Android 项目持续集成系统设计与实现

黄俊伟，贾云伟

（重庆邮电大学信息与通信工程学院，重庆 400065）

摘　要：持续集成是软件开发中的一个重要的实践，决定着软件开发的质量。分析持续集成工具 Jenkins 的工作原理，结合 Android 项目提出一种基于 Jenkins 的持续集成方案，该方案以自动化测试为核心。详细说明了持续集成方案的实施步骤和过程。通过设计测试用例，验证了整个系统的可行性。

关键词：持续集成；自动化测试；安卓项目

随着 Android 手机的市场份额不断的扩大，Android 应用程序大规模出现。由于 Android 应用程序变更性比较大且需求易变化，这些变化可能会对整个 Android 程序带来风险，为了保障产品的质量，很多大公司已经开发出适合 Android 产品的持续集成系统，但这些系统普遍存在开发周期长而且成本比较高。为了缩减开发周期、降低开发成本，帮助中小企业也能实施持续集成，利用开源的工具设计并实现一个持续集成系统。

1　持续集成

集成是软件开发过程中一个重要的环节，如果不能早集成、常集成，那么等到最后再集成会出现很多问题，给项目带来不可预期的麻烦，为了提升开发的效率、降低项目的不可预见性，Martin Fowler 提出持续集成的观点（Continous Intergration ,CI），该观点指出在实践中项目成员频繁地进行集成，每次集成都通过自动化构建来验证，从而尽快地发现集成错误。

持续集成里的构建不仅仅包含编译、链接、组成可执行程序，还应包括对这个可执行程序进行的自动化测试。对于一次成功的构建，开发人员都被要求在这个自动化过程中的每一步都不能出错，而最重要的一步就是自动化测试，可以这么认为：自动化测试是持续集成的核心。

2　总体设计方案

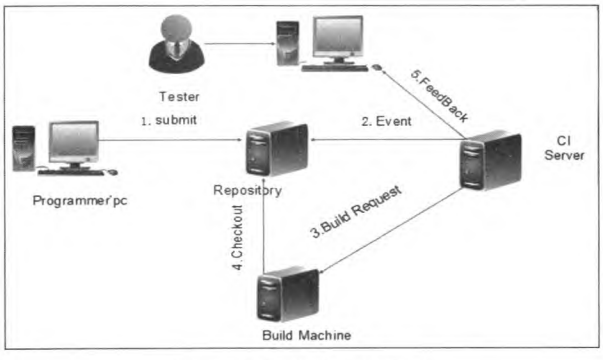

图 1　持续集成的场景图

持续集成的场景通常是这样的：首先开发者向版本控制库提交代码，同时持续集成服务器不断地检测版本库是否有变更，若有变更，则持续集成服务器执行构建脚本，最后把构建的结果反馈给开发人员。分析整个持续集成的过程，整个持续集成系统应该包含以下 3 个实践：第一是统一的源码库；第二构建自动化；第三自动化测试。图 1 为持续集成的场景图。

基于以上分析，结合 Android NotePad 项目，持续集成方案如下：对于统一的源码库，采用 git 工具来管理。之所以选择 git，是因为 git 是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理；对于构建的自动化，因为 Ant 基于 Java 与平台无关的构建工具，脚本的格式是基于 XML 的，比 make 脚本来说还要好维护一些，因此采用开源的工具 Ant 完成；对于自动化测试，由于本方案需要应用在 Andorid 项目中，对于不同的需求测试，采用不同的工具或框架。整个系统以开源的可以扩展插件的持续集成工具 Jenkins 来完成，它是一个支持自动化构建的工具。Jenkins 的工作流程图如图 2 所示。

