

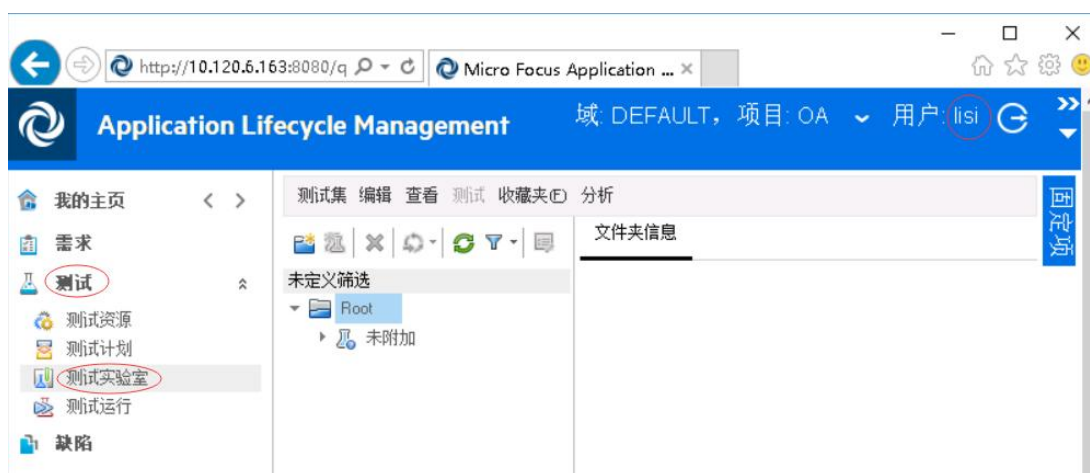
第 6 章 测试用例执行

测试版本搭建完毕，测试组长通知测试工程师进行项目测试。根据测试任务分配，每个测试工程师打开 ALM 进行对应模块的用例执行。

1. 测试集创建

测试工程师 lisi 登录 ALM，根据测试任务创建一个测试集（测试集是单次测试任务中的待执行测试用例的集合。）

(1) 测试工程师“lisi”登录 ALM，单击“测试”→“测试实验室”



(2) “测试集”→“新建文件夹”，在“测试集文件夹名：”中输入测试集的文件夹名称，如此处的“OA 系统”。单击【确定】按钮，完成“OA 系统”测试集文件夹的创建。

