

ICS 35.080
L 77



中华人民共和国国家标准化指导性技术文件

GB/Z 18493—2001
idt ISO/IEC TR 15271:1998

信息技术 软件生存周期过程指南

Information technology—Guide for software life cycle processes

2001-11-02 发布

2002-06-01 实施

中 华 人 民 共 和 国 发 布
国家质量监督检验检疫总局

前 言

本指导性技术文件等同采用 ISO/IEC TR15271:1998《信息技术 软件生存周期过程指南》。

本指导性技术文件由中华人民共和国信息产业部提出。

本指导性技术文件由中国电子技术标准化研究所归口。

本指导性技术文件由上海计算机软件技术开发中心和中国电子技术标准化研究所负责起草。

本指导性技术文件主要起草人：刘光龙、黄嘉启、宿为民、冯惠、谢晓燕。

ISO/IEC 前言

国际标准化组织(ISO)和国际电工委员会(IEC)是世界性的标准化专门机构。国家成员体(它们都是 ISO 或 IEC 的成员国)通过国际组织建立的各个技术委员会参与制定针对特定技术范围的国际标准。ISO 和 IEC 的各技术委员会在共同感兴趣的领域内进行合作。与 ISO 和 IEC 有联系的其他官方的或非官方的国际组织也参与国际标准的制定工作。

对于信息技术领域,ISO 和 IEC 建立了一个联合技术委员会,即 ISO/IEC JTC 1。

技术委员会的主要任务是起草国际标准,但在例外情况下,技术委员会可以提出下列类型之一的技术报告。

- 类型 1:虽然一再努力,但仍不能获得出版一项国际标准所需要的支持时;
- 类型 2:所讨论的项目仍处于技术发展阶段;
- 类型 3:当技术委员会所搜集到的各类数据与正式出版为国际标准的数据不同(例如:技术发展水平)时。

第 1 和第 2 种类型的技术报告在出版后 3 年内应提交复审,以决定是否将它们转成国际标准。第 3 种类型的技术报告不是务必要进行复审,除非它们提供的数据已被认为不再有效和不再有用。

ISO/IEC TR 15271 技术报告属于上述的第 3 种类型,它是由 ISO/IEC JTC1“信息技术”联合技术委员会的 SC 7“软件工程”分技术委员会制定的。

1 范围

1.1 目的

本指导性技术文件的目的是为软件生存周期过程标准(GB/T 8566)的应用提供指南。

本指导性技术文件对应用 GB/T 8566 时必须考虑的各种因素作了详细说明,并就可以应用 GB/T 8566 的各种不同场合作了说明。本指导性技术文件并不打算对 GB/T 8566 标准的基本原理加以阐述。

本指导性技术文件讨论了三种主要的生存周期模型并且给出了一些剪裁示例。

1.2 读者

本指导性技术文件是为在签定合同时(不考虑项目的大小或复杂度),或在一个组织内进行自我评估时,或在进行软件过程的改进时,使用或应用 GB/T 8566 标准的人员而写的。

本指导性技术文件主要讨论对于不同类型的软件如何使用 GB/T 8566 标准,并且指出每种情况与哪些过程有关。

当 GB/T 8566 被用作为需求文档和作为指导模板使用时,本指导性技术文件对其提供支持。(GB/T 8566 自己被用作为过程改进活动的一部分,就是后者的一个例子。)整个标准都必须理解,但可通过参照特定的条款把它应用于各种特定的情况。

1.3 先决条件

使用本指导性技术文件的先决条件有:

- a) 能得到 GB/T 8566;
- b) 熟悉 GB/T 8566;
- c) 熟悉相关组织的政策;
- d) 具有软件管理、软件工程和软件生存周期模型的一般知识。

2 引用标准

下列标准所包含的条文,通过在本指导性技术文件中引用而构成为本指导性技术文件的条文。本指导性技术文件出版时,所有版本均为有效。所有标准都会被修订,使用本指导性技术文件的各方应探讨使用下列标准最新版本的可能性。

GB/T 8566—2001 信息技术 软件生存周期过程(idt ISO/IEC 12207:1995)

GB/T 16260—1996 信息技术 软件产品评价 质量特性及其使用指南
(idt ISO/IEC 9126:1991)

ISO/IEC TR 15504(所有部分) 信息技术 软件过程评估

3 标记法

图1给出了按 GB/T 8566 中使用的风格绘制的过程和活动的图形标记。

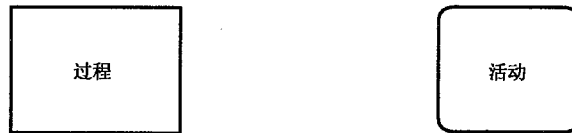


图1 图形标记法

4 GB/T 8566 中的一些基本概念

4.1 工程规范

与一些传统的工程学分支相比,软件工程的应用与实践还是一门比较年轻的学科。因此,传统工程项目所常用的控制手段,不一定总能在软件中实现。

GB/T 8566 的基本原理是,对于如软件开发与维护这些活动应当按工程规范的方式进行。遵循这一方法,就可以建立一个与系统工程环境有着明显联系的框架,也就是一个包括软件、硬件、人员和业务实践的框架。

4.2 软件生存周期的体系结构

GB/T 8566 建立了从概念开始直到退役为止的软件生存周期的顶层体系结构。此体系结构由一组过程和这些过程间的相互关系组成。这些过程基于两个基本原则:模块化和责任。

4.2.1 模块化

GB/T 8566 标准的过程是模块化的,因为它们是:

- 1) 强内聚的。一个过程的所有部分都紧密地相互联系。
- 2) 低耦合度的。过程之间的接口保持在最少数量。

原则上,每个过程在生存周期内只专注于一个单独的功能,并可以使用关于专用功能的另一个过程。下面介绍关于过程的确立、范围与构造的规则:

- a) 一个过程必须是模块化的。即在整体生存周期中一个过程宜且只宜执行一个功能,而且任意两个过程之间的接口宜达到最少;
- b) 每个过程应在该体系结构中被引用;
- c) 如果过程 A 只被过程 B 引用,则过程 A 属于过程 B;
- d) 如果一个功能被多个过程引用,则该功能本身成为一个过程;
- e) 在软件生存周期模型中,任何功能都必须是可以验证的;
- f) 每一个过程宜具有充分确定的内部结构,以便于执行。

4.2.2 责任

在 GB/T 8566 中,“组织”和“参与方”这两个术语的含义几乎相同。组织是因为某种特定的目的组成的团体,其形式可以多种多样,如公司、代理机构、协会、联盟或俱乐部。“组织”的大小可从一个人到许多人。当“组织”整体或部分地进入一个合同时,它就成为“参与方”。“组织”是单独的实体,但“参与方”可能来自同一个“组织”或来自多个分离的“组织”。

GB/T 8566 中的每一个过程都被看作“参与方”的职责。一个组织可以执行一个或多个过程。一个过程也可以被一个或多个组织执行,其中一个组织被认定为责任方。执行一个过程的责任方需要对整个过程负责,即使各个任务的执行是由不同的人员进行的。

软件生存周期体系结构的责任特征,使得那些可以由许多人正式参加的项目剪裁和应用 GB/T 8566 标准更容易。

4.3 过程的性质

过程被分成三大类:

- a) 基本过程;
- b) 支持过程;

c) 组织过程。

4.3.1 基本过程

基本过程有：

- a) 获取过程；
- b) 供应过程；
- c) 开发过程；
- d) 运作过程；
- e) 维护过程。

在实践中，获取过程启动整个软件生存周期。供应过程通过使用开发、运作和(或)维护过程来对获取过程作出响应。

4.3.2 支持过程

支持过程有：

- a) 文档编制；
- b) 配置管理；
- c) 质量保证；
- d) 验证；
- e) 确认；
- f) 联合评审；
- g) 审核；
- h) 问题解决。

一个支持过程可以被另一个过程所使用，这样使前者得到了某种特定用途的支持。

4.3.3 组织过程

组织过程有：

- a) 管理；
- b) 基础设施；
- c) 改进；
- d) 培训。

一个组织可以在组织范围内使用这些过程，以建立、实现和改进生存周期过程。

4.3.4 过程的细化

每个过程可用其构成的活动来定义，而每个活动又可以其构成的任务来定义。过程内的活动是一组紧密相关的任务。在 GB/T 8566 内，它们如表 1 所示：

表 1 过程的分解

类 型	过 程	活 动	任 务
基本类	5	35	135
支持类	8	25	70
组织类	4	14	27
合 计	17	74	232

任务以需求、自声明、建议或可允许动作等形式表示。为此，GB/T 8566 谨慎使用某些助动词以区别任务的不同表达方式。

——“**Shall**”(应)表示双方或多方之间有约束力的规定。

——“**Will**”(意愿)表示一方作出的目的或意向申明。

——“**Should**”(宜)表示多种可能性中的一种建议。

——“May”(可以)表示 GB/T 8566 范围内所允许的动作。

4.4 过程与项目

GB/T 8566 描述了用于大且(或)复杂的软件项目的一组过程。然而,GB/T 8566 被设计成可剪裁的,以适用于任何类型以及规模较小、复杂性较低的软件项目。并且在设计时已考虑到,不管软件是一个独立的实体,还是作为整个系统的一部分,都可使用 GB/T 8566。

在 GB/T 8566 中,过程、活动和任务以最一般和最自然的位置顺序安排。这个位置顺序并不表示就是生存周期模型的次序,其意图是要让软件项目合适地或适当地选择、整理、剪裁和重复这些过程、活动和任务。

对于同一个项目,GB/T 8566 可以分别地多次使用。比如,在一个给定的软件项目中,需方可能要求供方同需方一起进行软件开发,并且要求供方应用 GB/T 8566。然后供方可以要求它的分承包商完成全部或部分软件开发。供方(此时处于获取地位中)和它的分承包商(此时处于供应者地位中)分别应用了 GB/T 8566。在两种情形下,均有必要对 GB/T 8566 进行剪裁,以反映各自的安排。

更详细的内容,请参阅第 6 章“应用于项目”。

4.5 过程与组织

一个组织(或参与方)从当前进行的过程中获得其称谓。例如,进行获取过程时它就叫做获取方或需方。

为了适应多种多样的组织,GB/T 8566 中的过程组成一个综合性的集合。无论大的或小的组织均可以依据它的业务目标,从这些过程(和相关的活动与任务)中选择合适的子集来实现其目标。GB/T 8566 的意图是应用于一个组织的内部或被两个或多个组织用于处理合同关系。为便于在内部和合同情况下应用 GB/T 8566,标准内的任务都是用合同式的语言表达的。在内部使用时,合同式的语言被解释成自己要求的任务,正如第 7 章(应用于组织内部)所述的。

GB/T 8566 要与组织已有的策略和标准协调一致。通常情况,组织一直都在使用它自己现存的标准和特定的软件开发技术。因此,在组织内部应用 GB/T 8566 时,重要的是要澄清 GB/T 8566、组织自己的标准,以及已使用的不同技术之间的关系。

图 2 给出了这种关系的一个可能的例子,它在组织内部应用 GB/T 8566 时或许有用。GB/T 8566 处于第一层,组织内部的标准处于第二层,第三层是用于特定项目的具体的开发活动、开发技术和开发工具。在第二、三层中定义和使用的术语都必须与 GB/T 8566 相一致。

任何冲突解决都留给应用 GB/T 8566 的组织来完成,可能包括建立映射,必要时还需填充缺少的部分。

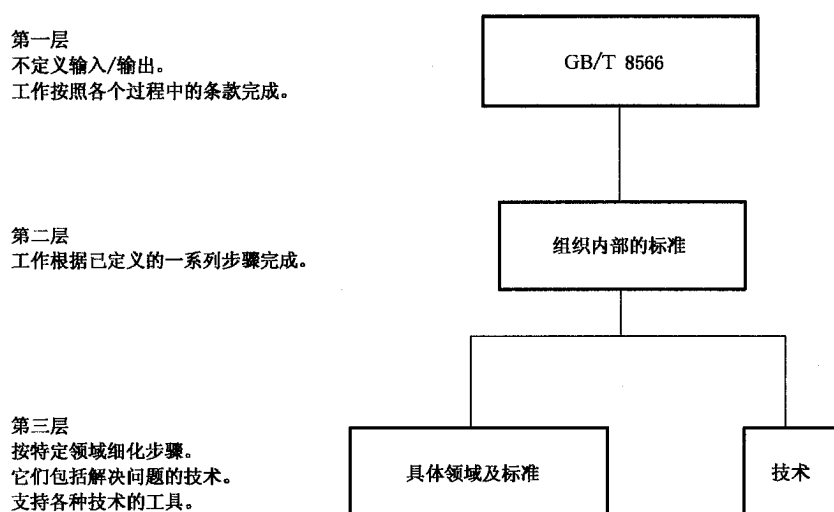


图 2 和现存文档的关系

4.6 软件与系统

4.6.1 与系统工程的接口

GB/T 8566 使整个系统和软件之间建立了很强的联系。这可能是由于 GB/T 8566 是建立在一般的系统工程的基础上的缘故。

就某种程度而言,GB/T 8566 的设计目标是要在系统工程过程的范围内起作用。当软件是整个系统的一部分时,软件首先要从系统中分离、然后生产出软件、并反过来集成到系统中。在没有可用的系统级标准时,GB/T 8566 的这一特点是有用的。当软件关系到全局利害时,系统级的任务可作为一种有用的指南。其他情况下,GB/T 8566 能使得软件工程有效地参与到系统工程中。

