、约束条件和其它特征，建立项目项目已定义过程，有效解决项目特殊问题。

b) 实践 2 使用组织度量库进行项目估算

项目应使用组织级度量库进行估算，提高项目估算的客观性。例如，组织定义的估算方法，组织建立的预测模型，组织级软件开发效率，单位成本等度量数据，历史类似项目的度量数据等。

c) 实践 3 使用组织过程资产策划项目

项目应使用组织级过程资产进行项目策划，控制项目风险。例如，基于组织标准规范与模版，借鉴历史项目的经验教训、度量数据、风险应对措施等，制定项目计划。

d) 实践 4 识别项目的关键依赖关系

项目应基于项目计划，借鉴组织的经验教训，识别项目外部和内部接口及其关键依赖关系，并与相关方协商管理这些关键依赖的措施，制定相关的沟通策略、管理方法和行动计划。

7.2.4.3 四级

实践包括如下：

a) 实践 1 使用统计和其它量化技术及方法，建立项目的质量与过程绩效目标

项目应使用统计和其它量化技术及方法，基于组织战略和绩效目标，识别顾客和其它相关方对项目过程绩效的要求，建立项目的质量与过程绩效目标。同时，项目应使用统计和其它量化技术及方法，基于历史绩效度量数据，分析实现这些目标的可能性或风险，并据此制定必要的措施，提高实现目标的可能性。

b) 实践 2 识别关键过程或子过程，及其过程绩效的统计和其它量化分析方法

项目应基于建立的项目的质量与过程绩效目标和项目已定义过程，选择实现目标的关键过程/子过程，识别其绩效属性，定义满足属性的测量项，并确定适合的统计和其它量化分析技术及方法。识别的绩效属性通常与绩效目标之间，存在直接或间接的统计相关性，它们包括但不限于：

- 1) 执行过程所消耗的工作量；
- 2) 执行过程的速度；
- 3) 执行过程所消耗的时间周期；
- 4) 作为过程输入的原材料数量或规模；
- 5) 执行过程人员的技术等级；
- 6) 执行过程所依托的环境类型或等级；
- 7) 过程输出的工作产品数量或规模；

- 8) 过程输出的质量属性，例如 Bug 数量、稳定性测量结果、性能测量结果等。
- c) 实践 3 组合项目过程，预测达成项目质量与过程绩效目标的可能性

7.3 监视与控制

使用统计和其它量化分析技术及方法，分析项目已定义过程满足项目质量与过程绩效目标的可能性，并选择项目已定义过程的最优组合，以此来策划项目，用最有效的方式控制实现绩效目标的风险。进行最优过程组合决策时，通常需要确定决策准则。

7.3.1 概述

监视与控制是通过各种监控方式了解项目进展，当发现项目实际状态与计划存在显著偏离时，采取合适的纠正行动，以提高实现项目目标的可能性。

监视和控制的方面包括但不限于：

- a) 规模；
- b) 工作量；
- c) 进度；
- d) 质量；
- e) 成本；
- f) 知识和技能；
- g) 利益相关方；
- h) 资源。

7.3.2 目标

监视与控制的目标如下：

- a) 监视项目进展，以实现项目目标；
- b) 让相关方及时了解项目状况；
- c) 客观地监视与控制项目各过程，支持项目决策。

7.3.3 相关实践域

监视与控制的相关实践域包括项目策划、风险管理、项目结项、组织绩效管理。

7.3.4 成熟度等级

7.3.4.1 二级

实践包括如下：

- a) 实践 1 跟踪项目总体计划及相关计划的执行情况

根据项目总体计划，跟踪各阶段目标、项目范围、任务执行、交付物等的执行情况。如果项目制定了一些子计划，这些子计划的执行情况应得到同样的跟踪。

- b) 实践 2 跟踪项目估算、计划数据、实际数据之间的偏差

对比估算数据，可包括规模、工作量、预算/成本、工期、进度、人力资源和其它资源等，跟踪实际度量结果，客观识别项目偏差。项目组在执行项目计划的过程中，收集数据，了解以下信息（但不限于这些信息）：