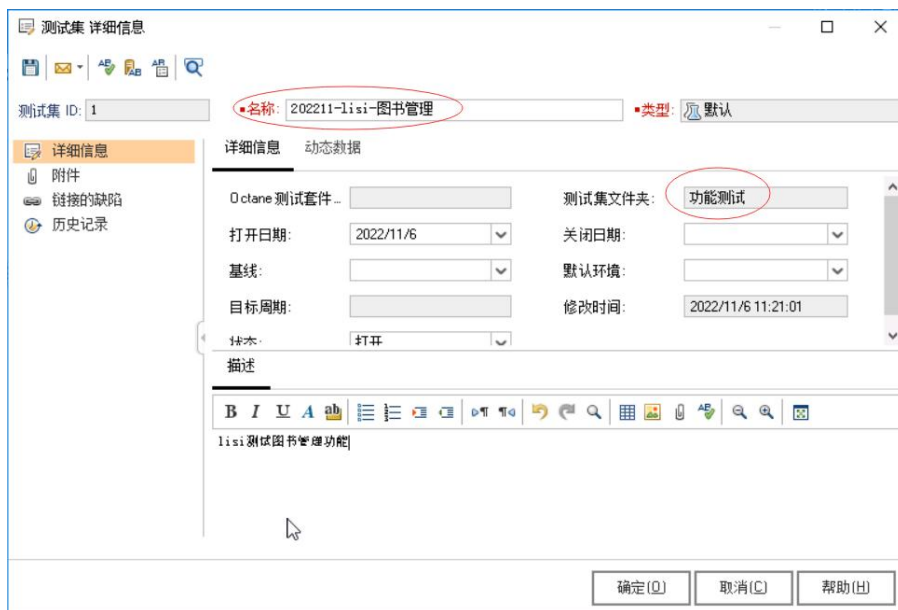


(3) 在“OA 系统”目录下创建“功能测试”，设置测试任务集。

“oa 系统”右键“新建文件夹”并命名为“功能测试”。



“功能测试”右键“新建测试集”并命名为“202211-lisi-图书管理”。

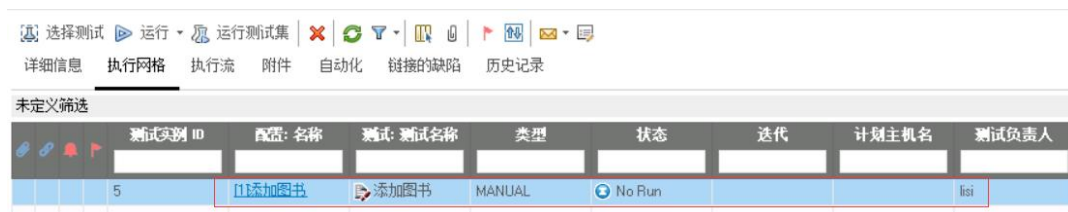


(4) 在测试集中设置待执行的测试用例。

选中“202211-lisi-图书管理”，在右边界面单击“执行网格”选项卡，单击“选择测试”。



在“测试计划树”中选择待测试的用例集合，拖入“执行网格”。



2. 测试集执行

(1) 分析待测用例，提取测试点

执行测试集之前，先执行一般常规测试。

【图书添加功能测试点】

通常情况下，按照“响应时间→标题栏→脚本错误→页面布局→图片→音视频文件→文字→功能”的顺序执行测试。

- ① 响应时间
- ② 标题栏。标题栏应当准确、正确地表述当前页面所需实现的功能或业务。
- ③ 脚本错误
- ④ 页面布局。系统整体风格一致。
- ⑤ 图片。图片是否完整？是否失真？如果是图片链接，则需验证是否实现了跳转功能。
- ⑥ 音视频文件。音视频文件能否正常播放、是否自动播放、是否清晰等。
- ⑦ 文字。
- ⑧ 功能。

(2) 执行测试集

单击菜单栏中的“运行”按钮。“运行名称”默认是当前时间，一般不用修改。“测试者”是测试人，会自动读取当前 ALM 登录账号。

【添加图书功能测试集执行】

手动运行器: 测试集 202211-lisi-图书管理, 测试 [1] 添加图书

开始运行 | 结束运行 | 取消运行 | 操作系统信息

运行详细信息

测试集名称:	202211-lisi-图书管理	测试实例:	[1] 添加图书
测试者:	lisi	运行名称:	Run_11-6_15-13-43
状态:	Not Completed	OS 内部版本号:	Build 14393
变更检测模式:		变更状态:	
操作系统:	Windows Server 2016	操作系统 Serv...	
测试: 测试名称:	添加图书	测试执行 ID:	

注释 | 测试描述

添加注释

测试详细信息

名称: 添加图书

测试详细信息

如果当前测试集有多个测试点，那么可以使用“运行测试集”功能。该功能在测试集中含有多个测试点时，第一个测试点测试结束后不退出，而是直接打开第二个测试点，直至所有测试点测试完成才停止。

