

软件过程规范

如果一个团队中有了统一的过程,那么,大家的行为就会符合规范,从而提高团队的整体能力。如果一个团队缺乏执行规范化过程的活动,就会导致整个组织的混乱,如出现下列情况。

- (1) 组织结构与过程活动之间不统一,这将导致成员不知是以过程为中心还是以任务为中心而造成混乱。
- (2) 成员缺少为执行有关活动所需的熟练程度和技能,从而导致过程效率低下。
- (3) 从管理的角度缺少对过程的信任。
- (4) 缺少对新成员的过程培训,从而导致新成员仍然采用过去自己所熟悉的知识与技能从事有关活动,容易与现有过程发生冲突。

为了消除软件过程所常见的问题,建立软件过程规范是必要的。软件过程规范可以确保过程活动的一致性、有效性和持续性等。

1.1 软件过程

在了解软件过程之前,首先要了解“过程”这个概念,然后再分析软件过程的特点。

1.1.1 过程

过程(process),在现代汉语词典中被解释为“事物发展所经过的程序”。实际上,不同的词典或标准,对“过程”概念就有不同的解释。

(1) 《牛津简明词典》中,“过程”被定义为活动与操作的集合。例如,一系列的生产阶段或操作。

(2) 《书氏大词典》定义“过程”是用于产生某结果的一整套操作、一系列的活动、变化以及作为最终结果的功能。

(3) IEEE-Std-610 定义“过程”是为完成一个特定的目标而进行的一系列操作步骤,如软件开发过程。

(4) 美国卡耐基梅隆大学软件过程研究所(Software Engineering Institute,SEI)的能力成熟度模型(Capacity Maturity Model,CMM)定义过程是用于软件开发及维护的一系列活动、方法及实践。

更为科学的定义,过程是指“一组将输入转化为输出的相互关联或相互作用的活动”。

从过程定义中,我们可以知道活动由输入、实施活动和输出三个环节组成,而且这些活动是围绕输出的结果(目标)而展开的,并伴随着时间先后次序的、不同的事件发生,这些活动之间存在着关系或相互影响,并且受到特定组织的控制、影响和管理,即各种相关的活动是在组织领导下开展的,如图 1-1 所示。

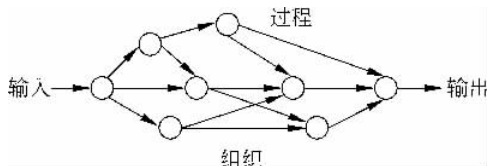


图 1-1 过程描述的示意图

过程一般可分为产品实现过程、管理过程和支持过程,如图 1-2 所示。

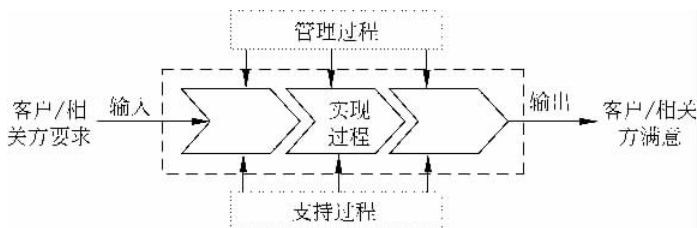


图 1-2 实现过程、管理过程和支持过程之间的关系

1.1.2 软件过程的分类和组成

软件过程(software process),是人们用来开发和维护软件及相关产品(如软件项目计划、设计文档、代码、测试用例及用户手册)的活动、方法、实践和改进的集合。根据 ISO/IEC12207 软件生命周期过程标准,软件过程被分为基本过程(可称为“实现过程”)、支持过程和组织过程,这里的组织过程包含了管理过程,如图 1-3 所示。