4.6.2 软件与系统的关系

一个系统是硬件、计算机、软件、材料、人员和设备的特定组合(如图 3 所示)。实际上,这正是一个系统所必须具备的要素。在父系统中存在一些过程,如业务过程。软件通过在计算机上执行这些过程的某些功能提供服务。软件可以常驻于计算机内、嵌入于一片固件或集成到一个硬件组件中。任何情况下,软件的获取、供应、开发、运作和维护都需要与父系统的这些过程相协调。

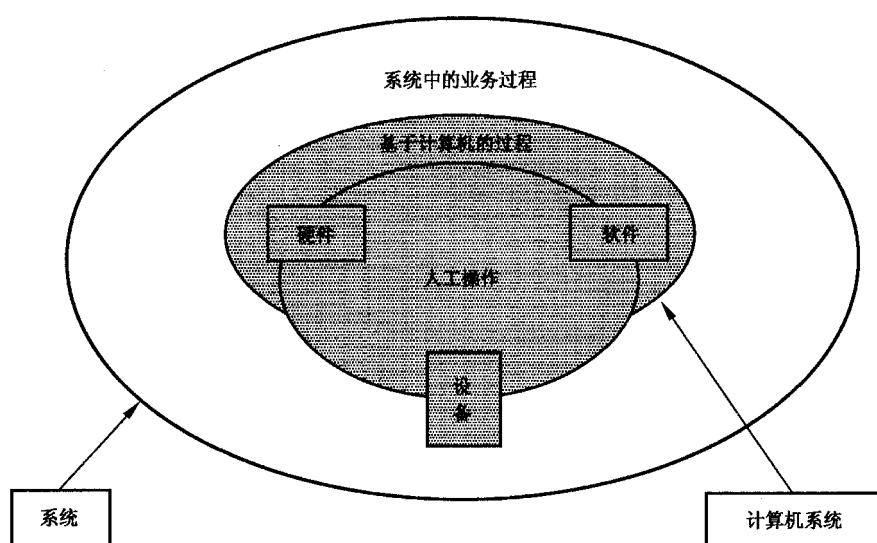


图 3 系统中的软件

在组织内部可能有许多计算机系统支持着业务过程(如图 4)。

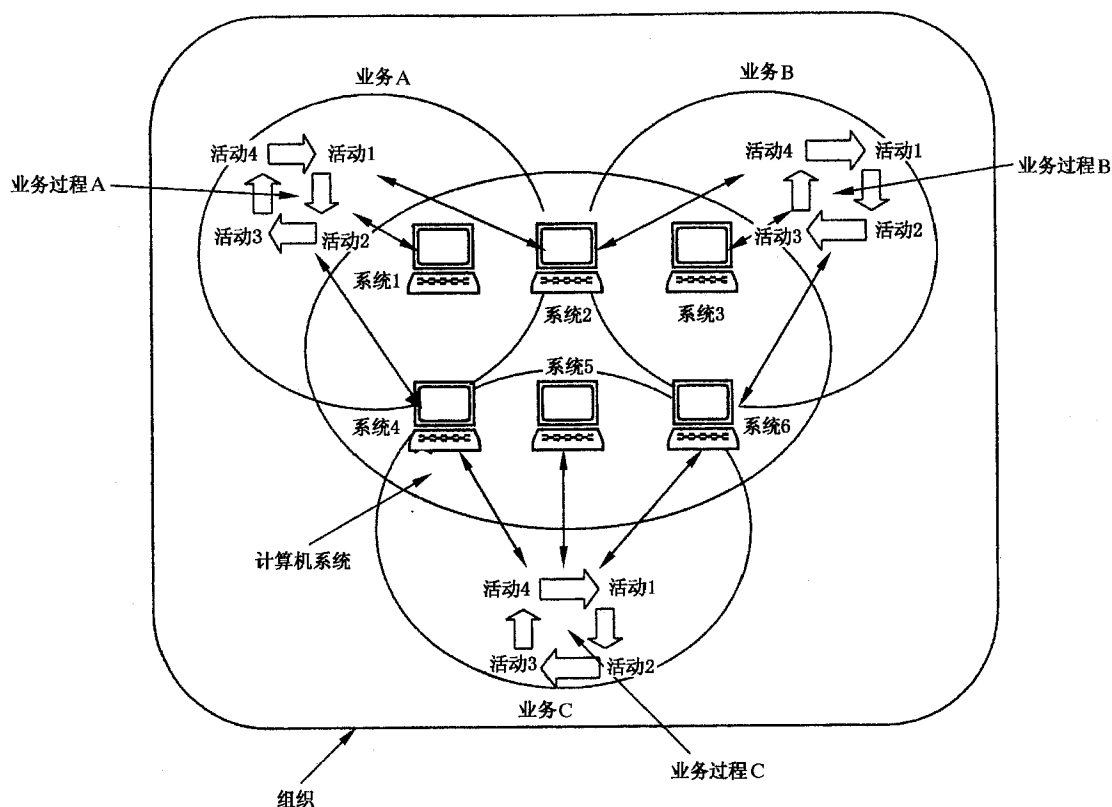


图 4 组织中的计算机系统

4.6.3 以软件为基础的系统

尽管 GB/T 8566 定义了系统,但它只包括了以软件为主的系统的生存周期过程(如开发过程、运作过程和维护过程)。因此,GB/T 8566 中没有关于硬件生存周期过程的定义。

4.6.4 系统与软件活动的分类

在 GB/T 8566 的开发过程中有两种类型的活动,它们是系统活动和软件活动。这些活动的名字反映了它们的范围。

图 5 把这些活动按类型分成两组,用“V”字形描述阐明系统活动和软件活动之间的对称及相关性。

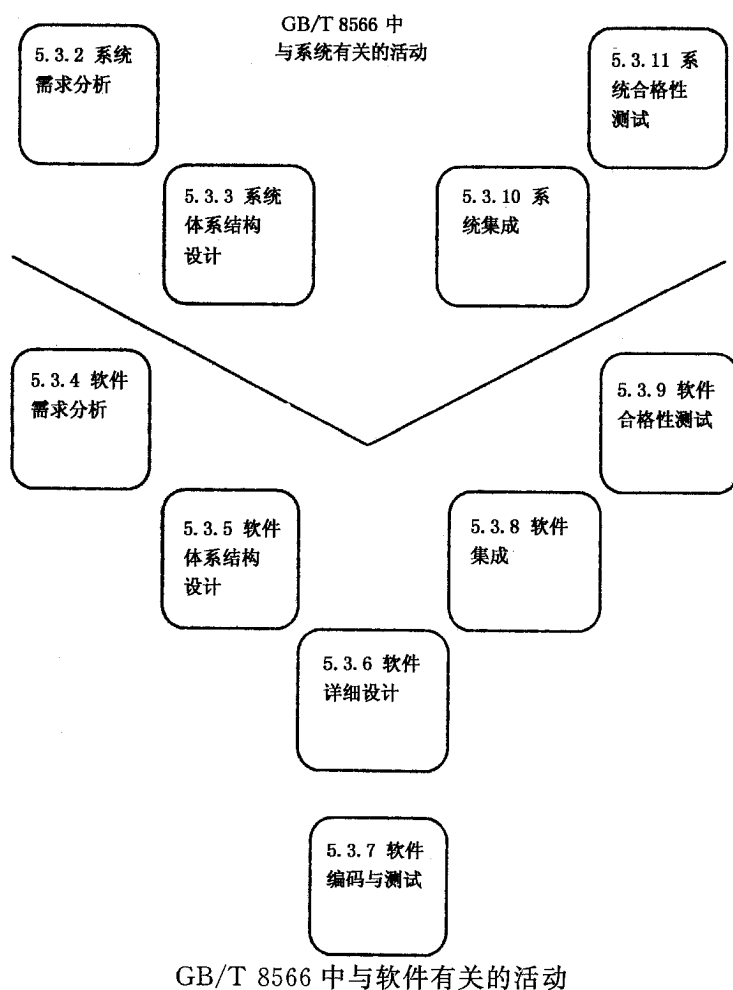


图 5 GB/T 8566 的活动分类

如图 5 所示,GB/T 8566 的开发过程的系统活动始于系统需求分析(5.3.2),止于系统合格性测试(5.3.11)。

本指导性技术文件的第 8 章描述了系统是怎样由硬件、软件和人工操作而组合成的。将系统分成这些要素是通过系统体系结构设计活动(5.3.3)完成的。从这个体系结构设计得出的软件活动从软件需求分析(5.3.4)开始直到软件合格性测试(5.3.9)为止。

一旦软件开发结束,硬件和人工操作就通过 GB/T 8566 的软件集成(5.3.10)进行集成,接着进行系统合格性测试(5.3.11)。根据以上描述的活动,可以得到推论:系统活动是软件活动的超集。

4.7 管理与规划

对于每一个基本过程和支持过程,项目级的过程管理是按照管理过程的以下实例进行的。正是通过这个管理过程,达到对所有其他计划好的事件的规划、执行和控制。GB/T 8566 的 7.1.2.1 中定义了计划所必须包含的条款,7.1.3 为过程报告作规定,7.1.3.3 则用于问题报告。

4.7.1 项目管理计划

在供应过程中,GB/T 8566 的 5.2.4.5 要求产生项目管理计划,并在 5.2.5.1 中执行和控制这个计划。在 5.2.5.3 中支持过程进一步规定了针对技术性能、成本和计划表的监督和报告。

4.7.2 附属计划

GB/T 8566 的 5.2.4.5 中列出的需要考虑的项,包括那些与支持过程类过程和组织过程类过程有关的项。许多这类过程要求制定计划,如质量保证、验证和培训。依据项目的规模和复杂程度,同时考虑到需要转包部分或全部工作,这些计划或者被合并到项目管理计划中,或者生成独立的附属文档。

当用到分包商时,建立适当的接口变得尤为重要,为此,这些附属计划应该通过 GB/T 8566 的 5.2.4.5 进行管理,以使计划同步。

这套附属计划的概要可以从表 B2 和表 B3 得到。

4.7.3 文档控制

关于管理文档(包括计划)的有关要求在文档编制过程中作了描述。

4.8 质量管理原则的实现

GB/T 8566 以下面三种基本方式实施质量管理原则。

4.8.1 质量集成到生存周期中

对覆盖整个软件生存周期的一组综合的、集成的过程,GB/T 8566 给出了有关需求。它通过改进过程为每个过程提供了对“计划—实施—检验—行动”环的访问。它将所有与质量有关的活动看作为整个软件生存周期的一部分,并且使那些活动与生存周期的相关过程相适应。也就是,对每个过程及负责执行过程的人员都分配了与过程内部质量有关的活动。

4.8.2 质量保证过程

质量保证过程是为了确保产品与服务遵从对它们所确立的需求和已建立的计划。负责这些过程的人员都被授予必要的组织上的自主和权力。组织上的自主是指相对那些直接负责生产产品的人员的自主,而权力意为进行评估与纠正行动的权力。

4.8.3 改进过程

GB/T 8566 包括一个改进过程,用于在组织范围内(如与合同要求无关的)进一步的质量改进。

4.9 对发展中的新技术的灵活性与响应能力

GB/T 8566 对发展中的软件工程规范来说是灵活的和易响应的。它通过提供顶层的开放式的体系结构实现这一点,即 GB/T 8566 是:

a) 对以下任何一种情况都可用的:

——生存周期模型(如:瀑布型、增量型、演化型);