(3) 运行测试

单击【开始运行】按钮，执行当前测试点。如果想退出，单击【停止】按钮。



ALM 将当前测试点的所有测试步骤都列出来, 测试工程师只需一边打开测试集, 一边打开被测软件, 一步一步按照测试用例中设计的步骤执行。

(4) 执行测试用例, 设置“通过”或“失败”。

在当前步骤测试通过时, 单击按钮, 将当前步骤设置为“通过”; 如果当前步骤执行失败, 可单击按钮, 将当前步骤设置为“失败”。

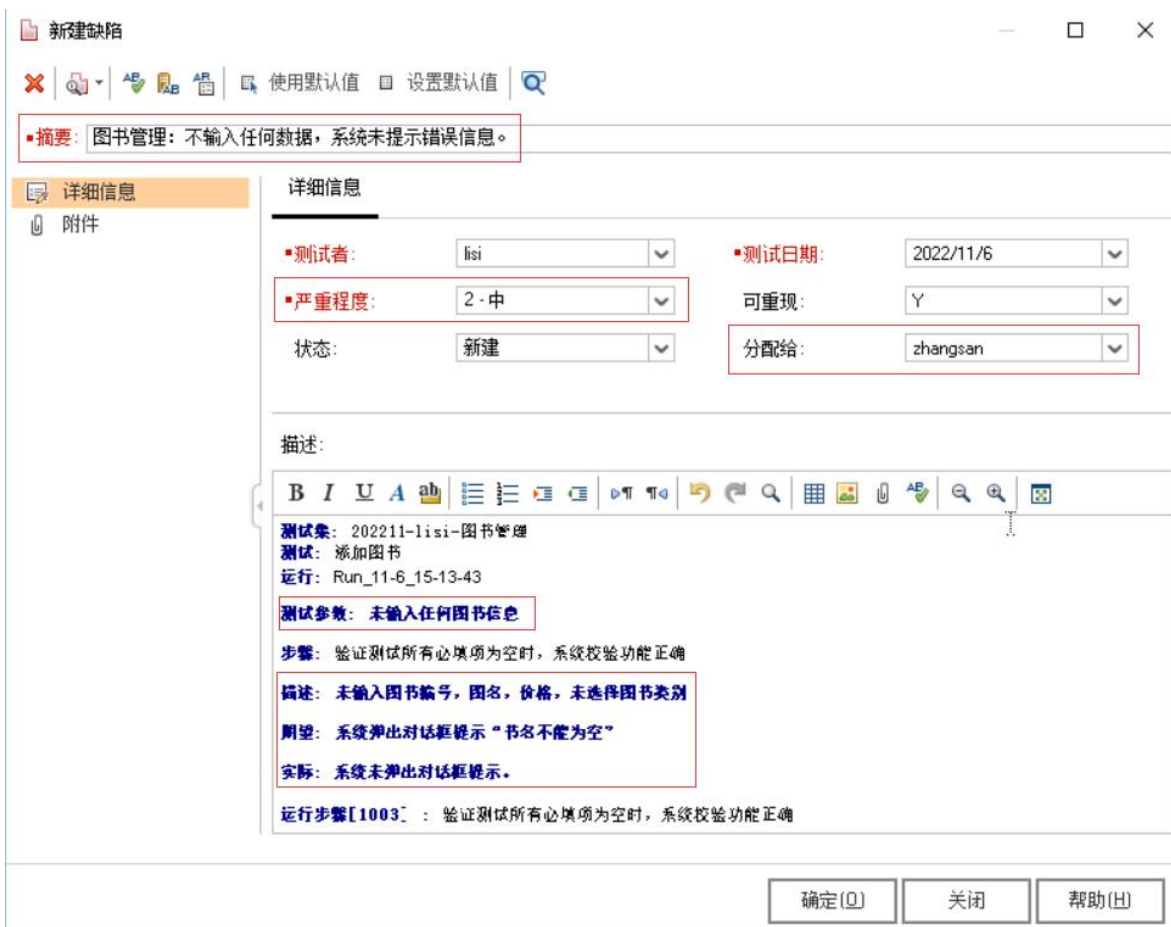


(5) 设置为“失败”的用例，则需添加缺陷。



在“实际结果”中填写与预期结果不相符的结果。单击“新建缺陷”按钮，ALM 将当前步骤中的所有信息默认读过去，只需添加对应的概要信息、严重度、执行的实际结果等即可完成提交缺陷操作。

此时缺陷为“新建”状态。将缺陷流转给 zhangsan（测试组长），测试组长审查缺陷是否规范。



(6) 所有步骤执行检查后，单击【停止】。显示测试完成后的结果界面。

手动运行器: 测试集 202211-lisi-图书管理, 测试 [1] 添加图书

结束运行 (Ctrl+Q)

步骤名	状态	执行日期	执行时间
验证测试所有必填项为空时, 系统校...	Passed	2022/11/6	15:22:28
验证测试所有必填项为空时, 系统校...	Failed	2022/11/6	15:23:45