- 1) 软件规模的增长；
- 2) 项目实际消耗的工作量；

- 3) 项目实际的费用支出;
- 4) 项目到达某里程碑式家消耗的工作日;
- 5) 项目实际进度滞后或超前于计划的工作日;
- 6) 人力资源实际投入数量和等级;
- 7) 其它资源到位可供项目使用的数量和时间等。

对比实际数据与计划数据之间的差别,可以客观了解项目计划执行是否存在显著偏离情况。对比实际数据与估算数据之间的差别,可以客观了解制定计划的一些假设前提或约束条件是否发生了显著变化。有时估算数据和计划数据之间存在不同。

c) 实践 3 跟踪项目相关方的参与和承诺

跟踪相关方的参与和承诺,识别由相关方引起的影响项目按计划完成的问题,并及时协调相关方解决。

d) 实践 4 采取措施,纠正项目的显著偏差

当项目的实际结果与计划之间发生显著偏差时,采取措施进行纠正,直到问题解决。

7.3.4.2 三级

实践包括如下:

a) 实践 1 按照项目计划和项目已定义过程来管理项目

按照项目计划管理项目,期望项目的实际结果与计划尽可能一致。按照项目已定义过程管理项目,期望利用过程工具来解决项目遇到的问题。例如,定义和使用任务开始和结束准则,控制进度偏差和交付质量;定义和使用监控阈值,确保在合适的时间采取合适的纠正措施;定义和使用里程碑评审准则,控制里程碑通过带来的风险;针对某外部相关方及其依赖关系,制定和实施该相关方的管理办法,控制与其相关的风险等(参考项目策划实践域)。

b) 实践 2 管理项目的关键依赖

按照项目计划,解决项目关键依赖相关的问题。在项目进行过程中,持续识别新的依赖关系,并协调相关方采取必要的措施进行解决。

c) 实践 3 与项目相关方一起管理和解决项目问题

项目发生问题时,确定解决问题的相关方,并与相关方沟通。协调相关方采取可能的行动,直到问题得到解决。当问题无法解决时,按升级机制处理。需要相关方解决的问题包括但不限于:

- 1) 需求无法实现,需要需求提供者一起解决;
- 2) 重大软件设计缺陷,需要软件架构组专家帮助解决;
- 3) 项目后期的大型需求变更,需要顾客,组织高层,和商务一起协商解决;
- 4) 缺少关键资源,需要提供关键资源的相关方解决等。

7.3.4.3 四级

实践包括如下:

a) 实践 1 使用统计和其它量化技术及方法,管理项目的关键过程和子过程的绩效

根据项目的质量与过程绩效目标,使用统计和其它量化技术及方法,跟踪和分析关键过程和子过程的绩效,持续预测项目质量与过程绩效目标达成的可能性。在无法达成时,采取有效的措施,提高目标达成的可能性。

b) 实践 2 分析并解决影响项目质量与过程绩效目标达成的原因

使用统计和其它分析方法,识别并分析影响项目质量与过程绩效目标达成的原因。并根据原因,计划和实施必要的纠正措施,提高达成项目质量与过程绩效目标的可能性。

7.4 项目结项

7.4.1 概述

项目结项是从项目质量、工期进度、人力投入、技术发展、成本预算、项目风险等方面进行项目工作总结总结，总结项目目标达成，决定项目是否可以结束。项目结项还是总结项目经验、教训和成果，向组织贡献过程资产的阶段工作。项目结项后，将释放所有资源。

7.4.2 目标

项目结项的目标如下：

- a) 总结项目目标达成和项目工作完成情况；
- b) 总结项目经验、教训和成果，向组织贡献过程资产；
- c) 通过项目总结，评价项目绩效结果。

7.4.3 相关实践域

项目结项的相关实践域包括项目准备与启动、项目策划、监控与控制、风险管理。

7.4.4 成熟度等级

7.4.4.1 二级

实践包括如下：

a) 实践 1 总结项目

根据项目目标、交付物和其它承诺，总结项目完成情况，并决定项目是否可以结束。项目启动时明确了项目目标、各相关方角色职责以及项目交付要求。项目结项时，对项目工作的实际达成情况进行总结，核查各项工作是否已经全部完成，可交付成果是否达到交付要求和预期的满意程度。对于未达到预期的或未完成的工作应记录在案，明确完成时间，以便按备案工作计划跟踪解决。