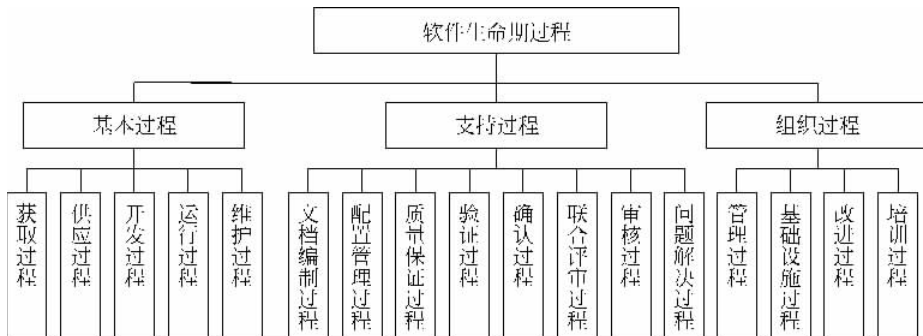


图 1-3 IEC12207 软件生命周期过程示意图

(1) 软件基本过程。软件获取过程、供应过程、开发过程、运行过程和维护过程,包括需求分析、软件设计和编码等过程。

(2) 软件支持过程。对软件主要过程提供支持的过程,包括文档编制过程、配置管理过程、质量保证过程、验证过程和确认过程(测试过程)以及评审过程等。

(3) 软件组织过程。对软件主要过程和支持过程的组织保证过程,包括管理过程、基础设施过程、改进过程和培训过程。

而在 ISO/IEC15504 软件过程评估标准中,软件过程被分为 5 个过程,工程过程、支持过程、管理过程、组织过程和客户—供应商的过程,其基础是组织过程,核心是工程过程,关键是管理过程,如图 1-4 所示。这 5 个过程可简单描述如下。

(1) 工程过程(Engineering Process,ENG)。软件系统、产品的定义、设计、实现以及维护的过程。

(2) 支持过程(Support Process,SUP)。在整个软件生命周期中可能随时被任何其他过程所采用的、起辅助作用的过程。

(3) 管理过程(Management Process,MAN)。在整个生命周期中为工程过程、支持过程和客户—供应商过程的实践活动提供指导、跟踪和监控的过程。

(4) 组织过程(Organization Process,ORG)。那些用于建立组织商业目标和定义整个组织内部培训、开发活动和资源使用等规则的过程,并有助于组织在实施项目时更好、更快地实现预定的开发任务和商业目标。

(5) 客户-供应商过程(Customer-supplier Process,CUS)。那些直接影响到客户、对开发的支持、向客户交付软件以及软件正确操作与使用的过程。

针对上述的各个基本过程,定义其具体的内容——即子过程,如图 1-5 所示。这些子过程的详细内容,会在后面的有关章节阐述。但从图 1-5 可以看出,ISO/IEC15504 对软件过程的划分,比 ISO/IEC12207 要细,除了支持过程一致以外,存在着较多的不同点。



图 1-4 软件过程的基本组成示意图

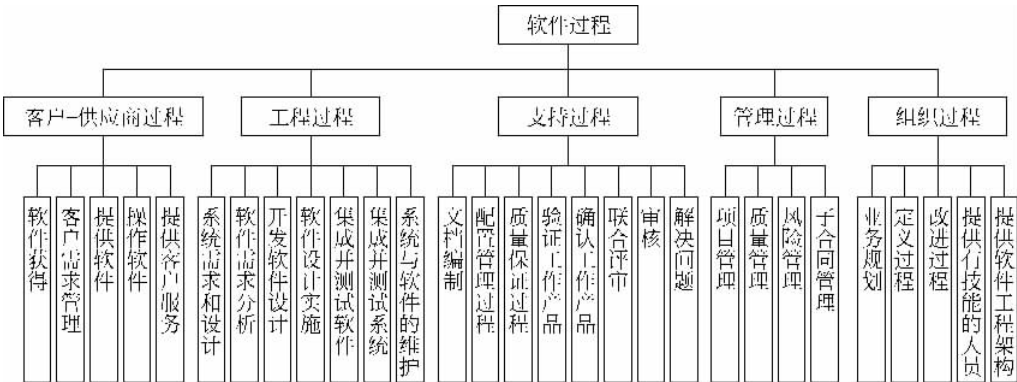


图 1-5 ISO/IEC15504 软件过程评估标准的组成

(1) IEC12207 的基本过程在 ISO/IEC15504 中被分为两个过程,客户—供应商过程和工程过程。

(2) ISO/IEC15504 的工程过程根据软件开发活动的特点分离出更多的子过程,而且区

别对待系统和软件。

(3) IEC12207 的组织过程在 ISO/IEC15504 中被分为两个过程,管理过程和组织过程。

(4) 在 ISO/IEC15504 的管理过程中定义了项目管理、质量管理、风险管理和子合同管理等,而且在其组织过程中也给出更具体的一些子过程,只有一个改进过程是 ISO/IEC12207 也拥有的。

