

学校编码: 10384

分类号 _____ 密级 _____

学号: X2007230031

UDC _____

厦 门 大 学

硕 士 学 位 论 文

统一软件生命周期
管理平台技术研究与应用

Research and Application of
the Unified Software Lifecycle Platform Technology

指导教师姓名: 陈海山教授

专业名称: 软件工程

论文提交日期: 2009年5月

论文答辩时间: 2009年6月

学位授予日期: 年 月

答辩委员会主席: _____

评 阅 人: _____

2009 年 5 月

厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下，独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合法律规范和《厦门大学研究生学术活动规范（试行）》。

另外，该学位论文为（）课题（组）的研究成果，获得（）课题（组）经费或实验室的资助，在（）实验室完成。（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明。）

声明人（签名）：

年 月 日

厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

1. 经厦门大学保密委员会审查核定的保密学位论文，
于 年 月 日解密，解密后适用上述授权。

2. 不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年 月 日

摘要

随着外资银行的进入，中国银行业的市场竞争不断加剧。如何快速交付一个高质量、让最终用户满意的金融产品系统，已成为能否占领市场先机的重要因素。为达到此目标，需要更为有效地进行银行软件开发项目的监控与管理。

随着银行软件开发团队规模的逐步扩大，系统可利用资源也日益增多。如何高效地利用系统资源，是企业当前面临的重要问题。统一软件生命周期平台作为提高系统资源整合的重要技术，是基于持续集成方法的软件生命周期管理技术研究的应用平台，它是以任务管理为中心，以需求为开发索引的统一变更管理平台。该平台以开发和测试模式为基础，实现多个生命周期不同阶段工具平台的集成，从而提高企业的生产力，有效降低企业运营成本和市场竞争力。

本文以持续集成方法为基础、以统一软件生命周期为模型进行研究，根据研究的方法开发一个统一软件生命周期平台，并付诸实施与应用。采用持续集成的方法论，分析和使用持续集成工具实现代码的快速集成，解决了传统集成模式的弊端，能够提高软件产品的整体交付质量。平台主要包括需求、开发、测试、项目监控和中间平台五大模块。需求模块将需求与开发过程紧密集成，实现需求开发及变更任务化管理、需求状态跟踪等功能；开发模块以任务为中枢，使开发人员在开发过程中，通过配置库操作实现新需求开发、需求变更开发、持续集成编译、缺陷修复等任务；测试模块实现测试需求与业务需求同步，按照测试需求、测试案例、测试缺陷的逻辑顺序关系，实现需求状态跟踪和缺陷任务化管理；项目监控模块设计与实现里程碑、风险、问题等管理流程电子化；中间平台是其他各模块之间的信息枢纽，实现任务调度和邮件转发等功能。

关键词：银行软件；统一管理；持续集成法

Abstract

Along with the foreign banks entering, the market competition of the Chinese banking became bigger and bigger continuously. How to deliver a high quality system that can satisfy the end-user fast has become an important factor to seize the earlier chance of market for us. In order to meet the need, we have to conduct the software development and management research for the project software development of the bank projects more available, by which we can raise the efficiency and quality of the software development.

For the team scale of the banks software development became larger gradually, the applied categories for the software development increased and its function got stronger, and the system resource also enriched. How to use the system resource efficiently and carry on the unified management to the system resource are very important to the enterprises, by which we can not only raise the productivity of the enterprises, but also can reduce the operation cost reasonably, and then to raise the competition of the enterprises.

The unified software lifecycle platform is an applied platform based on the software lifecycle management technology using the continuous integrated method, which takes the tasks management system as the center, and integrates the tool platforms in many lifecycles at different stages as an integral body, to implement the development and testing model for the unified change management in which it views the demand as the development indexes. We take the integrated methodology, analysis and use the continuous integrated tool to implement the fast code integrating, which also solved the drawbacks of the traditional integrated model and can raise the integral delivery quality of the software products. The unified software lifecycle platform is composed of five function modules including requirements module, development module, test module, project monitor module and Intermediate platform.

Requirements module tightly integrates requirements process with development process, and implements requirements development, task management of changing and requirement state monitoring. Development module builds task centre, so that developers can develop new requirement, and develop changed requirement, and compile program in continuous integration, and repair defect by add or modify the code through the configuration database in the development process. Test module synchronizes test requirement with business requirement, and implements requirement state monitoring and defect task management by the logical order of relations of test requirements, test cases and test defects. Project monitor module implements milestone management, risk management and defect management on electronic mode. Intermediate platform is the center module of other module, and implements task scheduling and message forwarding functions.

