# 论文题目

软件持续集成平台的设计与实现

## 研究内容

一、 选题背景

现在的软件开发模型的不断发展，不论传统的瀑布式开发模型，还是敏捷迭代开发方式，软件过程都需要经过不同的阶段，包括需求分析、设计、编码、测试、部署、运维等活动，都面临着如何缩短项目的周期时间，提高项目的代码质量的挑战。

软件开发过程中的不同阶段，都对项目的整个生命周期产生了影响，瀑布式开发模型由于自身阶段之间的反馈沟通僵硬不畅，对于整体的结果的产出无法及时掌控，只能通过完成日期和里程碑来跟踪每个阶段的成果，无法真正对软件的质量和产出进行评测，失去及时纠正的机会，而且整个流程单一，不可逆，极度依赖于早起的需求调查。

敏捷开发模式给软件开发带来了新的内容，敏捷开发强调改进，核心是迭代开发，强调最终目标是让客户满意，主动接受需求的变更，采用迭代、循序进行的方法进行软件开发，在项目初期就切分成多个子项目，各个子项目的成果都经过测试，具备可视、可集成和可运行的特性，在开发过程中，软件一直处于可使用的状态。从源头上解决了瀑布式开发模型中对于需求的依赖，保证了项目质量的可控性。

敏捷开发的核心是迭代，迭代的成果需要持续集成来支持，持续集成本身就是一种软件开发实践，要求团队成员经常集成自己的工作，并且利用自动化构建的方式来验证集成，这些构建能够自动运行代码更新、静态代码扫描、单元测试、自动化回归测试，尽快检测出来软件集成的错误，更快地实现有凝聚力的软件开发。

基于持续集成在当前公司软件开发流程中具有的重要作用，运用持续集成平台完成软件开发过程中的集成工作已经是流程中必不可少的环节，帮助开发人员定期持续地从代码库更新代码，自动进行编译、部署以及测试，形成安全稳定运行的代码库，为项目组的软件系统质量带来保护，并且能够将所有的质量工作都集中到一个系统中进行记录管理，对软件系统的质量有全面的评测

二、 研究内容

持续集成现在作为一种很好的软件工程实践被很多项目团队使用，尤其在需求变化频繁的项目团队中，持续集成的实践有比较好的反馈。在实际使用中，一般会按照自身项目特点和实际需求实现持续集成的自动化。持续集成方案一般下面几个方向，研究内容也主要覆盖这几个方面：

1、 代码版本管理

版本控制(Revision control)是一种软件工程技巧，一般使用在代码开发过程中，保证所有开发人员在开发过程中，都能同步获取其他人所开发提交的代码。目的就是让所有参与开发的人的开发过程都有据可查，从而实现多人合作开发。

版本管理分两步工作：

第一， 分支策略设定，一个分支等于一套代码，分支越多，维护成本就越高。所以分支的数量不能太多，存在的周期也不宜太长。拉出了分支以后要尽快地收编合入主干版本，减少维护的工作量

第二， 管理策略，一般我们应用的版本策略有：不稳定主干策略、稳定主干策略、敏捷发布策略。使用不同的版本策略，再结合研发团队的具体情况，达到代码稳定，发布版本稳定迅速，减少工作量等目的。

版本管理工具Git

Git是目前世界上最先进的分布式版本控制系统之一，与SVN的集中式版本管理系统不同，Git的版本库分布每个人的本地电脑上，不仅有一个中心的服务器控制最新版本代码，而且每个开发者自己还有个本地仓库，所以在开发过程中都是先将代码提交到本地仓库再推送到中心服务器上的，这样的好处就是每个人都依赖于中心服务器来实现交互，但又不会被中心服务器限制，就算中心服务器挂了，也能很容易的找到最新版本的代码，而且我自己的工作依然可以顺利进行，提交到本地仓库，当中心服务器修复之后，再将自己仓库的东西推送到中心服务器。

2、 构建工具

Jenkins, 是一个开源的实现持续集成的软件工具，由于功能和体验方面的特点，应用广泛。

Jenkins 是一个持续集成工具，通常用于软件开发工作。这种自动化的框架可运行重复的作业。Jenkins 可以管理和监视远程系统上的命令启动，还能执行可通过命令行运行的任何操作。Jenkins 将电子邮件、TestNG 和其他工具与支持插件集成起来。

