

集成测试计划

1 引言

1.1 目的

本文是“XP在线考试”系统的集成测试的大纲，主要描述如何进行集成测试活动、如何控制集成测试活动、集成测试活动的流程以及集成测试活动的工作安排等，以保证程序集成起来能正常工作，保证程序的完整运行。

1.2 范围

本测试计划主要是针对软件的集成测试:不含硬件、系统测试以及单元测试（完成单元测试是前提）。

主要的任务:

- 测试在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失;
- 测试各个子功能组合起来，能否达到预期要求的父功能;
- 一个模块的功能是否会对另一个模块的功能产生不利的影响;
- 全局数据结构是否有问题;
- 单个模块的误差积累起来，是否会放大，从而达到不可接受的程度。

主要测试方法是黑盒测试方法。必要的集成测试是回归测试。

本文主要的读者对象是项目负责人、集成部门负责人、集成测试设计师。

1.3 术语

和业务及技术相关的术语。

1.4 测试环境

序号	描述	配置
1	客户端浏览器	Chrome, Microsoft Edge, Firefox
2	输入习惯	中文, 英文
3	服务端操作系统	Windows 10
4	测试工具	JUnit, Selenium, github, swagger
5	数据库	MySQL

1.5 参考文献

开始测试涉及以下文档：

- 《XP在线考试系统需求规约》
- 《XP在线考试系统需求分析规约》
- 《XP在线考试系统概要设计说明书》
- 《XP在线考试系统详细设计说明书》
- 《单元测试报告》

2 集成策略

采用自底向上的集成方式。自底向上集成方式从程序模块结构中最底层的模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块（包括子模块的所有下属模块）事前已经完成组装并经过测试，所以不需要再编制桩模块。选择这种集成方法，管理方便，测试人员能较好地锁定软件故障所在位置。

2.1 集成内容

1. 函数集成
如函数间接口、函数是否调用正常。
2. 功能集成
如不同函数间实现的业务功能。
3. 数据集成
如数据传递是否正确，对于传入值的控制范围是否一致等。
4. 子系统集成
如把不同通信子系统、业务子系统及报表子系统进行集成。

2.2 集成策略

假如系统的集成测试采用自底向上的集成（Bottom-Up Integration）方式。自底向上集成方式从程序模块结构中最底层的模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块(包括子模块的所有下属模块)事前已经完成组装并经过测试，所以不再需要编制桩模块。选择这种集成方法，管理方便，测试人员能较好地锁定软件故障所在位置。集成测试中的主要步骤：

- 制订并审核集成测试计划。
- 测试用例分析、设计及评审。
- 测试的实施。
- 测试的执行。
- 测试的分析和评估。

如下所示。

活动	输入	输出	职责
制定并审核集成测试计划	概要设计说明书等	集成测试计划	制定测试计划
测试用例分析和设计及评审	集成测试计划概要设计说明书	设计测试用例	设计测试用例并评审
测试的实施	集成测试用例测试过程	测试脚本、测试环境、测试驱动或桩	开发测试脚本、搭建测试环境、开发驱动或桩
测试的执行	测试脚本	测试结果	记录结果、跟踪缺陷
测试的分析和评估	集成测试计划测试结果	测试分析和评估报告	会同开发人员评估测试结果，得出测试报告

2.3 集成顺序

1. 软件集成顺序
自底向上，先函数、数据、功能再子系统。
2. 软件/硬件集成顺序
无。
3. 子系统集成顺序
略。

3 测试过程描述

3.1 软件集成测试

以下举例说明。

在本项目中，集成测试的主要过程如下：

1. 设计集成测试用例。
自底向上集成测试的步骤：
 - 步骤1：按照概要设计规格说明，明确哪些是被测模块。在熟悉被测模块性质的基础上对被测模块进行分层，在同一层次上的测试可以并行进行，然后排出测试活动的先后关系，制订测试进度计划。
 - 步骤2:在步骤1的基础上，按时间线序关系将软件单元集成为模块，并测试在集成过程中出现的问题。这里可能需要测试人员开发一些驱动模块来驱动集成活动中形成的被测模块。对于比较大的模块，可以先将其中的某几个软件单元集成为子模块，然后再集成为一个较大的模块。
 - 步骤3:将各软件模块集成为子系统(或分系统)。检测各自子系统是否能正常工作。同样，可能需要测试人员开发少量的驱动模块来驱动被测子系统。
 - 步骤4:将各子系统集成为最终用户系统，测试各子系统能否在最终用户系统中正常工作。
2. 集成测试：组织人员按照(1)中的集成测试用例测试系统集成度。
 - 测试人员按照测试用例逐项进行测试，并且将测试结果填写在测试报告上(测试报告必须覆盖所有测试用例);
 - 测试过程中发现Bug,将Bug通过Issue发送到github仓库中;

