

第三部分 测试实施

前期所做的所有工作包括测试计划、整理测试需求、设计测试用例等均为为测试实施做准备的。测试实施阶段有几项工作需要进行,首先将前期所设计的测试用例进行测试实施,并记录下测试执行人、测试时间、是否发现缺陷等相关信息,其次如果发现缺陷,测试执行人员需要填写相关的缺陷报告,最后测试执行人员需要对本版本的测试进行总结。所以测试实施阶段主要有执行测试、编写缺陷报告、编写测试总结等相关工作。

3.1 执 行 测 试

执行测试任务

1. 任务类型定义

执行测试任务类型见表 3-1。

表 3-1 执行测试任务

任务				
执行测试任务类型描述:				
测试作业类型		保存	模 板	备 注
计划任务	业务理解	△		
	执行测试记录作成	○		
	执行测试记录评审	○		在原文件基础上添加标注
	执行测试记录修改	○		
日常任务	问题交流	△		
	会议	○	附录 A 中的“1. 会议模板”	
	培训/学习	△		
	邮件联络	△		
	周报	○	附录 A 中的“2. 周报模板”	
	自定义任务 1	△		
	自定义任务 2	△		
	自定义任务 3	△		

续表

- 任务说明：
- (1) “保存”列，○表示必须、△表示可选；
 - (2) 计划任务“执行测试记录作成”及“执行测试记录修改”主要针对界面测试用例和功能测试用例的执行。执行测试记录是在界面测试用例文档和功能测试用例文档的基础上补充执行记录；
 - (3) 日常任务“会议”，以项目组为单位制成，保存为 Word 格式，项目测试组长负责；
 - (4) 日常任务“周报”，以个人为单位制成，保存为 Word 格式，个人负责，每周五提交

2. 任务计划安排

每个项目组完成“学创购书网”所有模块的界面测试用例和功能测试用例的执行。模块的界面测试用例设计 and 功能测试用例设计已经完成，所以在执行测试时要与前期的测试用例设计严格保持一致，并在界面测试用例设计文档和功能测试用例设计文档的基础上添加测试执行记录内容。测试经理负责本项目组工作任务分配，分配形式见表 3-2。

表 3-2 执行测试任务计划

作业内容	模块 ID	担当者	作业类型	预定作业时间(H)	计划开始日	计划结束日
执行测试	M1～M3	测试员 A	界面测试用例执行测试作成			
			功能测试用例执行测试作成			
		测试经理	执行测试评审			
		测试员 A	界面测试用例执行测试修改			
			功能测试用例执行测试修改			
	M4～M7	测试员 B	界面测试用例执行测试作成			
			功能测试用例执行测试作成			
		测试经理	执行测试评审			
		测试员 B	界面测试用例执行测试修改			
			功能测试用例执行测试修改			
	M8～M12	测试员 C	界面测试用例执行测试作成			
			功能测试用例执行测试作成			
		测试经理	执行测试评审			
		测试员 C	界面测试用例执行测试修改			
			功能测试用例执行测试修改			
	M13～M15	测试员 D	界面测试用例执行测试作成			
			功能测试用例执行测试作成			
		测试经理	执行测试评审			
		测试员 D	界面测试用例执行测试修改			
			功能测试用例执行测试修改			

续表

作业内容	模块 ID	担当者	作业类型	预定作业时间(H)	计划开始日	计划结束日
执行测试	M16~M18	测试员 E	界面测试用例执行测试作成			
			功能测试用例执行测试作成			
		测试经理	执行测试评审			
		测试员 E	界面测试用例执行测试修改			
			功能测试用例执行测试修改			

执行测试任务说明：

(1) 测试执行范围达到 100％测试需求覆盖；

(2) 参照测试用例设计中每个测试用例的优先级，所有优先级为高、中的测试用例均被执行，优先级为低的测试用例执行率达到 60％

3. 任务工作量汇总

以项目组为单位进行工作量汇总,测试经理负责,汇总形式见表 3-3。

表 3-3 执行测试任务工作量汇总

计划任务工作量汇总(H)——按人						
任务类型	测试员 A	测试员 B	测试员 C	测试员 D	测试员 E	合计
业务理解						
执行测试记录作成						
执行测试记录评审						
执行测试记录修改						
合 计						
日常任务工作量汇总(H)——按人						
任务类型	测试员 A	测试员 B	测试员 C	测试员 D	测试员 E	合计
问题交流						
会议						
培训/学习						
邮件联络						
个人工作(计划、总结)						
自定义任务 1						
自定义任务 2						
自定义任务 3						
合 计						
本周总工作量						