——软件工程方法或技术(如:面向对象的设计、结构化编码、自顶向下的测试、原型法);

——编程语言(如:COBOL、Ada、汇编);

这些将依赖于项目和技术现状,故把它们的选择留给 GB/T 8566 的使用者。

b) 灵活的(从高层次的观点来看),即软件生存周期过程的活动和任务是“做什么”的项,而不是“怎么做”的项。也就是说,一个任务可能是“开发和编制体系结构设计文档”,但不是“用自顶向下的功能设计方法开发或编制体系结构设计文档”。这种方案为需方提供了一个途径来规定最终的产品和服务,而同时允许供应商有创新性并可选择适当的方法、技术以及工具来生产产品和提供服务;

c) 适应于任何诸如军事或商业应用等具体产业的习惯做法,或者适用于任何国家或组织的文化。

4.10 过程与文档编制

GB/T 8566 不是一个文档编制标准,即尽管 GB/T 8566 要求有过程的输出文档,但它没有规定文档的格式和内容,也没有说明怎样将性质相似的输出文档(如计划、规格说明书、测试需求)相关联起来。关于文档编制需求的详细介绍,可参阅附录 B。

4.11 软件度量

GB/T 8566 没有用特定的度量术语和指标项来定义或规定软件的属性(如:可靠性和可维护性)。它的确给出了规定这种软件属性的方法,但是把细节留给了 GB/T 8566 的用户。

4.12 依从性

可以按照以下方法遵守 GB/T 8566:

a) 组织公开声明以 GB/T 8566 的一套过程、活动和任务作为交易的条件,该组织的供应者都要遵守;

b) 软件项目剪裁合适的过程、活动和任务,它们依照合同建立的准则来执行。

注：GB/T 8566 包含一套以“应”形式陈述的需求。GB/T 8566 的用户不一定遵从整个标准，因为 GB/T 8566 中还有在遵从整个标准时要求的不带“应”形式的陈述。这可以由双方协商确定。各参与方可以同意将 GB/T 8566 本身作为协议的基础，或者同意为满足他们特定的需求剪裁 GB/T 8566。

4.13 小结

GB/T 8566 不是软件系统的系统化的、规范的管理和工程的替代品。GB/T 8566 提供了能合适地表示标识、计划和执行同软件相关的过程、活动和任务的框架。GB/T 8566 包含一套已定义好的积木块（过程）。GB/T 8566 的用户必须选择、剪裁并装配它们，使之对于组织和项目是适合、有效的。剪裁 GB/T 8566 时宜保持它的体系结构、目的及完整性。如：包含该要素，但注明“不适用”并且加上不应用的原因。

5 GB/T 8566 的实施

5.1 概述

实施 GB/T 8566 可能有各种各样的原因，其中包括：

- 用于特定的项目，定义其所需要的软件过程、活动和任务；
- 被一个组织用来改进其软件过程；
- 作为更大的系统生存周期模型过程的组成部分。

无论实施 GB/T 8566 的原因是什么，建议实施 GB/T 8566 的策略包括以下几点：

- a) 制定实施计划；
- b) 剪裁 GB/T 8566；
- c) 开展试验性项目；
- d) 方法的定型；
- e) 方法的制度化；

此策略是在组织或项目进行变革时，必须遵循的典型方法。当项目或组织需要采用一些其他附加的过程和（或）该过程被改进时，上述的实施策略可以进行多次重复。

当项目或组织已经处在稳定状态时，即 GB/T 8566 的过程已经建立并且制度化了，实施策略就可以被缩短，可能仅包括：

- a) 制定实施计划；
- b) 剪裁 GB/T 8566（按工作的风险级别）；
- c) 开展项目。

5.2 制定实施计划

实施 GB/T 8566 宜作为一个专门的项目来考虑，并为之制定计划。

以下是制定项目实施计划时应该考虑的条款的例子：

- a) 确定项目的范围。可能包括：

——一个单独的项目（或者是组织内部的，或者是双方的合同的一部分）；

——在期望为组织带来收益的地方，集中注意一些关键的过程甚至某个单独过程。这个方法可以在事先已发现薄弱点且将来可能促成全面实施 GB/T 8566 之处使用；

——可能以逐步引入的方式，对一系列项目采用 GB/T 8566。这里该组织可能没有或几乎没有已定义的过程，故应按照 GB/T 8566 来标准化。

——对所有项目和组织的所有部分，采用 GB/T 8566。除了非常小的组织外，很少有组织会采取这种方法。事先将 GB/T 8566 应用于工作实践是明智的，哪怕只是对于某个现存组织的新子公司。

b) 确定项目目标并决定它们如何适应整个组织范围的业务目标。如果在该项目和组织业务重点之间没有建立明显的连接，则实现项目目标的承诺将是困难的，或不可能维持；

- c) 确定项目小组或组织的角色与责任，同时为每个过程指定唯一的责任。许多情况下，一个组织或

个人可负责多个过程,特别是在小的项目或组织中;

d) 确定实施 GB/T 8566 可利用的资源,如:时间、资金、人员和设备;

e) 创建并编制实施 GB/T 8566 的项目管理计划文档。

5.3 剪裁 GB/T 8566

当需要进行剪裁的时候,就利用在 GB/T 8566 附录 A 中定义的剪裁过程。关于剪裁 GB/T 8566 用作特定用途的建议包含在本指导性技术文件的第 6 章(应用于项目)、第 7 章(在组织中的应用)和第 8 章(使用系统生存周期模型的应用)中。

本指导性技术文件中关于剪裁的建议,必须结合 GB/T 8566 附录 B 中的通用剪裁建议阅读,同时使用与环境相应的材料。例如,应用可以在组织内部并且包括系统生存周期模型的各方面。

本指导性技术文件的图 6 显示了剪裁过程的事件序列,这一点在 GB/T 8566 的附录 A 中有详细的讨论。本指导性技术文件的附录 D 中有专门关于剪裁的例子。这些例子:

- 利用具体场景来定义项目环境;
- 必要时,定义附加的活动和任务;
- 概述剪裁决定并给出剪裁的基本原理。

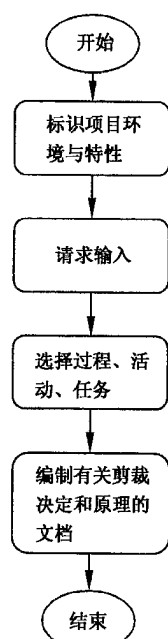


图 6 剪裁活动

5.3.1 标识项目环境与特性

组织的特性可以通过考虑以下问题决定:

——哪些过程、政策和程序已经就位? (标识现有的成功过程和实践是至关重要的。这些都需要合并到所需的整套过程中去);

——这个过程对达到组织的目标是主要的吗?

——存在高的业务风险吗?

——问题区域在哪里?

——组织的文化是什么? (倡导改革者和保守者?)

——需要哪些支持?

项目特性可以通过考虑以下问题决定:

——将采用什么系统或项目生存周期模型?

——特殊过程的成熟度是什么?

——技术风险有哪些?

——这是一个安全性十分重要的系统吗？

——采用了新技术吗？

5.3.2 请求输入

从相关业务和合同要求中得到的需求，是剪裁 GB/T 8566 的主要驱动源。例如，GB/T 8566 可以根据产品的供应者和购买者之间的合同进行剪裁。客户可能只要求进行软件的设计，而不是软件系统的全部开发。又如，如果客户的需求是安全性十分重要的软件，而不是消费性软件，那么从 GB/T 8566 中得到的任务的数目就大不相同了。

有关各方宜都参与剪裁决定。这些人员可协助确保经过剪裁得到的过程是可行的和有用的。如果可能，还包括以前项目的反馈。

5.3.3 选择过程、活动和任务

标识那些准备实施的 GB/T 8566 过程或其子集，并给定优先权。通常，可从那些会产生最重大效益的过程开始，而不是企图一下子实施整个 GB/T 8566。

GB/T 8566 没有定义过程、活动及任务的顺序，也没有规定任何特殊的软件生存周期模型。在这个阶段，一种可取的方法是把当前的过程、实践和(或)方法映射到 GB/T 8566 的过程、活动及任务上。

这种映射可用于验证方法的完整性，即标识当前情况与目标情况(即使用了 GB/T 8566 的过程之处)之间的差距在哪里。

5.3.4 编制有关剪裁决定和理由的文档

在应用 GB/T 8566 时，宜把从所定义的过程、活动和任务到已选的软件生存周期模型的映射，同已决定的剪裁关系及采纳它的理由一起编制成文档。若需评估所采用的方法(或过程)成功与否，还要将这些文档合并到有关项目管理计划中，以提供评估的参照框架。

附录 D 中的例子解释了如何编制映射文档。

5.4 进行试验性项目

当一个承担很多项目的组织引入 GB/T 8566 时，在关键领域内和为某关键过程进行试用将帮助该组织降低风险。成功的引入通常包括以下途径：

- a) 确定能利用所选过程的试验性项目。必须根据高优先权工作选择这些试验性项目，即它们能导致重大的改进，成功的可能性大，并且可以期望带来迅速而明显的效果；
- b) 选择一个志愿小组来进行试验性项目，然后宣传并酬谢他们的努力；
- c) 培训所有涉及的人员。除了正规的培训之外，在实施过程中定期进行交流将有助于提高认识；
- d) 制定试验性项目的计划和标识成功的关键因素；
- e) 对每个试验性项目，将所选的经过剪裁的过程纳入项目管理计划。适当地引用或包括本指导性技术文件的 5.3.4 中描述的文档。

f) 实施试验性项目，跟踪并记录所有关键成功因素的性能。把整个试验性项目取得的教训编成文档，并将这些取得的教训纳入到修正过程中。

5.5 方法的定型

方法的定型涉及一个新过程的引入，该新过程应跨越几个项目和(或)面向整个组织。诸如培训、文档编制、提供过程支持工具以及对新过程的使用与验收情况的跟踪观察等。宜对任何已经结束和正在进行的项目向新过程的转移进行规划。

注意：可以通过项目级的监督活动来对一个项目进行改进。也可以通过项目之间的比较决定哪些方法是成功的、哪些宜放到将来的项目中去来实现改进。

5.6 方法的制度化

制度化的重点是确保一个过程能够在整个项目或组织中得到一致、自动的贯彻。这也涉及到过程性能的度量，以及必要时所实施的过程改进。还可能包括对 GB/T 8566 中 7.3 所描述的改进过程的引用。

6 应用于项目

这一章给出将 **GB/T 8566** 应用于项目的附加指南。然而,该指南并非详尽无遗的,因为项目的情况各有不同。

影响如何获取、开发、运作和维护软件的因素包括如下几方面:

- 组织政策与规程的变更;
- 项目专用的获取/供应策略;
- 项目的规模和复杂度;
- 系统需求。

GB/T 8566 是为适应这种变更和为任何项目制定的。然而,为降低成本、改进质量,可能需要针对各特定的项目对 **GB/T 8566** 加以剪裁。宜让参加该项目的所有人员都有机会参与剪裁工作。

6.1 应用 **GB/T 8566** 的要素

下面的若干条考察在项目应用 **GB/T 8566** 时必须考虑的关键要素。这些要素不是详尽无遗的,只是代表当前的想法,并且会依赖于软件项目相互作用、相互影响。

若考虑具体的项目环境,则一种有效的方式是将这些要素组合到以下几方面中:

- 组织问题;
- 项目风险;
- 资源的能力/成熟度。

6.1.1 系统生存周期模型

使用 **GB/T 8566** 使之作为对软件项目的要求或指南,并对其加以剪裁的程度,依赖于软件项目在系统生存周期模型中的相对地位(参阅本指导性技术文件的附录 C“典型生存周期模型概述”)。

要标识软件在系统生存周期模型中的当前角色(参阅本指导性技术文件的 8.1)。这种判定有助于:

- 区别该软件是系统的一部分还是独立的;
- 标识 **GB/T 8566** 是否被用作进行计算机建模和模拟的方法;
- 标识 **GB/T 8566** 是否用来进行软件的开发、运作和维护。

参阅本指导性技术文件后几章的相关部分,并确保必要的接口均已在系统生存周期模型中建立。

6.1.2 组织的策略与规程

标识软件项目所必须遵从的涉及组织(尤其是需方与供方)的相关策略和规程。与策略和规程相关的例子有:

- 保密性;
- 安全性;
- 私密性;
- 风险管理;
- 独立的验证和确认机构的使用;
- 特定的计算机编程语言的使用;
- 硬件资源。

标识所有相关的法律和法规,包括那些可能影响软件项目的,与环境、公共安全及私密性相关的法律和法规。为确保系统依从这些法律和法规,可能有必要在项目进行过程中对这些外部原始资料加以不断的监控。

