

基于可复用技术的敏捷开发流程研究

贾 宁 郑纯军 高志君
(大连东软信息学院 大连 116023)

摘 要 随着软件开发规模的扩大和复杂度的不断提高,软件开发效率成为各个开发团队尤为关注的课题。可复用技术是提高效率的有效途径之一,设计将可复用技术与敏捷开发相互融合,结合跨平台的移动应用程序开发模式,可帮助开发人员快速、高质量地完成批量程序的开发。完成此种开发模式需要制订统一的移动应用程序开发标准和规范,并遵循指定的开发流程,在具备基础条件和正确性的基础上进行资源复用。通过在跨平台上的实际项目开发,验证此种开发流程的可靠性和高效性。

关键词 复用技术,敏捷开发,开发规范,开发流程

中图法分类号 TP368.1 文献标识码 A

Research of Agile Software Development Process Based on Reusable Techniques

JIA Ning ZHENG Chun-jun GAO Zhi-jun
(Dalian Neusoft University of Information, Dalian 116023, China)

Abstract With the expansion of software size and complexity increase, the development team is particularly concerned about the efficiency of software development. As one of the effective ways to improve the efficiency, reusing technology is combined with agile development and cross platform APPS development mode, and a lot of high quality programs can be developed rapidly. In order to complete the program development, unified APPS standards and regulations need to be worked out. In addition, according to formulation of development process, reusing resources must be carried out in the basic condition and correctness. With the help of cross project development platform, reliability and efficiency of the development process can be proven through the verification.

Keywords Reusable techniques, Agile software development, Development norms, Development process

1 引言

随着知识经济的快速发展,高度密集的信息知识革命时代已经到来,在这场以信息网络为标示的革命中,信息技术已经成为各个领域发展成败的关键因素^[1],其中,软件技术一直位于主导核心的地位。然而,随着软件开发需求的增加,软件的复杂度在增加,其代码量也在急剧上升,成千上万行的代码对软件的开发和维护产生了一定的影响^[2],从而导致软件本身的质量逐步下降,限制了软件功能的再扩充能力^[3]。

基于此种开发怪圈,各种软件开发模式和方法学不断涌现出来,例如经典的瀑布模型^[4]、喷泉模型、螺旋模型、增量模型等,每一种开发模式和开发方法都是当时开发行为中最佳算式的描述^[5],然而,伴随开发的深入,程序员逐渐发现瀑布模型太过于理想化,喷泉模型不利于项目管理^[6],螺旋模型存在各种条件限制,增量模型要求缜密的整体模型分析^[7],随着这些开发方法的缺陷的显现,普遍适用的方法被确认不符合软件开发的国情,基于特定条件下的各种方法学的优势融合成为程序员们的首选开发方案^[8]。

本文主要针对密集的、周期性的、开发周期较短的软件系统,利用可移植程序的跨平台特性,采用可复用技术和快速的敏捷开发方法,设计一个高效、高性能、高可靠性的开发流程,并制定在特定技术条件下的敏捷开发规范^[9],从而避免了大量重复性的劳动。可复用技术并非一切从零开始,而是以现有的工作为基础,进行初步整合,在保证正确性的基础上,进行重复使用,结合敏捷开发策略,可以随时响应客户变迁的应用需求和随机产生的功能。

2 基于可复用技术的 APPS 开发规范

可复用技术被权威地通用地理解为在新的项目中使用已有的构件或对象,并重复事先进行的软件开发的过程。事实上,可复用的目标更加优越于其过程,即用系统化的方法有效地组织和管理软件产品^[10]。由此可见,复用的前提是软件产品的积累过程和组织形式,以及该产品是否达到可进行复用的标准。

移动应用程序 (Applications, APPS) 作为当下最流行的开发对象,因其具有超大规模使用群体,稳定的开发团队和顾

本文受辽宁省教育厅科学技术研究一般项目 (L2012493) 资助。

贾 宁 (1985—), 女, 硕士, 讲师, 主要研究方向为嵌入式软件、硬件开发, E-mail: jianing@neusoft.edu.cn; 郑纯军 (1976—), 男, 硕士, 副教授, 主要研究方向为嵌入式软件、硬件开发, E-mail: zhengchunjun@neusoft.edu.cn; 高志君 (1980—), 女, 硕士, 副教授, 主要研究方向为嵌入式软件、硬件开发。