Keywords: Software of the Bank; the Unified Management; Continuous Integrated Method.

目 录

第1章 绪论	1
1.1 研究背景及意义	1
1.2 研究现状与存在的问题	1
1.3 主要内容及特色	2
1.4 本文的结构安排	3
第2章 统一软件生命周期管理平台总体分析	5
2.1 基本概念	5
2.1.1 软件生命周期	5
2.1.2 统一变更管理	5
2.1.3 持续集成方法	7
2.1.4 CMMI 模型简介	11
2.2 平台总体目标	12
2.3 平台需求分析	13
2.3.1 建立贯穿生命周期管理的统一平台	13
2.3.2 引入持续集成方法	15
2.3.3 为量化管理提供过程数据	17
2.3.4 项目监控电子化管理	18
2.4 设计方案分析	19
2.5 整体架构分析	21
2.6 功能分析	22
2.7 流程分析	23
第3章 统一软件生命周期管理平台设计	26
3.1 需求模块设计	26
3.1.1 需求任务管理	26

3.1.2 需求状态跟踪.....	27
3.2 开发模块设计	27
3.2.1 任务管理.....	27
3.2.2 统一变更管理.....	28
3.2.3 代码持续集成.....	28
3.3 测试模块设计	30
3.3.1 测试需求管理.....	30
3.3.2 测试缺陷管理.....	30
3.4 项目监控模块设计	31
3.4.1 问题与风险管理.....	31
3.4.2 里程碑管理.....	33
3.4.3 数据采集与展现.....	33
3.5 中间平台设计	34
3.5.1 邮件转发代理.....	34
3.5.2 任务调度.....	35
3.5.3 监控平台.....	36
第4章 统一软件生命周期管理平台实现	37
4.1 平台实现运用接口技术	37
4.1.1 ClearQuest 接口技术.....	37
4.1.2 Mercury Quality Center 接口技术.....	39
4.1.3 POP3/SMTP 协议.....	39
4.1.4 Socket 消息处理技术.....	41
4.2 需求模块实现	41
4.2.1 需求同步.....	41
4.2.2 需求和变更任务化.....	43
4.2.3 需求状态跟踪.....	44
4.3 开发模块实现	44
4.3.1 开发任务模式库实现.....	44
4.3.2 统一变更管理技术实现.....	46

4.3.3 代码持续集成实现.....	47
4.4 测试管理模块实现.....	49
4.5 项目监控管理模块实现.....	52
4.5.1 问题与风险模块实现.....	52
4.5.2 里程碑管理模块实现.....	53
4.5.3 数据采集.....	57
4.5.4 报表展现.....	58
4.6 中间平台实现.....	63
4.6.1 邮件转发代理.....	63
4.6.2 实时调度.....	63
4.6.3 定时调度.....	66
4.6.4 监控平台.....	67
第5章 总结与展望.....	69
参考文献.....	71
攻读硕士期间参与的课题和项目.....	74
致谢.....	75

CONTENTS

Chapter 1 Introduction	1
1.1 Research Background	1
1.2 Existing Problems	1
1.3 Purpose and Significance of the Research	2
1.4 Main Topics	3
Chapter 2 Analysis of the Unified Software Lifecycle Platform.....	5
2.1 Basic Concept.....	5
2.1.1 Software Lifecycle	5
2.1.2 Uniform Change Management.....	5
2.1.3 Continuous Integration Methods.....	7
2.1.4 Introduction of CMMI Model.....	11
2.2 Overall Target of the Platform	12
2.3 Analysis of the Platform Requirement	13
2.3.1 Establishment of Unified Platform Throughout the Lifecycle Management	13
2.3.2 Introduction of Continuous Integration Methods	15
2.3.3 Provide Process Data for Quantification Management.....	17
2.3.4 Electronic Management of Project Monitoring	18
2.4 Architecture Design of the Platform	19
2.5 Integral Analysis to the Architecture.....	21
2.6 Function Analysis	22
2.7 Process Analysis	23
Chapter 3 Design of the Unified Software Lifecycle Platform.....	26
3.1 Design of the Requirement Module	26
3.1.1 Requirement Task Management.....	26
3.1.2 Requirement State Monitoring.....	27