1.1.3 软件过程定义的层次性

在制定软件组织的标准过程时,首先应当参照有关软件过程的通用模型及国际、国内标准的内容。所谓公共过程模型及标准是指 CMM/CMMI、ISO 9000、ISO/IEC 15504、ISO/IEC 12207 以及 MIL-STD-498 等之类的过程规范或标准。然后,根据软件组织自身所处的行业、组织规模和组织类型(中间产品供应商、产品供应商和软件服务提供商等),选择合适的软件过程标准并进行剪裁,以建立适合自己组织的软件过程规范。通常,这些公共模型与标准会附有相应的裁剪指南。软件组织有时可能会被要求强制遵守一些特定的标准,这时就必须要以相应的强制标准为基础并在其上制定组织的标准过程。同样,项目经理也要以组织的标准过程为基础并参照相关裁剪原则和标准来制定项目的过程。整个软件过程的层次关系如图 1-6 所示。软件过程的定义具有 3 个层次上的内涵。

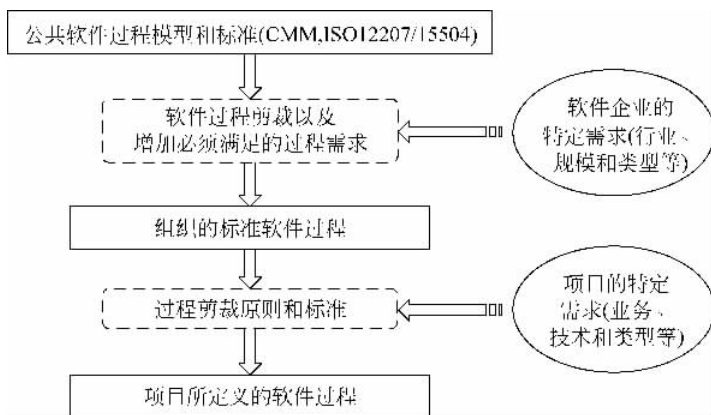


图 1-6 软件过程定义三个层次

- (1) 公共(通用)软件过程。
- (2) 组织标准软件过程。
- (3) 项目自定义的软件过程。

在后面讨论软件过程时,不同的场合涉及不同的层次。CMM/CMMI 软件过程能力成熟度是指公共(通用)的软件过程,而在项目管理中主要在项目层次上讨论软件过程,多数情况下(没有特别说明下)是在组织层次上讨论软件过程。

1.2 过程规范

软件过程管理的目的就是最大限度地提高软件产品的质量与软件开发过程的生产率。产品的质量和过程的生产率依赖于三个因素,过程、人和技术,因此要实现软件过程管理的目的,除了加强技术创新和引进、培育优秀的人才之外,规范和改进软件过程是十分重要的管理手段。所以,软件过程管理中的一个很重要的工作就是制定组织和项目的软件过程规范。

1.2.1 什么是过程规范

“规范”一词被解释为“明文规定或约定俗成的标准”,或理解为“用来控制或治理一个团队的一系列准则与章程,以及团队成员必须遵守的相关的规章制度”。因此,如果一个过程可以指定一系列的规程用以约束和规范成员的行为,则此过程称之为一个“规范”的过程。对于绝大多数软件组织,致力于建立一个有规范的过程,对团队成员进行过程规范化的培训,对所有过程的规范化内容进行文档化,并最终在软件开发和维护过程中实施规范化管理。

1. 软件过程规范的概念

过程规范就是对输入/输出和活动所构成的过程进行明文规定或约定俗成的标准。软件过程规范是软件开发组织行动的准则与指南,可以依据上述各类过程的特点而建立相应的规范,如软件基本过程规范、软件支持过程规范和软件组织过程规范。在软件过程管理中,人们会更更多地关注软件管理过程规范、开发过程规范、维护过程规范、配置管理过程规范和质量保证过程规范等。而软件开发过程规范,进一步可分为需求分析和开发的过程规范、设计过程规范、测试过程规范和发布过程规范等。