客群等特点,成为众多技术公司的研究对象,APPS 的开发周期较短,产品性能在一定范围内存在交叉,完全符合可复用开发和敏捷开发的标准,因此,在跨平台的移动应用开发框架形成的基础上,应设计符合标准的 APPS 开发规范,协助跨平台的 APPS 开发能够更加高效地进行。

可复用的对象主要为 APPS 开发过程中所使用的可复用资产,包括相似的开发流程、框架、文档、工具及模块等,目前,尚未有针对此类资产的开发规范和标准,而传统的软件工程开发规范过于复杂,对于此类软件开发并不适用,因此一种适用于资源复用的开发规范是软件开发的关键之需。制定 APPS 开发规范的基本原则如下:目的性、完整性、操作性、简要性^[11]。本着上述的原则,本文主要针对以下几个方面制定开发规范。

2.1 软件界面开发规范

软件界面属于图形界面的范畴,是功能组件的渲染结果,应具备简洁、直观、灵活性强等特点。基于此,软件界面应方便与用户进行直接的通信,无需更新就可实时显现,能够方便地满足不同用户的要求。用户可以按需定制界面的样式,如首页功能、修改命令等。

此外,应该在保持界面风格一致的前提下,尽量降低界面的复杂性和嵌套深度,同时保障长时间可靠的无故障的使用,以及保障用户信息的安全性。

指导性说明也是界面必备的功能之一,跨平台的平台展示和框架应保持一致,便于再次的功能开发。

开发平台上配有跨平台界面生成器,它可以对多平台(如 Android、MeeGo、Brew、iOS)进行界面元素的开发与转换,界面设计器主要包括事件处理机制、图形渲染、布局系统、风格和主题等与用户交互相关的内容,也可定制常见的风格和主题,便于界面开发。

2.2 软件编码规范

软件开发的质量取决于软件设计的质量,因此清晰的编码结构作为编码规范的重要标准之一^[12],要求源程序易于代码的翻译与移植,可利用本系统提供的代码生成工具获得较为规范的代码样本,在开发新的模板结构时,尽量采用较好的数据结构,减少程序错误,便于代码的后期复用。

此外,清晰的排版结构、重要的注释环节、一目了然的命名都将增加软件的可读性和可维护性。移动应用开发平台也提供了一系列模板和标准文档,帮助开发人员快速地熟悉开发规范。

2.3 文档制作规范

由于资源、编码在一定程度上存在复用关系,因此,需要设计一系列的跨平台 APPS 开发的相关文档,用于约束开发人员的各项行为,降低在软件模块衔接处可能出现的风险。

可复用平台相关文档主要包括需求说明、设计模式、测试用例、使用说明、二次开发说明、复用接口说明等,依据复用资源的不同,文档将有些许的调整。其中,二次开发和复用接口说明文档的描述对象为重点复用的资源,包括组件、模块、架构、关键技术等。例如,动态变化的表格组件需要相关文档描述其功能、实现方法、如何调用,并给出相关实例和编写模板。

除此之外,对于开发团队的组织结构、人员调配、模块分工等均需要一系列的文档和制度进行规范。在 APPS 开发起步后,需要设定时间节点和生存周期等信息,并要求相关人员

严格遵循部署安排。

2.4 可复用资产管理

APPS 开发平台支持复用资产的发布、分类、储存和查询等基本需求,目前,在该平台已经配备若干个复用资产,每个资产的基本信息包括资产名称、提供者、发布时间、版本信息等。在管理过程中,仍需添加资产构成、资产规约、引用规约等成分。

可复用资产在创建时可由人工输入若干个关键词、参数信息,与相关文档一起形成资产的实体描述文件^[13],并生成相应的资产包,上传至 APPS 开发平台。表 1 是资产储存时包含的部分信息。

表 1 资产存储信息

| 资产标识信息 | 资产描述文件 | 资产所含信息 |
|--------|---------------------|-----------------------------------|
| 资产 ID | 关键词、参数信息、 实体描述信息 | 需求文档、使用说明、测试文档、 二次开发、接口文档、程序源码 |