在软件的开发、运作及维护过程中,如果合适的话,上述策略与规程也需加以考虑。例如:如果有安全性和保密性策略,那么,开发过程的需求分析活动和运作过程的系统操作活动需要纳入这些策略。

6.1.3 系统特性

将子系统和配置项标识到适当的详细程度。标识系统的特性,尤其是那些分配给软件的特性,以及

那些对系统的运行起关键作用的特性。

宜考虑的系统级特性(如果分配给软件)的例子包括:

- 系统间和系统内部接口;
- 用户接口;
- 软件错误对系统安全性和保密性的影响;
- 计算机规模和时间方面的制约;
- 固件配备情况;
- 可提供的合适计算机。

如果系统有许多子系统或配置项,那么宜仔细地为每个子系统和配置项执行开发过程中的系统级活动。宜考虑所有的接口需求和集成需求。对于小规模的系统就不用如此严格。

6.1.4 软件特性

标识软件级的特性。例如包括:

- 软件配置项的潜在编号;
- 软件的类型、规模和关键性;
- 技术风险;
- 文档的类型、范围和介质;
- 是新开发? 修改? 或重用?
- 包含在 GB/T 16260 中的特性,如:可靠性;

如果软件有许多软件配置项、部件和单元,宜仔细地为每个软件配置项执行开发过程中的软件级活动。宜考虑所有的接口和集成需求。对于小规模软件就不必这么严格。

考虑到系统和软件特性的关键性,应确定软件所需的管理控制的程度,以及与评价相关的活动的范围。

6.1.5 软件维护策略

标识该软件维护应考虑的问题。应考虑的典型问题包括:

- 维护的期望周期;
- 变动/时间比;
- 该应用的重要性;
- 谁来进行维护;
- 培训的范围和程度;
- 软件维护(包括测试)所需要的环境。

宜考虑所有的文档编制需求,在软件预期有较长的维护周期或预期会有重大的变动时尤其需要。如果有可能,应当有文档编制环境,以保存和便于将来的检索。

在整个维护期间得到电子方式的文档是有益的。

6.1.6 项目的生存周期模型

为项目选择一个或多个合适的生存周期模型(请参阅本指导性技术文件的附录 C)。确定软件生存周期模型是系统生存周期模型的子部分,还是一个完整的生存周期模型。

附录 C 中的每个生存周期模型包括一些顺序、重复和联合执行的过程和活动。GB/T 8566 中的过程的活动宜被映射到所选择的生存周期模型中。对于增量型、演化型、构造型和预先策划好的产品改进等方法,生存周期模型某一活动的输出将反馈到下一次活动的执行,在这种情况下,相关的文档编制宜在该活动结束时(即在下一个活动开始前)完成。

6.1.7 参与方的多样性

标识项目中将会涉及到的各方,及他们将对哪些过程负责。对 GB/T 8566 中所有与各方之间的接口有关的活动和任务都宜予以考虑。

一个包含许多人员的大的项目需要有效的管理监督和控制。例如：对于大项目来说，内部的和独立的评价、评审、审核、检查以及报告生成都是很重要的手段。对于小项目来说，这些控制可能过多，宜谨慎运用。

6.1.8 软件类型

由于不同类型的软件要求有不同的剪裁决定，所以需要确定所涉及的软件类型。

软件类型的例子如下：

- a) 新开发的软件；
- b) 现货软件；
- c) 固件；
- d) 独立软件；
- e) 非交付的软件。

下面列出了关于关键软件类型的基本指导性技术文件。必须牢记的要点是：不同类型的软件是在开发过程的不同活动点进入开发过程的。

a) 新开发的软件

这类软件在一开始就进入开发过程，宜考虑开发过程的所有要求。

b) 现货软件

——原封不动地使用现货软件。这种类型的软件已经被设计、编码并经过测试。但是，根据诸如应用的关键性和使用历史等因素可能需要进行进一步的测试。此时走全部的开发过程可能过于烦琐，但至少不应迟于软件合格性测试进入开发过程。宜评价与软件有关的性能、文档、版权和今后的支持。

——不经过修改，只需进行必要的应用参数的配置即可使用的现货软件。例如：必须设置币种、日期格式和纸张大小等项参数。这类软件可根据情况在软件单元测试或集成后，进入开发过程。此时走全部的开发过程可能过于烦琐。宜评价与软件类型有关的性能、文档、版权和今后的支持。

——需修改的现货软件，如需增加更多合适的报表格式等。根据软件的关键性和所预期的改变，宜通过维护过程来引用开发过程。应在软件编码和测试时进入开发过程。宜评价与软件有关的性能、文档、版权和今后的支持。

c) 固件

嵌入或集成到系统中的软件或固件。因这种软件是大系统的一部分，故宜考虑开发过程中与系统相关的活动。在与系统相关的活动中，只有一个动词“执行”或“支持”需要选择。若软件或固件将来不大会被修改，宜仔细核查文档编制需求的范围和程度。

d) 独立软件

独立的软件。因这种软件不是最后系统的一部分，故不要求开发过程的相关活动。宜仔细检查文档需求，特别是关于维护的部分。

e) 非交付的软件

因为没有什么软件项正在被获取、供应或开发，只需要考虑 GB/T 8566 中的开发过程的 5.3.1.5。但是，如果需方出于将来运作和维护的考虑，要求获得一个这样的软件，那么这个软件宜按上述的 b) 到 d) 处理。

6.1.9 项目规模

与比如说只有三人的小项目相比，一个包括几十或几百个人员的大项目，管理问题就显得特别重要。大的项目或有分承包商的项目需要谨慎的管理监督和控制。一些项目通过联合评审、审核、验证、确认和质量保证实现这一点。对于小的项目这些控制就过多了。

6.1.10 项目关键性

越是依赖于软件的正确运作和按时完成的系统，就越需要过程可见性和过程控制。相反，对非关键软件的过分的监督和控制可能不能做到成本有效性。（请参阅 GB/T 8566 的 6.4.1 对关键性的解释。）

6.1.11 技术风险

软件开发可能会有技术风险。如果使用的软件技术不成熟,正在开发的软件又是前所未有的或十分复杂的,或者软件包含有安全性、保密性或其他重要的需求,那么可能需要严格的规格说明、设计、测试和评价。独立的验证和确认是重要的。

7 组织内的应用

7.1 考虑和技术

一个组织通常会应用 GB/T 8566 帮助其改进与软件有关的过程。该标准可以单独使用,或与现有的过程评估和能力决定方法(如 ISO/IEC TR 15504 技术报告)一起使用。

在整个组织内对 GB/T 8566 的应用,可按与应用于项目相同的方法进行。需考虑的因素已包含在第 6 章中提出的问题上,而本指导性技术文件的第 5 章中描述的策略是组织在使用 GB/T 8566 时应遵循的。

7.2 应用时机

要在整个组织内部应用 GB/T 8566 的理由可能包括以下几方面:

a) 验证现有方法的充分性。当方法是内部开发,或虽采用标准但已被大量修改时,这种验证是必要的;

b) 使现有的方法适应于进入新市场的风险。因为已意识到风险,需要有更严格的;

c) 开发新的方法以满足新组织的需求。这包括通过合并或业务联盟创立的组织。它可能有必要去维护几个过程模型以适应某些特定的活动;

d) 对引用新技术进行管理。例子包括现存人工过程的自动化或用于实现软件产品的技术的改变。

GB/T 8566 定义了可用于技术改变前后对该方法的完备性进行基准测试的准则;

e) 评价组织的内部能力以满足某一准则。如:作为投标评审过程的一部分;

f) 建立可开发改进程序的基准程序。如:针对 GB/T 8566 进行审核并使用改进过程本身。

7.3 管理者承诺

对于任何会导致工作惯例改变的计划,要显式地责成受影响的组织的管理者来实现和支持这种变动。当涉及到两个组织时,可以合同方式加以制约,然后就像一般的组织应用惯例那样,建立覆盖相关各方的策略。

8 系统生存周期模型的应用

本章具体说明了系统或项目的一般生存周期模型,并描述了怎样在这个系统生存周期模型内应用 GB/T 8566。参阅本指导性技术文件的附录 C 有关典型生存周期模型的概述。

8.1 系统生存周期模型

一个典型的系统生存周期模型从一个想法或一个需求开始,贯穿系统的开发、生产、运作和维护,在退役时结束。生存周期模型一般用时间段或里程碑来划分。每个时间段都代表将要被执行的主要的、不同的活动和任务,在从一个时间段向另一个时间段过渡时可能需要某种授权。

例如,系统生存周期的一个模型可按如下方式进行划分,每种不同方式的划分适应于不同的系统生存周期模型:

- a) 需求判定;
- b) 概念探索和定义;
- c) 论证和确认;
- d) 工程实施/开发;
- e) 生产/制造;
- f) 提交试用/销售;

- g) 运作；
- h) 维护和支持；
- i) 退役。

8.2 软件生存周期模型

一个典型的软件生存周期模型是由若干活动组成的。它从软件产品或服务的一个构想或概念开始，经过系统工程和软件工程阶段，然后进行运作、维护和支持，到退役结束。GB/T 8566 将这些活动及其相关活动条理化，为软件生存周期模型的基本过程、支持过程和组织过程等几个大的过程类型。

8.3 在系统生存周期的一般模型中 GB/T 8566 的应用实例

本指导性技术文件的图 7 显示了 GB/T 8566 在一个实例系统的生存周期模型中应用的要点，简要陈述了其基本目的，然后说明 GB/T 8566 的使用方式。表 2 给出系统生存周期的各项活动和适用的软件生存周期过程。

对 GB/T 8566 的任何活动或者整个生存周期模型而言，一个组织既可以在其内部使用，又可要求供方部分或全部按标准的要求提供产品或服务。

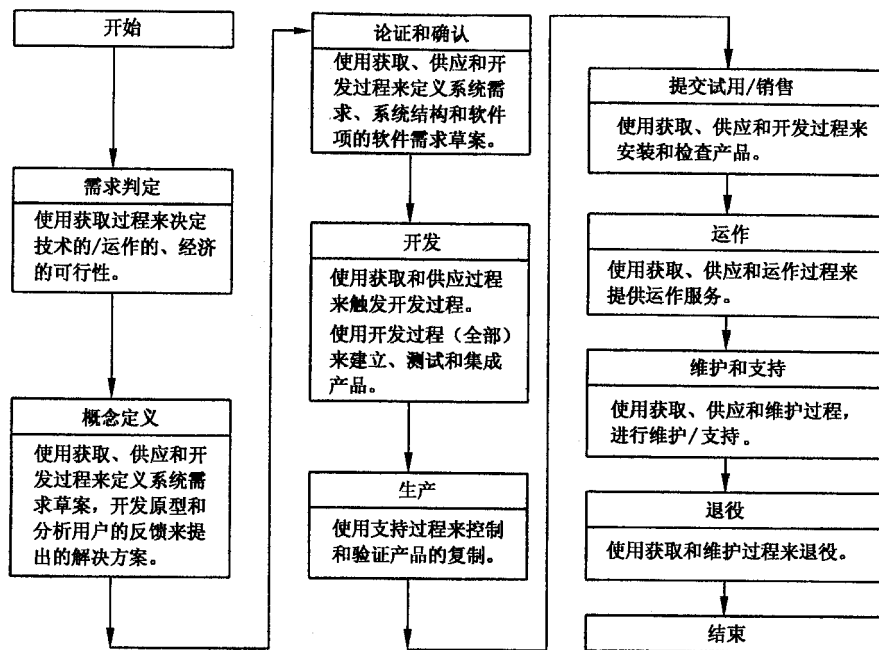


图 7 利用 GB/T 8566 来支持一个系统的生存周期模型

8.4 确定要求的活动

在这个活动中，识别和确定新的或者需改进的系统要求或构想。应当陈述最关键的一些要求，并对诸如系统成本、关键性和可行性等指标进行评审。

在进行进一步的研究、开发和承诺前，可利用获取过程来帮助做出决定，以确定要求是否在技术上和运作上都是可行的，甚至可利用开发过程来开发决策所需要的软件、方法或模型。

8.5 概念探索和定义活动

这个活动是初始的计划阶段。在这个阶段，通过综合的研究、试验性开发和概念评价对技术的、策略的和经济、市场的基础进行评估。为满足所确定的需求而提出的解决方案可以被细化，或者通过可行性评价、估算（如：成本、计划进度、市场、智力和后勤）、折中方案研究和分析来选择备选的方案。这个活动的输出通常是初步系统需求，还可能有软件的原型，它们将被输入给下一个活动。

