



中华人民共和国国家标准

GB/T 41865—2022/ISO/IEC 26550:2015

软件与系统工程 产品线工程与管理参考模型

Software and systems engineering—Reference model for product
line engineering and management

(ISO/IEC 26550:2015, IDT)

2022-10-12 发布

2023-05-01 实施

国家市场监督管理总局 发布
国家标准化管理委员会

目 次

前言 III

引言 IV

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 从单系统工程与管理到产品线工程与管理 5

5 产品线工程与管理参考模型 8

6 产品线工程与管理的双生存周期和双过程组 9

7 领域工程和应用工程内部及相互关系 19

附录 A（资料性） 有关产品的更多信息 26

参考文献 27

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。

本文件等同采用 ISO/IEC 26550:2015《软件与系统工程 产品线工程与管理参考模型》。

本文件做了下列最小限度的编辑性改动：

——根据国际标准的最新版本以及国家标准的采标情况，本文件在全文范围内对原文中引用的标准编号进行了部分修改，并补充对应国家标准编号，原引用 ISO/IEC 12207:2008 已更新为 ISO/IEC/IEEE 12207:2017，国家标准已等同采用为 GB/T 8566—2022，原引用国际标准 ISO/IEC/IEEE 15288:2015 国家标准已等同采用为 GB/T 22032—2021，原引用国际标准 ISO/IEC 15940:2006 已更新为 ISO/IEC 15940:2013，国家标准等同采标为 GB/T 30972—2014，这些修改不影响本文件技术内容；

——第3章术语和定义原文缺少 3.22，本文件将原 3.23～3.29 的条目编号改为 3.22～3.28。

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：中国电子技术标准化研究院、北京航空航天大学、江苏赛西科技发展有限公司、深圳赛西信息技术有限公司、中国船舶重工集团公司第七〇九研究所、中国科学院软件研究所、国家应用软件产品质量监督检验中心、重庆市软件评测中心有限公司、上海市软件评测中心有限公司、成都信息工程大学、北京无线电测量研究所、中国航天科工集团第二研究院七〇六所、成都淞幸科技有限责任公司、山东山科数字经济研究院有限公司、浪潮通用软件有限公司、北京工业大学、苏州赛腾精密电子股份有限公司、山东省计算中心(国家超级计算济南中心)。

本文件主要起草人：于秀明、张旻旻、黄继红、王威、张文渊、杨立、舒红平、赖祥伟、朴希望、孙金洋、刘岩、楼莉、李旺、张丽、王冠群、李刚、刘潇健、崔晓峰、张星星、吕雪、刘文红、蒋建民、王铁军、王亚强、李敏、郝琳、侯敏、孙海龙、刘言维、王腾江、孙海旺。

引言

自 20 世纪 90 年代以来,软件与系统产品线(Software and System Product Line,SSPL)越来越受到全球关注。SSPL 工程与管理创建、利用和管理一个通用平台,以更低的成本、更短的上市时间和更好的质量开发一系列产品(如软件产品、系统架构)。

本文件提供一个参考模型,包括软件与系统产品线工程与管理的关键过程以及过程之间关系的抽象表示。产品线工程与单系统工程的区别在于它有两个关键特性,即需求领域工程和应用工程两个生存周期过程,以及需要明确的可变性定义。领域工程的目标是定义和实施产品线中成员产品共同使用的领域资产,应用工程的目标是利用包括通用和可变资产在内的领域资产的开发应用。领域工程明确定义反映不同市场和细分市场特定需求的产品线可变性。可变性可嵌入到领域资产。在应用工程中,领域资产根据已定义的可变性模型进行配置。

SSPL 工程与管理的参考模型可用于后续的标准化工作,为软件与系统产品线工程创建:高度抽象的标准(如产品管理、范围划分、需求工程、设计、实现、验证与确认、组织和技术管理),中等抽象的标准(如配置管理、可变性建模、风险管理、质量保证、测量、评估和资产储存),以及详细抽象的标准(如构造、配置机制和资产挖掘)。

软件与系统工程

产品线工程与管理参考模型

1 范围

本文件是软件与系统产品线工程与管理系列标准的基础。

本文件的范围包括：

- 为软件与系统产品线工程与管理提供特定的术语和定义；
- 为软件与系统产品线工程与管理的总体结构和过程定义一个参考模型，描述产品线参考模型组件如何组合；
- 定义产品线参考模型组件之间的关系。

本文件不描述任何与软件与系统产品线工程与管理相关的方法和工具。这些方法和工具在产品线工程与管理系列标准(ISO/IEC 26551～ISO/IEC 26556)中进行描述。本文件不涉及 ISO/IEC/IEEE 24765:2010 所述的术语和定义，该文件提供了适用于所有系统和软件工程工作的通用词汇表。

本文件中“产品”指软件系统，或硬件和软件系统组成的“系统级产品”。本文件对仅涉及硬件系统的产品线工程与管理可能有用，但其并非特定为硬件产品线而制定。尽管硬件通常与软件一起构成产品，但本文件不旨在为构成产品的硬件的工程化、生产、存储、物流和管理提供帮助，此类过程属于其他学科(如机械学、电子学)。

注：附录 A 提供有关产品的其他信息。

本文件包括产品线参考模型、术语和定义，并以 GB/T 8566—2022/ISO/IEC/IEEE 12207:2017、GB/T 22032—2021/ISO/IEC/IEEE 15288:2015、GB/T 30972—2014/ISO/IEC 15940:2013、ISO/IEC 14102:2008 的结构作为基线。

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

应用架构 application architecture

约束产品线内特定成员产品的架构性结构和规则(如通用规则和约束条件)。

注：应用架构描述产品线内特定成员产品的高级设计。产品线中的成员产品的应用架构重用(可能经修改后)领域架构的通用部分，并绑定领域架构的可变部分。大多数情况下，成员产品的应用架构需开发特定应用的可变性。

3.2

应用资产 application asset

特定应用工程过程(如应用实现)的输出。此类资产可用于应用工程的其他生存周期过程，并可根
据产品管理决策调整为领域资产。

注 1：应用资产包括需求、架构设计、组件和测试。与需要在产品线内支持多种应用的大规模定制领域资产不同，多数应用资产不包含可变性。然而，应用可具有可变性（例如，最终用户可以通过在运行时绑定应用可变性，从而能够批量定制他们所使用的应用）。应用资产也就相应地具备了可变性，但应用资产的可变性仅服务于特定的应用，应用资产就是为这个应用而创建的。所以，应用资产可变性的范围通常比领域资产可变性的范围窄得多。

注 2：应用资产并非现货或随时可供使用的物理产品，物理产品（如机械零件、电子元件、线束、光学镜片）根据其各自学科的最佳实践进行存储和管理。应用资产有其自身的生存周期；GB/T 22032 可用于其生存周期管理。

3.3

应用设计 application design

导出符合领域架构的单个应用架构的应用工程过程。

3.4

应用工程 application engineering

通过重用符合领域架构的领域资产和绑定平台的可变性，实施和管理应用资产和产品线成员产品的一系列过程组成的生存周期。

注：传统意义上的应用工程指开发单个的产品，不需要领域资产的战略重用，也不需要明确的可变性建模和绑定。

3.5

应用实现 application realization

基于应用架构、应用资产和领域资产开发应用资产（其中一些应用资产可能来自于领域资产）和成员产品的应用工程过程。

3.6

资产库 asset base

从领域和应用工程产生的可重用资产集合。

3.7

资产范围界定 asset scoping

确定潜在领域资产和评估资产投资回报的过程。

注：资产范围界定过程中产生的信息，以及产品范围界定和领域范围界定过程中产生的信息，可用于确定是否将产品线引入组织。资产范围界定在领域范围界定之后进行。

3.8

绑定 binding

根据领域可变性模型和应用可变性模型，对领域资产和应用资产的相应变体进行实例化的任务。

注：执行绑定是应用绑定定义，使用领域和应用可变性模型从领域和应用资产中生成新的应用。

3.9

通用性 commonality

产品线所有应用共有的功能性和非功能性特性的集合。

3.10

领域架构 domain architecture

参考架构 reference architecture

产品线架构 product line architecture

描述软件与系统产品线高层级设计的核心架构，包括约束软件与系统产品线内所有成员产品的架构性结构和构造（如通用规则和约束条件）。

注：产品线成员产品的应用架构重用（可能经修改后）领域架构的通用部分，绑定领域架构的可变部分。成员产品的应用架构可（但不要求）提供可变性。

3.11

领域资产 domain asset

核心资产 core asset

领域工程生存周期过程的输出,在应用工程中可被重用于产品的生产。

注 1: 领域资产可包括领域特征、领域模型、领域需求规格说明、领域架构、领域组件、领域测试用例、领域过程描述,以及其他资产。

注 2: 在系统工程中,领域资产可以是在后续系统设计中可重用的子系统或组件。领域资产需考虑其原始需求和技术特征。领域资产包括但不限于用例、逻辑原则、环境运行数据,以及从以前项目中获得的风险或机遇。领域资产并非现货或可随时供使用的物理产品。物理产品(如机械零件、电子元件、线束、光学镜片)根据其各自学科的最佳实践进行存储和管理。领域资产有其自身的生存周期;GB/T 22032 可用于其生存周期管理。

3.12

领域工程 domain engineering

用于规定和管理产品线通用性和可变性的一系列过程组成的生存周期。

注 1: 领域资产是在领域工程过程中开发和管理,在应用工程过程中复用。

注 2: 根据领域资产的类型,即系统领域资产或软件领域资产,所使用的工程过程可由相关规范决定。

注 3: IEEE 1517—2010 第 3 章将领域工程定义为基于可复用的方法,用于为一类系统、子系统或成员产品定义范围(即领域定义)、描述结构(即领域架构)和创建资产(如需求、设计、软件代码、文档)。

3.13

领域范围界定 domain scoping

识别和界定功能领域的子过程。该子过程对构思产品线,以及提供足够的复用潜力以调整产品线创建是非常重要的。

3.14

特征 feature

最终用户和其他利益相关方可理解的,所关注系统抽象的功能特性。

注: 在系统工程中,特征是利益相关方需求的综合。这些特征是建立技术需求基线的输入之一。

3.15

成员产品 member product

应用 application

属于产品线的产品。

3.16

产品线 product line

产品族 product family

一组产品或服务,具有相同的明确定义和管理的通用和可变的特征,且依赖相同的领域架构满足特定市场的通用和可变需求。

3.17

产品线架构 product line architecture

同领域架构。

3.18

产品线平台 product line platform

由产品线架构、配置管理计划和领域资产组成,使应用工程能有效地重用并构建一组衍生产品。

注: 平台有其特定的生存周期。GB/T 22032 可用于生存周期管理。

3.19

产品线参考模型 product line reference model

领域和应用工程生存周期过程、过程的角色和关系,以及在产品线工程与管理中生产、管理和使用的资产抽象的表示。

3.20

产品线范围界定 product line scoping

定义产品线内的成员产品以及产品间主要的通用和可变特征的过程;从经济视角分析产品,控制和规划产品线及其产品的开发、生产和销售。

注:产品管理首先负责产品线范围界定。

3.21

产品范围界定 product scoping

产品线范围界定的子过程,即确定产品路线图,包括:目标市场;产品线组织开发、生产、营销和销售的产品类别;为产品线组织实现长期和短期商业目标,产品宜具有的公共和可变的特征;产品上市时间表。

