

《软件测试》实验二测试管理报告

- 姓名：郑伟林
- 班级：1619303
- 学号：061920125

目录

《软件测试》实验二测试管理报告

目录

- 一、实验题目
- 二、git仓库
- 三、持续集成
- 四、缺陷管理
- 五、README
- 六、总结

一、实验题目

1. 搭建一个包含以下模块的开发环境

- Git代码仓库（本次实验选择Gitee）
- Jenkins或其它持续集成与测试工具
- issue tracking模块（本次实验采用Gitee自带的Issue追踪模块）

2. 采用Ant、Maven等配置工具配置软件项目，项目中必须包含 build 和test两个目标，build用来导出程序的完整jar包，test用来 对程序进行测试。（本次实验采用Maven来配置项目）

3. 在持续集成（如Jenkins）中应用项目配置中设定的test任务，完成冒烟测试。（Jenkins日志中可以看到执行测试用例的提示）

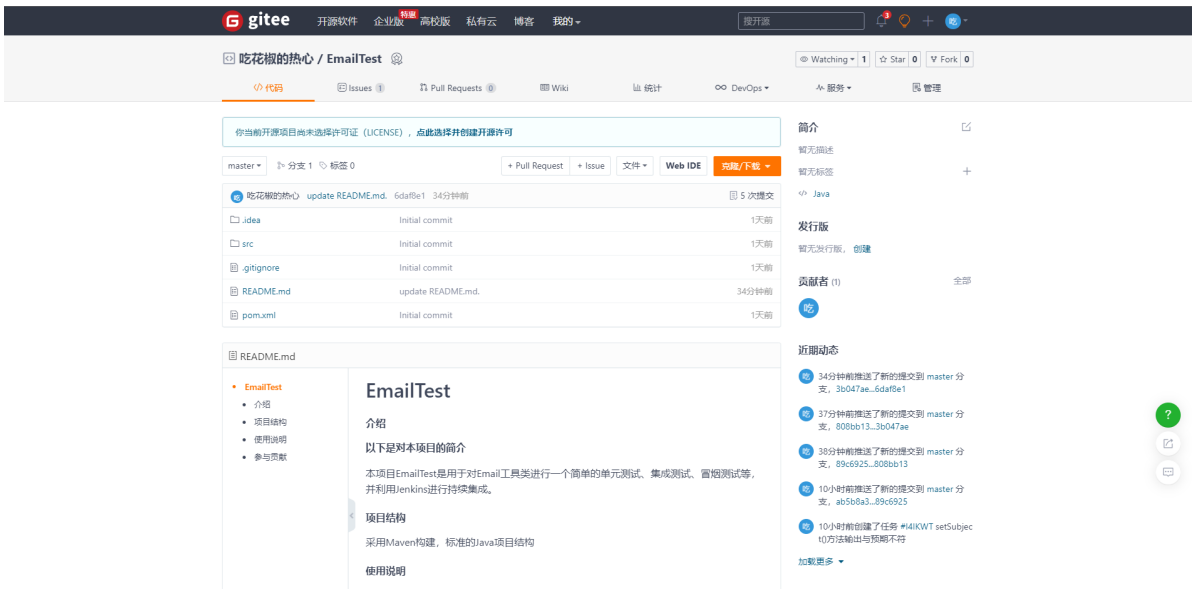
4. 调研缺陷提交的一般性流程和格式要求，在缺陷管理工具中登记发现的bug，填写缺陷信息。