获取、供应和开发过程可以用来：

- 帮助判定初始的系统需求；
- 开发原型；

——分析和把用户的反馈纳入到所提出的解决方案。

开发过程本身可成为一种方法,以开发用于进行相关的决策支持的分析/模拟模型软件。

8.6 论证和确认活动

在这个活动中,系统特性、有关概念,以及解决方案(包括计算机资源)将通过实施系统工程、配置主要设备、开发原型软件、测试以及评价等方式得到进一步的定义。系统特性及其相关概念、所有解决方案都得到确认,以便论证该系统(包括硬件资源和软件资源)是否适合于工程化的开发方式。然后,系统需求将被设定为系统的基线,并初步分配给系统的相关部件(如:硬件、计算机、软件和人员)。这些活动的输出将被输入给下一个活动。

获取、供应和开发过程可以用于分析和定义系统需求、最高层的系统设计方案和系统各部件(包括软件)的初步需求。开发过程可以作为进行需求分析、论证、确认、测试、制作原型以及设计解决方案的一种方法。

8.7 工程化/开发活动

该活动是系统硬件、计算机、软件、设备、人员子系统、培训和支持项被设计、制作、集成、测试和评价的阶段。输出是一个很接近目标产品的系统、下一活动所必需的文档和证明所生产的系统合格的测试结果。

GB/T 8566 完全适用于该活动。获取、供应和开发过程的过程、活动和任务宜经过适当的选择、剪裁,以实施软件的开发或升级。该活动所涉及的开发过程是由软件与其他系统部件共同协调实施的,可以是开发过程的一次执行或多次迭代执行。其输出是关于软件需求、设计和编码的基线。

如果要开发的软件将要成为系统的一部分,那么开发过程的所有活动都是必需的。此外需要搞清楚开发者是否将执行或支持与系统有关的活动。如果将要开发的软件是独立的而不是系统的一部分,那么就不需要进行与系统有关的活动,但宜加以考虑。

8.8 生产/制造活动

在此活动中,设计和开发好的系统要经历为需方(用户)进行生产或为进入市场(消费者)进行制造等阶段。生产阶段包括批准生产、生产过程、直到系统被交付和验收,目标是有效地为需方(用户)生产并交付可正常运行且享有支持服务的系统。制造阶段是从批准制造和制造过程到系统被重新设计或者退役。目标是有效地为消费者制造和交付可正常运行的、并享有支持服务的系统。

与硬件相比,软件生产/制造的工作量是极小的。它包括,为各种用户/消费者将开发好的软件和文档复制到适当的介质上,对此在 **GB/T 8566** 中没有明确的任务,可利用当地业界的惯例和政府法规。配置管理过程的发行管理和交付活动可以用来控制相关的任务。其他的活动(如:完整性验证)可以在适当的地方使用。

8.9 提交试用/销售活动

在此活动中,系统经历向需方(用户)提交试用或向客户销售等阶段。提交试用阶段从向需方(用户和维护者)交付第一个可运作的系统开始。销售阶段从向用户发放第一批系统开始,直到系统退出市场为止。

获取、供应和开发过程可以用于安装和检查已开发的或者经修改的软件。

8.10 运作活动

该活动包括操作、运行或用户和消费者对系统的使用,并在系统撤离运作时结束。

获取、供应和运作过程可以用于软件的运作和为关联的用户提供运作支持。

8.11 维护和支持活动

在该活动中,依据错误、不足、问题、用户请求或组织的改编、改进需要修改系统。该活动包括向用户(或消费者)提供后勤、技术和修复支持。

获取、供应和维护过程可以用于软件的维护和向组织、用户和消费者提供支持服务。

与开发过程的所有接口需要被决定。根据该项工作的影响,考虑到适用的情形,需要的开发过程活

动可以有所不同。

8.12 退役活动

在本阶段中,系统从正常的服务中退役。它包括将退役的系统存档和在给定的一段时间内向它的用户提供有限的支持。

获取过程和维护过程的退役活动可以用于软件退役和在特定的一段时间内向组织、用户和消费者提供支持服务。

8.13 一般系统生存周期模型中的软件生存周期过程

表2给出了系统生存周期各阶段中的软件生存周期过程布局的一个例子。只给出了GB/T 8566的基本过程。可通过基本过程使用支持或组织过程。其中“S”代表GB/T 8566中一个过程的使用;“M”代表方法的使用;“(S)”或“(M)”代表可能的使用。

表2 一般的系统生存周期模型中的软件生存周期过程

		软件生存周期过程				
		获 取	供 应	开 发	运 作	维 护
系统生存 周期过程 的阶段	需求判定	S		(M)		
	概念探索与定义	S	(S)	(S),M		
	论证和确认	S	S	S,M		
	工程化和开发	S	S	S,M		
	生产/制造					
	提交试用/销售	S	S	S		
	运作	S	S		S	
	维护和支持	S	S			S
	退役	S				S

附 录 A
(提示的附录)
质量过程和评价需求

与质量相关的支持生存周期过程,在 GB/T 8566 中的图 1 中用结构图分组显示。这些过程是:

- 质量保证;
- 验证;
- 确认;
- 联合评审;
- 审核。

在每个基本过程内,特定的任务引用这些过程来实施评价或评估,但是就任务实施者的责任而言,还有一些附加的评价任务,这些附加任务的目的是逐渐改进任务、活动和过程的质量。对某些项目而言,这将导致比生产一个质量合格产品需要双倍甚至更多的工作量。而对另外一些项目(如负有关键使命的项目)而言,将需要所有的评价过程、活动和任务。因此使用 GB/T 8566 的一个关键方面是剪裁与质量相关的过程来适合于正在从事的项目,以及把特定的质量过程映射到执行项目的各个角色。GB/T 8566 将这个重要的任务作为质量计划的准备,必要时由其他有关计划支持,如验证和确认计划。

对与质量有关的任务的剪裁可能导致将一些特殊的任务委托给其他的一些人。例如,在为公司开发一个内部使用的管理信息系统的小项目中,所做的质量计划可能允许由项目组自己执行验证和确认工作,并只执行一个简单的管理评审过程。而对于为客户开发的大的负有关键使命的系统,则项目计划可能要求安排独立的验证和确认小组,并进行联合评审和审核。这里行使驱动权力的是项目的大小和复杂度以及所生产的应用或系统的完整性级别。

本指导性技术文件的表 A1 给出了与产品、服务和过程有关的评价需求。在项目或过程的生存周期的活动中,评价者评价组织自己的或者其他组织的软件产品和服务。GB/T 8566 将这些评价归成下面列出的五种类型。前面四种评价类型是项目级的,最后一种是组织级的。这些评价必须根据项目的或组织的策略和需要的范围、数量、复杂度和关键性按比例进行选择和剪裁。评价所产生的问题、不一致性和改进报告输入给问题解决过程。

表 A1 产品、服务和过程的评价需求

评价类型	参考 GB/T 8566	目 的	实 施 者	备 注
过程内评价	5.1 到 5.5	日常活动	执行过程内指定任务的人员	—
质量保证	6.3	独立地确保软件产品及服务与合同的需求一致,并遵守所建立的计划	组织内独立于软件开发人员的人	可以使用其他活动的结果作为输入;可以将这些活动和其他评价活动协同进行
验证和确认	6.4 和 6.5	依据项目在不同的深度验证和确认产品	需方、供方、开发者、运作者、维护者、中立方或第三方	不可以重复或取代其他评价,即它们是补充的
联合评审和审核	6.6 和 6.7	基于已经同意的计划表,评价产品和活动的状态和依从情况	评价方(评审者或审核者)和被评价方(受评审者或受审核者)的联合	—
评估和改进	7.3	有效的管理和自我改进	管理者	—

附 录 B
(提示的附录)
过程输出分类

本附录标识 GB/T 8566 所要求或推荐进行文档编制的过程的输出。只列出了 GB/T 8566 中那些要求输出的章节。这些输出的文档宜根据项目或组织的范围、数量、复杂性和关键性按比例选择和确定。在表 B1 中,“输出”栏中不一定是文档的名称或标题。

表 B1 生存周期基本过程的输出

过 程	条 目	输 出	输出类型
获取过程	5.1.1.8	获取计划	计划
	5.1.1.9	验收策略和条件	描述
	5.1.2.1	建议书(或标书)	描述
	5.1.2.1	获取文档	描述
	5.1.3.1	供方选择规程	程序
	5.1.3.4	合同	合同
供应过程	5.2.2.1	建议书	建议书
	5.2.4.5	项目管理计划	计划
	5.2.6.4	评价、评审、审核、测试和问题解决报告	报告
开发过程	5.3.1.2	问题和不一致记录	记录
	5.3.1.4	开发计划	计划
	5.3.2.1	系统需求规格说明书	描述
	5.3.3.1	系统体系结构文档	描述
	5.3.3.1	系统分配文档	描述
	5.3.4.1	软件规格说明书	描述
	5.3.4.2	软件规格说明书评价	记录
	5.3.5.1	软件配置项	软件
	5.3.5.1	体系结构规格说明书	描述
	5.3.5.2	接口软件规格说明书	描述
	5.3.5.3	数据库顶层设计	描述
	5.3.5.4	用户手册	手册
	5.3.5.5	测试需求	描述
	5.3.5.6	设计评审	记录
	5.3.6.1	详细设计	描述
	5.3.6.2	详细接口软件规格说明书	描述
	5.3.6.3	详细数据库设计	描述
	5.3.6.5	软件单元测试需求	描述
	5.3.6.7	详细设计评审	记录
	5.3.7.1	软件单元和数据库	软件
	5.3.7.1	测试程序	程序
	5.3.7.2	软件单元测试结果	记录
	5.3.7.5	软件编码和测试结果评审	记录

表 B1(完)

过 程	条 目	输 出	输出类型
	5.3.8.1	软件集成计划	计划
	5.3.8.2	软件集成和测试结果	记录
	5.3.8.5	软件集成计划和文档评审	记录
	5.3.9.1	软件配置项合格性测试结果	记录
	5.3.9.3	软件集成评审	记录
	5.3.9.4	软件集成审核	记录
	5.3.10.1	系统集成和测试结果	记录
	5.3.10.2	系统合格性测试需求	描述
	5.3.10.3	系统合格性测试评审	记录
	5.3.11.1	系统合格性测试结果	记录
	5.3.11.3	系统审核结果	记录
	5.3.12.1	软件安装计划	计划
	5.3.12.2	软件安装事件和结果	记录
	5.3.13.1	软件验收评审和测试	记录
运作过程	5.4.1.1	运作计划	计划
	5.4.1.2	问题报告程序	程序
	5.4.1.2	问题报告	记录
	5.4.1.3	操作环境的测试程序	程序
维护过程	5.5.1.1	维护计划	计划
	5.5.1.1	维护程序	程序
	5.5.1.2	问题和修正请求程序	程序
	5.5.2.4	问题/修正请求记录	记录
	5.5.3.1	修正记录	记录
	5.5.3.2	修正测试结果	记录
	5.5.5.2	移植计划	计划
	5.5.6.1	退役计划	计划

表 B2 生存周期支持过程的输出

过 程	条 目	输 出	输出类型
文档编制过程	6.1.1.1	文档编制计划	计划
配置管理过程	6.2.1.1	配置管理计划	计划
	6.2.4.1	配置管理报告和记录	记录
质量保证过程	6.3.1.3	质量保证计划	计划
	6.3.1.4	质量保证记录	记录
验证过程	6.4.1.5	验证计划	计划
确认过程	6.5.1.4	确认计划	计划
联合评审过程	6.6.1.4	联合评审结果	记录
审核过程	6.7.1.5	审核结果	记录
问题解决过程	6.8.1.1	问题报告	记录

表 B3 生存周期组织过程的输出

过 程	条 目	输 出	输出类型
管理过程	7.1.2.1	管理计划	计划
	7.1.3.3	问题分析	报告
基础设施过程	7.2.1.2	基础设施计划	计划
	7.2.2.1	基础设施配置	描述
改进过程	7.3.1.1	组织过程程序	程序
	7.3.2.1	过程评估程序	程序
培训过程	7.4.1.1	培训计划	计划
	7.4.3.1	培训记录	记录