执行测试规范

执行测试记录是在测试用例设计文档的基础上添加执行记录,所以执行测试记录所使用的模板就是测试用例文档所使用的模板,即附录 A 中的“5. 界面测试用例模板”、附录 A 中的“6. 功能测试用例模板”。在此模板之上补充执行测试的记录内容即可,说明见表 3-4。

表 3-4 执行测试说明

序号	记录内容	说 明
1	测试结果	(1) ○: OK; (2) ×: NG(Not Good); (3) △: 不可实施
2	实施担当	实施测试人员姓名
3	实施日期	实施测试日期(××年××月××日)
4	测试版本	测试系统版本号
5	对应 Bug ID	如果发现缺陷,所对应的缺陷 ID 号
6	备注	对其他情况的补充说明

执行测试示例

示例 1: “会员登录”界面测试用例执行记录见表 3-5

表 3-5 “会员登录”界面测试用例执行记录

测试设计									测试记录					
Case ID	界面测试类型	子分类	测试项	检查内容	操作步骤	测试数据	优先级	正确结果	测试结果	实施担当	实施日期	测试版本	对应 Bug ID	备注
UI_P2_1	控件	EditBox	用户名	默认值	确 认 默 认 状态	—	高	[用户名] 为空	○	李文	2009-12-10	V1.0		
UI_P2_2	控件	EditBox	用户名	必须输入项为空	(1) 在 [用户名] 中不进行输入 (2) 单击 [登录]	—	高	[用户名] 不允许为空, 错误信息提示	○	李文	2009-12-10	V1.0		
...														

示例 2: “会员登录”功能测试用例执行记录见表 3-6

3.2 编写缺陷报告

在编写缺陷报告之前,首先总结一下软件缺陷的生命周期与状态。

软件缺陷的生命周期大致分为四个阶段,每个阶段又有不同的状态,所以一个缺陷随着生命周期的进行会存在多个不同的状态,缺陷报告的状态是与缺陷保持一致的。软件缺陷及缺陷报告的生命周期及周期内的状态见图 3-1。