3.22

可变性 variability

可区分产品线成员的特性。

注1:成员产品间的差异包含多个方面,例如功能性、质量属性、成员产品的使用环境、用户、约束条件以及实现功能性和质量属性的内部机制。

注2:区分系统和软件可变性与产品线可变性的概念很重要。由于软件系统具有内在可塑性、可扩展性或特定使用周境的可配置性,因此部分或全部由软件构成的系统都被认为具有软件可变性。产品线可变性关注的是产品管理明确定义的可变性。本文件主要涉及产品线可变性。

3.23

可变性约束 variability constraint

变体与可变点之间,两个变体之间,以及两个可变点之间的约束关系。

3.24

可变性依赖 variability dependency

可变点和变体集之间的关系,这种关系表明可变点包含了关于变体的一种决策。

注:可能存在两种可变性依赖:1)可选的可变性依赖:可选择地依赖于可变点的变体可以是产品线成员产品的一部分;2)强制的可变性依赖:若为成员产品选择了可变点,则需为成员产品选择依赖于可变点的变体。

3.25

可变性管理 variability management

可变性的管理任务,包含两个维度:可变性维度和资产维度。

注:可变性维度的可变性管理包括监控整个产品线级别的可变性,创建和维护可变性模型,确保可变性模型之间的一致性,管理产品线所有可变性和约束依赖,以及管理可变性模型与相关领域和应用资产(如需求模型、设计模型)之间的可追溯性。资产维度的可变性管理包括管理每个领域和应用资产中的可变性影响,即在资产的何处产生特定的可变性,以及资产在该处可以采用何种不同状态。这两个维度在本质上是互补的,即二者都是可变性管理所需要的。

3.26

可变性模型 variability model

产品线可变性的明确定义。

注:可变性模型提出了可变点、可变点的变化类型、可变点上可以采用的变体、可变性依赖以及可变性约束。可变性模型可与其他模型(如需求或设计模型)正交或集成。可变性模型包含两种类型:应用可变性模型和领域可

变性模型。

3.27

可选项 variant

用于实现特定可变点的替代方案。

注：每个可变点需对应一个或多个变体。每个变体需关联一个或多个可变点。特定产品变体的选择和绑定决定其特定可变性的特性。

3.28

可变点 variation point

产品线环境中，产品、领域资产和应用资产的特定可变特性的表示。

注：可变点表明产品线的变化。每个可变点宜至少包含一个变体。

4 从单系统工程与管理到产品线工程与管理

单系统工程与管理是软件与系统产品概念化和开发的主要方法。本章首先概述了软件与系统产品公司在单系统工程与管理方法时面临的一些主要挑战。可变性管理被认为是最具挑战性的领域，相关内容见 4.1。本章还阐述了单系统工程与管理与产品线工程与管理之间的主要差异，理解这些差异是组织级从单系统工程与管理向产品线工程与管理成功转变的关键。

4.1 单系统工程与管理面临的挑战

在不适用的环境下过度使用单系统工程与管理，给客户、最终用户、供应商带来各种各样问题。例如，客户可能因误判其需求的独特性而申请使用价格昂贵的定制系统，实际上，价格合理的商业化产品可满足其需求。对终端用户而言，软件系统复杂、提供的功能繁多，可能导致难以找到实际需要的功能；供应商可能销售几个相互关联的产品，这些产品的界面和体验完全不同，且不能交互操作，甚至是对于相同客户也是如此。

当使用单系统工程时，单个产品的供应商经常面临以下问题：工作量和成本被低估、生产力被高估、必备功能缺失、产品进度或质量目标未实现、客户满意度总是低于预期。工作量被低估或生产力被高估的原因是组织没有类似产品的开发经验，或者即使曾开发过类似产品，由于组织内部门之间的竞争原因，也可能导致开发过类似产品的部门不愿共享其经验和其他可重用资产。成本估算不准确和预算通常固定导致进度波动、功能缺失和/或质量问题。质量问题也可能是由于缺乏重用文化，因为相对于重用经充分测试的组件开发的软件，新开发的软件通常缺陷密度更高。

单个产品供应商面临最重要的问题通常是如何应对适当的可变性。在此情况下，可变性需求一般产生于与不同用户的交互活动。供应商常运用一种或多种看似简便但无效的策略来应对出现的可变性。例如，供应商可能会在单个产品中引入越来越多的参数（部分是最终用户可见），在源代码中通过越来越多的 if-then-else 条件选择语句来处理运行时参数，从而为产品引入可变性。结果源代码行数增多，源代码愈发复杂，难以理解和维护，软件的可测性（通常还有性能）也会变差。此外，供应商可从现有产品基础上创建新产品分支来应对新客户的变化性需求，修改新产品的源代码，在时间和其他资源允许的时候将修改后的源代码合并回主线，最后删除原分支。但这种方法成本昂贵、容易出错，并且分支的改动及合并会使得主线的源代码更加复杂，从长期看，这种方式需要进行昂贵的周期性重构。最坏的情况是，供应商最终得到众多部分克隆的产品，却没有可维护的主线源代码。这种应对可变性的无效策略也使软件开发人员的工作变得乏味，使得员工流动率增高。

总之，使用单系统工程与管理方法的产品公司，可能最终获得高复杂度、低质量的产品，低生产率、高员工流动率和低于预期的客户满意度。

产品线工程与管理是解决此类问题的一种可行方案，但并不能解决所有问题。若对产品线工程与

管理理解或执行不当,可能导致大量投资却无法实现预期收益。因此,本文件概述了产品线工程与管理是什么,供应商如何利用它建立和管理可变性、降低成本和产品复杂性,通过战略性、程序化地创建使用领域资产来提高生产率和产品质量,缩短上市时间,通过大规模产品定制和更准确地估算进度与成本以提高客户满意度。

4.2 可变性管理

在单系统工程与管理中,知识重用非常重要。产品线工程与管理与单系统工程与管理从根本上来说是不同的,产品线工程与管理要明确考虑多种产品,以及产品内部和产品之间的差异。对可变性进行百分百的事先规划是不可能的,仍然可能出现新的可变性需求(例如竞争者的预料之外的产品),但是多数可变性需求宜基于目标市场、可用技术、竞争产品和其他因素的详细分析。产品线各成员的共性与可变部分的区别对产品线工程与管理造成多方面影响,具体示例如下。

- 开发领域架构:在产品线的领域架构中,应明确区分产品线中各产品的共性与可变性部分。
- 确保可追溯性:产品线成员内部和成员之间的可变性造就了各种领域资产和应用资产,包括可变性模型。在整个领域和应用工程生存周期中,应分别追溯领域资产和应用资产。应用资产还应能够对领域资产进行双向追溯,因为应用资产可能直接或者经修改后复用领域资产。因此,可追溯性有必要贯穿领域工程和应用工程的生存周期。

综上,可变性是单系统工程与管理与产品线工程与管理之间的主要区别,应对其进行定义、建模、实现、版本化、验证与确认。可变性也应能够在领域工程与应用工程的生存周期内部及之间进行追溯。管理可变性的准则称为可变性管理。产品线工程与管理中最核心的概念与可变性管理密切相关,详见4.3。

4.3 单系统工程与管理与产品线工程与管理的关键区别

识别并分析单系统工程与管理与产品线工程与管理之间的关键区别,能帮助组织理解产品线参考模型(见5.2),制定产品线工程与管理的正确实施策略。因此,产品线组织宜设计其结构和过程,从而处理这些区别。

- 应用工程:一个过程生存周期,通过重用与领域架构一致的领域资产、绑定产品线的可变性,实现和管理产品线的应用资产和成员产品。因此,领域工程和应用工程这两个生存周期的存在,区分了单系统工程与管理与产品线工程与管理。
- 绑定:一种将产品线工程与单系统工程区分开来的决策任务。它决定了由领域资产和应用资产提供,并由可变性模型表示的各种可选或可替代行为,以创建应用资产或成员产品。以下场景宜考虑“绑定”,包括在领域工程中引入变体时,以及在应用工程中绑定变体时和绑定变体后。应用工程过程中可变性的静态绑定发生在程序运行之前。可变性的动态绑定可以在程序运行时使用。它使得系统能够根据预先指定的规则自适应其行为,并由用户进行调整(例如绑定额外的变体来增强使用特性)。
- 协作:由于产品线工程中存在两个并行的生存周期,负责领域工程的人员不仅需要在领域工程内相互协作,也要与相关应用工程的人员进行协作。
- 配置管理:产品线配置具有时间多维性和空间多维性。应用资产、领域资产、已发布平台和成员产品都有众多版本,每一版本又有自己的配置。成员产品的版本取决于资产及平台发布的版本,平台发布的版本又取决于领域资产版本。例如,领域资产中的任一更改都可能影响众多成员产品。任一给定时间下,任一成员产品可能存在多个配置,且其可能的配置会随着时间的推移而改变。这些维度都需要得到管理。配置管理对于任何包含软件的系统来说都是必要的,但其对于产品线工程与管理的重要性尤为突出。
- 领域架构:描述产品线的高层级设计,包含领域工程所定义的可变性。它被用作产品线内所有

成员产品的架构和构造设计的蓝图。领域架构(或参考架构)的需求性是产品线工程与单系统工程的重要差异之一。

- 领域资产:软件与系统产品线中的领域资产,是产品线工程和单系统工程的又一差异所在。
- 领域工程:领域工程定义、实现、验证和确认领域资产。通过该过程生存周期构建产品线的领域可变性模型,这亦是产品线工程与单系统工程的差异之一。针对生存周期内的任一相关过程,都宜对必要的子过程、角色、程序进行描述。
- 使能技术支持:产品线工程与管理所需的使能技术应可用于成功的实现产品线。由于软件产品线工程管理比单系统工程管理更具知识密集性,知识管理基础设施显得尤为重要。相较于单系统,产品线包含更多的资产、多的不同资产类型、更多的资产之间的依赖性。因此,产品线工程中对有效资产配置的管理比单系统工程更具挑战性。
- 测量和跟踪:产品线工程与管理中要求多维测量。数据的收集、处理和跟踪既要针对领域工程及应用工程生存周期,也要针对产品线的组织技术管理,因此需在两者间寻求平衡。例如,对负责应用工程和领域工程的相关组织单位的测量不恰当或者有偏倚,很可能导致争执以及资金及其他资源的分配不当。这种争端可能损害领域工程的资金投入,因为大部分或全部组织收入来自于成员产品销售及相关服务,而不是领域资产的销售。缺乏合理的资金投入模式和支持的测量跟踪系统,可能妨碍甚至破坏产品线工程与管理的组织实现及制度化。
- 平台:产品线工程是以平台为基础的。没有有效的平台,就不可能实现大规模产品定制。因此,平台是产品线工程与单系统工程的重要差异,对产品线组织具有重要战略意义。平台的引入及淘汰对产品线组织存在极大影响。
- 产品管理:关注产品线工程与管理及其产生的产品线的经济和业务问题。产品管理尤其注重市场战略和竞争战略。产品线应随着新的创新型产品,以及产品线内的产品能够利用的应用资产、市场的变化和新的竞争品的出现而持续演化。有效举措包括:在迭代周期中开发产品线、为每个周期确定明确目标、完成每一周期后进行绩效检查。产品管理部门负责这些实践,并根据审查结果对产品组合和平台投资进行适当调整。单系统工程与管理也需要产品管理,但产品线工程与管理中的产品管理有着更重要的作用。
- 可重用性:领域资产的可重用性是产品线成功的关键因素之一。可重用性要求对关键领域进行战略性长期关注,以保证领域开发相关投资的可行性,以及整个产品线工程与管理过程中技术及管理的有效性。
- 构造:包含共用规则和约束(例如,用于特定解决方案的通用架构风格或设计模式、用于架构组件组合的共用黏合成分)的架构性构造,用于产品线中的所有成员产品以及领域资产和应用资产的设计与实现。构造在单系统工程中也很重要,但它在产品线工程中作用更突出,因为领域架构的构造决定了众多成员产品的架构,例如用于确保产品线中的成员产品具有共同的外观和使用感受。
- 可追溯性:产品线工程与管理通常比单系统工程与管理更具知识密集性。产品线工程组织需要管理的知识不仅与更多的资产相关,而且与资产之间的更多关联相关。例如,可变性模型可能与领域及应用资产存在双向关联,领域资产与应用资产之间也可能存在双向关联。因此,产品线组织需要开发并维护知识管理基础设施,并且该基础设施应包含对可追溯性链接的管理。
- 验证与确认:通过提供客观证据来确认领域资产及成员产品的所有需求是否得到满足。例如,对领域资产的验证与确认可分别确保资产达到指定的服务质量,帮助需要该资产服务的系统(例如成员产品)达到其目标。产品线工程与管理环境下的验证与确认与单系统工程与管理中的验证与确认有着根本区别。
- 可变性:由产品管理定义并由产品线平台实现的产品线可变性,使得产品线中的成员产品能够进行灵活有效的大规模定制,是产品线工程与管理与单系统工程与管理的重要区别。