我国已经建立了一系列软件过程规范,如下所示。

- (1) 计算机软件需求说明编制指南 GB9585-88(ANSI/IEEE829)。
- (2) 计算机软件测试文件编制指南 GB9386-88(ANSI/IEEE830)。
- (3) 软件配置管理计划规范 GB/T12505-90(IEEE828)。
- (4) 软件开发规范 GB8566-88。
- (5) 计算机软件质量保证计划规范 GB12504-90(ANSI/IEEE829)。

2. 软件过程规范的建立

一般依据软件过程标准(国际标准和国内标准),参考其他软件组织已有的软件过程规范来建立软件过程规范。软件过程规范是建立在软件组织之上,充分地结合软件组织的实际情况(如规模、行业 and 开发模式等),尽力地吸收先进的软件过程模型、过程框架或过程模式所包含的软件过程思想、方法及实践,引入适用的技术和工具,为软件开发和维护建立一部详细、可操作的过程指南。

组织过程规范的建立,除了国家制定的软件过程规范之外,可以借鉴的过程模型、框架

或模式主要有如下几种。

(1) 软件能力成熟度模型(CMM)。适用于评估和改进软件组织的过程能力,提供了关键过程域、过程活动等指导。

(2) 个体软件过程(PSP)。帮助软件开发团队中的个体——软件工程师改善其个人能力和素质的组织过程,内容丰富,具有良好的实践性。

(3) 团队软件过程(TSP)。建立在个体软件过程之上,致力于开发高质量的产品,建立、管理和授权项目小组,改善开发团队过程、提高开发团队能力的指导性框架。

(4) 能力成熟度模型集成(CMMI)。是在 CMM 的基础上,试图把现有的各种能力成熟度模型(包括 ISO 15504),集成到一个框架中去,包含了健全的软件开发原则。

(5) IBM-Rational 的统一过程(RUP)。定义了一系列的过程元素,如角色、活动和产物,通过适当的组合,能够帮助软件开发组织有效地管理软件过程。

(6) 极限编程(eXtreme Programming,XP)方法。为适应快速的需求变化而积累的最佳实践,但需要适度借鉴。

(7) 微软解决方案框架(MSF)。基于一套制定好的原理、模型、准则、概念和指南而形成的一种成熟的、系统的技术项目规划、构建和部署的指导体系。

可以为软件过程的各个方面建立规范,但也不可能一气呵成,需要分阶段逐步建立起来。过程规范的建立,先从软件开发的基本过程开始着手,然后向一些关键过程(如项目管理过程、需求变更管理过程和配置管理过程等)推进,最后使过程规范能覆盖所有过程。

1.2.2 过程规范的内容和示例

软件过程规范的内容是非常丰富的,不仅描述了对各个软件子过程的准入、准出的条件和要求,而且还常常包含系统设计的原则、特定程序语言(C++、Java 等)的编码规范、开发模式的选择标准、软件过程支持和管理中要求遵守的规章制度,以及软件过程工具选择和使用的要求和评价标准。

软件规范一般会规定设计原则的简单性、可靠性和复用性,要求不存在单点失效的关键组件,反对复杂性设计,强调越简单越好。在项目管理规范中,对过程计划管理、过程风险管理、里程碑管理都会有明确的、具体的要求和标准,如过程约束、控制手段和过程结果的评估方法等。

软件规范中,一般还规定一些通用的事项,如下所示。

(1) 任务规范。对任务的定义、安排、资源分配、跟踪、完成和总结都要有清楚的说明和指导。

(2) 日常规章制度。如制定每周项目例会、每日缺陷审查会、设计评审会、需求分析会和项目组长日报、周报等相关方面的制度。

(3) 软件工具。如任务跟踪工具(MS Project、ProManager 和 Xplanner 等)、版本控制工具(CVS、ClearCase 和 Subversion 等)、持续集成工具(Ant+Junit、Maven+CC)以及 Bug 跟踪工具(BugZilla 和 Jira 等)。