在建立关键词索引机制的基础上,将由相关人员验证此类资产是否符合模型标准,检验条件包括:资产的完整性、文档的正确性、描述是否匹配等。符合模型标准的合格资产可以放入资产库中存储,资产验证的环节至关重要,其流程如图 1 所示。需要说明的是,进入资产库中储存并不代表资产的全部功能准确、合理、符合实际需求,只是表明此资产的关键词、相关文档等信息齐全,外部描述正确,后续仍需要大量资产提取、使用过程验证此资产的有效性。

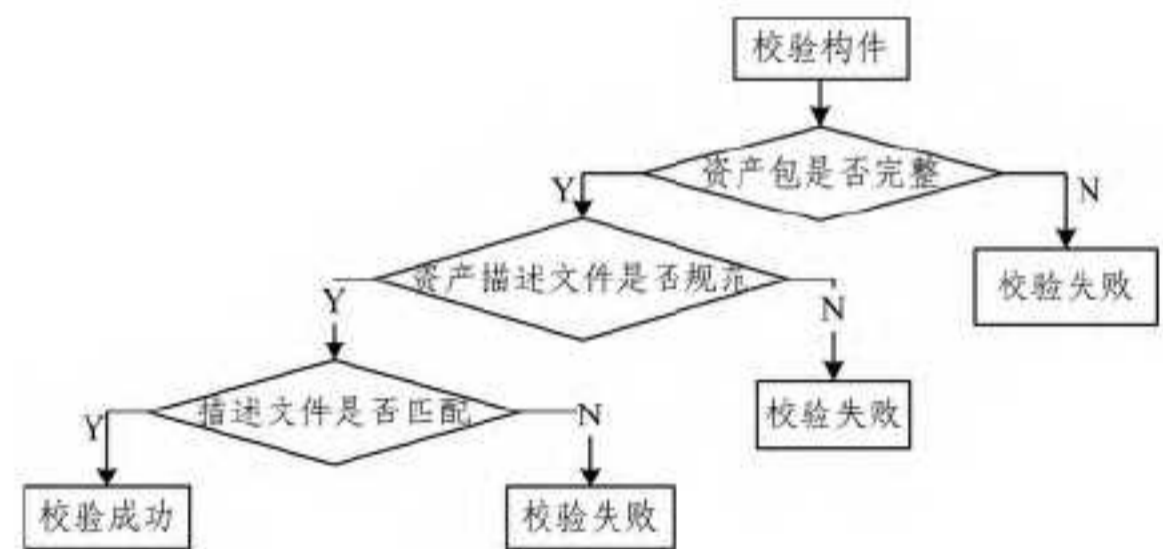


图 1 资产验证流程

3 基于可复用技术的 APPS 开发流程

上文提到的 APPS 开发规范作为软件资源复用过程中的指导性标准,在遵循开发标准的基础之上,制定可复用的 APPS 开发流程。本节将重点描述利用可复用资产进行软件开发的过程,该过程的目标是尽量复用已有的软件资产,将其应用在不同的系统之中,把从零开始开发软件作为最后的选择,这样能够在软件开发中获利。

与传统的开发过程不同,利用可复用资产进行 APPS 开发的主要活动包括:需求分析和构件的评选、构件的剪裁和扩展、构件的装配和集成、构件的测试、构件的部署和演化。

3.1 需求分析和构件的评选

需求分析作为 APPS 架构的抽象解决方案,可描述为对系统问题的定义,利用 APPS 游戏或应用的特点,可将需求划分为若干个子模块的拼凑与组合,其中子模块划分的前提是能够满足移动应用开发平台已有构件或需求构件的要求,基于此,已有构件的验证工作或需求构件的开发工作均可以完成。此外,需求分析要有一定的变化性,即可以满足不同用户的后续需求。

构件的评选工作由开发人员完成,可根据模块的需求在

开发平台上搜索已有的匹配构件。

3.2 构件的剪裁和扩展

平台上的构件往往无法做到功能上的完全匹配,因此需要对现有的构件进行一定程度的改造。首先,可以依据复用接口说明文档和实体描述文件确定现有构件的使用方法,如果涉及框架类的构件,直接套用即可。如果是功能组件,则需要确定保留的功能,并将其从源程序中剥离。剪裁时需要注意避免裁出量过大,尤为注意新老构件的接口要保持一致。