5 产品线工程与管理参考模型

本章定义了产品线工程与管理的参考模型,该参考模型构成了设计和执行产品线工程过程以及实现方法和工具能力的基础。产品线参考模型在较高级别上定义了产品线工程与管理的基本要素及其关系。

5.1 概述

产品线参考模型同时涉及工程与管理过程,涵盖了产品线工程与管理的关键特征。

产品线参考模型对系列标准(即 ISO/IEC 26551~ISO/IEC 26556)及模型的结构进行概述,其中:

- 产品线范围界定、领域需求工程和应用需求工程方法和工具的过程和功能在 ISO/IEC 26551 中描述;
- 领域设计和应用设计方法和工具的过程和功能在 ISO/IEC 26552 中描述;
- 领域实现和应用实现方法和工具的过程和功能在 ISO/IEC 26553 中描述;
- 领域验证与确认以及应用验证与确认方法和工具的过程和功能在 ISO/IEC 26554 中描述;
- 技术管理方法和工具的过程和功能在 ISO/IEC 26555 中描述;
- 组织管理方法和工具的过程和功能在 ISO/IEC 26556 中描述。

5.2 产品线参考模型

软件与系统产品线(SSPL)的工程与管理应包括领域工程和应用工程生存周期,以及组织管理和技术管理过程组(图 1),它们之间宜松耦合。领域工程和应用工程生存周期需要同步并适用于不同的生存周期模型,因为生存周期是在各种组织和技术环境中执行的,以满足不同的质量准则并实现不同的组织目标。组织管理和技术管理过程组对于帮助组织建立和改进培育从概念到退役的产品线的能力,并建立和管理与客户、供应商以及其他主要利益相关方的关系是必要的。

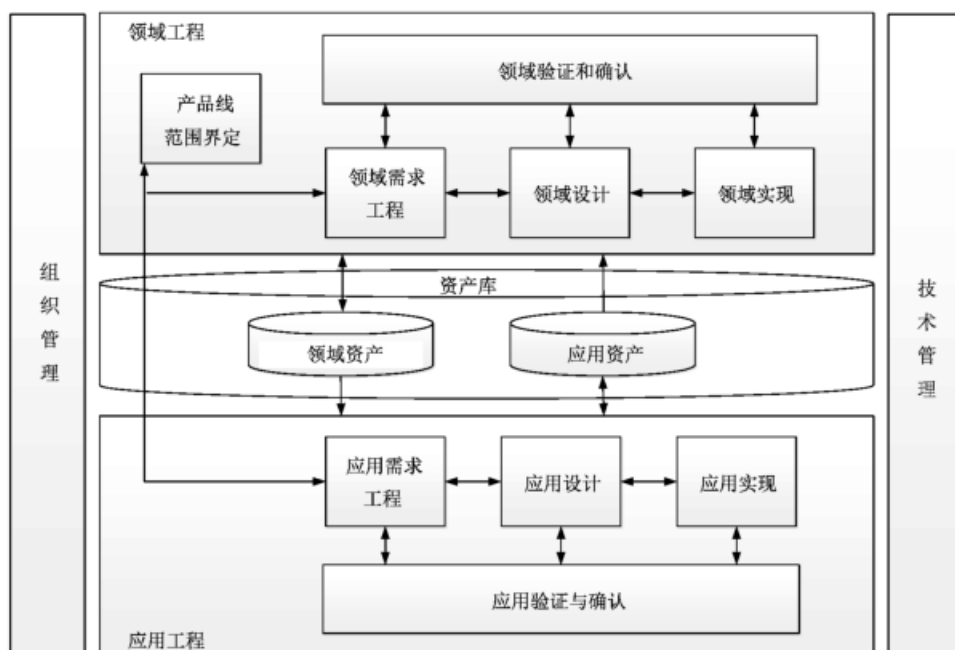


图 1 SSPL 工程与管理的参考模型

图 1 的两个不同的工程生存周期和两个过程组兼容 GB/T 8566/ISO/IEC/IEEE 12207 和

GB/T 22032/ISO/IEC/IEEE 15288 的过程组。存在两个工程生存周期是产品线工程与管理同单系统工程与管理的主要区别之一。资产库同时存储领域资产和应用资产。它分离了两个工程生存周期,并协调和同步了生存周期内和生存周期之间的过程。技术管理过程组与 GB/T 8566/ISO/IEC/IEEE 12207 的项目过程组、软件支持过程组以及部分软件重用过程组保持一致。组织管理过程组与 GB/T 8566/ISO/IEC/IEEE 12207 和 GB/T 22032/ISO/IEC/IEEE 15288 的组织项目支持过程组和协议过程组保持一致。

注 1: 该产品线参考模型不适用于处理诸如电路板、机械零件或人工操作者之类的物理制品。它涉及系统和软件级别的制品,如,需求文档、架构数据、验证计划 and 行为模型。当存在属于较大系统的软件制品时,可以应用两次。它首先用于处理产品线的系统级制品,然后处理产品线的软件制品。此外,产品线参考模型在软件与系统工程级别内和不同级别之间都是递归的。

注 2: 产品线参考模型和本文件并未规定任何特定的软件与系统工程方法的使用。组织可以决定将哪种方法用于领域工程和应用工程。一些组织可能对领域和应用工程使用相同的方法,而另一些组织可能选择混合和匹配多种方法(如,特定领域建模、敏捷和精益)。所选择的方法不仅宜促进领域实现资产,还宜促进其他领域资产(如,需求、设计和测试资产)的战略创建、重用和配置。

6 产品线工程与管理的双生存周期和双过程组

根据软件与系统产品线工程与管理的参考模型(见图 1),本文件着重于领域工程和应用工程生存周期以及组织管理和技术管理过程组。本章对上述内容进行说明。

6.1 领域工程生存周期

领域工程识别领域中关键的通用性和可变性,建立一个实现通用性和可变性的产品线平台,并在整个产品线的生存周期中管理这一平台。因此,开发方法应支持更大、更复杂平台的长期开发和维护。领域工程的方法和工具有助于实现 6.1.1~6.1.5 中描述的过程。

6.1.1 产品线范围界定

产品线范围定义产品线的成员产品及其主要(外部可见)的通用和可变特征,从经济角度分析产品,并控制和规划产品线及其产品的开发、生产和市场营销。产品管理主要负责此过程,由三个任务组成,具体如下。

- 产品范围界定,决定产品路线图,即目标市场;产品线组织宜开发、生产、上市和销售的产品类别;为了达到产品线组织的长期和短期业务目标,产品宜提供的通用和可变特征;产品投放市场的时间表。
- 领域范围界定,识别功能域,这些功能域对预期的产品线是重要的,并提供足够的复用来校正产品线的创建。领域范围界定构建在产品范围界定所产生的产品类别定义的基础之上。
- 资产范围界定,识别潜在的可重用资产,估算每个领域资产的成本和收益,并将这些结果作为资产建议书提供给产品线经理,由他们决定实现哪些领域资产。这些被采纳为领域资产的现有资产记录在资产列表中。

注: 产品线范围界定是一个组织的产品线的规划和管理的过程。可以将其放置在组织管理过程组(6.3)中。在本文件中,由于它从战略上指导了领域和应用工程生存周期的执行,因此将其归为领域工程过程。

6.1.2 领域需求工程

领域需求工程使用产品线范围界定的结果(例如,产品路线图,产品线的高层特征说明以及现有资产的文件资料)作为确定产品线中成员产品的通用需求和可变需求的起点。它产生了需求规格说明和

相关的领域可变性模型,并在必要时向产品管理提供有关产品线范围界定结果中所需更改的反馈。

有必要对通用需求和可变需求进行提取、分析、描述、验证与确认,以便于后续领域工程生存周期过程,即领域设计、实现、验证与确认。此过程包含以下任务。

- 领域需求获取,确定合适的领域利益相关方,使他们参与到单个系统工程的需求获取活动中,并使用适当的技术获取产品线在可预见生存周期内的期望范围和变化。
- 领域需求分析,利用已定义的产品线范围和其他相关材料来执行单系统工程的需求分析活动,找到共同点并识别变化。它也可以向应用工程师及客户阐明,某个特定成员产品是如何通过使用更通用、更少特定需求来实现经济可行性的。同时还构造了领域可变性模型。
- 领域需求规格说明,记录经过分析的产品线需求,应用需求规格说明可以在应用工程中对这些需求进行选择或修改。
- 领域需求验证与确认,确保在产品线的整个生存周期中,完整、正确、一致且无歧义地分析和描述所有相关的产品线需求。
- 领域需求管理,计划、定义和管理领域需求工程过程,并与负责应用需求管理的产品经理和其他利益相关方一起协调产品线的需求工程活动。它针对领域需求基线中的变更制定了一个正式的变更管理机制,以支持对所提议变更的影响域分析。领域需求和其他资产(包括需求源、应用需求,领域架构和领域测试资产)之间应保持可追溯性链接,以促进因需求变更而导致的影响域分析以及适当的资产变更。

6.1.3 领域设计

领域设计开发包含结构和构造,可实现所有成员产品的产品线架构。领域架构不仅反映了外部可变性,还反映了由所选技术解决方案引入的内部可变性。

领域设计大致分为四个部分:领域设计中的通用性和可变性分析、领域架构设计、领域架构评估和领域架构管理。

- 领域设计中的通用性和可变性分析,是指分析并考虑需求的通用性和可变性,以辅助设计领域架构。需求的通用性和可变性通过领域架构中关于软件和系统的结构和构造阐述。
- 领域架构设计,获取产品线的高层级设计。它宜满足通用需求并支持关键的可变需求,以促进成员产品的大规模定制。应定义架构级的构造体系并提供给所有成员产品,以便建立处理领域架构的通用方法。该体系包含用于设计、实现、编码和集成测试的通用规则。该框架可以提供满足其他约束条件的允许范围内的组件配置。
- 领域架构验证与确认,评审领域架构,并确认领域架构是否支持功能和非功能领域需求。包括检查包含组件框架在内的架构级的构造体系,并确保领域架构涵盖了领域需求的通用性和可变性,并且在领域设计过程中引入的新的(大多数是内部的,市场不可见的)变化是灵活的。包括对从质量属性的角度评估设计决策的架构级评价。
- 领域架构管理,在产品线的全生存周期中维护和管理领域架构以及相关的领域资产。管理领域架构的主要任务可以分为三类:配置管理,变更管理和可追溯性管理。配置管理至关重要,因为在应用工程实施期间,领域架构会被绑定到众多配置中。这些配置应经过验证与确认,因为不正确的配置可能会为后续应用引入问题甚至导致产品失效。建立配置管理时,应从一开始就考虑可变性模型。否则,之后的配置管理将变得过于复杂。总而言之,所有配置管理要求都应由同一个配置管理过程来处理。此过程是技术管理过程组(6.4.4)中的支持过程。变更管理应精心策划,因为领域架构应具备长生存周期,并且能对市场和技术的长期发展做出响应,同时尽可能不受产品线中各个成员产品的市场状况变化和组件级技术发展的影响。领域需求资产与领域架构之间,领域实现资产与领域架构之间,领域验证与确认资产与领域架构之间,以及领域架构与由此衍生的应用架构之间的可追溯性链接宜保持在可被理解的水平,以便