b) 实践 2 跟踪并处理项目遗留问题

根据未达到预期的或未完成的备案工作，识别所有必须处理的遗留问题，协调相关方按备案工作计划的限期进行处理，跟踪问题处理过程直至问题关闭。

7.4.4.2 三级

实践包括如下：

a) 实践 1 总结项目经验教训和成果

开展项目回顾，复盘项目从开始到结束的完整过程，分析项目数据，归纳总结形成优秀实践和项目案例，反思项目过程中产生的严重的问题或错误，提出改进建议。

b) 实践 2 向组织提交项目过程资产

基于项目回顾、复盘和总结，贡献项目过程资产，包括但不限于项目文档、程序代码、度量数据、经验教训、工作成果等。

c) 实践 3 按项目评价准则，评价项目绩效

项目绩效评价的主要工作内容包括：在总结回顾项目工作全过程的基础上，分析项目的绩效和影响；评价项目的目标实现程度，对总结的经验教训提出对策和建议，持续改进提升项目管理工作。

7.5 风险管理

7.5.1 概述

风险管理是指提前识别潜在风险，分析风险的影响，计划并实施缓解风险的措施，规避和降低风险发生的可能性，从而降低风险的危害，或推迟风险的发生。

7.5.2 目标

风险管理的目标如下：

针对项目风险，项目组或组织采取既定的有效措施，降低不利影响，以保证提高实现项目或组织目标的可能性。

7.5.3 相关实践域

风险管理的相关实践域包括项目准备与启动、项目策划、监视与控制、项目结项。

7.5.4 成熟度等级

7.5.4.1 二级

实践包括如下：

a) 实践 1 识别潜在的风险

在风险管理中，首先识别并记录潜在的风险，以使项目或组织提前制定风险处置措施，避免或最大限度地减少风险的影响。

b) 实践 2 分析已识别的风险

基于已识别的风险，分析风险发生的可能性、预期发生的时间、影响范围、风险起因等因素。

c) 实践 3 监视与控制已识别的风险

对已识别的风险进行监控，掌握风险发生的条件，并及时与相关方沟通状态。

7.5.4.2 三级

实践包括如下：

a) 实践 1 根据已定义的风险来源和类别，识别和管理风险

基于项目风险管理经验，通过多种方法对风险进行分类和判断，充分有效识别潜在风险，提高管理风险的效率。例如，持续多方面收集组织内各个项目或者是同类项目的问题和风险数据，建立组织级风险库，供新项目参考使用，形成风险管理机制和规程。

b) 实践 2 定义和使用优先级管理风险

基于组织级风险库等风险经验，定义和使用一定的参数系数，来分析和处理风险，将风险按照优先级设置，集中有限资源缓解或降低高级别风险，提高风险管理的资源利用效率。

c) 实践 3 制定风险管理策略和计划

基于风险管理机制和规程，制定和实施风险管理的策略，形成风险管理计划，提高风险管理的效率和效果。

d) 实践 4 实施风险管理策略和计划

开展风险管理工作，基于风险监视和控制，掌握风险发生概率，降低风险发生的可能性，并能够及时处置潜在风险及发生的风险，以减小风险给项目带来的负面影响，或者推迟风险发生的时间。

e) 实践 5 风险管理持续改进

开展风险处置工作总结，基于风险处置结果，及时优化和调整风险处置策略，完善组织级风险库。

8 支持保障

8.1 配置管理

8.1.1 概述

配置管理是通过技术手段对软件产品及其开发过程和生存周期进行控制、规范的一系列措施，配置管理的目标是记录软件产品的演化过程，保证软件配置项的完整性、一致性和可追溯性。配置管理主要包括配置项标识、配置管理策划、配置管理系统准备、配置项和基线管理、配置审核、配置状态报告等活动。