5. 按项目将要给其它人接手的角度，为其编写一个简要的 README.md来说明项目开发、测试与缺陷提交步骤、方法

二、git仓库

选择Gitee的远程仓库对项目代码进行管理。先在远程仓库建立一个新仓库，然后在本地项目中初始化git，并添加提交，push到远程仓库。

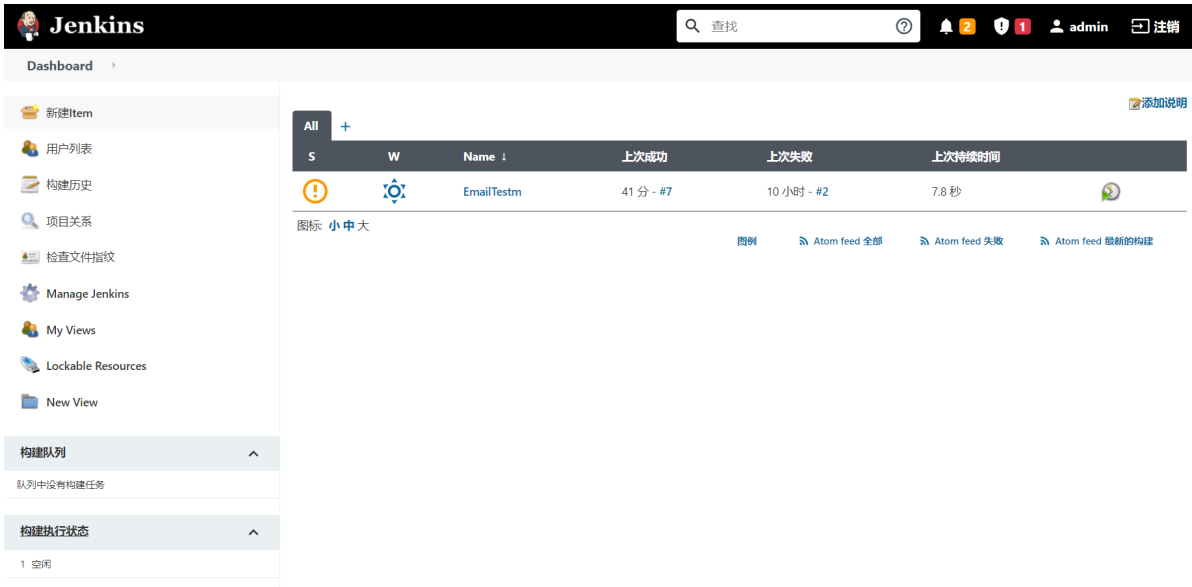
仓库地址：<https://gitee.com/rlin1538/email-test>



三、持续集成

选择jenkins来对远程仓库的代码进行自动构建。

安装完Jenkins服务后，创建一个Maven项目Item，配置好仓库信息和构建信息，进行持续测试。



配置轮询构建或监听提交进行构建，Jenkins将自动对仓库里的项目进行构建、测试。

若注入缺陷，则测试会失败，并给出错误信息：

Dashboard	EmailTestm	#7
		<div>Results :</div> <div>Failed tests: setSubject(org.apache.commons.mail.SimpleEmailTest): expected:<TestMail[]> but was:<TestMail[##THERE IS A BUG##]> setSubject(org.apache.commons.mail.SimpleEmailTest): expected:<TestMail[]> but was:<TestMail[##THERE IS A BUG##]></div> <div>Tests run: 12, Failures: 2, Errors: 0, Skipped: 0</div> <div>[ERROR] There are test failures.</div> <div>Please refer to C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\target\surefire-reports for the individual test results.</div> <div>[JENKINS] Recording test results</div> <div>[INFO]</div> <div>[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ TestEmail ---</div> <div>[INFO] Building Jar: C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\target\TestEmail-1.0-SNAPSHOT.jar</div> <div>[INFO]</div> <div>[INFO] --- maven-install-plugin:2.4:install (default-install) @ TestEmail ---</div> <div>[INFO] Installing C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\target\TestEmail-1.0-SNAPSHOT.jar to C:\WINDOWS\system32\config\systemprofile\.m2\repository\org\apache\commons\TestEmail\1.0-SNAPSHOT\TestEmail-1.0-SNAPSHOT.jar</div> <div>[INFO] Installing C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\pom.xml to C:\WINDOWS\system32\config\systemprofile\.m2\repository\org\apache\commons\TestEmail\1.0-SNAPSHOT\TestEmail-1.0-SNAPSHOT.pom</div> <div>[INFO] -----</div> <div>[INFO] BUILD SUCCESS</div> <div>[INFO] -----</div> <div>[INFO] Total time: 3.156 s</div> <div>[INFO] Finished at: 2021-11-17T20:34:10+08:00</div> <div>[INFO] -----</div> <div>Waiting for Jenkins to finish collecting data</div> <div>[JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\pom.xml to org.apache.commons/TestEmail/1.0-SNAPSHOT/TestEmail-1.0-SNAPSHOT.pom</div> <div>[JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\target\TestEmail-1.0-SNAPSHOT.jar to org.apache.commons/TestEmail/1.0-SNAPSHOT/TestEmail-1.0-SNAPSHOT.jar</div> <div>channel stopped</div> <div>Finished: UNSTABLE</div>