于变更管理和配置管理。

6.1.4 领域实现

领域实现处理通用和可变领域资产的详细设计和实现。它包括组件以及支撑的基础构造的构建和购买。应通过充分的实施机制来实现计划内的可变性,并且应基于领域架构和构造体系验证与确认核心组件和接口。领域实现涉及五个主要问题:接口实现;商业现货(Commercial off-the-shelf,COTS)组件的识别、评估、选择和集成;组件实现;领域实现资产的验证与确认和领域实现管理。

- COTS 组件的识别、评估、选择和集成 COTS 组件(包括开源组件)可以极大地加快领域实现速度,降低成本并提高整个平台的质量。应识别和评估合适的组件。在产品线工程环境中,COTS 组件是否能很好地满足已知可变性需求以及在必要时添加新可变性需求的难易程度,是评估 COTS 组件的重要标准。如果无法找到或购买到合适的 COTS 组件,则应实现新的组件。技术管理过程组中的支持管理(见 6.4.4)支撑了关于 COTS 组件相关的决策制定。
- 接口实现,包括通用和可变组件接口的内部设计和编码,以确保接口展示了通用性和可变性相关的适当信息。宜评审提供组件和所需组件的接口详细程度。
- 组件实现,如领域架构中所描述,组件是用于构成整个成员产品的单元。程序编码是基于领域架构的构造体系以及组件及其接口的设计完成的。此外,组件实现还包括诸如配置机制之类的资产,这些资产可以帮助应用开发人员选择和绑定变体,并使用领域组件和接口构建成员产品。
- 领域实现资产的验证与确认,检测领域组件和接口中的失效,这些失效会影响所有依赖领域资产的成员产品,宜通过确保领域实现资产的质量来防止失效。这个任务将审查已实现的组件和接口是否符合领域架构,是否遵守架构级的构造体系和组件框架的配置机制,以及编程风格是否符合适用的行业惯例或为产品线选择的标准。宜指出,领域验证与确认过程(见 6.1.5)负责该任务的一部分,这里讨论该任务是为了强调其对于产品线实现的重要程度,以及领域实现专家验证与确认已实现资产的责任。
- 领域实现管理,在软件产品线整个生存周期中,系统地控制变更并维护组件和接口的完整性和可追溯性。由于组件的多个版本和变体被使用,因此配置管理在此尤为复杂和重要。

领域实现与单个系统实现之间的区别在于,领域实现的可重用组件是松耦合和可配置的,并且不构成可执行、可测试的产品。另外,领域实现包括实现可变性的配置机制。

6.1.5 领域验证与确认

领域验证与确认确保按照领域生存周期过程中规定的正确方式对正确的领域资产进行建模、规格说明、设计、构建、验证和测试。该过程还生成领域验证与确认资产,这些资产可以在应用验证与确认中重用。领域验证与确认是局部的,因为通常没有单个可执行成员产品可全面覆盖待测试的域。而且,通常缺少许多变体,因为仅当一个或多个成员产品确实需要时才会去创建。因此,在领域验证和确认过程中,应制定适当的策略以测试通用领域资产,并至少测试最重要的可变领域资产。并且,准备领域验证和确认资产,以便在应用验证和确认过程中重用。

领域需求工程、领域设计和领域实现中固有的验证与确认活动见 6.1.2~6.1.4。本条包括如下问题。

- 领域测试规划,文档化领域验证与确认活动的范围、策略、资源和进度。领域测试计划宜包括详细的领域验证与确认需求,为处理产品线的可变性,测试活动和测试完成标准而选择的领域验证与确认策略。验证与确认时间表依赖于产品路线图,而产品路线图决定了产品线投放市场的时间。
- 领域测试设计,描述考虑了领域资产可变性的领域测试规程、测试用例、测试数据和测试环境。

它利用从其他领域工程过程中获得的各种领域资产来构建适当的领域验证与确认资产。选择和构建领域验证与确认资产的任务在整个领域工程全生存周期中进行。根据每个领域资产涉及的可变特性的种类,对领域资产的测试可以进行补充和裁剪,对相关应用资产的测试也一样。领域测试设计确定要在何处进行何种测试,并相应地构造领域验证与确认资产。

- 领域测试执行,根据领域验证与确认策略将构造的验证与确认资产应用于测试对象,例如,组件,子系统(如组件分组)和已实现的产品线平台。在执行期间和执行之后,将创建验证与确认结果,以可重复和可验证的方式记录所应用的测试用例和场景,被测对象以及预期和实际的验证与确认结果。如果在领域验证与确认期间已构建样例应用,则仅执行与示例应用相关的通用领域验证与确认资产以及可变域验证与确认资产。具体的领域测试用例通常仅适用于这些项目。领域工程中无法验证与确认任何样例应用中都不存在的产品的可变性。
- 领域测试关闭和报告,完成验证与确认后,将对结果进行分析,以发现领域资产中的缺陷及其根本原因。最后,将创建一个领域测试总结报告,记录已测试的领域资产、已使用的验证与确认资产,以及得到的结果。领域测试执行未覆盖的测试将由应用验证与确认实施。在领域验证与确认中执行的测试用例通常应在应用验证与确认中重复。
- 领域测试管理,跟踪领域验证与确认资产,如,缺陷和资产的更改历史记录。

6.2 应用工程的生存周期

应用工程在平台之上开发应用资产和单个系统。与单系统工程相比,应用工程是有效和高效的,因为根据平台的范围和成熟度,大部分产品线工程的工作量和复杂性已分配给领域工程,从而降低了应用复杂性并缩短了应用开发时间。应用工程通常涉及客户,因此需要应对不断发展的市场需求。

6.2.1 应用需求工程

应用需求工程重用了在领域需求工程中定义的通用需求和可变需求,以开发特定于应用的需求。它还为领域需求工程提供认识和反馈,以指导平台开发,尤其在产品线创建的早期阶段,所发挥的作用尤为重要。

- 应用需求获取,识别与成员产品相关的利益相关方,从应用利益相关方中获取出初始需求,使用可变点和变体与利益相关方进行沟通,让利益相关方选择最能满足其需求的变体,并根据选择结果绑定适当的产品线领域需求。
- 应用需求分析,首先通过抽象、建模、原型设计、仿真和其他方式,确保理解应用利益相关方的所有初始需求,仔细审查需求的不正确、遗漏和不一致之处。然后分析和协商那些无法通过领域需求重用来满足的利益相关方的需求,即领域和应用需求之间的差距。
- 应用需求规格说明,通过将应用特定需求添加到所选的、可能经过调整的和绑定的领域需求规格说明中来文档化分析后的应用需求。应用需求宜具有良好的文档结构,以便在后续应用工程生存周期过程中使用,并在必要时合并到领域资产中。
- 应用需求验证与确认,确认特定应用的需求是完整、正确、一致且无歧义的,并且绑定的变体与特定产品需求有关。宜指出,此任务部分是应用验证与确认过程的职责。在此介绍它是为了强调它对应用需求工程的关键性质,以及应用需求工程师的责任,即与产品管理及应用验证与确认专家协作来验证与确认应用需求。
- 应用需求管理,计划、定义、管理和协调应用需求工程过程,并为应用需求设定基线。它为应用利益相关方提供了一种正式的机制,用于提出新的应用需求和基于应用需求基线的变更,并评估所提议的变更的影响。应维护应用需求和其他资产之间的可追溯性链接,包括应用需求源、领域需求、应用架构以及应用验证与确认资产等,因为应用需求的变更可能会影响相关资产,而其他资产的变更可能会影响应用需求。

6.2.2 应用设计

应用设计从领域架构派生出应用架构以满足应用需求。宜进行相应的产品特定更改,以满足产品特定需求。应用架构应遵循领域架构的结构和构造。此过程具有以下任务。

- 领域架构的变体绑定,是应用架构师的责任,为应用架构建立基线。应根据应用可变性模型和应用需求工程的绑定结果来绑定领域架构变化点的变体。否则,应用架构可能缺乏对应用需求要求成员产品应具有的和特征和质量属性的支持。
- 应用特定架构设计,应用架构宜尽可能从领域架构派生,否则无法获得领域架构的全部益处。例如,某成员产品可能与其他成员产品差异太大,并被证明不适用于该产品线。于是应用架构师通常应设计新的应用设计资产。通过将可变与领域架构的选定变体绑定,将适当的领域设计资产合并到成员产品中。应用架构是应用设计资产和其绑定的领域架构的组合。需要重点注意的是,新的应用设计资产可以在以后概括为领域设计资产。
- 应用架构验证与确认,可确保应用架构正确组成,符合架构规则并满足应用需求。该过程审查在应用设计期间引入的可变性是否可行,是否应用可变性模型中进行了适当记录,以及是否可被正确绑定。还从应用特定质量属性的角度评估了应用架构。通过领域验证与确认资产的规格说明、实施和执行,领域验证与确认宜已经确保可以正确绑定领域架构的适当可变性。但是,应用验证与确认需要设计和实现与应用设计资产相对应的新应用验证与确认资产,并将它们与领域验证与确认资产一起执行以确保应用架构的完整性和有效性。宜指出,应用验证与确认只是部分负责此任务。在此说明此任务是为了强调应用架构师需要负责应用架构的验证与确认,并与应用测试人员合作以确保应用架构符合期望。
- 应用架构管理,在成员产品的全生存周期中管理和维护应用架构以及相关的应用资产。应用架构师提出的与领域架构相关的反馈也在此处进行管理。管理应用架构的主要任务可以分为三组:配置管理,变更管理和可追溯性管理。配置管理很重要,因为成员产品可能为批量定制提供大量可变性。因此,每个成员产品可以以多种配置存在,应进行验证与确认。变更管理通过处理产品线中各个成员的市场状况的变化以及组件级技术的发展来支持并延长应用的生存周期。应用需求资产与应用架构之间,应用实现资产与应用架构之间,应用验证与确认资产与应用架构之间以及领域架构与应用架构之间的可追溯性链接宜保持在可被理解的级别,以实现变更管理,配置管理以及向产品线架构师提供反馈。

6.2.3 应用实现

应用实现通过利用应用需求和架构,重复使用和配置领域组件和接口,识别、选择和集成适当的商用货架组件,以及构建新的组件和接口来实现成员产品并使其具备指定功能。它承担以下任务。