- 对应责任人接到Bugzilla通过email发过来的Bug信息;
- 对于明显的并且可以立刻解决的Bug,将Bug发给开发人员(Bug状态为ASSIGNED,对于不是Bug的提交, 集成部经理通知测试设计人员和测试人员, 对相应文档进行修改(Bug状态为RESOLVED,决定设置为INVALID);对于目前无法修改的, 将这个Bug放到下一轮次进行修改(Bug状态为REOPENED, 决定设置为REMIND)。

3. 问题反馈:反馈Bug给开发人员

- 开发人员接到发过来的Bug立刻修改(Bug状态为RESOLVED,决定设置为FIXED);
- 测试人员接到 Jira 通过email发过来的Bug更改信息, 应该逐项复测, 填写新的测试报告(测试报告必须覆盖上一次中所有REOPENED的测试用例)。

4. 回归测试

- 重新测试修复Bug后的系统。重复(3), 直到(4)中回归测试结果到达系统验收标准。
- 如果复测有问题返回第(2)步(Bug状态为REOPENED),否则关闭这项Bug(Bug状态为CLOSED)。
- 本轮测试中测试用例中有90%一次性通过测试, 结束测试任务; 本轮测试中发现的Bug有95%经过修政并且通过再次测试(即Bug状态为CLOSED), 返回进行新一轮测试。

5. 集成测试测试总结报告:完成以上4步后,综合相关资料生成报告。

6. 进入系统测试: a测试、β测试。

3.2 软件\硬件集成测试

主要涉及硬件和软件间集成, 硬件和硬件间集成这里一般不涉及, 集成关注:

- 功能点:根据用户文档列出所有功能点, 检验其正确性。
- 接口:根据用户文档列出所有接口, 检验其正确性。
- 流程处理:根据用户文档列出所有流程, 检验其正确性。
- 外部接口:根据用户文档列出所有外部接口, 检验其正确性。

3.3 子系统集成测试

完成子系统间集成。

4 集成测试验收标准

4.1 模块（指单元集成后的模块）验收标准

接口：接口提供的功能或者数据正确。

功能点：验证程序与产品描述、用户文档中的全部说明相对应，一致性。

流程处理：验证程序与产品描述、用户文档中的全部说明相对应，一致性。

外部接口：验证程序与产品描述、用户文档中的全部说明相对应，一致性。

4.2 集成测试验收标准

首先, 集成测试用例中所设计的功能测试用例必须全部通过,性能及其他类型测试用例95%以上通过。在未通过的测试用例中, 不能含有“系统崩溃”和“严重错误”这两种错误, “一般错误”小于1%。测试结果与测试用例中期望的结果一致, 测试通过, 否则标明测试未通过。

5 工具

5.1 测试工具

测试中心平台: JUnit Echarts Vue;

性能测试工具: JMeter;

集成测试工具: Swagger;

5.2 其他工具

电子表格软件: Excel

缺陷跟踪工具: Github Issues

6 挂起、恢复和退出条件

6.1 挂起

- 进入第一轮集成测试，测试人员大体了解一下产品情况，如果发现在单元内存在三个以上错误或缺陷以及操作性的错误，退回单元测试组测试；
- 遇到有项目优先级更高的集成测试任务；
- 在复测过程中发现产品无法运行下去；
- 人员、设备不足；
- 重大突发紧急情况。

6.2 恢复

- 符合进入集成测试条件；
- 项目优先级更高的集成测试任务暂告完成；
- 复测过程中产品可以运行下去；
- 人员、设备到位；
- 突发事件处理完成；
- 退出；
- 项目因故终止；
- 不可抗力：合同专用条款中约定等级以上的自然灾害也属不可抗力；
- 其他原因的测试工作频频被挂起或者挂起后迟迟恢复不了，并超过了客户要求的期限。

7 责任人和时间表

7.1 责任

1. 测试负责人：沈天宇

控制并完成测试任务和测试过程，决定测试人员提交上来的Bug是否需要修改。

2. 测试设计人员：姜昱九 余杰 匡政

设计集成测试用例。

3. 测试人员：沈天宇 姜昱九 余杰 匡政

按照测试用例进行测试活动。

4. 开发人员：沈天宇 姜昱九 余杰 匡政

程序Bug修改，开发人员之间协调。

8 缺陷跟踪

记录：利用Github平台的issue记录Bug，并指定相关责任人。
解决问题：小组会议以及开发人员协调负责人，协调测试开发之间的工作。

9 集成测试用例与报告

9.1 User Controller 模块

User APIs : User Controller		Show/Hide	List Operations	Expand Operations
GET	/user/info	获取用户的详细信息, 包括个人信息页面和操作权限		
POST	/user/login	根据用户名或邮箱登录,登录成功返回token		
POST	/user/register	注册		