对于各个子过程规范,其内容应包含“责任人、参与人员、入口准则、出口准则、输入、输出和活动”等基本内容。例如,项目实施过程规范可定义如下。

- (1) 参与人员。项目经理、开发组长、测试组长和项目组其他成员。
- (2) 入口准则。项目计划已批准,项目资源和进度已确定,任务安排完毕,项目计划基线已建立,并通过配置管理组的确认。
- (3) 出口准则。项目通过用户验收,《验收报告》经用户代表、项目经理、开发组长和测试组长等签字确认。
- (4) 输入。《市场需求文档》、《软件需求规格说明书》和《软件项目计划书》等。
- (5) 输出。通过验收测试的、可交付的程序、源代码及相关文档。
- (6) 在项目实施期间的主要活动包括如下。
 - 项目经理、开发组长和测试组长需要提交每日、每周报告,包括存在的问题、缺陷状态、任务进度和资源等。
 - 项目经理、开发组长和测试组长要定期审查项目计划的执行情况,若发现进度延误,应及时采取措施以加快进度或及时调整项目计划。
 - 项目经理、开发组长和测试组长审查各类问题并及时解决这些问题,清理软件缺陷,决定哪些缺陷要优先修正,哪些缺陷可以留到下一个版本。
- (7) 相关模板。《软件项目计划》、《项目日报》、《缺陷报告单》、《缺陷状态跟踪表》以及《项目进度周报》等模板。

1.2.3 过程规范的影响和作用

一个规范的过程可以很好地帮助团队的成员在一起进行有效的、创造性的工作。但是,过程的规范或多或少存在着某些规则,它们会限制团队中少数人的创造性发挥的空间。

1. 消极影响的存在和消除

过程规范是人们需要遵守的约定和规则,包括已定义的操作方法、流程和文档模板,不会对过程的创新、技术创新有约束,产生消极的影响呢?回答是不肯定的,这取决于过程规范是如何制定的。

如果过程规范是从国家相关规范或其他组织生搬硬套过来的,这种消极的影响肯定存在而且比较严重,会限制创造力的发挥,降低过程的生产率。所以,在制定过程规范时,要结合软件组织的实际情况,要让过程的执行者(真正执行过程的开发人员、测试人员和项目管理人员等)参与到过程规范的设计中,使过程规范满足各方面的需求,发挥良好的作用。同样,在过程的执行过程中,应该不断收集各方面的反馈意见,用以判断过程规范的实施是否真正有助于人们的工作、是否明显地降低了工作效率,从而决定是否对过程规范进行修改、或者采取相应的措施预防规范所带来的负面影响,如限制过程规范的实施范围。

在一个创造性的环境中如何把握并保证规范强制性的实施?例如,可以成立过程改进小组或召开头脑风暴(brainstorming)会议,让参与人员打破条条框框,对过程规范带来的利弊进行充分的、自由的探讨,各抒己见,从而获取有关过程规范修改、剪裁等的各种建议,寻求最有效的过程规范及其实施办法。

弗雷德·布鲁克斯(Fred Brooks)在其著名的《神秘的人月》一书的20周年纪念版里写道:“创造力来自个人,而不是组织结构或者过程”。他提醒人们,在设计软件过程时必须深

思熟虑,使过程成为一个鼓励软件人员发挥创造力的过程,而不是使之成为一个抑制并约束创造力发挥的过程。这也是一个软件管理者常常要面对的问题,即如何设计组织结构与过程,从而能使其提高而不是抑制人们的创造力和主动性的发挥。

2. 规范存在的必要性

在一个小规模的开发团队中,如果开发成员个个都经验丰富、技术过硬并且工作积极性很高,即使没有明确的、文档化的规范以及强制实施的过程,也有可能开发出高质量的软件产品。但是从长远的角度来看,这种状态是不稳定的、不可靠的。这是因为,软件的工作质量受团队成员情绪影响比较大,而工作质量必然影响到产品的质量。一个缺乏规范的环境意味着无法预测有关产品的质量,也很难判断产品质量的水平。没有规范的过程,不仅无法对质量进行计划与管理,一旦出现质量问题,也很难作出解释并找到问题发生的根本原因,也就无法保证以后同类问题不会再次发生。