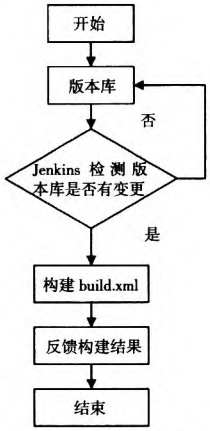

图 2　Jenkins 工作流程图

3　自动化测试框架及相关技术

自动化测试是持续集成的核心部分，一个持续集成系统应包含不同类型的自动化测试。在此只对单元测试、功能测试和压力测试进行展开。在 Android SDK 中已经给出了如何在系统上进行测试的方法，测试框架图如图 3 所示。

收稿日期：2014-09-24

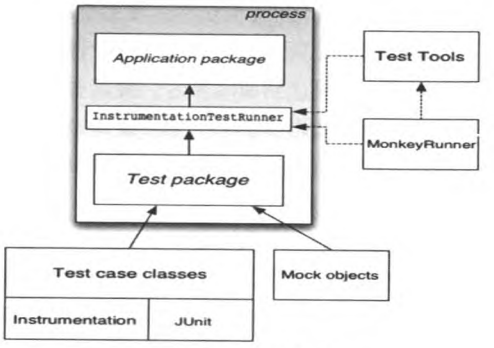

图 3　测试框架图

由图 3 知 Android 的测试方法有：

（1）Android 中的 JUnit:Android 上实现了 Junit 单元测试，利用 JUnit 等单元测试框架进行单元测试对于 Java 程序员并不陌生，利用这些非常有效的工具，使得代码的质量得到有效的监控和维护。

（2）Android 中的 Instrumentation 类:在图 3 的测试包中除了 JUnit 外，还有 Instrumentation 类。使用 Instrumentation，可以在主程序启动之前，创建模拟的系统对象，如 Context；控制应用程序的多个生命周期；发送 UI 事件给应用程序；在执行期间检查程序状态。 Instrumentation 框架通过将主程序和测试程序运行在同一个进程来实现这些功能。

此外，Android 测试对 Junit 进行了扩展，扩展如图 4 所示：

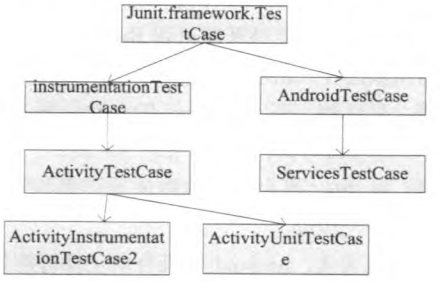

图 4　JUnit 中的测试扩展

图 4 中所示的是比较重要的几个扩展。比如 NotePad 的客户端首页的 Activity 是 NotesList，想测试它上面的一些功能，就可以实现 ActivityInstruemtationTestCase2<NotesList>的子类，而当这个子类运行的时候，NotesList 就会自动启动，而不需要调用 Intent 的方法。然后用 Instrumentation 来模拟用户的点击、拖拉等行为，这样就实现了自动化测试。从所建的工程来看，将 setup() 方法重写，然后在 setup() 方法里面调用自己封装的方法，为了使测试代码简洁，在代码中封装了 box 方法，来调用一些用户的基本行为，如点击、拖拉等。代码片段如下：

```
    private Box box;
    public NotePadTest() {
        super("com.example.android.notepad", NotesList.class);
    }
    @Override
    public void setUp() throws Exception {
    box = new Box(getInstrumentation(), getActivity());
    }
    public void test() throws Exception
    { }
}
```

可以使用 JUnit 进行单元测试，用 ActivitityInstruemtation-TestCase2 类进行功能测试，其他的方法进行补充。上述提供的方法中并没有提到压力测试，压力测试使用自动化测试工具 Monkey 来完成。图 5 是自动化测试的流程图。