描述

预期:

实际:

选择测试 运行 运行测试集 详细 自动化 链接的缺陷 历史记录

未定义筛选

测试实例 ID	配置: 名称	测试: 测试名称	类型	状态	迭代	计划主机名	测试负责人	执行日期	时间
5	图书添加图书	添加图书	MANUAL	Failed			lisi	2022/11/6	15:41:28

上次运行报告

步骤名	状态	执行日期	执行时间
验证测试所有...	Passed	2022/11/6	15:22:28
验证测试所有...	Failed	2022/11/6	15:23:45

步骤详细信息

描述:

测试数据: 无

操作步骤:

不输入任何数据, 直接点击“确定”按钮

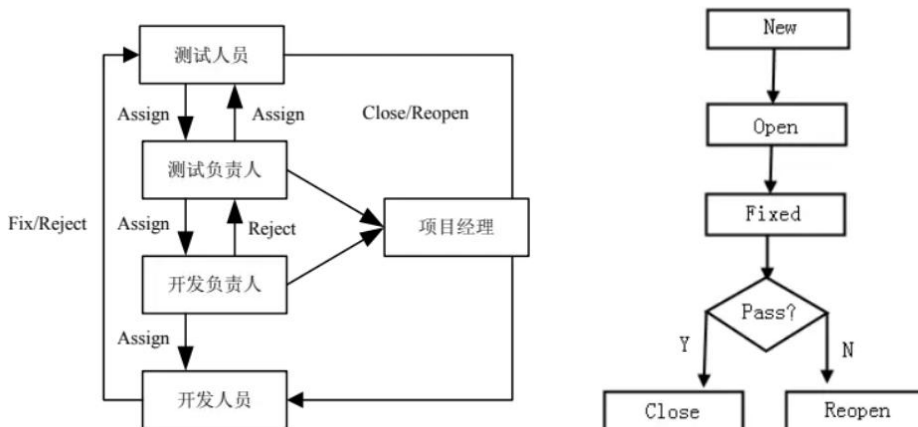
预期为:

系统弹出对话框提示“书名不能为空”

实际为:

3. 缺陷跟踪处理

缺陷管理流程和缺陷状态转换图, 如下:



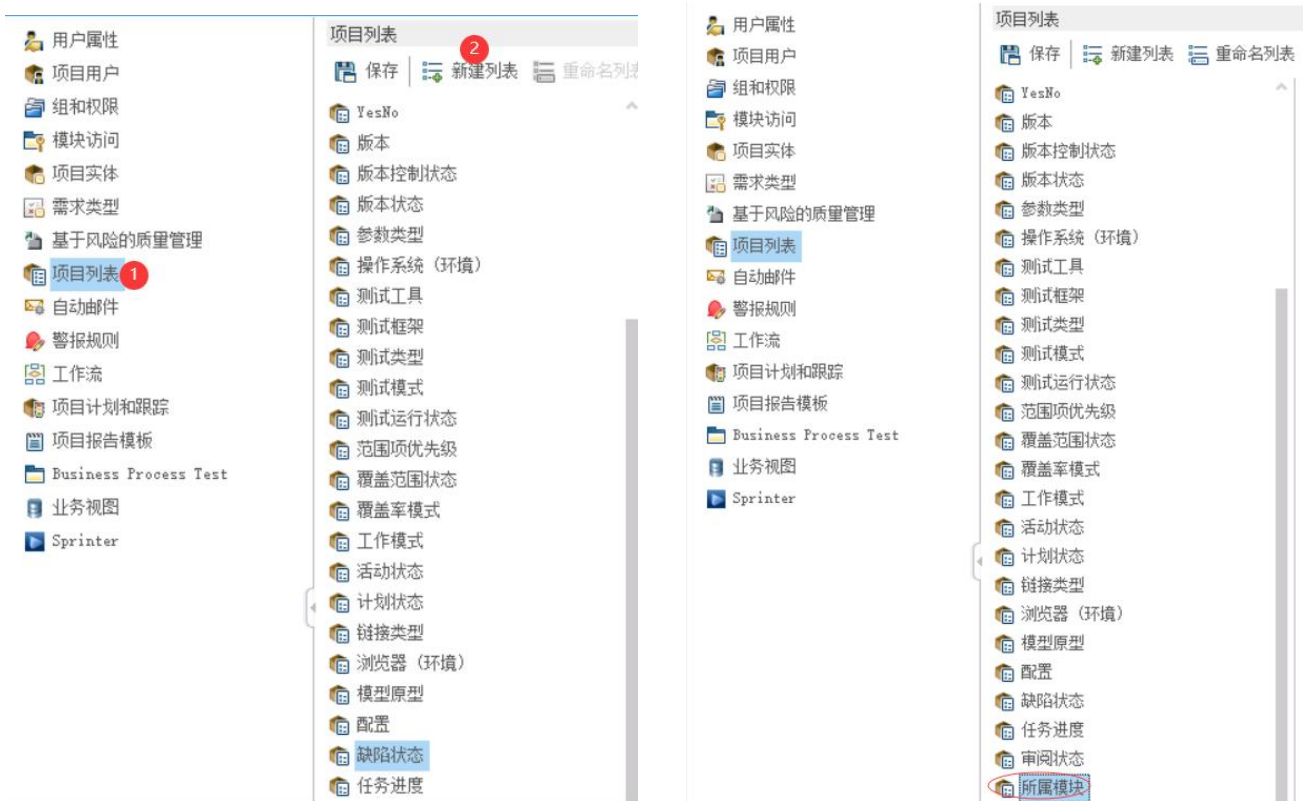
为了更好地分析管理缺陷, ALM 项目管理员 (或项目管理员) 需在“项目自定义”功能中给“缺陷”模块添加一个缺陷“所属模块”自定义字段, 用此字段来标识缺陷的所属模块, 便于在测试结果分析时统计出各个功能模块的缺陷情况。