- 绑定组件级别的可变性,绑定领域组件的内部可变点。在领域实现中所实现的接口宜重复使用而无需更改。否则,领域架构的构造可能会被破坏,并且阻碍领域资产的战略性重用。
- COTS(包括开源)组件的识别、评估、选择和集成,当领域组件不足以支持应用实现时,COTS组件通常是一个可行的选择。它们可以加快应用的实现速度,降低成本并提高成员产品的质量。与专用组件相比,如果有必要的话,COTS组件以后也可能更易于概括为平台的一部分。应识别并评估适当的COTS组件。如果无法找到或购买到合适的组件,则应实现新组件。
- 专用接口实现,包括专用组件接口的内部设计和编码。仅当领域实现未提供适当的接口以供重复使用时才宜执行此操作。宜仔细检查提供的组件和所需组件的接口详细信息级别。
- 专用组件实现,当没有合适的领域资产和COTS组件可用来满足应用需求时,与在单系统工程中一样精确地实现。
- 应用实现资产验证与确认,同时检查与绑定变体相关联的领域实现资产和应用组件和接口,以

查看;绑定变体和已实现的组件和接口是否符合应用架构并遵循配置机制和架构级构造体系;编程风格是否符合适用的行业惯例或为产品线选择的标准。宜指出,应用验证与确认部分负责此任务。提出此任务为了强调应用实现工程师需要负责应用实现资产的验证与确认,并与应用测试人员协调工作。

- 应用实现管理,定义、计划和协调应用实现过程,系统地控制变更,并在应用全生存周期内维护应用组件和接口的完整性和可追溯性。配置管理很重要,因为每个成员产品在其整个生存周期中不断演化,并且拥有了成员产品用户等不同利益相关方可以绑定的可变性。因此每个成员产品可以在其生存周期中以多种配置存在。为了维护,应了解成员产品中使用的领域组件和接口的版本。领域实现可以为成员产品提供新的领域资产版本,从技术或其他角度来看,这些版本可以显著地更新成员产品(尤其是具有相对较长生存周期的成员产品)。应保持应用实现资产和其他资产(包括应用架构,应用可变性模型,领域实现资产以及应用验证与确认资产)之间的可追溯性链接。

6.2.4 应用验证与确认

应用验证与确认可确保对正确的成员产品和正确的应用资产进行了正确的建模、说明、设计、构建、验证与确认。该活动根据应用需求验证最终产品及其架构和实现资产。利用领域验证与确认资产并创建应用验证与确认资产,以确保成员产品充分的质量保证。

由于在 6.2.1~6.2.3 中已经阐述了应用工程生存周期其他过程中固有的验证与确认活动,因此本小节仅包括以下任务。

- 应用测试规划,生成一个应用测试计划、记录范围、测试策略、测试完成标准、资源以及应用验证与确认活动的进度。针对直接可重复使用的领域需求,领域测试计划的相应内容可以复用,并在应用测试计划中予以必要的调整。针对修改的领域需求和新的专用需求,应重新做测试规划,并将其纳入应用测试计划中。
- 应用测试设计,通过重复使用和修改领域验证与确认资产来指定并构建应用测试规程,测试用例,测试数据和测试环境。领域验证与确认资产的通用部分可以按原样重复使用。对于包含可变性的验证与确认资产,应基于应用可变性模型以及来自应用需求工程、应用设计、应用实现以及领域验证与确认的输入来绑定变化点。对于应用特定部分,应构建新资产。生成应用验证与确认资产所需的工作取决于成员产品的大小和范围,以及应用工程过程中领域资产被重用的程度。如果成员产品从产品线平台获得大多数功能,并且已经针对该平台广泛建立了相应的领域验证与确认资产,则应用验证与确认将花费相对较少的工作量。
- 应用测试实施,根据应用验证与确认策略将应用验证与确认资产应用于各种测试对象,例如组件,子系统和成员产品。除了验证成员产品的正确性和完整性外,还应评估成员产品与架构级构造体系的一致性。成员产品中绑定的可变性和配置的约束应进行验证与确认。预期的和实际的验证与确认结果与已执行的应用验证与确认资产一起文档化。
- 应用测试关闭与报告,当完成上述应用验证与确认任务后,将对结果进行分析以发现应用资产中的缺陷及其根本原因。最后,将创建一个测试总结报告,它记录已验证与确认的资产、已使用的验证与确认资产,以及测试结果。领域测试总结报告可在报告中重复使用。
- 应用测试管理,保持跟踪缺陷、应用验证与确认资产,以及应用验证与确认资产的变更历史记录。

6.3 组织管理过程组

组织管理过程对于构建产品线组织是需要的,在组织中引入和制度化产品线战略需要持续地准备、规划、执行和改进。

6.3.1 组织级产品线规划

组织级产品线规划属于组织级战略规划。组织产品线规划涉及各类计划,如产品线迁移计划、采购计划和领域资产投资计划,其最重要的职责之一是通过分析投资回报率,做出是否采用产品线战略策略的决定。宜从以下几方面分析产品线带来的预期商业价值,如成本降低、生产率提高、质量改进、商业风险降低、上市时间缩短以及市场份额增加。基于前述信息,组织的商业管理宜设定引入产品线方案的商业价值目标,并通过测量来检查这些商业价值目标是否达到,若未达到,组织应采取必要的纠正措施。

在某些情况下,存在仅在规划阶段不能独立完成该分析的情况,需基于计划的执行结果完善计划。通常情况下,宜快速迭代一个领域工程过程周期,尽快开发出首个产品以占领市场份额,同时测试实施产品线的可行性。通过反馈,建立和完善具有进一步的应用资产(如需求、设计和软件部件),并纳入平台,进入下一个领域工程过程迭代周期,应在组织级商业管理明确该策略计划。过程管理宜确定领域和应用工程过程是否适应于修改过的平台。

该过程有如下任务。

- 商业机会分析,帮助利益相关方决定是否启动产品线方案,以及是否在现有产品线中增加特定的产品。这包括成本/收益估算和商业价值分析。建立来自高级管理层、销售、产品管理、工程与用户关系管理和用户组等各方代表间的协作关系,是成功开展商业机会分析的必要条件。首先应分析市场,以做出初步可采纳的决策,并指导产品线的演进,将新产品引入产品线并持续完善。商业机会分析还应确定建立和运营领域资产库、知识管理基础设施,以及在预期的产品线生存周期中实施适用的产品线工程过程和管理过程所需的投资规模。只要能从市场和其他来源获得收益,且收益大于投资,就存在商业机会。
- 客户关系管理,指产品线组织(如销售、产品经理、领域专家、用户组协调员)与客户一个或多个特定利益组(如法律、财政或技术实体、操作、培训)的委派代表间交换信息和其他资源,这些客户代表负责验收产品线组织的产品,负责在客户组织中导入产品并在产品生存周期中提供支持。产品线组织宜跟踪和协调客户需求,及时响应和变更产品以满足客户认为必要的需求更改,按 B2B 模式设置产品线的组织和他们的客户,宜致力于构建互利的长期关系。
- 开发采购策略,应建立可行的计划,通过与供应商签订产品与服务合同达到特定产品线的目标和结果。采购有助于获取新的领域资产,有助于在产品线中融合为新的产品。该策略应包括对 COTS 部件和其他第三方产品供方进行管理的计划,例如宜确定能满足产品线预期的可变性、使用期和演进要求的关键供方,并将合同发包给它们。产品线中若使用开源 COTS 部件,宜指派组织角色/职责及明确定义流程,以管理和开源社区的关系。最后,开发采购策略时,还要考虑产品线中部分或全部成员产品安装或使用了第三方应用的情况,宜确定是否允许用户和其他第三方提供、销售并分销该第三方应用,如果允许,采购策略则应包括为第三方建立适当的组织、过程和技术支持(如应用工程工具和质量保证、销售以及在线分销机制)的计划。
- 组织迁移规划,建立培育产品线的组织能力,它规划产品线的启动,进度表以及建立和管理产品线所需的资源,如组织、人员和预算,它还定义了采用产品线方案实现的目标,迁移计划是组织迁移规划过程的主要交付物。
- 组织运营规划,概述特定组织单位如何生产领域资产、如何定义生产计划、如何使用领域资产和生产计划来构建产品,以及如何度量和监视生产运营。产品线组织竞争力来自于比竞争对手更低的生产成本,为此,运营计划是尤其重要的制品。运营管理章节(6.3.3)详细描述了运营计划及其执行。
- 组织产品线改进规划,定义领域工程过程和应用工程过程如何持续改进以使产品线的效能和效率达到平衡。
- 价值管理规划,提供集成的度量方案以提高迁移计划和运营计划的效能,宜规划存在于商业机

会中的商业价值及其应达到的目标。

注：以智能移动通信终端产品线的例子来解释“开发采购策略”作用。智能移动通信终端产品线的制造商能向客户提供大量廉价但高质量的第三方应用，如智能移动通信终端产品线的成员产品上运行的游戏，对于移动通信终端产品线的成功至关重要。市场倾向于欢迎能够最广泛地提供高质量第三方应用的智能移动通信终端。本例中，智能移动通信终端是成员产品，第三方应用通常并不是智能移动通信终端制造商产品线的成员产品，它们扩展了智能移动通信终端的特征，但移动通信终端制造商的产品管理部门往往对这些特征的控制有限。第三方应用的供方可以独立于移动通信终端，建立自己的产品线，并充分考虑智能移动通信终端平台。例如智能移动通信终端产品线的成功将吸引越来越多的第三方游戏供方提供支持智能移动通信终端的游戏产品线，从而促进智能移动通信终端产品线的成功。本例中，第三方游戏是游戏产品线的成员产品之一。

6.3.2 组织产品线使能管理

组织产品线使能管理规划了产品线的启动和演进，以及为建立和管理产品线所必要的组织过程、授权和职责结构以及基础设施。培训对于产品线实施和长期演进起核心作用。产品线会要求产品线生存周期内大量技术和组织的变更；因涉及众多组织单元和外部合作伙伴，需要组织内和跨组织、职能内和跨职能、项目内和跨项目的协调。

该过程具有如下任务。

- 产品线组织结构化，明确组织章程及范围；定义领域工程、应用工程间和它们之间的交互、领域工程和应用工程的组织管理、技术管理相关的角色及职责；安排与分配资源；监视组织效率；改进组织运营；建立组织内部关系；管理组织迁移。
- 培训与人力资源管理，有助于确保负责创建、运营和演进领域资产及成员产品的组织单位接受正确的培训，为每个工作角色提供有资质的人员。培训活动应与产品线实施及演进所涉及的其他活动相协调。
- 组织质量管理，保证产品线平台、成员产品以及领域工程过程及应用工程过程均满足组织质量目标。

6.3.3 组织产品线管理

组织级产品线规划启动了组织向产品线工程的迁移。在组织迁移期间，组织产品线管理在产品线工程制度化过程中逐渐承担更大的作用。组织产品线管理关注产品线组织和产品线从当前状态到期望状态的系统化演进。组织产品线管理保障必要的资金和其他资源；维持产品线的预算、授权结构权限和职责分配；定义和维持使能产品线工程与管理的基础设施及过程活动；并确保产品线组织中的每个人员清楚其职责以及履行职责的可用方法。组织产品线管理监控产品线；定义并维护生产计划和进度安排，以便基于产品线策略、工作量估算、预算、客户需求及市场环境的持续分析，对应用工程产品开发、领域工程资产创建及平台进行协调；管理产品配置；监视采购契约并在必要时修改采购策略；为现有客户用户提供部署产品所需的支持和其他服务。