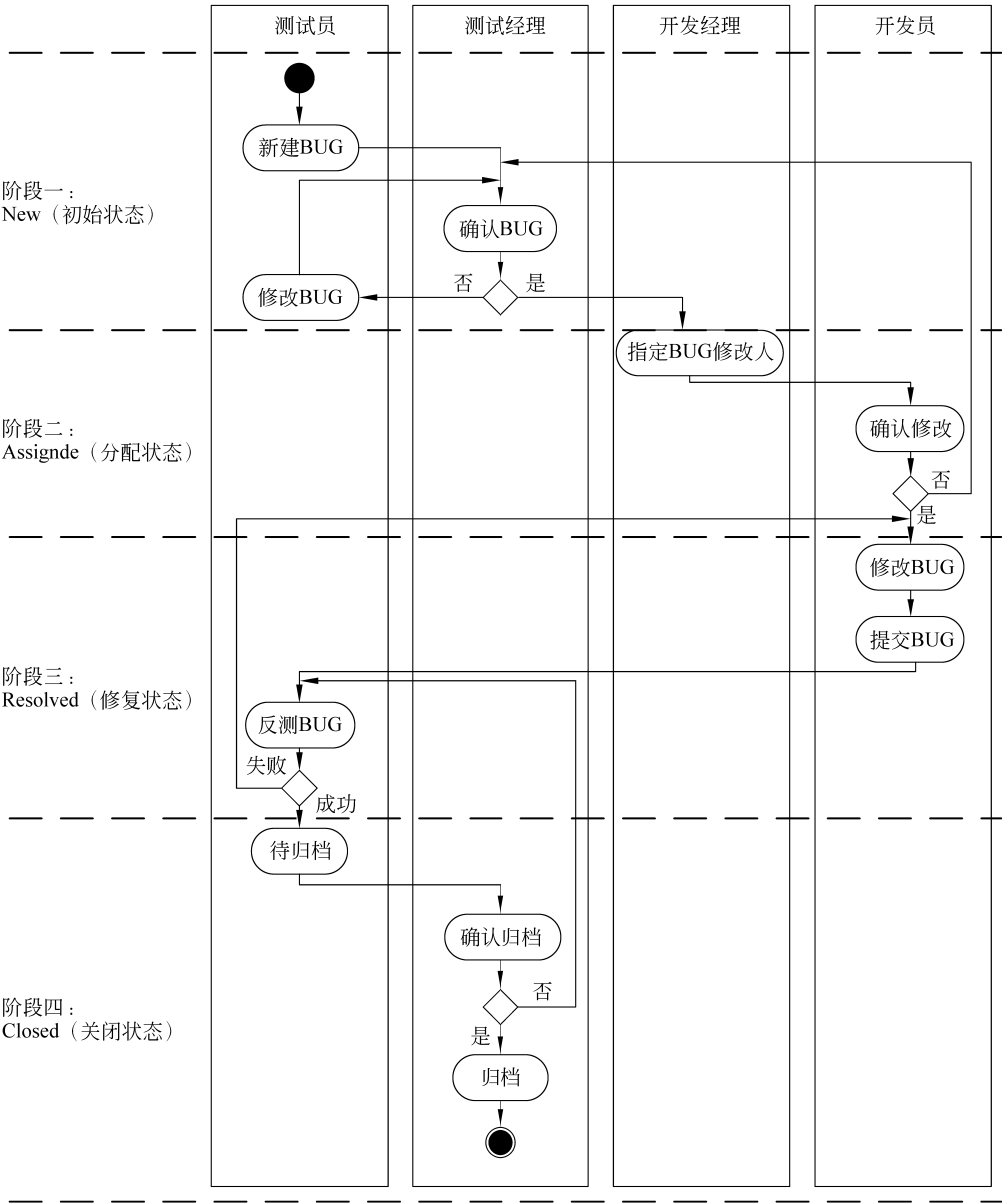


图 3-1 缺陷生命周期

阶段一,该阶段主要包括新发现的缺陷。当前被测版本发现一个新的缺陷时将其置为初始状态(New)。由测试实施人员提交缺陷报告,测试经理确认缺陷报告。经过测试经理确认的缺陷进入阶段二。如果确认时发现该缺陷报告未描述清楚或描述错误,则将修改信息传递给测试实施人员进行缺陷报告的修改,修改后该缺陷才能进入阶段二。

阶段二,缺陷进入该阶段后,首先接受缺陷报告的是软件开发经理。软件开发经理将缺陷指定软件修正人员,该阶段缺陷处于分配状态(Assinged)。被指定修复软件缺陷的人员通常是该缺陷部分的开发人员。经过分配状态后缺陷分为两种情况,如果开发人员接受该缺陷并进行修改,此时该缺陷置于确认分配状态(Assigned),如果开发人员不接受该缺陷,他需要将不修改的原因反馈给测试经理,此时该缺陷置于重新分配状态(Reassigned)。

阶段三,开发人员对分配的缺陷进行修改并重新提交到软件下一个版本中,该阶段缺陷置于修复状态(Resolved)。测试人员对被修复的缺陷所在的软件升级版本进行反测。反测工作通常由最初发现该缺陷的测试人员承担。经过第三个阶段后缺陷又会分成两种情况进行处理,如果被修复的缺陷反测成功,则该缺陷置于已经修复状态(Fixed),并且进入第四个阶段,如果反测不成功,测试人员会将不成功信息反馈给修改该缺陷的开发人员进行重新修改。

阶段四,缺陷经过修改后进入关闭状态(Closed),测试人员将缺陷进行提交待归档,测试经理进行再确认。如果验证后已经确认缺陷被解决,则标记该缺陷为关闭状态,如果验证未通过,则将该未通过的信息反馈给测试人员进行重新反测。

其实在软件测试生命周期中,软件缺陷及缺陷报告还包括很多状态,例如重新启用状态(Reopen)、下版本解决(Later)、问题归档状态(WorkFilling)等,我们在上面列举了最基本的几种状态。

对缺陷报告进行管理的工具有很多,例如 MI 公司开发的缺陷管理工具 TestDirector,还有一些开源缺陷管理工具例如 Mantis 与 BugFree,项目组可根据实际情况采用不同工具进行缺陷管理。有关缺陷管理工具的相关信息,在这里不再展开。

编写缺陷报告任务

1. 任务类型定义

编写缺陷报告任务类型见表 3-7。

表 3-7 编写缺陷报告任务

任务				
编写缺陷报告任务类型描述:				
测试作业类型		保存	模 板	备 注
计划任务	业务理解	△		
	编写缺陷报告作成	○	附录 A 中的“7. 缺陷报告模板”	
	编写缺陷报告评审	○		在原缺陷报告基础上追加
	编写缺陷报告修改	○	附录 A 中的“7. 缺陷报告模板”	

续表

测试作业类型		保存	模 板	备 注
日常任务	问题交流	△		
	会议	○	附录 A 中的“1. 会议模板”	
	培训/学习	△		
	邮件联络	△		
	周报	○	附录 A 中的“2. 周报模板”	
	自定义任务 1	△		
	自定义任务 2	△		
	自定义任务 3	△		

任务说明：

(1) “保存”列，○表示必须、△表示可选；

(2) 计划任务“编写缺陷报告作成”及“编写缺陷报告修改”保存为 Word 格式。建议使用缺陷管理工具例如开源 Mantis 或 BugFree 进行缺陷的记录；