(1) 缺陷管理流程自定义字段实现

项目管理员 zhangsan 登录，“自定义”：

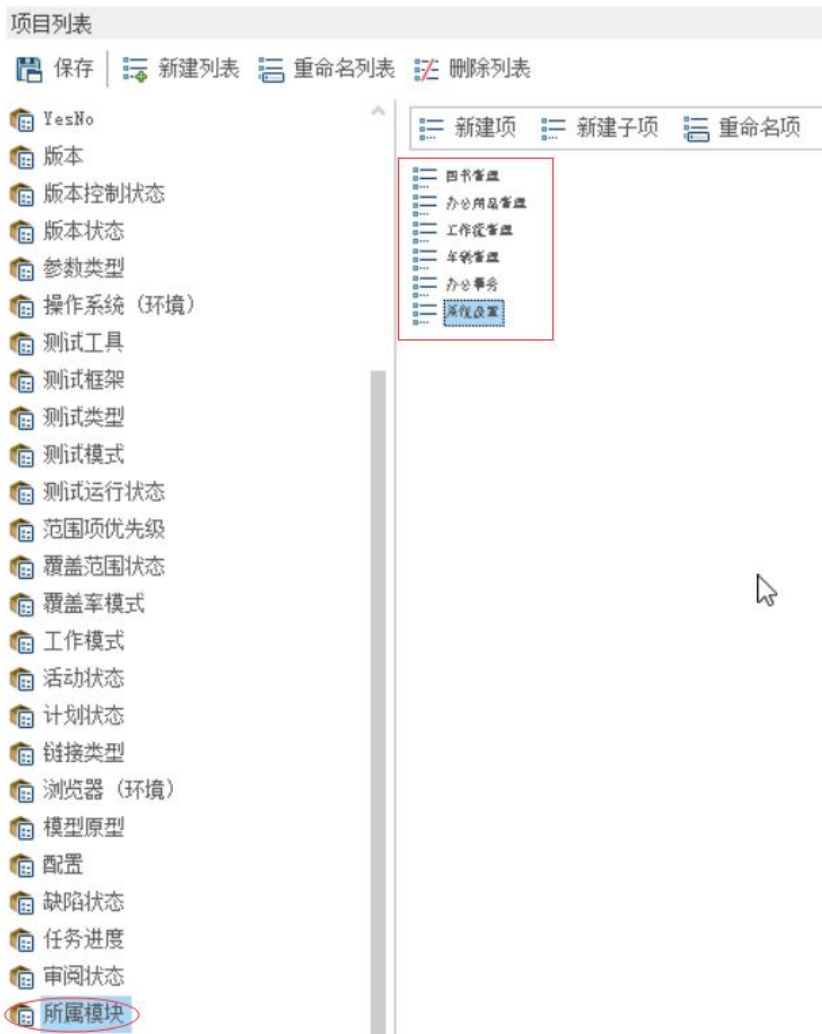


“项目列表”->“新建列表”，输入列表名称“所属模块”，新建成功。



选中“所属模块”，“新建项”。根据测试任务重的功能划分，依次创建“所属模块”的列表值：图书管理、办公用品管理、 workflow 管理、车辆管理、办公事务、系统设置。