该过程具有如下任务。

- 产品线演进管理，定期分析用户需求、关键竞争对手、技术和其他市场环境的状态及变化趋势，产品线宜能通过演进产品线平台或调整成员产品以适应这些变化。
- 部署和创新管理，通过有效评估、改进和激活产品线计划，促进产品线方案成功实现。在产品线方案广泛部署之前，可先行开展试验性项目以评估产品线组织的能力和是否就绪。组织宜验证与确认能否通过产品线策略达成目标。
- 运营管理，一般设计、度量并监视特定组织单位如何生产和演进领域资产、如何定义并完善生产计划、如何利用领域资产和生产计划构建产品。运营管理遵循并执行采购策略，以定义从第三方采购哪些资产和服务。形成的运营设计是重要的领域资产。关键测量宜有助于运营管理

和产品线组织能够从整体上跟踪产品线工作进展及其交付物,必要时采取纠正措施以实现组织的商业价值目标。

- 组织风险管理,组织风险管理与技术风险管理不同,它管理多组织单元、职能和产品内部以及跨组织单元、职能和产品间的风险。诸多风险会阻碍组织实现产品线工程与管理,应提前进行风险管理,以便尽可能消除风险的发生,并在风险发生时能够更高效地处理。如,产品线工程与管理需要跨组织、跨职能和跨产品的大量沟通、协调、协同和控制,如果失效将成本巨大。
- 组织级监控,用于测量引入和演进产品线的实际进展和效果,为产品线组织设定的关键测量指标宜基于已定义的商业价值,聚焦于测量目的。

6.4 技术管理过程组

技术管理是产品线过程、领域资产和产品的创建与演进所需要的,由过程管理、可变性管理、资产管理和支持管理组成。

6.4.1 过程管理

不同组织单位的人员共同工作来设计和管理产品线,这需要大量的协作。产品线过程宜能够跨职能和跨组织。应用工程和领域工程产生交付物种类不同,宜采用不同的开发方法,因此产品线过程的设计非常具有挑战性。例如,成员产品需满足市场需求,上市时间通常应短,开发应用的生存周期通常也较短。领域资产则需要比较长的生存周期,领域资产一般被开发为通用化的,以满足不同成员产品的需求,同时应满足严格的质量要求。

“6.3.1 组织级产品线规划”中所描述的战略着重于应用工程,产品线战略实施初期着重测试推向市场的首个产品,在首批产品被证明是成功的之后,将重点逐步转移到领域工程。这种安排是高效的,原因是在产品线创建初期可充分利用单系统工程,降低应用工程和领域工程两种方法并行实施的需要,以及市场和产品线组织开发允许的范围内,有利于尽快迁移到产品线工程。

过程管理对于建立和管理产品线的组织能力,实施适用的产品线过程是需要的,任务如下。

- 产品线实施过程化,有助于建立并管理过程实施的基础设施,以保障产品线过程定义、过程评价和改进的领导及资源。
- 领域工程过程定义,建立和维护产品线工程与管理的组织标准过程集和裁剪指南。组织定义了所有产品线参与者宜遵循的共用过程,也建立了领域工程整个生存周期中使用的过程和开发方法。不同组织单位宜协同工作,为此,需在不同层级定义产品线组织标准过程并跨越职能。
- 应用工程过程定义,建立和维护成员产品开发适用的过程,用于定制在产品线环境下开发成员产品的特定过程,也包括所有成员产品均适用的应用工程过程、开发方法和工具。应用工程过程可直接使用通用过程或对其进行裁剪,同时还宜遵循领域工程过程定义的约束。
- 关系管理过程及支持信息系统的定义与应用,用于处理采购策略(6.3.1)的实施,以便与外部供方和开源社区协作。该任务将关系管理过程、组织和技术支持过程以及负责过程的组织实体制度化,以与利益相关方(如 COTS 部件提供商和开源社区)进行合作。
- 实施产品线过程监控,有助于监视领域/应用工程过程的性能,有助于控制纠正措施,以纠正实际性能与计划的偏离。为监视过程性能,宜选取合适的度量指标,并根据测量结果控制过程的实施。
- 实施产品线过程改进,有助于基于测量结果管理组织过程评价和过程改进。宜系统化进行组织过程评价和过程改进,使组织过程从当前状态向期望状态变迁。

6.4.2 可变性管理

产品线的可变性定义了成员产品间如何区分。可变性管理要求通过领域和应用可变性建模,将可

变性明确地文档化。以下可变性管理任务提供对领域和应用工程生存周期可变性建模、文档化和演进的支持。

- 可变性建模,利用领域和应用工程与可变性相关的信息,支持详细可变性模型的创建及维护,可变性建模技术使用一致的标注。有两种可变性模型:领域可变性模型和应用可变性模型。领域工程一般提供充分的信息以结构化领域可变性模型,应贯穿于领域工程生存周期对可变性模型期进行完善和管理。因产品线中的每个成员产品均提出大量的可变性需求,应用工程也以可变性模型的形式提供可变性信息。同样,这些模型贯穿于应用工程生存周期过程并不断完善。不同过程的可变性信息详细程度不尽相同(如应用需求工程)。
- 可变性机制,是实现产品线可变性的途径,因为在整个领域工程生存周期各个阶段均可引入可变性,且在整个应用工程生存周期中绑定,所以宜提供可变性机制以支持确定绑定的次数、规则和约束,同时提供领域制品。即使在相同的生存周期阶段所用的可变性机制不同,领域制品也不同。不同可变性机制的实现方式(如不同的编程语言)、安装和部署方法也不相同。本任务支持领域工程师根据确定的绑定次数、领域制品、选择的部署方法等选择正确的可变性机制。
- 可变性绑定,支持在应用工程实施期间可变性的持续利用,促进主动地、正确地重用领域资产。维护信息使领域和应用工程师及自动化工具以合适的方式解决可变性问题;解决在领域和应用工程实施期间所确立的可变性;并记录可变性问题的解决方案形成文档,以便日后查看(如某个成员产品选择了哪一个变型)。可变性绑定通过文档,准确地记录成员产品绑定了哪些可变性及如何绑定,有助于成员产品的维护。
- 可变性文档化,支持利益相关方通过标注详细记录可变性模型,帮助利益相关方在产品线工程实施期间了解和绑定可变性。标注可描述采用领域资产提供的特定可变性而忽略其他可变性,以及支持跨产品线成员产品领域资产重用的理由。领域可变性模型标注应能帮助做出绑定决策。应用可变性模型标注还应能够提供特定成员产品的绑定结果、新增变化点、变型、依赖性和约束的理由。
- 可变性追溯,使可变性模型元素与其关联的领域和应用资产间建立并维护追溯性关系。宜在追溯性的成本与收益分析的基础上,确定合适的追溯性管理颗粒度。过度的追溯或较弱的追溯代价都很大。
- 可变性控制与演进,应对资产库中存储的可变性模型不同版本及其关联的追溯性关系进行管理,应对可变性模型变更进行控制,将产品线可变性作为组织战略性资源资产进行管理。

6.4.3 资产管理

资产管理负责存储、挖掘并管理领域资产和应用资产,资产分别由平台的领域工程和单个产品的应用工程产生。产品线工程需对领域资产、应用资产以及它们之间关系进行综合管理,因此,产品线环境中的资产管理比单系统工程中的资产管理更为复杂。资产管理实现对平台和其他资产的控制,包括挖掘应用资产的属性(用于将应用资产泛化为领域资产)、重用领域资产所需的标注(如,附属于领域资产的技术映射、过程以及领域资产的说明,包括规定领域资产如何使用的指南)、变更请求和反馈、可追溯性以及领域和应用资产的基线版本的管理。

资产库是领域和应用资产的通用知识库,包括资产间追溯关系、资产与平台发布版本的追溯以及资产与成员产品的追溯。产品线平台由领域资产、领域架构和配置管理计划组成,存储在资产库中。应用资产也存储在资产库中,部分应用资产可以泛化并作为领域资产进行重用。资产库支持领域和应用资产的创建、存储、恢复、更新、删除、版本控制及追溯性管理。资产管理工作确立并管理资产库。

该过程具有下列任务。

- 资产识别,明确并评估候选领域资产(如特征、模型、规格和测试用例),例如从应用工程或遗留

应用项目识别。应选择通用化和可复用的模块作为最佳候选,以充实资产库。值得注意的是领域工程仅创建领域资产,并将其存储在资产库中。

- 资产库实现,用于构建资产库。资产库宜能够配置资产,以易于挖掘、恢复和管理,同时宜包括标注以提供必要信息,便于资产集成和条理化,使资产能被系统化重用。
- 资产验证与确认,确保领域和应用资产能反映已定义的资产结构,便于挖掘、恢复和管理。
- 资产演进,管理和控制资产建立基线后的变更请求、追溯性以及版本更新。

6.4.4 支持管理

支持管理涉及如下任务,用于支持其他过程,促进其他过程的成功并提高过程质量。支持管理职能包括技术质量管理、配置管理、决策管理、技术风险管理和工具管理。

- 技术质量管理,保障领域工程与应用工程所实现的工作产品及过程符合预定义的条款和计划。通过技术质量管理方案(如质量保证)评价整个产品线工程中能够达到的功能性和质量属性。
- 配置管理,控制每个应用和领域资产的版本及发布。产品线工程与管理中的配置管理相比单系统工程与管理更复杂,宜处理产品线平台和成员产品的可变性。
- 决策管理,有助于在商业或技术决策选择时做出最有利的选择。在选择最佳方案的决策期间,宜考虑每个备选方案隐含的优点和不足。决策管理服务于所有产品线的决策,包括领域工程、应用工程、组织管理和技术管理。
- 技术风险管理,用于处理可能危及商业价值目标和其他产品线目标实现的技术风险问题。如果没有充分的技术风险管理,很可能会发生重大技术风险(如领域知识的缺乏、领域需求的不确定或多变、工作量估算历史数据的缺乏),因此,产品线组织应制定出缓解和应急计划以应对这些风险。
- 工具管理,需要工具管理以使领域工程和应用工程生存周期过程、领域和应用资产的创建、配置、追溯、验证、确认、维护和演进自动化或半自动化。

7 领域工程和应用工程内部及相互关系

本章主要阐释产品线工程与管理参考模型(见图1)中,领域工程和应用工程过程内部和过程之间的关系。

7.1 产品线范围界定和领域需求工程之间的相互关系

产品线范围界定的主要成果是产品路线图和资产建议书。产品路线图定义了包括产品线及其主要资产在内的各种产品(如,直接影响领域和应用需求工程的高层级公共和可变特征),以及其量化成本和估计收益。产品路线图可以定义用于将特定的成员产品交付给客户或将其推向市场的时间表。这个时间表是战略推理和由产品线或产品经理进行工作评估的结果。每个资产建议书都包含一个源于领域需求和其他领域资产的资产清单。产品线范围界定和领域需求工程之间的相互关系见图2。

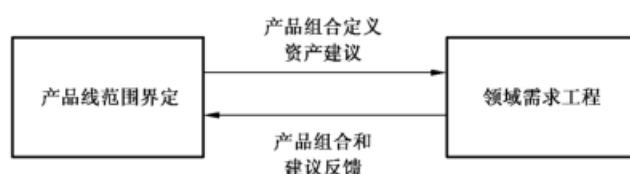


图2 产品线范围界定和领域需求工程之间的相互关系

产品线产品的特征在领域需求工程中定义。这些特征主要处理外部的可变性。根据对现有产品、

利益相关方需求、法律、约束和其他需求来源的分析,将这些信息作为对新增和变更的产品或特征以及特征完善的建议反馈至产品线范围界定。该分析可能涉及与客户和提供者的重大交互。宜建立相应的(处理)流程。