9.1.1 /user/login接口

3个参数, loginType,username/email和password, 根据loginType等于1或者2, 决定用户采用用户名还是邮箱来登录

用例编号	USER_TEST_CASE_1			
优先级	高			
测试覆盖的系统功能	用户通过用户名/邮箱和密码登录功能			
前置条件	无			
用例设计方法	强健壮等价类测试			
用例序号	输入	期望结果	实际结果	测试结果
1	loginType=1, username=teacher, password=admin123	登录成功	登录成功	通过
2	loginType=1, username=teacher, password=adad	密码用户名或者密码错误	密码用户名或者密码错误	通过
3	loginType=1, username=120@qq.com, password=admin123	密码用户名或者密码错误	密码用户名或者密码错误	通过
4	loginType=1, username=1455@qq.com, password=adad	密码用户名或者密码错误	密码用户名或者密码错误	通过
5	loginType=2, username=teacher, password=adad	密码错误	密码错误	通过
6	loginType=2, username=teacher, password=admin123	密码用户名或者密码错误	密码用户名或者密码错误	通过
7	loginType=2, username=120@qq.com, password=admin123	登录成功	登录成功	通过
8	loginType=2, username=120@qq.com, password=adad	密码用户名或者密码错误	密码用户名或者密码错误	通过

9.1.2 /user/register接口

4个参数, 邮箱, 手机号, 密码和确认密码
邮箱和手机号不能与数据库中重复
密码和确认密码需要相同

用例编号	USER_TEST_CASE_2			
优先级	高			
测试覆盖的系统功能	用户输入邮箱, 密码, 确认密码和手机号进行注册			
前置条件	无			
用例设计方法	强健壮等价类测试			
用例序号	输入	期望结果	实际结果	测试结果
1	e-mail=120@qq.com, password=123456, password2=123456, telephone=19956869829	注册失败	注册失败	通过
2	e-mail=45454@qq.com, password=123456, password2=123456, telephone=19956869829	注册失败	注册成功	不通过
3	e-mail=120@qq.com, password=258741, password2=123456, telephone=19956869829	注册失败	注册失败	通过
4	e-mail=120@qq.com, password=123456, password2=123456, telephone=18800002222	注册失败	注册失败	通过
5	e-mail=456662@qq.com@qq.com, password=123456, password2=123456, telephone=19956869829	注册成功	注册失败	不通过
6	e-mail=456662@qq.com@qq.com, password=258741, password2=123456, telephone=19956869829	注册失败	注册失败	通过
7	e-mail=120@qq.com, password=258741, password2=123456, telephone=18800002222	注册失败	注册失败	通过
8	e-mail=444778@qq.com@qq.com, password=258741, password2=123456, telephone=18800002222	注册失败	注册失败	通过

9.2.1 /exam/detail/{id}

9.2.2 /exam/record/detail/{id}

用例编号	USER_TEST_CASE_4
优先级	高
测试覆盖的系统功能	获取考试记录详情
前置条件	
用例设计方法	
用例序号	
	<div> <div>健壮性等类测试</div> <div>期望结果</div> <div>实际结果</div> <div>测试结果</div> </div>
1	<div> <div> 输入 <div> 29ee2a5e43cb45e59e2f6ae82b9ead58 </div> </div> <div> <div> 登录成功 <div> 期望结果 实际结果 测试结果 </div> </div> <div> <pre> { "code": 0, "msg": "获取考试记录详情成功", "data": { "examRecord": { "examRecordId": "29ee2a5e43cb45e59e2f6ae82b9ead58", "examId": "7f36f47a75184adeb3df4fb8c4058a5a", "answerOptionIds": "38085807d06948ca8ad68eaca522c85f8f", "answerOptionIds": "38085807d06948ca8ad68eaca522c85f8f" } } } </pre> </div> </div> </div>
2	<div> <div> 输入 <div> 29ee2a5e43cb45e59e2f6ae82b9ead54 </div> </div> <div> <div> 登录成功 <div> 期望结果 实际结果 测试结果 </div> </div> <div> <pre> { "code": -1, "msg": "获取考试记录详情失败", "data": {} } </pre> </div> </div> </div>