若无缺陷，则给出构建成功信息：

Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\pom.xml to org.apache.commons/TestEmail/1.0-SNAPSHOT/TestEmail-1.0-SNAPSHOT.pom
[JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\EmailTestm\target\TestEmail-1.0-SNAPSHOT.jar to org.apache.commons/TestEmail/1.0-SNAPSHOT/TestEmail-1.0-SNAPSHOT.jar
channel stopped
Finished: SUCCESS

四、缺陷管理

利用Gitee提供的Issue追踪管理，可以很方便的对项目缺陷进行追踪、管理。

缺陷管理一般流程：

1. 测试人员发现缺陷
2. 测试人员对缺陷进行重现
3. 测试人员与开发人员进行沟通，核对需求，确认缺陷
4. 测试人员提交问题单
5. 开发人员修复问题，并进行自验证，然后提交新版本
6. 测试人员验证问题是否修复，修复则

缺陷提交格式

运行环境——简述运行代码的环境

bug描述——bug的直观简短的描述

bug复现步骤——提供复现问题的步骤。

错误代码——发生错误的代码部分

预期结果——描述需求预期的结果，必要时辅以截图说明

实际结果——描述项目的实际结果，必要时辅以截图说明

gitee

开源软件 企业版 高校版 私有云 博客 我的

搜索

3

吃花椒的热心 / EmailTest

Watching 1

Star 0

Fork 0

<> 代码

Issues 1

Pull Requests

Wiki

统计

DevOps

服务

管理

Issues / 详情

email.setSubject() 方法输出与预期不符

待办的

#14IKWT

吃花椒的热心 拥有者

创建于 2021-11-17 11:21

编辑 删除

运行环境

Java 1.8.0_211

IDEA 2021.1

JUnit 4

Maven 3.8.3

bug描述: Email类中的setSubject()方法设置的邮件标题与预期不符合, 后面多加了一串字符。

bug复现步骤: 在SimpleEmailTest类中, 测试setSubject()方法, 调用email.setSubject("TestMail"), 检查输出是否与预期一致

错误代码:

```
public void setSubject() {  
    email.setSubject("TestMail");  
    assertEquals("TestMail",email.getSubject());  
}
```

预期结果:

TestMail

实际结果:

TestMail##THERE IS A BUG##

0 0 +

+ 添加评论

状态

待办的

负责人

未设置

标签

bug

里程碑

未关联

Pull Requests

未关联

关联的 Pull Requests 被合并后可能会关闭此 issue

分支

未关联

开始日期 - 截止日期

未设置 - 未设置

置顶选项

不置顶

优先级

不指定

参与者 (1)

吃

Gitee

开源软件企业版高校版私有云博客我的

搜开源

3

+

吃

README.md

EmailTest

- 介绍
- 项目结构
- 使用说明
- 参与贡献

EmailTest

介绍

以下是对本项目的简介

本项目EmailTest是用于对Email工具类进行一个简单的单元测试、集成测试、冒烟测试等，并利用Jenkins进行持续集成。

项目结构

采用Maven构建，标准的Java项目结构

使用说明

- 在src/main目录下的TestEmail.java中是手工进行一个单元测试
- 在src/test中，采用JUnit生成单元测试用例，并写了一个批量单元测试的方法，如需要测试其它方法，请用IDEA中的JUnit插件生成
- Jenkins已配置一个项目，能够成功构建并发现test中的错误

参与贡献

- Fork 本仓库
- 新建分支
- 提交代码
- 新建 Pull Request
- 缺陷提交

缺陷格式为：

bug描述——bug的直观简短的描述

bug复现步骤——对于操作步长超过3的，且RD难以理解怎么复现的问题，提供复现问题的步骤。

预期结果——描述需求预期的结果，必要时辅以截图说明

实际结果——描述RD实现后的实际结果，必要时辅以截图说明

近期动态

吃

1小时前推送了新的提交到 master 分支。3b047ae...6daf8e1

吃

1小时前推送了新的提交到 master 分支。808bb13...3b047ae

吃

1小时前推送了新的提交到 master 分支。89c6925...808bb13

吃

10小时前推送了新的提交到 master 分支。ab5b8a3...89c6925

吃

10小时前创建了任务 #14IKWT setSubject()方法输出与预期不符

加载更多

本次实验学会了项目管理与集成持续测试，通过git建立代码仓库并上传到远程仓库可以便捷地对项目进行管理，Jenkins自动构建可以简化开发者的工作，只需将改好的项目上传提交即可进行测试。缺陷管理现在一般都用远程仓库所带的缺陷追踪，易于使用。