然而平台上的复用构件往往无法达到需求功能,此时需要开发人员进一步的程序开发。要特别留意二次开发文档的注意事项,避免出现系统漏洞。改进后的构件与原有构件将作为两个不同的资源储存在复用平台中。

3.3 构件的装配和集成

构件装配的前提是 APPS 框架的搭建,移动应用开发平台提供若干个程序框架供开发人员选择和修订。在程序框架中的固定位置提供可调用的构件接口,利用该接口可将已经成形的构件加入其中。

此外,APPS 的界面布局和功能需要进一步的集成,利用界面生成器获得的界面暗含各类事件响应机制,利用模板进行编写即可将界面操作与实际功能相结合。同理,界面首页与其他页面之间的转换关系也可在此实现。

3.4 构件的测试

在 APPS 基本功能搭建的基础上,进行程序的严格测试,以检验构件的可复用性。测试环节既要保证构件的质量和强壮性,又要保证满足整个项目的复用需求,每一个构件均将进行单独的测试,并且给出测评方案,其中,以下几个方面将是测试的重点环节:对于父类进行细化的部分、构件进行剪裁后的接口使用、构件扩展后的原有功能体现、构件集成过程中的界面响应、跨平台的程序功能响应等。在测试结束后,优秀作品将提供 APPS 合格性测试评价,否则将进行二次修改。

3.5 构件的部署和演化

构件的部署是对现有构件的一次跨平台检验,将构件置于多种平台、操作系统或硬件系统之中,观察构件的功能是否基本实现,是构件部署存在的重要意义。

构件的演化是在现有构件的基础上,对现有的构件进行一定的修改,以满足系统的新需求。这是继软件提交后的收官之步。在对构件进行修改的过程中,主要修改的对象是新增或扩展的功能点,并非是原有构件的搭建框架,基于此,构件演化应该遵循以下的修改法则:一致性,即保留旧的控件,新增的控件采用扩展的方式,与原有框架的接口保持一致;保留性,即旧构件中的控件,如非必要删除,可以原样保留,且不再进行修改,保留原有功能和接口;开放性,即保留原有的扩展功能,新演化的构件在具备新功能的基础上,仍可进行下一次的功能扩展和开发。

图 2 描述了基于可复用技术的 APPS 开发流程与开发标准之间的紧密联系。此开发方法实现了分析、界面、功能等多层次的集成,在分析层面上,可复用过去 APPS 的开发思路;在界面层面上,可使用界面生成器快速地获得跨平台的界面,并可实现界面的自定义;在功能层次上,利用过去的构件、模块、类可以实现快速的框架搭建。由于文档标准的制定,构件接口、构件上下文及运行平台等问题均不再成为复用的困扰。