在产品线的生存周期中,产品线范围界定应对客户需求、可用技术、竞争对手产品以及其他市场状况等变化做出响应。根据响应需要对产品路线图和资产建议书进行调整,例如引入新特征介绍或从产品组合中删除过时的成员产品。

7.2 领域需求工程和领域设计之间的相互关系

领域需求工程应遵循产品线高级别特征的规格说明,这些由产品线范围提供的特征,详细描述足以支持领域设计、实现、验证与确认的公共和可变需求。

领域需求工程为领域设计提供了所有明确的领域需求,以及在可变性模型中的产品线可变性定义。领域设计人员可以决定包含在领域架构中的技术解决方案。源于领域设计的内部可变性通过领域需求工程添加到了可变性模型中。最终的可变性模型定义了领域架构的可变性。

领域设计根据新的、修订的或更详细的需求给领域需求工程提供反馈。领域需求工程和领域设计之间的相互关系见图 3。

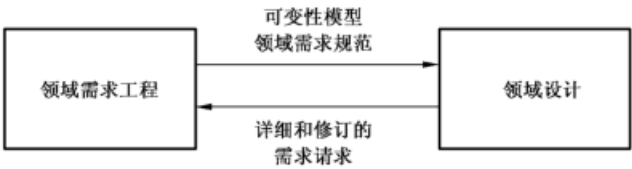


图 3 领域需求工程和领域设计之间的相互关系

7.3 领域设计和领域实现之间的相互关系

领域设计为领域实现提供领域架构,领域架构是构造所有成员产品的基础和构建重用组件和接口的构造。

出于对领域设计改进的目的,需要将问题报告和领域实现过程中出现的问题作为反馈,提供给领域设计。领域设计和领域实现之间的相互关系见图 4。

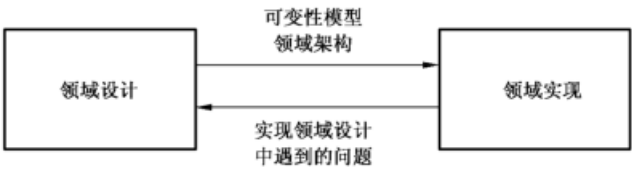


图 4 领域设计和领域实现之间的相互关系

7.4 领域需求工程和领域验证与确认之间的相互关系

领域需求工程给领域验证与确认提供领域需求资产,指定公共和可变的领域需求,以及可变性模型。领域验证与确认给验收测试提供了系统测试和验收准则,并检查特定的领域需求是否可测试。

平台包含松耦合的组件集,但没有完整的成员产品。因此,领域验证与确认通常只能在实现公共需求且不受产品线可变性影响的子系统上执行系统测试。为了找到不包含可变性的测试用例,需要配置变体。但是,在可变点数量有限且已知变体数量相对较少的情况下,可以依次测试每个变体。例如,对于具有三个变体的可变点,需要规划三个测试。需要大量的测试运行才能在不同的可变点测试所选变体的所有组合,但这在某些情况下可能是合理和可行的。

缺陷应被报告给领域需求工程,例如,来自不完整或模糊的领域需求定义的缺陷,以及在领域系统测试中发现的缺陷,以便领域需求资产中的缺陷可以被纠正。领域需求工程和领域验证与确认之间的相互关系见图 5。

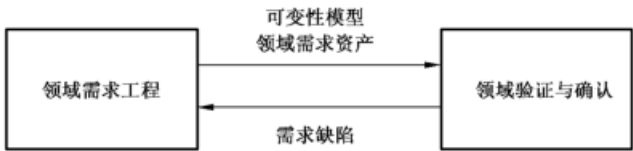


图 5 领域需求工程和领域验证与确认之间的相互关系

7.5 领域设计和领域验证与确认之间的相互关系

领域设计为领域验证与确认提供领域架构,并选择已在平台中实现的可重用资产。集成验证与确认资产是为公共组件交互以及那些与基于领域架构实现的组件有少量可变交互的组件而创建。同样还宜至少为可变组件的最通用交互创建集成验证与确认资产。验证与确认资产应提供与组件间交互的可变性相匹配的可变性。领域设计和领域验证与确认之间的相互关系见图 6。

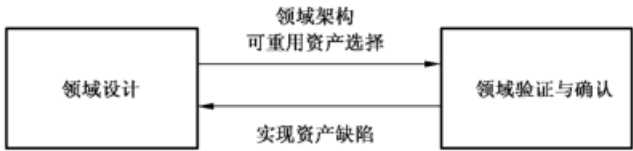


图 6 领域设计和领域验证与确认之间的相互关系

集成测试中会发现诸如不完整和模棱两可的领域设计资产等缺陷,这些缺陷妨碍了验证与确认资产的定义。领域设计中产生的缺陷需提交给领域设计。

7.6 领域实现和领域验证与确认之间的相互关系

领域实现向领域验证与确认提供已实现的可重用组件和接口,以及用于单元测试的接口规格说明。集成测试是使用在领域设计中形成的集成测试规格说明来执行对形成领域架构中特定配置的组件的测试。领域测试只能对已经实现的组件执行测试。领域实现和领域验证与确认之间的相互关系见图 7。

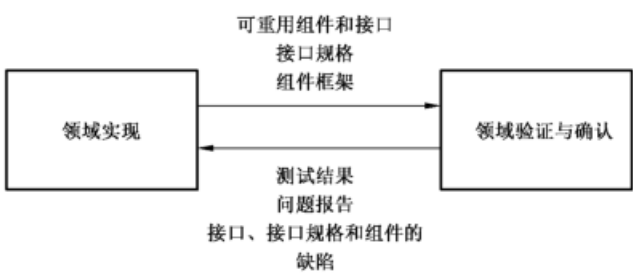


图 7 领域实现和领域验证与确认之间的相互关系

领域验证与确认为领域实现提供验证与确认结果。验证与确认结果包括三部分内容:表明验证与确认的对象是否通过测试,描述验证与确认对象以何种方式失败的问题报告,以及缺陷描述。

7.7 产品线范围界定和应用需求工程之间的相互关系

产品线范围界定通过指定产品宜具有哪些公共和可变特征规定了宜从应用需求工程中产生哪些产

品。应用需求工程细化了产品线范围界定所规定的特征。

应用需求工程为产品线范围界定提供了关于新增或变更特征建议的反馈,这些反馈来源于所获得的新的需求。

产品线范围界定和应用需求工程之间的相互关系见图 8。

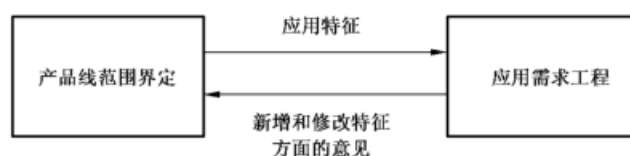


图 8 产品线范围界定和应用需求工程之间的相互关系

7.8 领域需求工程和应用需求工程之间的相互关系

领域需求工程为应用需求工程提供了预定义的公共和可变需求资产以及可变性模型。可变性模型支持产品线可变性的表示和领域需求资源的重用。领域需求工程和应用需求工程间的相互关系见图 9。

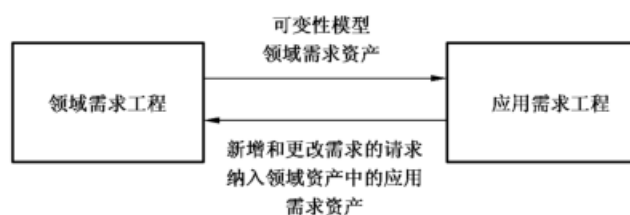


图 9 领域需求工程和应用需求工程之间的相互关系

应用需求工程可提供新增和变更的需求以及应用需求资产,这些应用需求资产可被纳入领域需求工程。因此,领域资产可能会变更,软件产品线也可能随着时间的推移而演化。产品经理和管理产品线演化的其他利益相关方将决定应用工程的输入是否影响领域资产。

7.9 领域设计和应用设计之间的相互关系

领域设计为应用设计提供领域架构,应用设计通过此领域架构实现单个产品的设计。应用设计应遵守产品线架构性构造中定义的规则。

应用设计为领域设计提供新增和变更的设计资产要求。当应用架构师发现架构性构造不足以满足特定产品时,应用设计可为领域设计提供将集成到平台中的设计资产。这些资产是应用架构中有助于产品线的新开发部分。将它们集成到领域架构中有助于产品线的演化。领域设计和应用设计之间的相互关系见图 10。

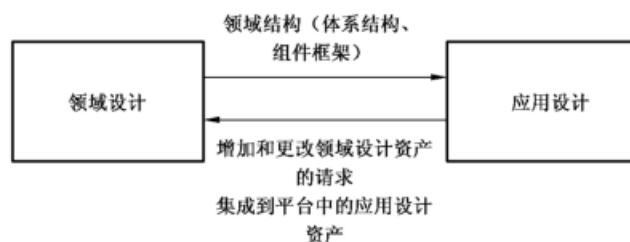


图 10 领域设计和应用设计之间的相互关系

7.10 领域实现和应用实现之间的相互关系

领域实现为应用实现提供已设计、实施、验证与确认的可重用组件和接口。它还为应用实现提供配置支持,以组装特定的成员产品(例如,通过提供配置管理工具)。

应用实现为新增和变更的领域实现资产提供领域实现要求。这涉及到宜由领域资产提供,却没有被很好地实现或根本没有被实现的功能或质量。应用实现为领域实现提供特定应用的组件和接口,这些组件和接口对于整个产品线的实现可能会起很大作用。领域实现决定是否将这些应用资产合并到产品线中。领域实现和应用实现之间的相互关系见图 11。

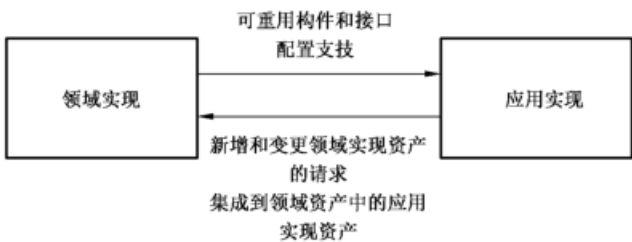


图 11 领域实现和应用实现之间的相互关系

7.11 领域验证与确认和应用验证与确认之间的相互关系

领域验证与确认提供可重用的应用验证与确认资产,例如测试用例。所有领域验证与确认资产,包括已在领域验证与确认中使用过的资产,将提交给应用验证与确认。同其他领域资产一样,领域验证与确认资产可能包含可变性。应用验证与确认过程应与可变性进行绑定,以获取特定成员产品的验证与确认资产。即使受绑定变体影响的部分已经在领域验证与确认中进行了测试,它们仍将根据影响分析的结果进行重新测试。

应用验证与确认可为领域验证与确认提供领域验证与确认资产中的缺陷,以及即将集成到领域验证与确认资产中的应用验证与确认资产。开发应用验证与确认资产主要用于测试特定应用特征等目的。如果应用验证与确认资产被集成到领域资产中,则相关的测试、设计和实现资产也应被集成进去。领域验证与确认和应用验证与检查之间的关系见图 12。