过程规范的制定应致力于创建一个鼓励并能引导人们提高创造力的过程、一个提高开发效率的过程,并以更具创造性的方式来提高整个过程的实施效率和质量。越来越多的证据显示,软件开发是一个团队协作的过程。对于一个团队的活动,如果没有约定,就会使团队的工作处于混乱的状态,而制定一个规范的过程是达到预期目标的最好方式。

3. 过程规范的作用

过程规范,无论对于大型项目的开发还是小型项目的开发,都是为了保证软件开发满足质量、进度和成本等的关键要求,其表现出来的作用主要有以下几个方面。

(1) 过程规范可以帮助团队实现共同的目标。每个过程都包含了一系列的活动,各个活动都有着自己具体的目标,要让这众多的目标朝着共同的目标前进,就需要规范这个过程。组织或团队要采用的过程必须能够规范其成员的行为与活动,使全体成员在过程规则、方法和最佳实践上认识一致,从而使团队朝着共同的方向而努力,从而保证目标的实现。只有过程规范的团队,才能具有协调的、互补的行为和工作方式,团队全体成员致力于同一个目标,更有效地、快速地实现目标。

(2) 人们都一致认可这样一个事实,软件的产品质量依赖于开发并维护此软件的过程质量。换句话说,一个规范的软件过程必将能带来稳定的、高水平的过程质量,进而确保产品的高质量。在一个规范的过程中,开发团队成员会去分析产品质量,挖掘低质量产品产生或产品缺陷产生的原因,识别出过程中的根本问题并制定相应的计划、措施来加以纠正。一个持续改进的过程环境中会存在着一个良好的反馈系统。对反馈的分析有助于开发团队以及管理者追踪低质量产品产生的原因,从而能够确定应从哪些方面着手改进过程并提高产品质量。另外一方面,团队中每个成员的能力和习惯等都不一样,但为了达到统一的高水平质量,也需要过程规范来弥补团队中某些成员能力不足的一面,消除不良习惯和能力不足对质量带来的影响。

(3) 过程规范可以帮助确定目标产品的质量标准与特性,与用户达成共识,并且让项目成员接受相应的培训,使每个成员都清楚知道产品的质量标准与特性,从而容易建立一致、稳定和可靠的质量水平。

(4) 过程规范使软件组织的生产效率更高,过程规范执行的结果使团队具有统一、协

调、规范的行动与工作方式,工作效率大大提高。团队应能实时监控过程的实施情况,以便掌握项目的实际完成情况。通过监控过程的实施状况来管理团队的方法比其他传统的管理方式更为有效。

当然,也应该看到规范的过程是达到软件开发目标、提高质量的必要条件,不是充分条件。实施了过程规范,如果没有制定正确的产品策略或市场战略,其开发的产品可能得不到客户的认可。过程规范可以保证组织正确地做事,但不能保证做正确的事。要做正确的事,需要正确的愿景、目标、策略和领导。而且制定了正确的过程规范,其结果还依赖于过程规范的执行,既要严格执行适用的规范,又能对规范进行必要的剪裁和持续的改进,在灵活性和创造性之间获得良好的平衡,使软件组织从“软件过程规范”中获取最大的收益。

1.3 软件生命周期的过程需求

软件生命周期是软件获取、供应、开发、运行和维护的过程,涉及软件过程中各个参与方或利益方(stakeholder),包括软件产品的需方、供方、开发者、操作者和维护者。

软件生命周期的过程需求,首先来自于软件的开发,而对于软件开发过程已进行了很多研究,建立了相应的软件过程模型。其次,在软件开发过程中所涉及的资源、培训、环境、成本和质量等各种需求引发了软件的支持过程和管理过程。除此之外,软件最终是为客户服务的,要处理和客户的关系,就必然要求建立和维护好客户—供应商过程,而这一切都要建立在软件组织之上,即还存在软件组织的过程需求。正如 ISO/IEC15504 所定义软件过程为“工程过程、支持过程、管理过程、组织过程和客户—供应商过程”等 5 大类需求,并进一步分为 29 个子过程的需求,如图 1-5 所示。