3.2 Design of the Development Module	27
3.2.1 Task Manager	27
3.2.2 Uniform Change Management	28
3.2.3 Continuous Integration in Coding Process	28
3.3 Design of the Testing Module	30
3.3.1 Testing Requirement Management	30
3.3.2 Testing Defect Management	30
3.4 Design of the Project Monitoring Module.....	31
3.4.1 Issues and Risk Management.....	31
3.4.2 Milestone Management.....	33
3.4.3 Data Collection and Display	33
3.5 Design of the Intermediate Platform	34
3.5.1 E-mail forwarding Agents.....	34
3.5.2 Task Scheduling	35
3.5.3 Monitoring platform.....	36

Chapter 4 Implementation of the Unified Software

Lifecycle Platform.....	37
4.1 The Interface Technology.....	37
4.1.1 ClearQuest Interface Technology.....	37
4.1.2 Mercury Quality Center Interface Technology	39
4.1.3 POP3/SMTP Protocol	39
4.1.4 Socket Message Processing Technology.....	41
4.2 Implementation of the Requirement Module.....	41
4.2.1 Requirement Synchronization.....	41
4.2.2 Task of Requirement and Changes.....	43
4.2.3 Requirement State Monitoring.....	44
4.3 Implementation of the Development Module.....	44
4.3.1 Implementation of Development Task Model Library	44

4.3.2 Implementation of Uniform Change Management Techniques	46
4.3.3 Implementation of Continuous Integration in Coding Process.....	47
4.4 Implementation of the Testing Module.....	49
4.5 Implementation of the Project Monitoring Module	52
4.5.1 Implementation of Issues and Risk Management	52
4.5.2 Implementation of Milestone Management	53
4.5.3 Implementation of Data Collection.....	57
4.5.4 Display of Report	58
4.6 Implementation of the Intermediate Platform.....	63
4.6.1 E-mail Forwarding Agents	63
4.6.2 Real-Time Scheduling.....	63
4.6.3 On-Time Scheduling	66
4.6.4 Monitoring Platform	67
Chapter 5 Conclusion and Expectation.....	69
References	71
Acknowledgements.....	75

第 1 章 绪论

1.1 研究背景及意义

随着软件开发团队规模逐渐扩大，软件开发的应用种类增多，高效地使用系统的资源，并能够统一地对系统资源进行管理对企业来说非常重要，统一软件生命周期平台作为提高系统资源整合的重要技术，是基于持续集成方法的软件生命周期管理技术研究的应用平台。

实施统一的软件生命周期管理的意义：其重点在于集成化，对系统资源实现集中管理，使得所有的项目团队都工作在同一个平台上，提高工作效率，增强团队内部的沟通。同时将软件资产纳入自动化工具的管理之下，提高软件项目的量化管理水平，帮助管理人员及时了解项目进展的情况，强化软件生命周期管理，提高软件质量，同时建立完整的需求到设计到源码到测试的跟踪关系。

引入持续集成方法的意义：传统的开发模式是项目一开始就划分模块，然后等所有的代码都开发完成之后再集成到一起进行测试。随着软件技术的发展，各种软件生命周期模型百花齐放，软件规模也在扩大，软件需求越来越复杂，软件已经不能简单地通过划分模块的方式来开发，需要项目内部互相合作，划分模块这种传统的模式的弊端也越来越明显。由于很多缺陷在项目的早期就存在，到最后集成的时候才发现问题，开发者需要在集成阶段花费大量的时间来寻找缺陷的根源，加上软件的复杂性，问题的根源很难定位，甚至出现不得不调整底层架构的情况。持续集成主张项目开发人员要频繁及时的将他们对源码的修改提交到一个单一的源码库，并验证这些改变是否对项目带来了破坏。持续集成的关键是自动化，即读取源代码、编译、连接、测试，整个创建过程都自动完成，其最终目的是得到一个可以使用的稳定的发布版本。持续集成方法的引入是鼓励并支持开发人员尽快的提交对源码的修改并得到尽快的反馈。

1.2 研究现状与存在的问题

目前，软件开发过程中主要在需求、代码、测试、计划管理和工具等方面存在一些问题：

- 需求

需求没有明确的计划与跟踪，对需求开发、测试和实现状态难以准确掌握。在日前的软件开发工作中，项目组根据已确认的需求进行分析、设计和实现。但由于对需求没有明确的规划和用户的不断变更，需求的实现状态将变得难以掌握，这将会给交付的产品质量带来危险。需求与代码之间的关系知情权往往只有开发作者可以说的清楚，这同时也会给后期的变更带来不可预知的风险。