Jenkins本身使用了master/slave架构，支持主从模式，Master提供web接口让用户来管理job和slave，job可以运行在master本机或者被分配到slave上运行。一个master可以关联多个slave用来为不同的job或相同的job的不同配置来服务，Jenkins 内置了 SSH 客户端实现，用于同远程 sshd 和从代理进行通信。还有多种在主从机器之间进行通信的方式。

Jenkins还提供了插件扩展功能，借助jenkins提供的功能点，可以进行定制化的插件开发，jenkins已经支持超过1000个插件，这个插件的功能包括了配置管理、自动测试、通知、报告等方面，jenkins通过单独的类加载器来加载插件，避免了插件之间的冲突，插件中的所有功能都可以无缝集成到jenkins上。

Jenkins 设计之时就已经支持了让我们通过 REST API 的方式拿到所有的对象的接口，对象的接口可以通过固定的URL进行访问查询，通过提供的接口，使用者可以根据需要来定制jenkins job的创建、设置和触发等功能，实现自身的需求。

3、 静态代码扫描

静态代码扫描从字面上理解就是检查项目的源码，直接从代码内容中，找出存在的缺陷，包括bug、冗余代码、不合理代码。

SonarQub它是一个开源的代码质量管理系统，支持 25+ 种语言，可以通过使用插件机制与 eclipse 和 JIRA 等其他外部工具集成，从而实现了对代码的质量的全面自动化分析和管理。SonarQube 并不是简单地将各种质量检测工具的结果（例如 FindBugs，PMD 等）直接展现给客户，而是通过不同的插件算法来对这些结果进行再加工，最终以量化的方式来衡量代码质量，从而方便地对不同规模和种类的工程进行相应的代码质量管理。支持 25+ 种编程语言，针对不同的编程语言其所提供的分析方式也有所不同，对于所有支持的编程语言，SonarQube 都提供源了代码的静态分析功能；对于某些语言提供了对编译后代码的静态分析功能，比如 java 中的 class file 和 jar 和 C# 中的 dll file 等，对于某些语言还可以提供对于代码的动态分析功能，比如 java 和 C# 中的单元测试的执行等。

Sonarqube对某一个项目进行代码质量管理时，通常可以采用三种不同的方法来发起代码分析，分别为 Analyzing with SonarQube Runner、，Analyzing with SonarQube Ant Task 和 Analyzing with Maven 方法。这三种方法的主要区别是 , 它们分别适用于不同架构和组织形式的项目，并且其相应的配置文件的写法也不尽相同。

FindBugs 是一个静态分析工具，它检查类或者 JAR 文件，将字节码与一组缺陷模式进行对比以发现可能的问题。Findbugs自带检测器，其中有60余种Bad practice，80余种Correctness，1种 Internationalization，12种Malicious code vulnerability，27种Multithreaded correctness，23种Performance，43种Dodgy。我们还可以自己配置检查规则(做哪些检查,不做哪些检查)，也可以自己来实现独有的校验规则(用户自定义特定的bug模式需要继承它的接口,编写自己的校验类,属于高级技巧)。

Sonarqube的findbug插件可以方便集成代码分析的功能，并且支持定制校验规则，让代码扫描更加精准。

4、 自动化测试

自动化测试是测试的一种形式的统称，是以人为驱动的测试行为转化为机器执行的一种过程。 从发展历程上经过了录制回访、脚本、数据驱动、关键字等不同阶段。

TestNG是一个测试框架，其灵感来自JUnit和NUnit，但引入了一些新的功能，使其功能更强大，使用更方便。它使用注解，支持综合类测试，拥有灵活的运行配置，并支持依赖的测试方法，并行测试，负载测试，局部故障重试，包括灵活的插件api和多线程测试，为自动化测试提供了支撑框架。