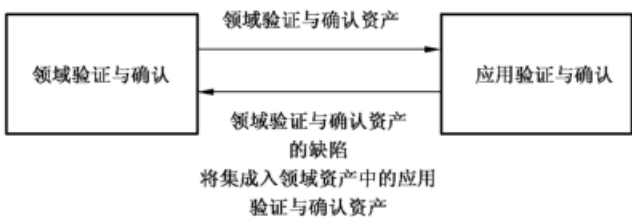


图 12 领域验证与确认和应用验证与确认之间的关系

7.12 应用需求工程和应用设计之间的相互关系

应用需求工程为应用设计提供应用需求规格说明,包括,通过领域可变性模型定义的可变性,以及结合新的可变特征得到的应用可变性模型,这些特征将成员产品与市场中的其他成员产品区分开来,并能够根据市场需求、应用需求资产和需求增量对成员产品进行大规模定制,其中部分应用需求资产可能来源于领域需求资产的重用或调整。需求增量是通过分析并比较客户或产品经理提出的应用需求与相关领域需求资产确定的。应用设计基于应用需求规格说明和领域设计提供的领域架构导出应用架构。

应用需求工程和应用设计之间的关系见图 13。

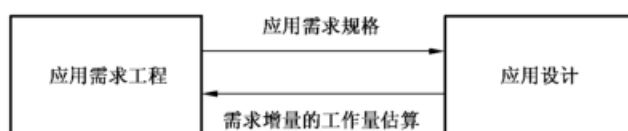


图 13 应用需求工程和应用设计之间的关系

应用需求工程通常包括根据客户提出的需求进行的权衡决策。需求实现的工作量取决于可重用的程度。需求增量可能导致对领域架构和可重用组件进行较大的修改,如,实现高于产品线工程师预期的性能需求。由于修改会影响应用工程的成本,所以有必要决定是否接受更高的价格、更长的时间以及放弃高成本的需求。

7.13 应用设计和应用实现之间的相互关系

应用实现基于应用架构和领域实现资产来构建成员产品。应用架构决定要构建的成员产品结构及构建规则,这些规则包含在某种构造文件中。应用架构还确定了作为成员产品一部分的重用领域组件和接口配置,以及它们与应用组件和接口的相互关系。

应用实现为应用设计提供了实现过程中出现的、应由架构师解决的设计错误。包括不可用的应用组件和接口、无法正常工作的配置等。应用设计和应用实现之间的关系见图 14。

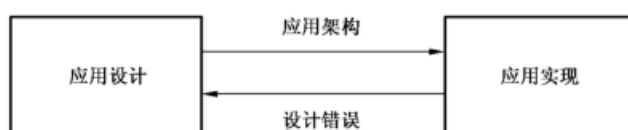


图 14 应用设计和应用实现之间的关系

7.14 应用需求工程和应用验证与确认之间的相互关系

应用系统测试根据应用需求规格说明验证成员产品。系统测试应确保成员产品实现所有的应用需求。根据应用可变性模型中的变体绑定,选择并绑定成员产品的变体后,应用验证与确认将绑定领域验证与确认资产中的各个可变点,并根据需要创建新的应用验证与确认资产,以验证应用需求。应用需求工程和应用验证与确认之间的关系见图 15。

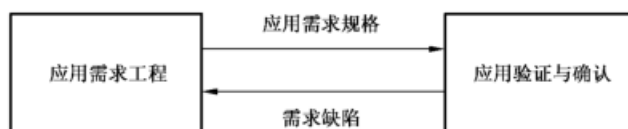


图 15 应用需求工程和应用验证与确认之间的关系

应用验证与确认应报告需求缺陷,如,不明确或不完整的需求,以及应用需求工程变体配置中的错误。

7.15 应用设计和应用验证与确认之间的相互关系

应用设计创建的应用架构用以驱动应用集成测试。组成成员产品的组件可以分为领域组件和应用组件,包括修改后的领域组件和新的特定应用组件。即使领域组件已经过测试,也宜使用适当的领域验证与确认资产来重新测试它们,因为组件的组合可能会导致不可预见的缺陷和影响。

应用设计在应用集成测试过程中得到确认。测试工程师应确保设计中的可变性已被正确绑定,并且应用设计是可测试的。每当测试工程师无法完全确定集成测试用例或执行测试所需的数据时,一种不完整性或模糊性就应该被检测出来。应用设计资产中的任何缺陷都应报告给应用设计。应用设计和应用验证与确认之间的关系见图 16。

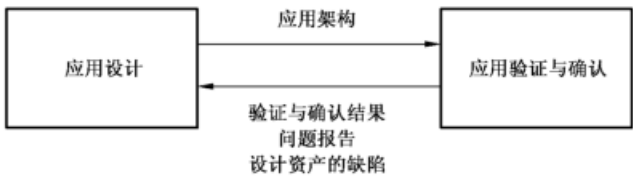


图 16 应用设计和应用验证与确认之间的关系

7.16 应用实现和应用验证与确认之间的相互关系

应用实现为应用验证与确认提供一个可执行的成员产品,以及相关的已设计实现并可以进行单元测试的应用组件和接口。

应用验证与确认为应用实现提供验证与确认结果、问题报告以及应用接口和组件中的缺陷。应用实现和应用验证与确认之间的关系如见图 17。

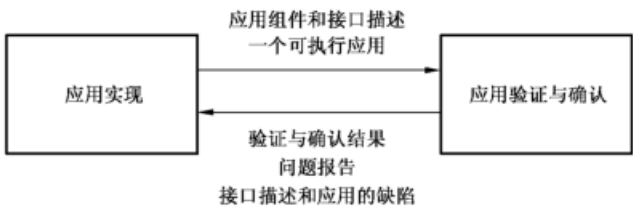


图 17 应用实现和应用验证与确认之间的关系

附 录 A
(资料性)
有关产品的更多信息

本文件适用于系统工程和软件工程领域。但是应注意的是,系统工程产品只涉及架构设计,并不包含与实现相关的细节。这些实现细节分属不同技术学科,如力学,电子学,化学和光学等。软件工程属于技术学科之一,它通过编写计算机程序的方式来实现。由于软件工程的非物理性质,该学科可以将相同的概念重新用于软件产品线工程与管理。处理其他技术学科的产品线问题不在本文件的范围内。

图 A.1 是参照其他标准进行资产分级分类管理的例子。

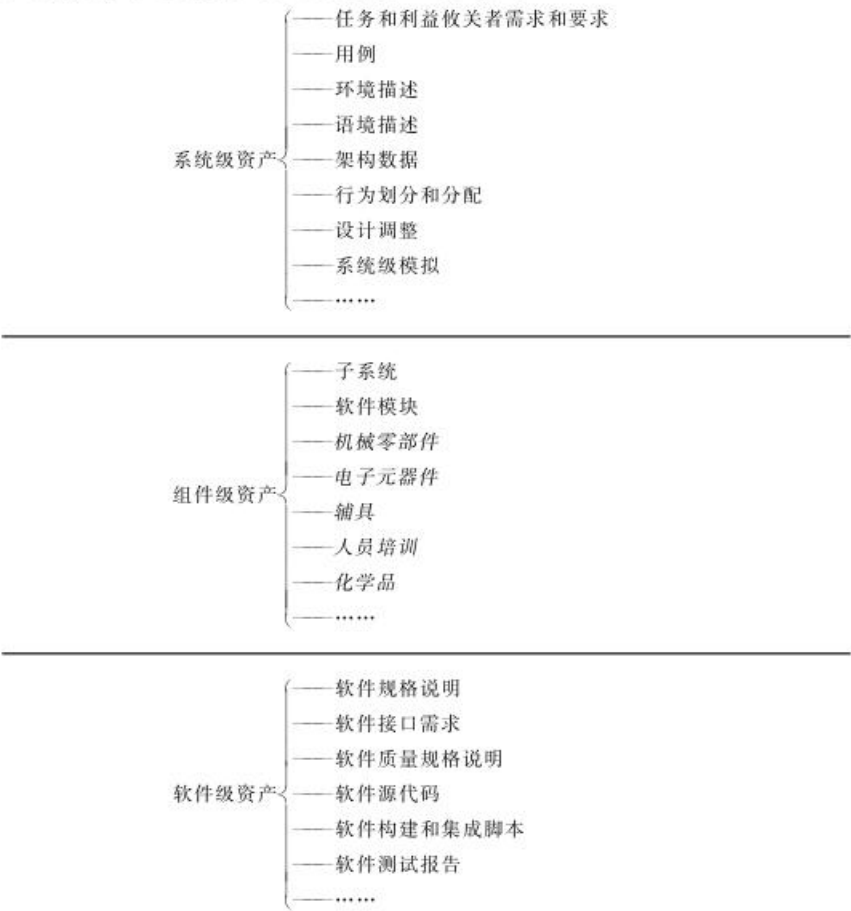


图 A.1 资产级别和类型示例

图 A.1 中斜体部分描述的资产超出本文件的范围。

参 考 文 献

- [1] GB/T 8566—2022 系统与软件工程 软件生存周期过程
- [2] GB/T 22032—2021 系统与软件工程 系统生存周期过程
- [3] GB/T 25000.1—2021 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第1部分:SQuaRE 指南
- [4] GB/T 30972—2014 系统与软件工程 软件工程环境服务
- [5] ISO/IEC 14102:2008 Information technology—Guideline for the evaluation and selection of CASE tools
- [6] ISO/IEC/TR 19759 Software Engineering—Guide to the Software Engineering Body of Knowledge (SWEBOK)
- [7] ISO/IEC/IEEE 24765:2010 Systems and software engineering—Vocabulary
- [8] ISO/IEC 26551:2016 Software and systems engineering—Tools and methods for product line requirements engineering
- [9] ISO/IEC 26552:2019 Software and systems engineering—Tools and methods for product line architecture design
- [10] ISO/IEC 26553:2018 Information technology—Software and systems engineering—Tools and methods for product line realization
- [11] ISO/IEC 26554:2018 Information technology—Software and systems engineering—Tools and methods for product line testing
- [12] ISO/IEC 26555:2015 Software and systems engineering—Tools and methods for product line technical management
- [13] ISO/IEC 26556:2018 Information technology—Software and systems engineering—Tools and methods for product line organizational management
- [14] ISO/IEC 26557:2016 Software and systems engineering—Methods and tools for variability mechanisms in software and systems product line
- [15] ISO/IEC 26558:2017 Software and systems engineering—Methods and tools for variability modelling in software and systems product line
- [16] ISO/IEC 26559:2017 Software and systems engineering—Methods and tools for variability traceability in software and systems product line
- [17] ISO/IEC 26560:2019 Software and systems engineering—Tools and methods for product line product management
- [18] IEEE 1517—2010 Information Technology—System and Software Life Cycle Processes—Reuse Processes
- [19] Klaus Pohl, Günter Böckle, Frank J. van der Linden. Software Product Line Engineering: Foundations, Principles and Techniques. Springer 2005
- [20] Northrop Linda M., & Clements Paul C. A Framework for Software Product Line Practice, Version 5.0. Software Engineering Institute, Carnegie Mellon University, July 2007
- [21] Käkölä T., & Duenas J. C. Software Product Lines—Research Issues in Engineering and Management. Springer, 2006
- [22] Bosch J. Design and Use of Software Architectures. Adopting and Evolving a Product-Line Approach. Addison—Wesley Professional, 2000

[23] Clements P. C., & Northrop L. M. *Software Product Lines: Practices and Patterns*. Addison-Wesley Professional, 2001

[24] van der Linden F. J., Schmid K., Rommes E. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Springer, 2007

中 华 人 民 共 和 国
国 家 标 准
软件与系统工程

产品线工程与管理参考模型

GB/T 41865—2022/ISO/IEC 26550:2015

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)

网址:www.spc.org.cn

服务热线:400-168-0010

2022年10月第一版

*

书号:155066·1-71147

版权专有 侵权必究



GB/T 41865-2022