GB/T 8566 中附录 A 中的剪裁过程是提供来作为规范的参考。引用时,需要下面的输出。

表 B4 剪裁过程的输出

过 程	附 录	输 出	输出类型
剪裁过程	A4.1	剪裁决策和理由	记录

附 录 C

(提示的附录)

生存周期模型

有许多软件生存周期模型。但是,其中有三种主要的模型,它们是:

- 瀑布型;
- 增量型;
- 演化型。

这些生存周期模型中的每一种都可以原封不动地使用,或者也可以把它们结合成一种混合型的生存周期模型。通过生存周期模型的选择,GB/T 8566 的过程、活动和任务被链接起来,并且它们的优先关系被定义。

本附录描述上述三种主要的生存周期模型,以及与它们相关的风险因素(原因)和时机(益处)。在为项目决定生存周期模型时,宜考虑这些风险因素和时机。

C1 瀑布型

瀑布型生存周期模型本质上是一次通过、每个活动做一次的方法。简单来说即:

- 确定用户要求;
- 定义需求;
- 设计系统;
- 制作系统;
- 测试;
- 改正;
- 交付或使用。

在这种方法中,随着每个软件项的开发,开发过程的活动和任务通常是顺序地执行。然而,在顺序的活动相交时,它们有可能部分平行执行。

当并发地开发几个软件项时,可以对所有的软件项平行地使用开发过程的活动和任务。维护和运作过程通常在开发过程后使用。获取、供应、支持和组织过程通常和开发过程平行地使用。

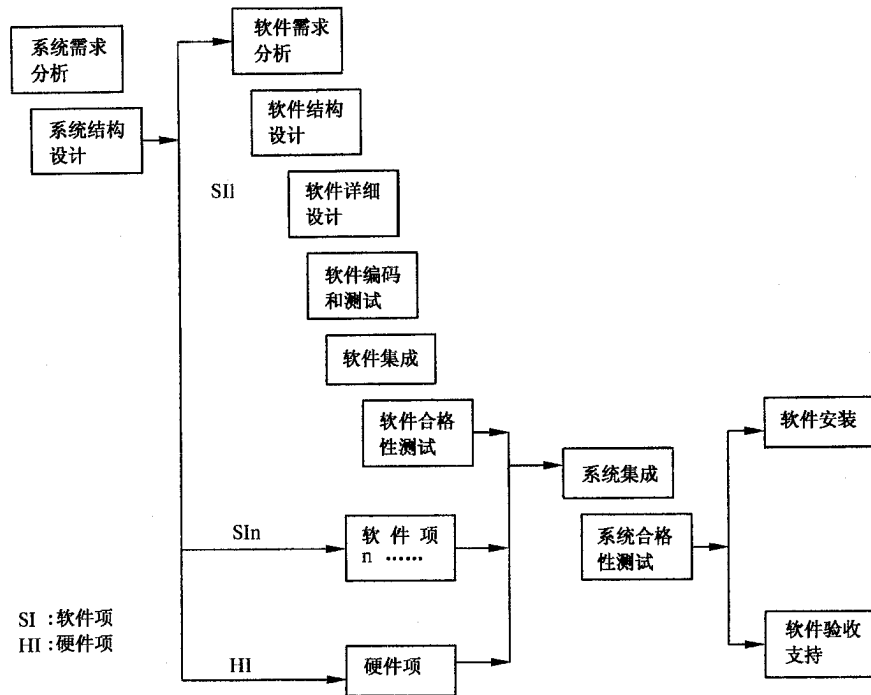


图 C1 瀑布型例子

C1.1 风险

在评价该方法时,要考虑以下相关的风险因素:

- 需求未被充分理解;
- 系统太大而不能一次做完所有事;
- 事先打算采用的技术迅速发生变化;
- 需求迅速发生变化;
- 有限的资源,如:现有的工作人员/资金;
- 无法利用某一中间产品。

C1.2 时机

可使用该方法的时机,即当希望:

- 所有的系统功能一次交付时;
- 必须同时淘汰全部老系统时。

C2 增量型

增量型生存周期模型,也叫作有计划的产品改进型,它从一组给定的需求开始,通过构造一系列的可执行工作版本来实施开发活动。第一个工作版本纳入一部分需求,下一个工作版本纳入更多的需求,以此类推,直到系统完成。每个工作版本都要执行必要的过程、活动和任务,如:需求分析和体系结构设计需要执行一次,而软件详细设计、软件编码和测试、软件集成和软件合格性测试在每个工作版本构造过程中都执行。

在这种方法中,在开发每个工作版本时,开发过程中的活动和任务顺序地或部分平行地使用。当相继的工作版本在部分并发开发时,开发过程中的活动和任务可以在各工作版本间平行地被采用。

在所有的工作版本中,开发过程的活动和任务通常按相同的顺序被重复使用。维护过程和运作过程可以与开发过程平行地使用。获取过程、供应过程、支持过程和组织过程通常与开发过程平行地进行。

C2.1 风险

在评价该方法时,有些相关的风险因素需要考虑:

- a) 需求未很好地被理解;
- b) 一次要求所有功能;
- c) 事先打算采用的技术迅速发生变化;
- d) 需求迅速发生变化;
- e) 长时期内有限的资源的保证(工作人员/资金)。

C2.2 时机

可使用该方法的时机,即当希望:

- a) 需要早期获得功能;
- b) 中间产品可以提供使用;
- c) 系统被自然地分割成增量;
- d) 工作人员/资金可以逐步增加。

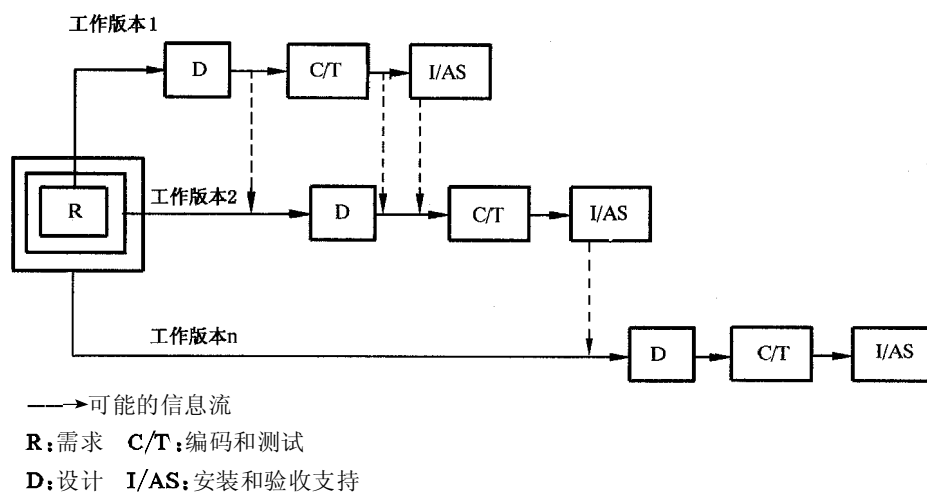


图 C2 增量型例子

C3 演化型

演化型生存周期模型也是通过构造系统的各个可执行的工作版本来开发系统的,但是,与增量型生存周期模型的区别是承认:需求不能被完全了解,且不能在初始时就确定。在该方法中,需求一部分被预先定义,然后在每个相继的工作版本中逐步完善。

该方法中,每个工作版本在开发时,开发过程中的活动和任务顺序地或部分重叠平行地被采用。

对所有的工作版本,开发过程中的活动和任务通常按同一顺序被重复使用。维护过程和运作过程可以与开发过程平行地使用。获取过程、供应过程、支持过程和组织过程通常与开发过程平行地使用。

表 C1 说明了怎样映射软件生存周期模型过程,来产生演化型生存周期的生存周期模型。这里只考虑了开发过程。做标志的条目表明一个活动或任务的实例,水平轴代表时间刻度。如果需要,该映射可以做得更详细。

C3.1 风险

在评价演化型方法时,有些相关的风险因素需要考虑:

- a) 一次要求所有功能;
- b) 长时期内有限的资源(工作人员/资金)的保证。

C3.2 时机

可使用本方法的时机,即当希望:

- a) 需要早期获得功能；
- b) 中间产品可以提供使用；
- c) 系统被自然地划分为增量；
- d) 工作人员/资金可以逐步增加；
- e) 需要通过用户反馈来理解全部需求；
- f) 便于对技术变化的监督。

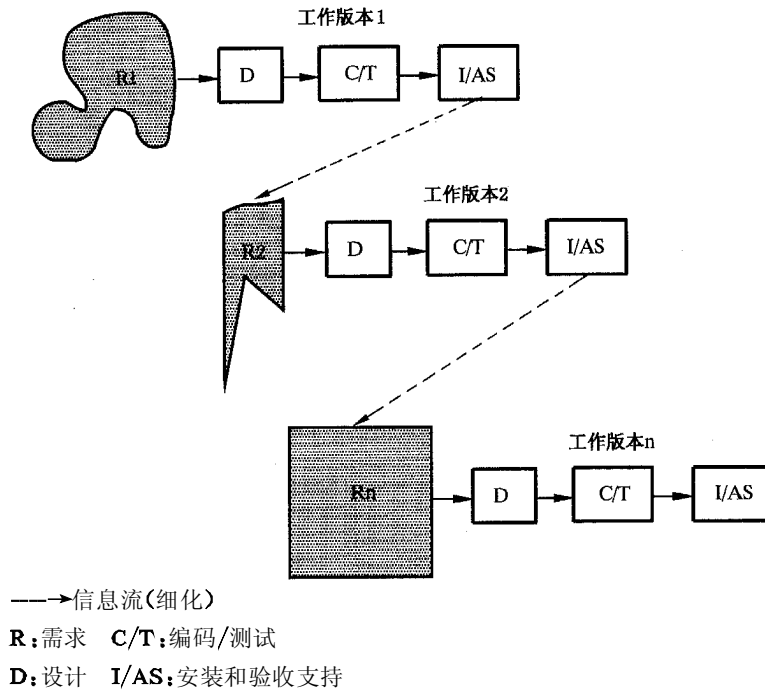


图 C3 演化型例子

表 C1 演化型开发映射例子

过程/活动/任务	时间刻度																							
过程实施	✓	✓							✓	✓							✓							
系统和软件需求分析		✓	✓	✓					✓	✓							✓							
系统和软件结构设计			✓	✓	✓					✓	✓							✓						
软件详细设计				✓	✓	✓					✓	✓							✓					
软件编码和测试					✓	✓	✓					✓	✓							✓				
软件集成和合格性测试						✓	✓	✓				✓	✓								✓			
系统集成和合格性测试							✓	✓	✓				✓	✓								✓		
软件安装和验收支持								✓	✓	✓							✓	✓					✓	
运作过程									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
维护过程									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
文档编制过程	✓	✓					✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
配置管理过程								✓			✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
质量保证过程						✓	✓	✓							✓	✓	✓		✓	✓	✓	✓	✓	✓
验证过程						✓	✓	✓	✓						✓	✓	✓		✓	✓	✓	✓	✓	✓

表 C1(完)

过程/活动/任务	时间刻度																							
确认过程						✓	✓	✓							✓	✓	✓		✓	✓	✓	✓	✓	✓
联合评审过程	✓	✓						✓	✓	✓						✓	✓				✓	✓	✓	✓
审核过程								✓								✓			✓		✓		✓	
问题解决过程				✓				✓				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
开发过程管理	✓							✓	✓		✓				✓	✓	✓	✓				✓	✓	✓
维护过程管理								✓	✓	✓					✓	✓	✓	✓				✓	✓	✓
基础设施过程	✓	✓	✓	✓				✓	✓	✓				✓	✓	✓				✓	✓	✓	✓	
培训过程	✓	✓	✓	✓					✓	✓	✓					✓								
改进过程	✓								✓							✓								✓

附 录 D

(提示的附录)

剪裁示例

D1 工作惯例的范围扩充

本示例解释了如何通过增加新的活动来剪裁 GB/T 8566,以扩展其范围,使之适应合同要求。同时包括设计成与 GB/T 8566 具有相同风格的详细文本。

D1.1 场景

需求分析的一个重要方面是决定应如何满足特定的需求,有可能仅用软件解决方案来满足需求,但改变现有的工作惯例可能会更有益。

注意:这里的“工作惯例”是指工作人员在日常的工作中执行任务的方式。一个例子是文档的处理,当系统改变时,可以不同的顺序对文档进行更加有效的处理。

在某些情况下,包括软件在内的系统的实施会对一个目标组织内的工作惯例产生重要的影响。经典的例子如:工资单/人员和信息管理系统,尤其是在从手工向电子运作转变的过程中。在所供应的软件是现成的软件包,使用时需要少量修改或不需要修改时,期望改变工作惯例也是合理的。