Selenium是一个用于Web应用程序测试的工具，直接运行在浏览器中,就像真正的用户在操作一样，selenium提供原生api来实现对浏览器页面的操作，在实际使用中，一般将原生api进行二次封装，形成操作关键字来简化编写步骤，并且增强代码的复用性，结合测试框架，给使用人员在web UI方面的自动化测试工作带来了效率。

5、 自动部署

在实际工作中，根据应用的不同类型来定制部署实现。按照应用的类型分为Web，App，在web应用中，利用Tomcat自动部署的机制，编写shell脚本实现web应用的定向复制，服务器启动，备份内容等步骤。

6、 构建方式

构建支持三种方式：命令行手工构建、脚本实现、自动或定时构建。

命令行，直接在工程中使用命令触发构建；

脚本实现，直接编写bat或shell脚本触发构建；

自动或定时，使用工具来定时或自动触发构建

7、 持续集成平台实现

分析持续集成技术，将研发过程各环节编译、连接、测试、部署和报告集成，结合扩展工具，实现研发过程持续集成自动化。将自动化的过程通过Web系统的方式实现，结合用户管理、权限管理、流程管理的实现，完成持续集成平台系统在实际工作中的落地，将持续集成自动化体系发挥到最大的价值。

三、 关键技术

J2EE技术、测试框架、Jenkins、sonarQube、selenium、findbugs

四、 论文计划

2016年6月，了解企业内部现在和行业背景，寻找参考文献，整理相关资料，了解业务需求。

2016年7月，建立理论模型、技术选型，确定实施方案，建立理论模型。技术选型，获取相关实施工具。

2016年9月，需求分析，确定需求方案，进行需求访谈及总结，形成需求文件。

2016年10月，概要设计，设计页面风格，设计功能模块，设计相关接口关系。

2016年12月，详细设计，成数据库表设计，完成接口工作设计，完成功能模块WBS拆分。

2017年2月，论文中期检查，完成中期报告、顺利通过中期检查。

2017年4月，系统实现，完成功能模块开发，接口功能开发，报表功能开发，数据处理SQL编写。

2017年5月，完成功能模块开发，接口功能开发，报表功能开发，数据处理SQL编写。

2017年7月，测试及应用，完成论文的编写。

2017年8月，论文答辩，编写相关文档，顺利完成论文答辩内容。

五、 论文进度

完成需求分析，根据某公司部门研发过程的持续集成的要求基础上，调研了业界中持续集成的实现落地过程和内容，集合自身需要，对于需要实现的功能、质量和时间进度有了整体的抱我，在此基础上完成了需求规格说明书、项目计划书，明确了项目干系人。

完成概要设计，生成概要说明书，某公司持续集成管理系统概要设计主要设计以下方面：数据库表设计、权限管理设计、用户管理设计、系统应用管理、持续集成任务流程设计。

完成详细设计，生成详细设计说明书。数据库表设计明确了数据库对象的定义，例如，表、索引、存储过程、函数、视图、包头、包体的命名规则，定义了存储过程、包体中程序模板。持续集成任务流程设计定义了流程的调用规范、流程各环节的扩展方式、对测试框架、持续集成工具的集成做了详细说明。权限设计中明确了数据权限和功能权限具体实现技术。

完成开发总工作量50%左右，生成报表开发说明书、数据库内容开发说明书。开发大致分为前台和后台工作，前台开发人员主要负责门户页面、用户权限管理、持续集成流程的交互与各功能与后台的数据核对；后台人员主要任务是编写数据服务，测试框架集成服务、静态代码集成服务、持续集成任务管理和触发服务，为前台页面提供数据支撑，也包括数据库各层的数据校验开发。