8.1.2 目标

配置管理的目标如下：

确保软件生存周期过程产生的工作产品的完整性、一致性、可追溯性。

8.1.3 相关实践域

配置管理的相关实践域包括项目策划、监视与控制、需求开发与管理、设计、实现与集成、测试、发布与部署。

8.1.4 成熟度等级

8.1.4.1 二级

实践包括如下：

a) 实践 1 识别和标识配置项

应制定配置项识别和标识策略，根据制定的策略识别出需要控制、管理和使用的配置项并进行标识。

b) 实践 2 使用配置管理工具实现配置项版本可追溯

对配置项进行版本控制，确保软件开发者在软件生存周期中各个阶段都能得到精确的软件产品配置。

c) 实践 3 建立和维护配置项状态信息

通过配置管理工具，建立并维护配置项信息，对相关方访问配置项的权限进行授权和管理，确保对配置管理信息的合理访问。

8.1.4.2 三级

实践包括如下：

a) 实践 1 建立配置管理系统

组织应建立配置管理系统，以实现配置项的统一管理和访问控制，系统应具备恢复先前版本、配置项或基线的能力。

配置管理系统可包括配置管理规程、职责权限、配置管理工具、硬件设备资源、配置库结构、备份及恢复方案、配置基线等。

b) 实践 2 策划项目配置管理

组织应编制项目配置管理计划，包括：必要的配置管理活动、实施配置管理活动的策略和进度安排、负责实施配置管理活动的组织等。配置管理计划应形成文档并加以实施。

c) 实践 3 建立和发布基线

组织应根据配置管理规程的要求建立和发布基线，提供给相关方使用，基线发布前应得到授权或批准。

d) 实践 4 管理配置项变更

配置项的变更应纳入变更管理的控制，组织应遵循变更管理规程对配置项进行变更，包括记录变更请求、分析和评估变更、变更审批、验证和发布等。

e) 实践 5 基于相关方需要，提供配置管理报告

组织应定期向相关方报告配置管理情况，报告内容至少包括配置管理基本情况、配置项和基线的内容及其变更状态、配置项交付记录、配置库备份情况等。

f) 实践 6 执行配置审核

组织应按计划的时间执行配置审核，以保持配置项、基线、变更和配置管理系统内容的完整性、一致性和正确性。

8.2 质量保证

8.2.1 概述

质量保证可以根据适用的过程描述、标准和规程，来检查已执行的过程和所产生的工作产品，并对发现的问题进行跟踪和解决，以最大限度地提高业务效益和顾客满意度。

8.2.2 目标

质量保证的目标如下：

验证并改进已执行的过程和所产生的工作产品的质量。

8.2.3 相关实践域

质量保证的相关实践域包括需求开发与管理、设计、实现与集成、测试、同行评审、项目策划、监视和控制。

8.2.4 成熟度等级

8.2.4.1 二级

实践包括如下：

a) 实践 1 制定项目质量保证计划

建立并维护质量保证活动和任务计划，确定项目质量保证人员和职责，并形成文档。

b) 实践 2 客观的评价所选择的已执行过程和工作产品

根据已定义的过程和适用的标准，在整个项目周期内，采用客观的评价方法和明确的评价准则，评价选定的已执行过程和工作产品，识别和记录发现的不符合，并对发现的不符合提出改进意见。

c) 实践 3 记录、沟通、跟踪过程和工作产品的不符合

与相关方沟通不符合，跟踪不符合的处理情况。制定不符合的解决方法，针对不符合记录分析质量趋势，形成报告。

8.2.4.2 三级

实践包括如下：

a) 实践 1 建立质量保证过程，确保质量保证目标的达成

建立并维护质量保证规程，保证软件以及相关过程符合规定要求。质量保证规程包括但不限于：质量保证人员和其他相关人员的职责，质量保证活动的资源、工具和方法，评审机制，项目质量监控项，项目结项的检查时间、检查项、项目结项所需的报告等。

b) 实践 2 分析质量保证活动记录，识别改进机会

记录并维护与质量保证活动相关的信息，提交改进建议。

8.3 同行评审

8.3.1 概述

同行评审是软件开发过程中一项重要而有效的验证、确认活动，通过审查、结构化走查、审计或其它评审方法来实施，客观地检查工作产品，尽早、有效地识别并清除工作产品中的问题和缺陷。