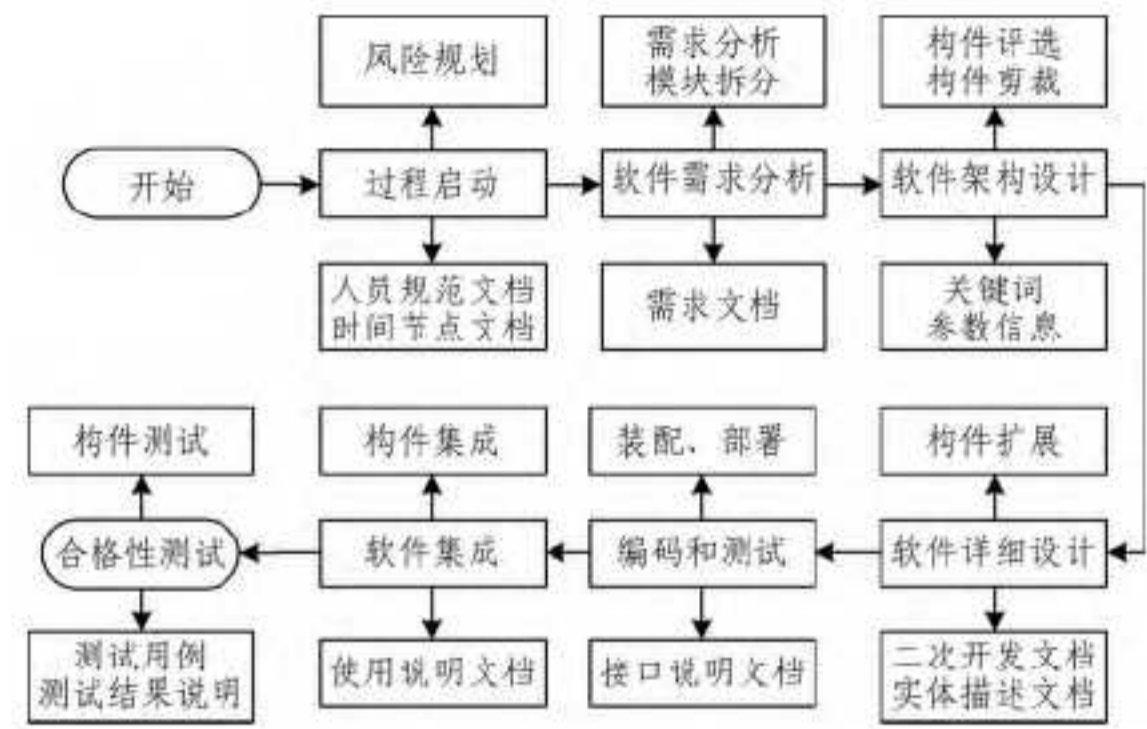


图 2 开发流程与开发标准的衔接

4 敏捷开发方法

APPS 项目主要针对新兴移动平台(iOS, Android 平台等)以及嵌入式设备平台开发的应用程序或游戏。主要特点是:(1)项目组成员一般只有几人,符合敏捷方法的应用情形。(2)项目符合迭代增量开发的要求。(3)项目开发周期短,对市场调研不完全,系统需求不明确。(4)项目属于新行业、新产品、新应用,采用的新技术比较多,为开发带来了一定的风险。因此,面向移动平台应用项目的开发与面向桌面系统的开发过程管理上有很大的区别。

敏捷开发(Agile Development)是一种面临迅速变化的需求快速开发软件的能力。它以人为本,重视交流、沟通。为获取这种敏捷性,使用一些必要的纪律和反馈的实践,遵循使软件保持灵活、可维护的设计原则、设计模式。敏捷开发方法非常适合 APPS 项目。

因此,将可复用资产与敏捷开发方法相结合,在移动平台开发应用项目将是一套全新的开发过程。基于可复用技术的敏捷开发流程合理地融合了可复用技术和敏捷开发的思想,它囊括了两者的优势,使得跨平台的软件开发更加高效、准确。此类开发流程的特点有:(1)开发初期只需客户提交基本需求即可进行软件开发,由于开发过程不断递进,客户可不断地修改需求,而开发人员可以依据需求的变化进行相应的处理。(2)构件的可复用化使得构件的使用是跨平台的,即一次开发,多次使用,这种开发方式降低了开发成本,最快解决迫切需求,提升开发效率,此外,通过大量的测试,缩短新平台和构件之间的磨合期。(3)开发人员与客户沟通能力的加强,导致产品需求的变化信息将以最快速度精确地传达,最大限度满足客户的软件需求。

此种开发方法依托于敏捷开发和可复用技术,利用敏捷开发可以缩减软件的开发周期,提高响应速度;采用可复用技术,可最大限度地利用平台上的现有构件,减少大量重复代码的编写,从而实现快速、高效的软件开发。

APPS 开发过程中,基于可复用技术的敏捷开发过程具体分为领域分析、体系结构开发、可复用构件开发、系统分析、结构设计、构件工程等阶段。在开发的各阶段充分利用可复用资产库中的资源,加快开发进度。按照敏捷开发的要求,每个阶段又需要进行多次迭代,每次迭代又可以分为需求、分析、设计、实现和测试 5 个具体工作流程。据此得到的改进后的开发过程如图 3 所示。