D1.2 剪裁决定

以下活动已经被添加到开发过程:

- 工作惯例的需求分析;
- 工作惯例的设计和文档编制;
- 工作惯例的集成。

以下活动已经被添加到运作过程:

- 工作惯例的评价。

D1.3 剪裁理由

在考虑应用到更宽的系统背景的同时,GB/T 8566 着重于软件的生存周期过程。尽管在 GB/T 8566 中,针对系统的一些活动和验证过程提供了对惯例工作惯例的支持,但如果更加突出工作惯例部分,就能获得更有效的全面解决方案。当由于需要良好的通信和组织一级的改进控制而使工作惯例显著改变时,这一点特别重要。

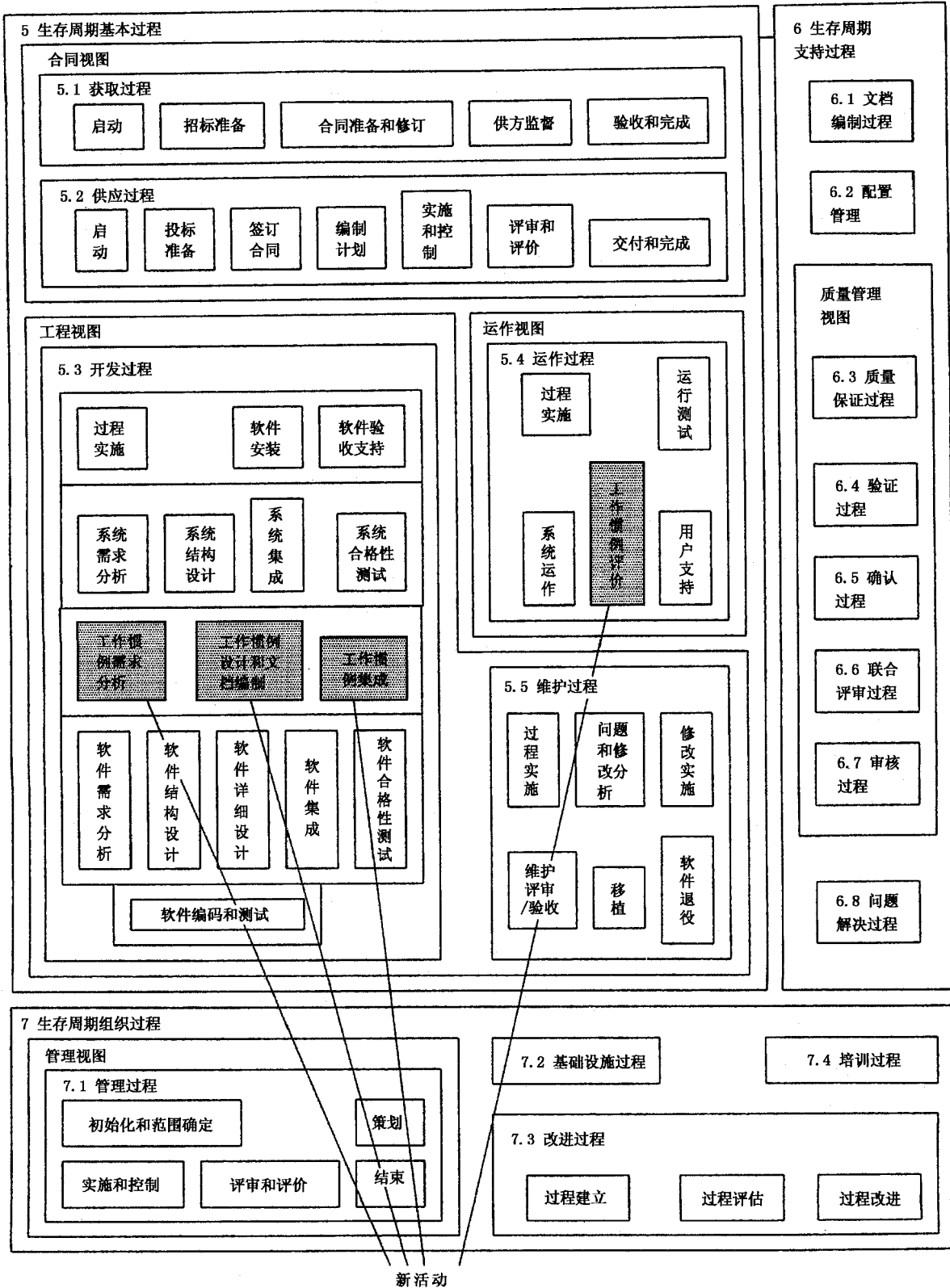


图 D1 工作惯例剪裁示例

以下 4 条是按照 GB/T 8566 的风格定义的。它们详细说明了在每个增加的活动中宜执行的任务。

D1.4 工作惯例需求分析

该活动由以下任务组成,开发者应按照合同要求执行或支持这些任务:

a) 应结合软件需求分析活动来分析现有的工作惯例,以确定实施系统的最有效的方法。它应包括对业务活动、组织结构和每个相关组织单元的权利与职责的评审。工作惯例的需求应编成文档。

b) 评价工作惯例应考虑以下列出的准则。评价的结果应编成文档:

- 系统和软件需求的可追溯性;
- 来自股东和各联合体的资源的充分性;
- 实施的可行性;
- 对规章制度要求的依从性;
- 系统的可审核性。

D1.5 工作惯例的设计和文档编制

对每个可标识的独立的工作惯例,本活动由以下任务组成:

- a) 开发者应协同利益相关小组决定与工作惯例相关联的过程流;
- b) 开发者应开发程序化文档的初步版本;
- c) 开发者应开发工作惯例的培训资料;
- d) 开发者应把系统的引入看作一个整体,开发用于与获取组织中的人员进行通讯的宣传资料。

D1.6 工作惯例的集成

对每个独立的可标识的工作惯例,该活动包括以下任务:

- a) 开发者应根据初步的程序化文档和有关的软件开发一个试验性现场培训计划;
- b) 开发者应使用程序化文档的初步版本进行培训,以决定文档和培训资料的有效性;
- c) 开发者应参考以下准则来评价程序化文档、培训资料和宣传资料。评价的结果应编成文档:
 - 系统和软件需求的可追溯性;
 - 与软件组件运作的一致性;
 - 对规章制度的依从性;
 - 操作简便性;
 - 通信有效性;
 - 系统的可审核性。
- d) 开发者应在必要时更新程序化文档和有关的资料。

D1.7 工作惯例的评价

本活动包括以下任务,运作者应按合同要求执行这些活动:

- a) 对软件产品或程序化文档的每一次发行,运作者都应确保系统是一致的;
- b) 应开展用户满意度的定期调查,并进行分析,以决定系统的适用性。这些分析应在适当时推荐文档化的活动进程。如:通过附加的培训、工作惯例评审或软件修正。

D2 小系统原型化举例

本例详细说明了如何剪裁 GB/T 8566 来适应小系统的原型化。

D2.1 场景

对小业务系统的开发,若完全覆盖 GB/T 8566,可能过于烦琐,这样会导致工作的浪费,所产生的文档意义不大,却不必要地增加了项目的成本。在这种情形下,原型化解决方案可能是达到需求目标的费效最佳的方法。

这种解决方案的关键目标是在项目的早期获得尽可能多的细节。这是通过软件开发者和熟悉相关业务过程的用户之间的直接通讯实现的。系统需求由用户说明,尤其是系统功能和外部接口,相关业务过程通过用户对原型的一系列评价加以细化。

对于小系统来说,硬件、操作系统软件和数据库包可以在市场上通过购买比较标准的产品而获得。软件开发者的使用带有第四代语言开发环境的数据库快速地设计屏幕和灵活地添加、修改和删除相

关项。用户也可有机会通过在真实环境下试运行,并凭经验来迅速验证该解决方案。

需求分析和合格性测试之间的时间间隔将大幅度减少,可以期望这种解决方案能更充分地满足需求,因为它因融入了用户的理解和投入而得到增强。

D2.2 剪裁决定

以下标准活动已经组合到利用所谓第四代语言进行软件编码的独立活动中:

- 软件体系结构设计;
- 软件详细设计;
- 软件集成。

该独立活动如图 D2 中所示的那样使用。图中,GB/T 8566 中有关的活动被醒目地显示并被连接到原型方法的一个图形描述。

时间盒方法是一种允许在一段规定的时间内制作原型,并可以进行任意次的迭代的方法。

D2.3 剪裁理由

需要原型化方法以便快速而精确地决定用户界面。一旦该界面被认可,系统的开发就可以开始考虑,如数据变换和性能特性等方面。

软件开发者通过以下几点控制原型的制作:

- 需求按优先权排序;
- 施加时间盒限制;
- 最终用户的介入。

D3 快速应用开发的例子

前面的例子介绍了原型化。本例在一个完整的系统开发生存周期模型中使用原型化,通常称作快速应用开发(RAD)。

注意:本 RAD 例子来自动态系统开发方法(DSDM),一个公共领域的 RAD 方法(www.dsdm.org)。

D3.1 场景

RAD 方法要获得成功,开发者必须和最终用户一起工作,必须在技术上以及对所使用的工具有经验,并且应用领域必须是非关键性的(商用的)。在这些前提条件下,RAD 还有以下重要的成功因素:

- 给业务需求安排优先权,正好与系统运行特性的质量相反;
- 进行基于产品的评审,该评审强调工作产生出什么,它比基于活动的评审更灵活,后者强调工作是怎样完成的;
- 使用强壮的配置管理程序,因为每个改变都可能被撤销;
- 激励团队实现业务目标,而不是简单地完成所分配的任务。
- 贯穿整个生存周期的集成测试;
- 把时间和成本估算建立在最终产品的功能上,而不是期望的开发活动;
- 风险评估关注系统功能,而不是系统的构造;
- 给高层次的需求划定基线,使之在开发过程中分解有足够的灵活性。

D3.2 剪裁决定

所采用的 RAD 生存周期模型包括如图 D3 中描述的各部分:

a) 可行性

审查项目是否满足 RAD 成功的准则;

注意:GB/T 8566 中的 5.3.1 中的开发过程——过程实施覆盖了这一点。

b) 业务研究

定义范围和开发原型计划;