- 代码

传统的集成模式导致编译、单元测试、代码扫描和集成测试实施不及时，常常开始集成时开发人员已经忘记当时的开发思路，对出现的问题需要很长时间才能进行定位，甚至会出现人员岗位调离的现象。同时，传统的代码集成模式对于代码的实现和修订情况也很难有效统计和进一步分析。

- 测试

开发与测试往往过于独立，导致交互困难，测试发现的问题需要二次手工处理或整理才能传至开发团队开始修复。有的项目使用了测试管理工具或电子表格对缺陷进行管理，但这种管理方式无法根据缺陷的关联关系进行源码的修复，导致缺陷的修复内容很难得到真实和完整的记录，给后续的代码集成工作带来很大的困扰。

- 计划管理

计划任务不能得到有效地追踪。计划总是在项目的开始被定义，但在实际的工作中却很难根据计划对项目工作进行监控，从而导致项目逐步变为无计划的工作，导致项目进度的延迟。

- 工具

在整个生命周期过程中软件开发团队面对着不同生命期阶段的辅助工具，如：配置工具 ClearCase，变更工具 ClearQuest，需求工具 CaliberRM，测试工具 Mercury Quality Center 等。这些工具相对独立，数据内容无法达到信息同步和共享，导致无法有效了解需求—设计—开发—测试之间的一致性。

1.3 主要内容及特色

本文主要从需求、开发、测试和实施环节入手，使用持续集成的方法论加快

产品集成，减少开发的存货；有效整合需求分析、开发、测试等环节的应用流程及管理工具，实现环节之间的无缝协作，分别从分析、设计和实现三方面对该平台技术研究和应用进行介绍。本文的主要内容如下：

1. 统一软件生命周期管理平台总体分析。首先对软件生命周期、统一变更管理、持续集成方法和CMMI进行概念引入介绍。通过现状及存在的问题，确定总体实现的目标和总体需求。在需求分析的基础上确定了统一软件生命周期管理平台的设计方案和总体框架。按照软件生命周期的各个阶段定义，将系统分为了需求、开发、测试、监控等功能模块。

2. 主要功能模块的设计和实现。需求模块主要针对需求如何有效及时地转化为开发任务、需求实现状态的跟踪等方面进行设计和实现。开发模块主要从各类开发任务管理、任务与配置项有效关联、代码自动持续集成等方面进行设计和实现。测试模块主要针对需求到测试需求的转换、测试缺陷在测试人员和开发人员之间的自动信息同步处理等方面进行设计和实现。项目监控模块则设计和实现了问题管理、风险管理、里程碑管理、数据集中展现等方面的电子流程化管理功能。中间平台模块主要设计和实现了工具之间的信息交互、邮件通知和任务调度等功能，也为以后系统的扩展打好了良好的基础。

统一软件生命周期管理平台的建设目标是通过建设一个整合平台框架，在原有的工具应用基础上，通过工具的接口技术和监控管理的电子流程化，将各个工具通过集成的方式进行互连互通，达到信息资源的及时同步与共享，使项目实施过程以需求为起点进行统一管理和监控，同时，通过引入持续集成方法，实现代码的及早集成，加快项目的整体交付能力。

1.4 本文的结构安排

本文共分为五章。

第1章，对当前银行软件存在的问题进行分析，阐明课题的研究意义与主要研究内容。

第2章，简要介绍统一软件生命周期管理平台的基本概念和总体目标，并对需求分析、设计方案和整体架构等问题进行探讨。

第3章，对统一软件生命周期管理平台各模块的具体设计进行描述。

第 4 章，对统一软件生命周期管理平台各模块的具体实现进行说明。

第 5 章，对本论文的一个总结和展望，对项目的主要工作，论文的主要内容进行总结，同时提出尚未完成的工作，并对统一软件生命周期管理平台的进一步研究进行展望。

厦门大学博硕士论文摘要库

Degree papers are in the "[Xiamen University Electronic Theses and Dissertations Database](#)". Full texts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to etd@xmu.edu.cn for delivery details.

厦门大学博硕士论文摘要库