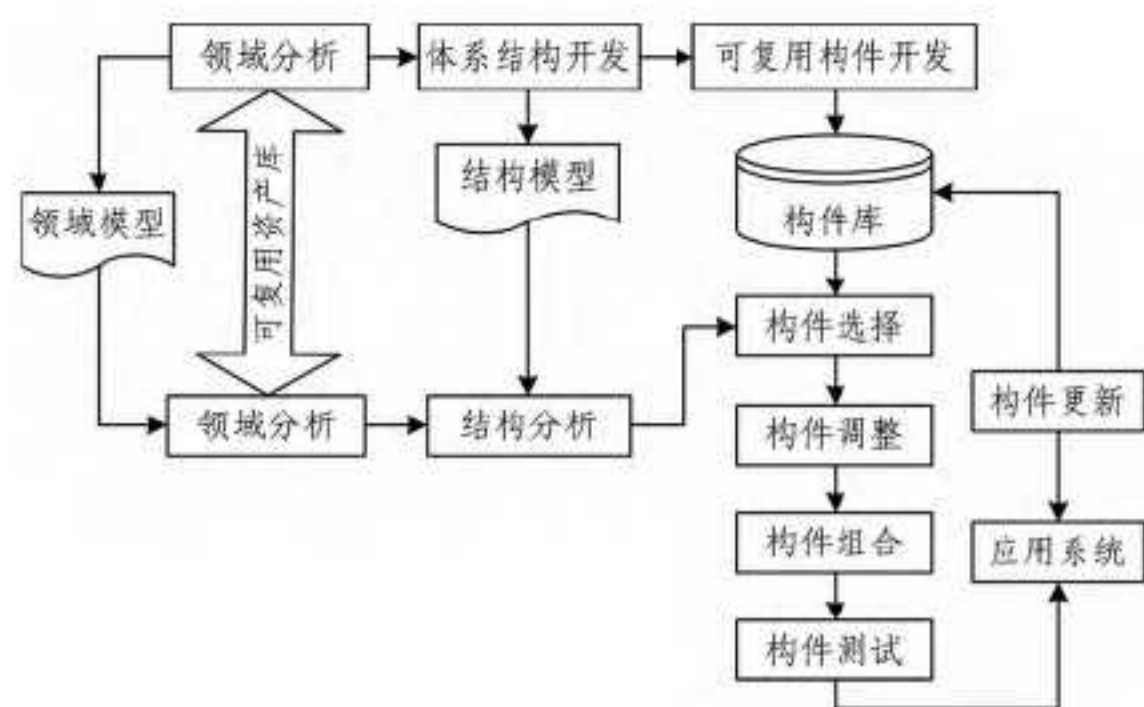


图3 敏捷开发过程

以常见的天气预报 APPS 为例,在规划 APPS 开发风险之初,拆分开发所需资源,并进行分类,发现此类 APPS 所需界面元素、整体架构、细分功能可基本复用于此平台,在确定接口的基础上,组建小型开发团队,分工进行界面设计、构件集成、功能实现等环节。此种开发形式可实现快速的 APPS 开发,在此基础上,进入程序测试环节,与此同时,开发团队将更新后的构件置于可复用平台,并完善文档,此时的构件可应用于下一次的 APPS 开发。

结束语 为提高移动应用程序开发的效率与质量,减少大量重复的构件设计过程,本文基于移动应用开发平台,设计了一种跨平台的快速 APPS 开发模式,采用可复用技术,同时结合敏捷开发模式,帮助软件开发人员快速、高效、高品质地完成程序开发,与此同时,利用具体的开发流程验证此平台界面生成器的使用效果,并验证平台中大量构件功能的正确性。

本文还介绍了基于可复用技术的 APPS 开发规范和标准,基于此,介绍了 APPS 的开发流程,可帮助开发人员快速且正确地掌握此平台的使用方法。

移动应用开发方向目前正处于不断完善和规范的阶段,大量不稳定且界面欠佳的软件被逐步淘汰,本文介绍的跨平台的快速 APPS 开发模式将帮助开发公司提高软件质量,提升市场的占有率。然而此种开发模式目前仍存在一定的局限性,只适用于可重复开发的 APPS 资源,且平台中构件的数量和功能的完备性有待提高,这些问题将在后期的项目开发中进一步完善。