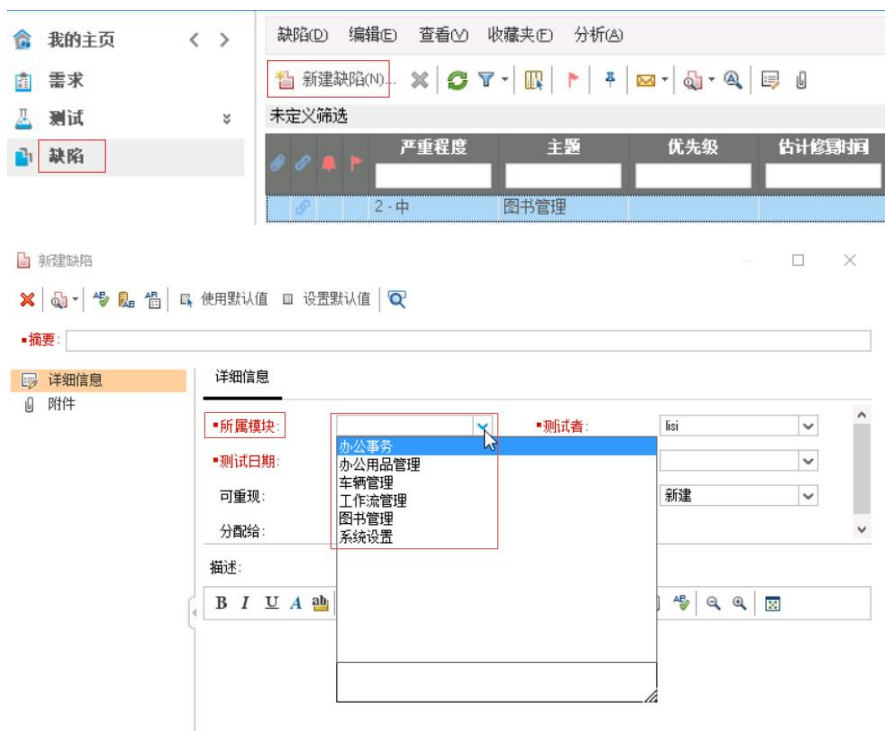
保存，完成“所属模块”项目列表的创新活动。



“项目实体”，到缺陷模块自定义字段设计页面。项目实体->缺陷->用户字段->“新建字段”：

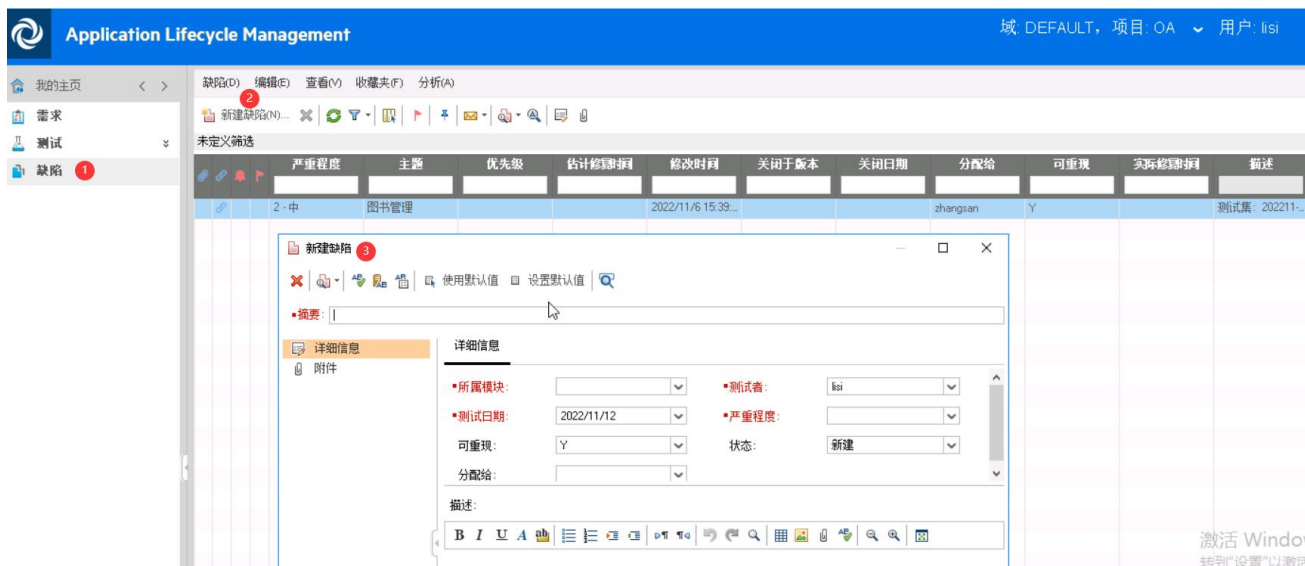


测试工程师 lisi 登录系统，查看“缺陷”模块，“新建缺陷”，将会出现项目管理员 zhangsan 添加的用户自定义字段“所属模块”。

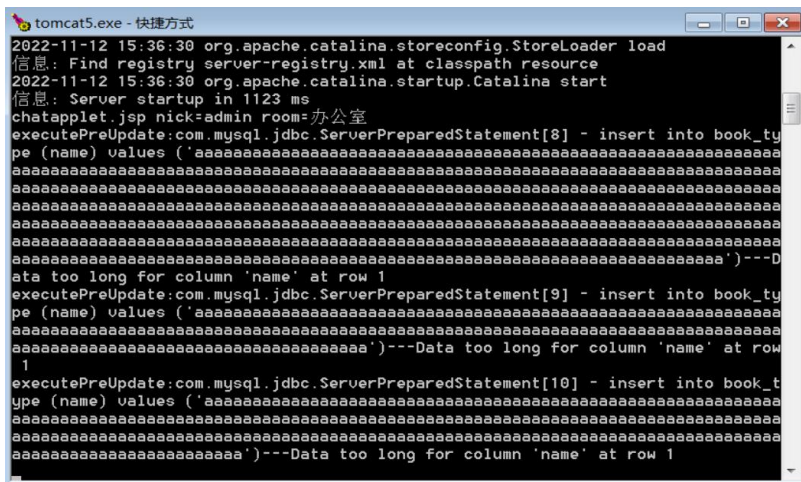


(2) 测试工程师提交缺陷

测试工程师 lisi 登录系统，进入“缺陷”页面单击【新建缺陷】按钮提交缺陷，打开添加缺陷的界面。



测试工程师 lisi 执行“图书管理”功能模块处的用例时，发现在图书类别添加，输入超过 150 个字符的类别名称时，Tomcat 的控制平台会报出 SQL 语句错误，而页面则会提示“数据库操作失败！”。对于用户而言，“数据库操作失败”是个相对专业的表述。上述表述改为“图书类别名称不能超过 100 个字符”，用户更容易理解其含义。



李四提交一个名为“图书类别添加功能处，输入超过 150 个字符的类别名称，Tomcat 控制平台报出 SQL 异常”的缺陷，此时该缺陷的“状态”为“新建”，“严重程度”为“中级”，具体缺陷信息：