对已经开发完成的功能进行测试，并生成测试文档，就流程性的功能，使用实际开发中项目进行实地校验，分析其功能是否满足，其数据是否正确存储，其状态是否准确展示。

2016年6月，了解企业内部现在和行业背景，寻找参考文献，整理相关资料，了解业务需求。

2016年7月，建立理论模型、技术选型，确定实施方案，建立理论模型。技术选型，获取相关实施工具。

2016年9月，需求分析，确定需求方案，进行需求访谈及总结，形成需求文件。

2017年3月，概要设计，设计页面风格，设计功能模块，设计相关接口关系。

2017年5月，详细设计，成数据库表设计，完成接口工作设计，完成功能模块WBS拆分。

2017年9月，论文中期检查，完成中期报告、顺利通过中期检查。

2017年10月，系统实现，完成功能模块开发，接口功能开发，报表功能开发，数据处理SQL编写。

2017年12月，完成功能模块开发，接口功能开发，报表功能开发，数据处理SQL编写。

2018年1月，测试及应用，完成论文的编写。

2018年3月，论文答辩，编写相关文档，顺利完成论文答辩内容。

六、 目标

基于目前持续集成在软件过程管理中具有重要的作用，运用自动化技术将持续集成自动化是各软件公司发展的客观需要，并考虑到之前在工作中经常遇到因为集成导致的质量问题和工作效率问题，就其根本原因展开研究，将自动化测试、自动扫描、自动部署、自动反馈等技术工具应用到持续集成过程中，以实现以下目标：

1、 整合现有自动化测试工具，建立持续集成平台，该平台以规范和标准的流程把控软件开发过程，统一代码版本入口和编译构建文件，开发、测试及部署在统一标准下运行，保证整体代码质量的统一和可控；

2、 平台采用定制化的方式支持整个持续集成过程的环节，将各阶段的自动化测试工具和框架接入其中，发挥各个的重要作用，提升软件开发效率；

3、 整合质量数据，通过平台的数据质量分析，优化持续集成各环节的技术实现方式，从而推动软件开发过程的质量和效率更加优秀。

## 进展情况

2016年6月，了解企业内部现在和行业背景，寻找参考文献，整理相关资料，了解业务需求。

2016年7月，建立理论模型、技术选型，确定实施方案，建立理论模型。技术选型，获取相关实施工具。

2016年9月，需求分析，确定需求方案，进行需求访谈及总结，形成需求文件。

2016年10月，概要设计，设计页面风格，设计功能模块，设计相关接口关系。

2016年12月，详细设计，完成数据库表设计，完成接口工作设计，完成功能模块WBS拆分。

2017年2月，论文中期检查，完成中期报告、顺利通过中期检查。

2017年4月，系统实现，完成功能模块开发，接口功能开发，报表功能开发，数据处理SQL编写。

2017年5月，完成功能模块开发，接口功能开发，报表功能开发，数据处理SQL编写。

2017年7月，测试及应用，完成论文的编写。

2017年8月，论文答辩，编写相关文档，顺利完成论文答辩内容。

实际进展情况：完成概要设计、详细设计、完成接口工作设计，完成开发工作的50%

工作成果

完成需求分析，根据某公司部门研发过程的持续集成的要求基础上，调研了业界中持续集成的实现落地过程和内容，集合自身需要，对于需要实现的功能、质量和时间进度有了整体的抱我，在此基础上完成了需求规格说明书、项目计划书，明确了项目干系人和项目模块的完成时间。

完成概要设计，生成概要说明书，某公司持续集成管理系统概要设计主要设计以下方面：数据库表设计、权限管理设计、用户管理设计、系统应用管理、持续集成任务流程设计。

完成详细设计，生成详细设计说明书。数据库表设计明确了数据库对象的定义，例如，表、索引、存储过程、函数、视图、包头、包体的命名规则，定义了存储过程、包体中程序模板。持续集成任务流程设计定义了流程的调用规范、流程各环节的扩展方式、对测试框架、持续集成工具的集成做了详细说明，并对集成的技术指定了明确的需求。权限设计中明确了数据权限和功能权限具体实现技术。

完成开发总工作量50%左右，生成报表开发说明书、数据库内容开发说明书。开发大致分为前台和后台工作，前台开发人员主要负责门户页面、用户权限管理、持续集成流程的交互与各功能与后台的数据核对；后台人员主要任务是编写数据服务，测试框架集成服务、静态代码集成服务、持续集成任务管理和触发服务，为前台页面提供数据支撑，也包括数据库各层的数据校验开发。