参考文献

- [1] Guo J, Lu Q. A survey of software reuse repositories[C]//Proceeding of IEEE International Conference and Workshop on the Engineering of Computer Systems(ECBS2000). 2000;92-100
- [2] Almeida, Alvaro, Garcia. Designing Domain-Specific Software Architecture(DSSA): Towards a New Approach [C]//The Working IEEE/IFIP Conference Software Architecture(WIC-SA'07). 2007;30
- [3] Shirogane J, Iwata H, Fukaya K, et al. Support Method for Changing GUIs according to Roles of Widgets and Change Patterns[C]//IMECS. 2007;1098-1103
- [4] Schobbens P Y, Heymans P, Trigaux J C. Feature diagrams: A survey and a formal semantics[C]//Glinz M, Lutz R, eds. Proceedings of 14th International Requirements Engineering Conference. 2006;139-148
- [5] Benavides D, Segura S, Ruiz-Cortés A. Automated analysis of feature models 20 years later: A literature review [J]. Inform Syst, 2010, 35(6):615-636
- [6] 杨美清. 软件复用及相关技术[J]. 计算机科学, 1999, 26(5):1-4
- [7] Czarnecki K, Helsen S, Eisenecker U. Formalizing cardinality-based feature models and their specialization [J]. Software Process Improve Pract, 2005, 10(1):7
- [8] 杨燕燕, 梅宏, 陈海文. 数据仓库技术和可复用构件库系统[J]. 计算机科学, 1999, 26(5):56-60
- [9] Clarke D, Helvensteijn M, Schaefer I. Abstract delta modeling [C]//Visser E, Järvi J. eds. Proceedings of 9th International Conference on Generative Programming and Component Engineering. 2010;13-22
- [10] 彭鑫, 赵文耘, 肖君. 基于本体的构件描述和检索[J]. 南京大学学报, 自然科学版, 2005, 41(Z1):470-476
- [11] 梅宏, 刘譞哲. 互联网时代的软件技术: 现状与趋势[J]. 科学通报, 2010, 55(13):1214-1220
- [12] Frakes W B, Kang K C. Software reuse research: Status and future[J]. IEEE T Softw Eng, 2006, 31(7):529-536
- [13] Apel S, Kastner C. An overview of feature-oriented software development [J]. J Object Tech, 2009, 8(5):49-84

(上接第 536 页)

- [6] Rodrigues G, Rosenblum D, Uchitel S. Using Scenarios to Predict the Reliability of Concurrent Component-Based Software Systems[M]. Fundamental Approaches to Software Engineering, Cerioli M, Springer Berlin/Heidelberg, 2005
- [7] Mohanta S, Vinod G, Ghosh A K, et al. An approach for early prediction of software reliability [J]. SIGSOFT Softw. Eng. Notes. , 2010, 35(6):1-9
- [8] Cortellessa V, Singh H, Cukic B. Early reliability assessment of UML based software models[C]//Proceedings of the 3rd International Workshop on Software and Performance. Rome, Italy; ACM, 2002;302-309
- [9] 颜炯, 王戟, 陈火旺. 基于 UML 的软件 Markov 链使用模型构造研究[J]. 软件学报, 2005, 16(8):1386-1394
- [10] Böhr F. Model Based Statistical Testing and Concurrent Streams of use[C]//3rd Workshop on Model-based Testing in Practice6th European Conference on Modelling Foundations and Applications(ECMFA 2010). 2010;41-50
- [11] Luo Y, Ben K. Scenario-Based Early Reliability Model for Distributed Software[C]//2010 Fifth International Conference on

- Frontier of Computer Science and Technology. 2010;201-207
- [12] Cheung L, Roshandel R, Medvidovic N, et al. Early prediction of software component reliability[C]//Proceedings of the 30th International Conference on Software Engineering. Leipzig, Germany; ACM, 2008;111-120
- [13] Nayak A, Samanta D. Synthesis of test scenarios using UML activity diagrams[J]. Software and Systems Modeling, 2011, 10(1):63-89
- [14] Musa J D. Operational profile in software-reliability engineering [J]. IEEE Trans Software, 1993, 10(2):14-32
- [15] Ouabdesselam F, Parissis I. Constructing operational profiles for synchronous critical software[C]//Proceedings of 6th International Symposium on Software Reliability Engineering. Los Alamitos, USA; IEEE Computer Society, 1995;286-293
- [16] Priya S S. Test Path Generation Using UML Sequence Diagram [J]. International Journal of Advanced Research in Computer Science and Software Engineering, 2013, 3(4):123-134
- [17] 张德平, 聂长海, 徐宝文. 软件可靠性评估的重要抽样方法研究 [J]. 软件学报, 2009, 20(10):2859-2866