8.3.2 目标

同行评审的目标如下：

尽早发现工作产品中存在的问题和缺陷，从而有效地提高工作产品质量，减少返工成本。

8.3.3 相关实践域

同行评审的相关实践域包括需求开发与管理、设计、实现与集成、测试、发布与部署、质量保证。

8.3.4 成熟度等级

8.3.4.1 二级

实践 1 对工作产品进行评审，记录并处理发现的问题和缺陷

应对工作产品进行评审，并记录评审结果，处理评审中发现的问题和缺陷。

8.3.4.2 三级

实践包括如下：

a) 实践 1 建立并维护同行评审规程

建立并维护同行评审规程，包括定义同行评审的类型、工作产品评估准则、选择参与者与分配角色的要求、同行评审的步骤、定义同行评审相关模板等内容。

b) 实践 2 制定同行评审计划

选择要进行同行评审的工作产品。在选择工作产品时应考虑工作产品的优先级别、采用的评审技术和方法等。应根据同行评审类型，制定同行评审计划。

c) 实践 3 解决同行评审过程中发现的问题和缺陷

对评审中发现的问题和缺陷提供解决方法，并在计划时间内解决，同时向相关方汇报问题和缺陷解决的进展和结果。

d) 实践 4 分析同行评审的数据

对同行评审的问题、缺陷等数据进行分析，根据分析结果改进同行评审过程，提升同行评审效率。

8.4 供方管理

8.4.1 概述

当组织开发的软件系统中的某部分需求需通过第三方的软件产品（或组件）或软件服务实现时，组织应确定并管理供方，对供方进行评价、选择、开发、使用和控制。

8.4.2 目标

供方管理的目标如下：

a) 通过供方协议管理，确保供方交付物满足要求；

b) 通过供方管理，提升供方满足交付要求的可能性。

8.4.3 相关实践域

供方管理的相关实践域包括需求开发和管理、设计、实现与集成、监视与控制、组织绩效管理。

8.4.4 成熟度等级

8.4.4.1 二级

实践包括如下：

a) 实践 1 建立供方协议

建立供方协议，通过协议和供方进行约定，供方协议可以是采购方和供方签署的合同、许可证、备忘录等。

b) 实践 2 执行供方协议

确保按照供方协议约定进行交付，对违反协议约定的，应及时采取纠正措施。供方协议需定义违反协议或不符合协议时所应采取的措施以及定义违约责任。

c) 实践 3 验收供方交付物

确保按照供方协议约定进行验收。

8.4.4.2 三级

实践包括如下：

a) 实践 1 选择满足服务要求的供方

组织在合格供方目录内选择能满足需求的供方。选择供方时应考虑：采购范围和要求、采购策略、组织内部政策、法规要求。在某些情况下，组织可根据经验、专业知识、过去的表现等预先验证首选供方的资格。从首选供方列表中选择最终供方可以大大减少招投标所需的工作量 and 时间。

b) 实践 2 评价供方实施绩效

组织根据供方承诺及协议监视供方的进度和绩效（如工作量、响应速度、技术能力），验证供方的交付物，形成并记录对供方的评价结果。

参 考 文 献

- [1] GB/T 8566-2007 信息技术 软件生存周期过程
- [2] GB/T 32423-2015 系统与软件工程 验证与确认
- [3] ISO/IEC/IEEE 12207:2017 Systems and software engineering - Software life cycle processes
- [4] ISO/IEC 33001:2015 Information technology - Process assessment - Concepts and terminology
- [5] ISO/IEC 33002:2015 Information technology - Process assessment - Requirements for performing process assessment
- [6] ISO/IEC 33003:2015 Information technology - Process assessment - Requirements for process measurement frameworks
- [7] ISO/IEC 33004:2015 Information technology - Process assessment - Requirements for process reference, Process assessment and maturity models
- [8] CMMI Model V2.0