(3) 日常任务“会议”，以项目组为单位制成，保存为 Word 格式，项目测试组长负责；

(4) 日常任务“周报”，以个人为单位制成，保存为 Word 格式，个人负责，每周五提交

2. 任务计划安排

每个项目组完成“学创购书网”所有模块的测试执行,将发现的缺陷填写缺陷报告。测试经理负责本项目组工作任务分配,分配形式见表 3-8。

表 3-8 编写缺陷报告任务计划

作业内容	模块 ID	担当者	作业类型	预定作业时间(H)	计划开始日	计划结束日	文档名称
编写缺陷报告	M1～M3	测试员 A	缺陷报告作成				BUG_ 模块 ID_1.0
		测试经理	缺陷报告评审				
		测试员 A	缺陷报告修改				BUG_ 模块 ID_2.0
	M4～M7	测试员 B	缺陷报告作成				BUG_ 模块 ID_1.0
		测试经理	缺陷报告评审				
		测试员 B	缺陷报告修改				BUG_ 模块 ID_2.0
	M8～M12	测试员 C	缺陷报告作成				BUG_ 模块 ID_1.0
		测试经理	缺陷报告评审				
		测试员 C	缺陷报告修改				BUG_ 模块 ID_2.0
	M13～M15	测试员 D	缺陷报告作成				BUG_ 模块 ID_1.0
		测试经理	缺陷报告评审				
		测试员 D	缺陷报告修改				BUG_ 模块 ID_2.0

续表

作业内容	模块 ID	担当者	作业类型	预定作业时间(H)	计划开始日	计划结束日	文档名称
编写缺陷报告	M16~M18	测试员 E	缺陷报告作成				BUG_ 模块 ID_1.0
		测试经理	缺陷报告评审				
		测试员 E	缺陷报告修改				BUG_ 模块 ID_2.0

任务计划说明：

(1) “模块 ID”列请与表 1_2 测试范围中的“模块 ID”列保持一致；

(2) “文档名称”列 BUG 代表为缺陷、1.0 表示作成版本号、2.0 表示修改版本号，模块 ID 和版本号请根据项目实际情况进行调整

3. 任务工作量汇总

以项目组为单位进行工作量汇总，测试经理负责，汇总形式见表 3-9。

表 3-9 编写缺陷报告任务工作量汇总

计划任务工作量汇总(H)——按人

任务类型	测试员 A	测试员 B	测试员 C	测试员 D	测试员 E	合计
业务理解						
编写缺陷报告作成						
编写缺陷报告评审						
编写缺陷报告修改						
合 计						

日常任务工作量汇总(H)——按人

任务类型	测试员 A	测试员 B	测试员 C	测试员 D	测试员 E	合计
问题交流						
会议						
培训/学习						
邮件联络						
个人工作(计划、总结)						
自定义任务 1						
自定义任务 2						
自定义任务 3						
合 计						
本周总工作量						

编写缺陷报告规范

缺陷报告的主体内容包括编号、版本、报告日期、报告人、缺陷分类、功能分类、重现率、严重等级、优先级、状态、概要、详细描述等相关信息，本书就严重等级和详细描述等内容加以说明。

1. 严重等级及优先级

严重级别和处理优先级是有区别的，例如一个严重级别比较低的缺陷有可能是处理级别比较高的，例如一个公司的标识文字上的细微错误或图片上小偏差属于严重级别为“一般”，但是发布时如果不修复会影响公司的形象，所以这个缺陷会作为优先处理级别“高”进行处理。

缺陷的优先级随着项目的发展会发生变化。例如，原来优先级为低的缺陷，随着版本的发布会调整优先级。作为缺陷提交人应对缺陷的状态进行关注与追踪，对不合理的进行调整，并及时与相关人员进行沟通，使缺陷得到合理的处理。

2. 详细描述

详细描述的格式如下：

操作步骤：
(1)
(2)
(3)
⋮
测试结果：…(NG)
预期结果：…

详细描述包含所有重现的必要操作，每个操作描述独占一行，对操作过程描述进行编号。

对操作设定的项目进行具体的描述：

- ① 操作是调用某个菜单项，描述为“[主菜单]→[子菜单]”的形式；
- ② 操作是对 Dialog 中设定值，描述为“[设定项目]=[设定值]”的形式；
- ③ 当发生的错误与特定值有关，需要用具体数据明确说明；
- ④ 如果任何数据都会出现错误，设定值不需要说明；
- ⑤ 采用特别的方式或标记将缺陷体现出来，例如 NG(not good)字样；
- ⑥ 当一个缺陷报告中出现多处与预期结果不符的地方请参考如下格式，但并不是将多个缺陷写在一个缺陷报告中。

操作步骤：
(1)
(2)
(3)