1.3.1 软件工程过程

工程过程是软件系统、产品的定义、设计、实现以及维护的过程。虽然在 ISO 12207 标准中,没有定义一个“工程过程”类别,但可以从其“主要过程”中抽取属于工程过程的 3 个子过程,即开发过程、运行过程和维护过程。

(1) 开发过程(Research & Development, R&D)。定义并开发软件产品的活动过程,包括需求分析、软件设计和编程等。

(2) 软件运行过程(software operation process)。在规定的环境中为其用户提供运行计算机系统服务的活动过程,包括软件部署(software deployment)。

(3) 软件维护过程(software maintenance process)。提供维护软件产品服务的活动过程,也就是通过软件的修改、变更,使软件系统保持合适的运行状态,这一过程包括软件产品的移植和退役。

根据 ISO/IEC 15504,软件开发过程可以进一步被分为 6 个子过程。开发系统需求并设计、开发软件需求、开发软件设计、实现软件设计、集成并测试软件和集成并测试系统。系统和软件,在许多情况下是要统一考虑的、不能分割的,所以,在这里将“开发系统需求并设计、开发软件需求”合并为大家熟知的软件子过程“软件系统需求分析”,将“集成并测试软件、集成并测试系统”合并为一个过程“集成并测试软件系统”。所以,开发过程被定义为“软

件系统需求分析、软件系统设计、编程和集成并测试软件系统”4个子过程,这样就与业界习惯、软件过程模型等保持一致。

1. 开发过程

开发过程包括需求分析、设计、编码、集成、测试、与软件产品有关的安装和验收等活动。收集各方面的用户需求信息(过程的输入),定义用户需求中产品的功能和特性,然后通过设计将用户需求转换成软件表示,在逻辑、程序上去实现所定义的产品功能和特性,设计的结果将作为编码的框架和依据。最后通过编程将设计转换成计算机可读的形式。整个开发过程,可以进一步分为4个子过程。

(1) 软件系统需求分析(requirement analysis)。定义软件系统的功能性需求和非功能性需求,涉及系统的体系结构及其设计,确定如何把系统需求分配给系统中不同的元素,确定哪些需求应该实现、哪些需求可以推迟实现。该过程的成功实施期望带来如下结果。

- 开发出符合客户要求的系统需求,包括符合客户要求的界面。
- 提供有效的解决方案以便确定软件系统中的主要元素。
- 将定义的需求分配给系统中的每个元素,了解软件需求受系统的制约、对操作环境的影响。
- 制定合适的软件版本发布策略,以确定系统或软件需求实现的优先级。
- 确定软件需求,并根据客户需求变化进行必要的更新。

(2) 软件设计(software design)。设计出满足需求并且可以依据需求对其进行测试或验证的软件组成和接口,包括数据结构设计、应用接口设计、模块设计、算法设计和界面设计等。该过程的成功实施期望带来如下结果。

- 设计和描述主要软件组件的体系结构,这些组件可以满足已定义的软件系统需求。
- 定义各个软件组件内部和外部的接口,包括数据接口、参数接口和应用接口等。
- 通过详细设计来描述软件中可构建和可测试的软件单元或组件。
- 在软件需求与软件设计之间建立可跟踪、可控制的机制,保证它们之间的一致性。

(3) 编程(coding, programming)。通过一系列编程活动,开发出可运行的软件单元,并检验其是否与软件设计要求相一致。该过程的成功实施期望带来如下结果。

- 根据需求制定出所有软件单元或组件的验证标准。
- 实现设计中所定义的所有软件单元或组件。
- 根据设计,完成对软件单元或组件的验证。

(4) 集成与测试软件系统。集成软件单元,将软件中的组件与其运行的系统环境集成在一起,最终形成符合用户期望(系统需求)的完整系统。该过程的成功实施期望带来如下结果。

- 根据已设计的软件体系结构制定出软件单元的集成策略和集成计划。
- 根据每个单元已分配的需求确定严格的集成验收标准,包括功能性和非功能性的需求验证、操作和维护方面的需求验证。
- 根据所定义的验收标准,测试并验证软件的集成接口和软件系统,记录测试结果,完成系统集成及其验证。
- 对发生必要变更的系统,需要制定回归测试策略,以重新测试得到验证。