注意:GB/T 8566 中的 5.3.1 中的开发过程——过程实施覆盖了这一点。

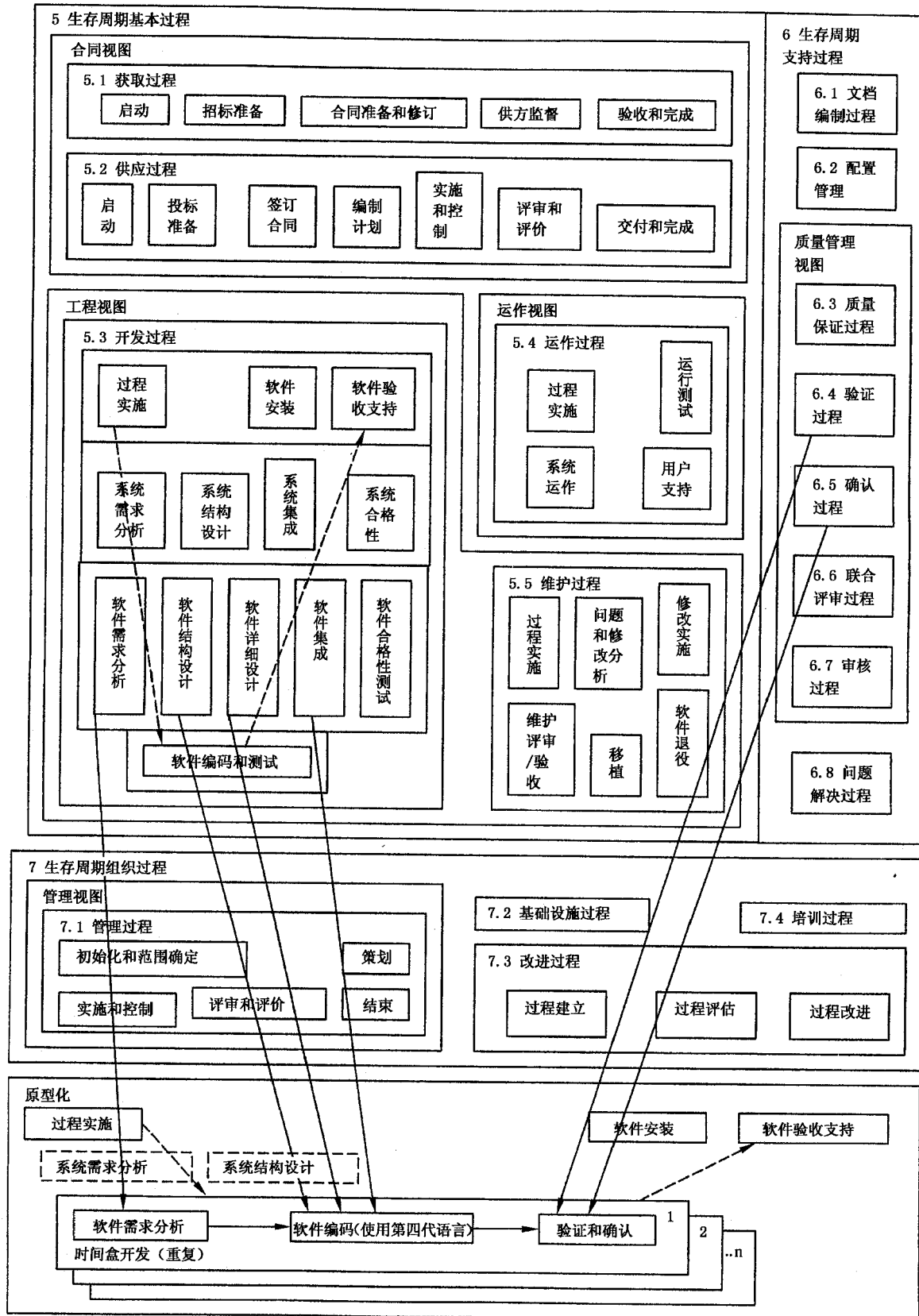


图 D2 原型化剪裁示例

c) 功能模块迭代

产生一个功能原型、非功能性需求的描述和实施策略；

注意：这可能是作为开发过程的附加的非交付项，通过 GB/T 8566 中的 5.3.1 过程实施调用。GB/T 8566 的其他方面，如验证和质量保证，可能是有用的。

d) 设计和建立迭代

生产经测试的满足所有功能性和非功能性需求的系统。

注意：这是 GB/T 8566 的 5.3.3 到 5.3.8——系统结构设计到软件集成的快速迭代。

e) 实施

此处，开发者在用户环境中实现系统，同时提供文档和培训。

注意：这是 GB/T 8566 的 5.3.12——软件安装与文档编制和培训过程的组合。在 c) 和 d) 中，开发者标识原型，确认计划表，然后创建和评审这些原型。该迭代周期的每个方面都按时间盒方式：即设置时限并按照初始调查、细化和强化三层次迭代规范。

注意：对于完整的剪裁，宜在任务级建立 RAD 方法和 GB/T 8566 之间的映照。

D3.3 剪裁理由

RAD 方法经常因为要迅速得到新系统或者在完整的系统开发期间紧急地要得到系统而被采用。

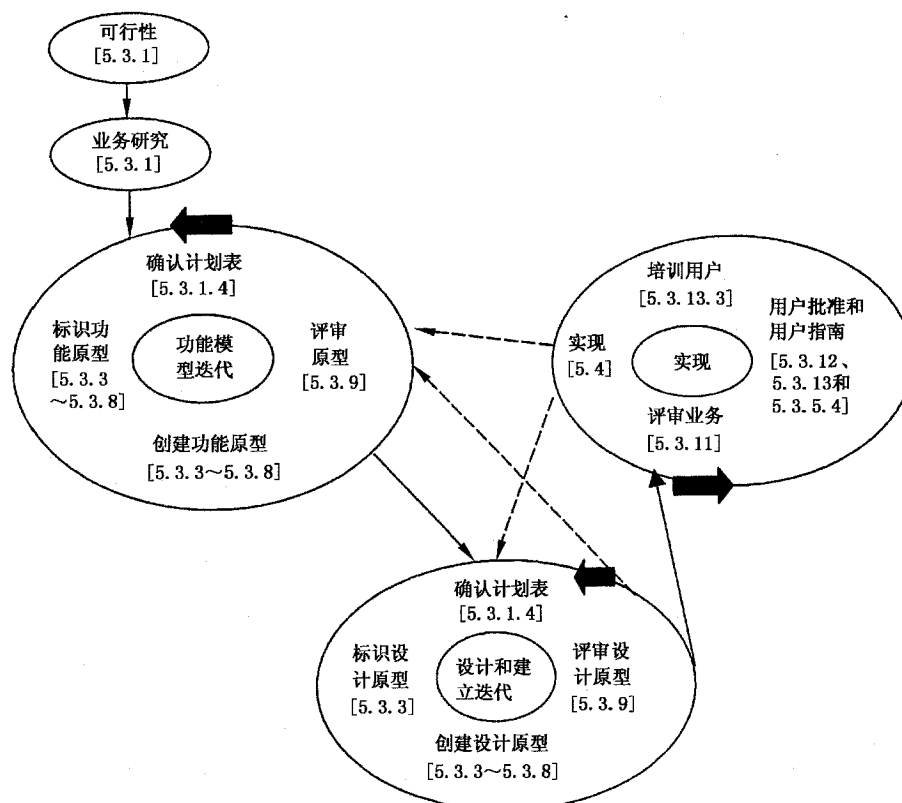


图 D3 快速应用开发示例

D4 维护举例

本例详细说明 GB/T 8566 可以怎样通过增加和删除活动进行剪裁以适应特定的维护环境。同时包括与 GB/T 8566 相同风格的详细文本。

D4.1 场景

在供应过程中对合同进行调查时，维护者可以希望向 GB/T 8566 中增加过程和任务。例如，在获取过程中，维护者可能与原先的系统供方不同，即维护者可能想要对所获取的将要维护的软件产品进行详细分析。根据这些分析，维护者可以在激活整个维护过程之前，对软件产品进行再工程。这就可以通过

向获取过程增加称作软件工程和评估的活动来适应。在获取过程中,重要的是,确保这些任务在合同中有所反映并且在验收和完成时执行。

如果决定要对软件产品进行再工程,维护者可能不想执行维护过程的所有活动。通常,在决定进行再工程时,软件产品的寿命就扩展了,有可能超出合同的持续时间。因此,剪裁掉退役活动(参阅 GB/T 8566 的 5.5.6)是适当的。

可以通过向联合评审过程(参阅 GB/T 8566 的 6.6)增加任务来实现附加的质量要求。维护者可能需要在开发过程和维护过程中由维护人员做非正式的同行评审,并且在交付软件产品之前进行。

维护者可以通过在所有过程中使用质量度量来增强改进过程。对改进过程(参阅 GB/T 8566 的 7.3)增加一个度量收集、分析和解释的任务将有助于改进。

图 D4 提供了本例中如何剪裁维护过程的图形描述。该例不宜作为维护剪裁的完整处理来理解,而仅是特定的添加和删除的例子。

D4.2 剪裁决定

命名为软件工程和评估的活动已经增加到获取过程。

软件退役活动已经从运作过程中删除。

D4.3 剪裁理由

在合同验收之前,维护组织需要对现有软件进行分析。合同规定的条件不包括处理退役的需求。

D4.4 软件工程和评估

该活动包括下列任务,维护者应该按照合同要求实施或支持这些活动:

a) 应分析所交付软件产品的代码来决定软件的可维护性。评价应特别考虑以下几点:

- 软件产品的规模;
- 注释代码行数和源代码行数;
- 软件组件的复杂度。

b) 应分析所交付软件产品的代码来决定是否需要再工程。应实施以下几点:

- 找出多余的代码;
- 标识从不执行的代码区域;

c) 基于所标识的不必要的代码的数量及其复杂度,应做出对软件产品再工程的决定。应实施以下几点:

- 删除多余的代码;
- 删除从不执行的代码;
- 重新设计和开发高复杂度的代码;
- 重新构造软件产品。

注意:本例的目的是,如果需要工程化,那么需要对开发过程的进一步的剪裁。

D4.5 软件退役

应从维护过程中删除退役活动,因为:

- 维护者将不执行维护过程的退役活动。

D4.6 同行评审

下列任务应增加到联合评审过程:

——在依照 GB/T 8566 中的 6.6,在软件开发过程的软件详细设计、软件编码和测试、软件集成和软件合格性测试活动期间,维护者应进行非正式的同行评审。

D4.7 质量度量

下列任务应在过程改进活动中增加到改进过程中(参阅 GB/T 8566 中的 7.3):

- 维护者应收集、分析和解释与维护过程有关的软件质量度量,以帮助过程改进。

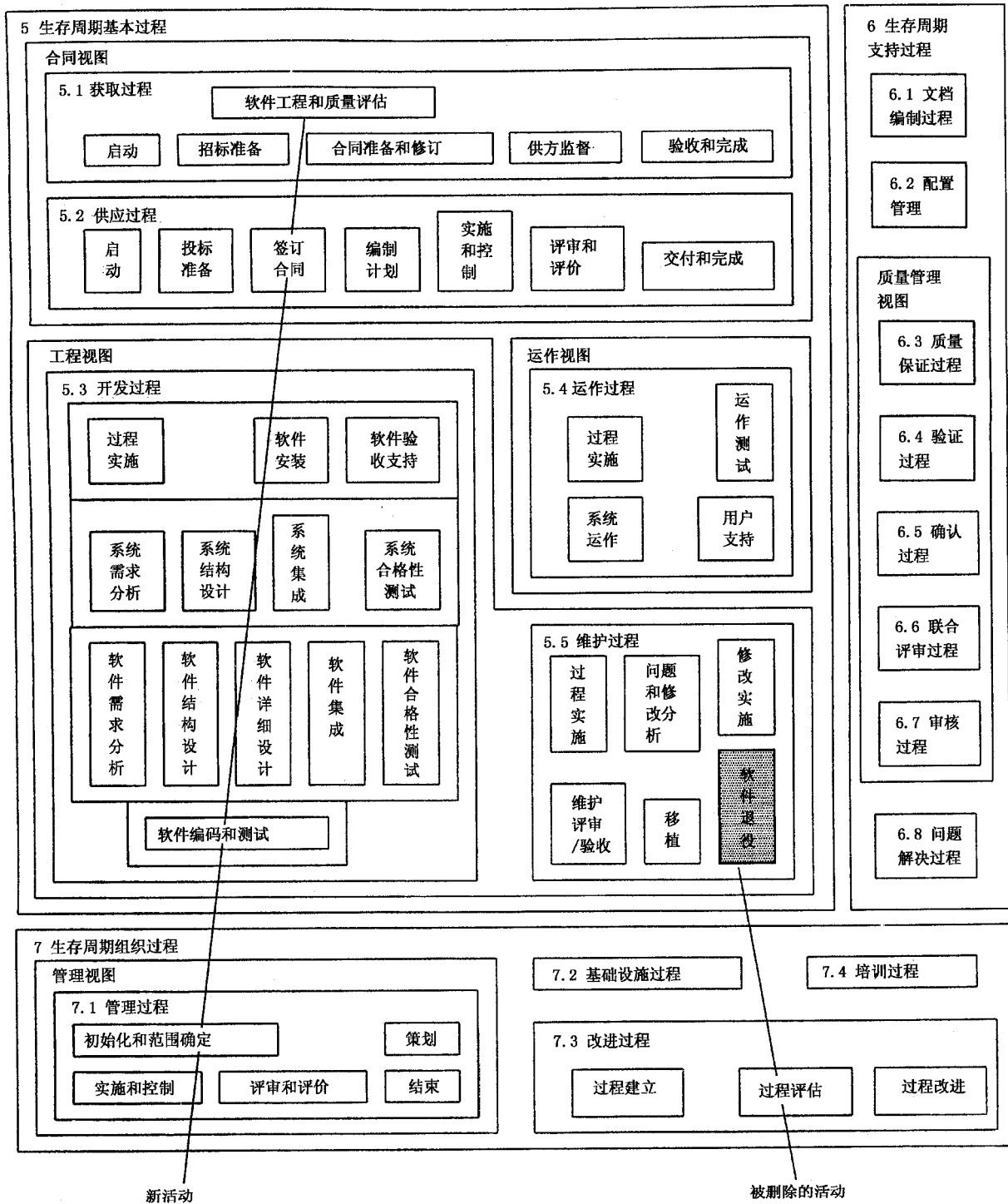


图 D4 维护示例