对已经开发完成的功能进行测试，并生成测试文档，就流程性的功能，使用实际开发中项目进行实地校验，分析其功能是否满足，其数据是否正确存储，其状态是否准确展示。

主要创新点

1. 先进性

为了保证公司软件开发的快速迭代和代码质量，软件开发过程应加入灵活的开发模式和成熟的实施经验，让用户的需求经过开发的快速迭代不断反馈出来，在这个过程中需要利用持续集成平台好的机制，并集成当前业界流程优秀的自动测试工具，结合适合当前研发的客观情况，制定相应的集成策略来，支持整体项目目标的实现。

2. 自动化

软件的快速迭代的核心就是持续集成，利用持续集成能够在软件快速迭代的过程中，保证每个环节以自动化的方式快速进行，无需人工干预，节省人力成本，提升项目团队的工作效率。

在自动化测试上面，集成已有的自动化测试框架，无缝运行已有的自动化测试用例，并且对自动化测试的执行结果搜集展示，通过对结果的展示来实现对软件过程的效率提升。

3. 可视化

持续集成的过程也需要及时的反馈给项目成员，让项目成员对整个持续集成过程有一个清晰的认识，这样才可以让团队成员有一个共同目标。在持续集成平台上有整个持续集成过程的跟踪，包括持续集成任务执行的过程、结果等。前端使用了业界流行的前端框架开发，用户体验度好，应用了强大的视图插件来帮助展示集成结果，使持续集成的成果更有说服力。

4. 扩展性和可实施性

系统架构和功能要易于扩展，随着业务的拓展、业务模式的转变、数据规模的增大，系统要能够轻松升级。持续集成平台采用前后端分离的架构，提高系统的可伸缩性，采用的三层结构是明确进行分割的，逻辑上各自独立，并且能单独实现。中间件封装的业务逻辑对应用来说是透明的，在对局部进行扩展和修改时不会影响到系统其他部分的正常，在新的功能需求提出时，更快更低成本地对其记性扩展开发。

5. 流程定制化

系统在持续集成的设计实现中加入了流程定制化的功能，持续集成的环节在一定的场景中也是随之改变的，不变的是整个持续集成的完整性，在某些环节的实现和落地上，可以记性流程的定制，按照实际情况来制定持续集成的内容，以防流程的限制导致实际实施的阻碍。

6. 服务化

在持续集成平台的后台设计中，采用服务化的方式，来达到前后台分离解耦的效果，并将持续集成以服务的方式开发，以便在不同场景以不同方式来共享持续集成的便利。

## 计划及进度安排

接下来工作主要有两方面：

1、开发持续集成流程功能，预计17年9月至11月完成，需要3人月工作量；

2、系统集成测试：预计17年12月完成，需要1人月工作量

3、系统推广培训：预计18年1月完成，需要1人月工作量

其中系统集成测试的工作比较重要，需要实际将研发工作在整个流程中进行试验，对前台后台进行功能测试，因为持续集成是一个创新项目，具体的业务功能一般的测试人员尚未了结清楚，需要与测试人员沟通好平台的功能需求，按照功能需求进行测试，方可对整体的平台质量进行把控。

## 问题及整改方案

对于持续集成平台的服务化的方式需要进一步调研服务化的内容，包括自动化测试服务、静态代码扫描服务、自动部署的服务，这些需要对具体的测试工具进行进一步的调查研究，了解工具本身提供的功能，并结合自身的需求来定制和扩展开发。

另外一方面是实施过程中的推广问题，如何对现有的自动化测试技术进行转换变更，让其无缝接入，需要进一步了解现有的自动化测试技术的细节和实现方式，制定相应的集成方式，并结合持续集成流程的定制策略，让实施起来更加顺利。

## 参考文献

《软件开发的过程和管理》 张湘辉等 清华大学出版社

《敏捷软件开发：原则、模式与实践》 Robert C.Martin 清华大学出版社

《单元测试之道》 Andrew Hunt,David Tomas 电子工业出版社

《软件开发过程与案例》 陈宏刚,熊明华,林斌等编著 清华大学出版社

《单元测试之道》 Andrew Hunt,David Tomas 电子工业出版社

《软件配置管理及其工具应用》黄军，刘晓梅. 人民邮电出版社

《单持续集成在现代软件开发中的应用与研究》 徐仕成 中南大学