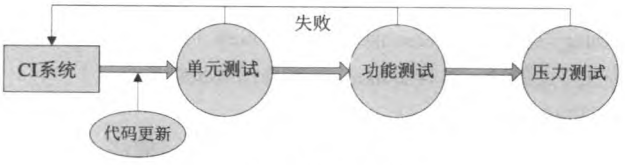

图 5　自动化测试的流程

4　持续集成方案的实现

基于以上对持续集成方案的分析，下文详细地介绍了持续集成方案的实施步骤、集成构建系统的配置和自动化构建的结果。

4.1　实施步骤

（1）环境的搭建。根据实施的目的，准备 2 台机器来搭建持续集成环境：1 台用作 Jenkins 服务器，1 台用作源代码仓库。

（2）编写并调试 ant 脚本 build.xml。该脚本是持续集成各个环节中最关键的部分，主要包含 5 部分的配置代码：清理、编译源代码、安装 apk、执行自动化测试以及测试用例覆盖率的检查。

（3）编写自动化测试脚本。测试脚本包含单元测试脚本、功能测试脚本以及压力测试脚本。测试用例用 TestLink 来管理。

（4）验证系统的可行性，图 6 为整个系统的实施图。

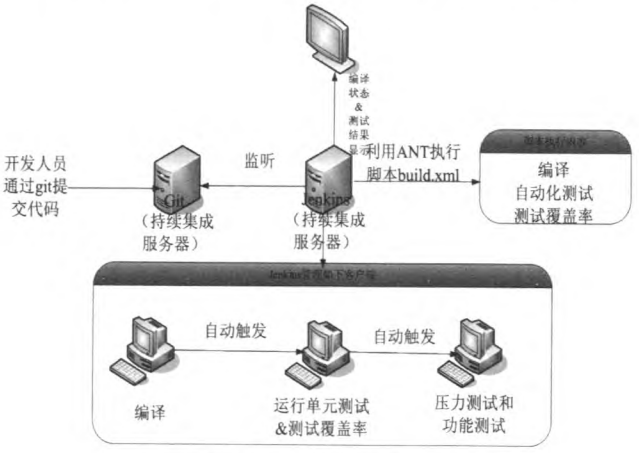

图 6　系统的实施图

4.2　基于 Jenkins 持续集成系统配置

4.2.1 创建 Job

进入 Jenkins 首页后，新建两个 Job:NotePad 和 NotePadTest，点击 NotePad 进入配置界面，在源码管理处选择 git 并填写 NotePad 项目的源代码的路径（源代码包含 NotePad 代码和 NotePad 的测试代码），构建项选择 Invoke Ant，在 targets 处填上 clean debug install，Job NotePad 主要完成 NotePad 的编译并将生成的 apk 安装到手机中。

4.2.2 TestLink 的配置

进入 Jenkins 首页后，点击 NotePadTest 进入配置界面，构建项选择 Invoke TestLink，按照 TestLink 新建的测试项目，填写 TestLink Configuration；Test Execution 选择 Invoke Ant，在 targets 处填上 clean emma debug install test；Result Seeking Strategy 选择 JUnit case name，在 Include Pattern 处填写测试报告文件的路径，这样 Jenkins 就能显示测试结果。

4.2.3 Emma 的配置

进入 Jenkins 首页后，首先要安装 Emma 插件，在构建后的操作中选择 Record Emma coverage report，在 Folders or files containing Emma XML reports 中填入测试用例覆盖率检查报告文件 coverage.xml 的路径，这样 Jenkins 就能显示测试用例覆盖检查的结果。

4.3 自动化构建的结果及分析

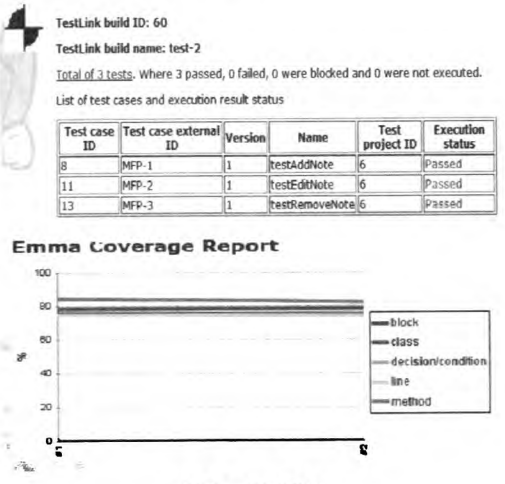

图 7 CI 报告

Jenkins 中的 Build History 会记录当前构建的状态和历史构建的结果，绿灯表示构建成功，红灯表示构建失败。根据上述的配置，提交变更代码触发 Jenkins 运行，Jenkins 会通过 html 页面显示测试报告和 Emma 测试用例覆盖率的检查报告，如图 7 所示。通过 CI 测试报告，可以看到 3 个测试用例全部成功，但代码的覆盖率只达到 80%左右，故需要继续完善测试用例，尽可能地提高代码覆盖率。如果编译未通过或者有失败的测试用例，则开发人员应当第一时间修改代码，再自动化构建一次，直到构建成功为止。

通过该持续集成方案，有利于提前发现项目中存在的问题，可以按预期发布版本，保证软件开发的质量。而且采用的都是开源的框架，该方案具有开发周期短、花费小等优点，对于正在开发 Android 项目的公司具有很大帮助。

5 结语

提出了 Android 项目持续集成方案，该方案主要以自动化测试为核心，以较少的成本实施了该持续集成方案，解决了 Android 项目的集成问题，可以帮助中小企业快速地实施持续集成。

不足之处是缺少持续部署，在软件较简单的情况下，无论在什么环境下，同样的测试用例运行的结果基本一样，但当软件项目比较复杂时，往往在不同的环境中，测试的结果不一样，故对于较大的项目，建议加入持续部署环节。

参考文献

[1] MatrinFowler. ContinuousIntegration. 2006.

[2] 陈婧欣. 基于 Hudson 的持续集成方案的研究与实践 [D]. 东北师范大学，2011.

[3] 金敏，周翔. 高级软件开发过程 Rational 统一过程、敏捷过程与微软过程. 北京：清华大学出版社，2005.

[4] 谢红霞，吴红梅. 基于 Android 的自动化测试的设计与实现 [J]. 计算机时代，2012，(2)：20-22.

[5] 宋春雨. Android 平台自动化测试的研究与实践 [D]. 北京邮电大学，2012.

[6] 杨怡君，黄大庆. Android 手机自动化性能测试工具的研究与开发 [J]. 计算机应用，2012.

(上接第 43 页)

[3] 杨俊宝. 基于 JSP 技术的学生宿舍管理系统的设计与实现 [D]. 沈阳工业大学，2006.

[4] 陈全，邓倩妮. 云计算及其关键技术 [J]. 计算机应用，2009，(9).

[5] 黄永峰，张久岭，李星. 云存储应用中的加密存储及其检索技术 [J]. 中兴通讯技术，2010，(4).

[6] 陈康. 云计算后台大规模数据处理技术探讨 [J]. 电信工程技术与标准化，2009，(11).

[7] 王晓霞，原晓燕，卢佩琴. 电子文件信息安全与保护措施研究 [J]. 兰台世界，2008，(10)：17-18.

[8] 赵铭伟，毛锐，江荣安. 基于过滤驱动的透明加密文件系统模型 [J]. 计算机工程，2009，35 (1)：150-152.

[9] 郑明雄，蒋朝根. Windows 下的文件操作监控与过滤 [J]. 现代计算机 (专业版)，2008，(11)：86-88.

[10] 罗谦，舒辉，曾颖. 二进制文件结构化比较的并行算法实现 [J]. 计算机应用，2007，27 (5)：1260-1263.

[11] Forman George，Eshghi Kave，Suermondt Jaap.Efficient Detection of Large-Scale Redundancy in Enterprise File Systems [J]. SIGOPS Operating Systems Review，2009，43 (1)：84-91.

[12] 石峰. 利用 Delphi7.0 存取配置文件实现 Oracle 数据库的远程连接 [J]. 电脑编程技巧与维护，2011，16.

[13] 郝艳芳，廉永健. JSP Web 应用性能优化的探讨 [J]. 机电产品开发与创新，2006，04.