确认无误后，单击【提交】按钮，提交该缺陷，并将其分配给测试组长 zhangsan。
缺陷列表显示刚才新建的缺陷。

缺陷(D) 编辑(E) 查看(V) 收藏夹(F) 分析(A)									
新建缺陷(N)...									
筛选: 分配给(wangwu)									
严重程度	主题	优先级	估计修复时间	修改时间	关于版本	关闭日期	分配给	可重现	
2 - 中	图书管理			2022/11/12 23:5...			wangwu	Y	
2 - 中				2022/11/12 23:5...			wangwu	Y	

开发组长逐个处理缺陷，将缺陷按模块分发给开发工程师（如 maliu）。如果开发组长不认为是一个缺陷，或者觉得不理解的时候，就加上“注释”指派给测试组长。

严重程度	主题	优先级	估计修复时间	修改时间	关于版本	关闭日期	分配给
2 - 中	图书管理			2022/11/13 0:08...			maliu
2 - 中				2022/11/13 0:08...			maliu

(5) 开发工程师处理缺陷

开发工程师需处理分配给自己的缺陷。开发工程师 maliu 进入 ALM 后过滤指派给自己的缺陷。

缺陷(D) 编辑(E) 查看(V) 收藏夹(F) 分析(A)									
新建缺陷(N)...									
筛选: 分配给(maliu)									
严重程度	主题	优先级	估计修复时间	修改时间	关于版本	关闭日期	分配给	可重现	
2 - 中	图书管理			2022/11/13 0:08...			maliu	Y	
2 - 中				2022/11/13 0:08...			maliu	Y	

根据缺陷的描述，开发工程师进行相关代码的修改，当修复完成后，需将对应的缺陷的状态改为“修复”，表示这个缺陷已经修改了，测试工程师可在下一个测试版本校验。

缺陷 ID: 1

摘要: 图书管理: 不输入任何数据，系统未提示错误信息。

详细信息

附件

链接的实体

历史记录

分配给: lisi

关闭日期:

计划关闭版本:

检测于发布:

检测于周期:

目标发布:

实际修复时间:

修改时间: 2022/11/13 0:08:46

主题: 图书管理

估计修复时间:

关于版本:

检测于版本:

检测于环境:

可重现: Y

目标周期:

项目:

优先级:

状态: 已修正

描述:

测试集: 202211-lisi-图书管理

测试: 添加图书

运行: Run_11-6_15-13-43

注释:

添加注释

确定(O)

取消(C)

帮助(H)

开发工程师如果不认为这是一个缺陷，可加上备注，指派给测试工程师，说明他不修改的理由。

一旦在测试过程中对缺陷的定义有争议，就需要根据约定召开项目组缺陷评审会议，对被“拒绝”的缺陷进行处理。

4. 回归测试

回归测试是对已被测过的程序在修复缺陷后进行的重复测试，以发现在这些变更后是否有新的缺陷引入或被屏蔽。当软件发生变更或者应用软件的环境发生变化时，需要进行回归测试。

回归测试套件一般都会执行多次，而且通常很少有变动，因此将回归测试自动化是很好的选择。

回归测试通常有完全回归和选择性回归测试两种策略。对于任何一个项目，前三轮测试版本迭代过程中，都建议使用完全回归测试策略，将所有测试用例全部回归。而被测对象是升级或者维护性的版本变化，则可采用选择性回归策略实施。

回归测试流程：确认缺陷是否修复和执行用例回归测试。

(1) 确认缺陷是否修复

测试工程师提交的缺陷，经过开发工程师处理，如果确实是缺陷，并且已经修复，则测试工程师需在下一个版本上确认缺陷是否已经修复完成，这个过程一般称为缺陷校验。

对于状态是“拒绝”的缺陷，测试工程师应当确认开发工程师拒绝的理由是否成立，如果不成立，则需重新打开缺陷，如果成立，则关闭缺陷。

(2) 执行用例回归测试

校验缺陷活动结束后，测试工程师根据测试任务分配进行执行用例活动，重新开展测试活动。

5. 功能测试报告输出

测试工作完成后，测试组长输出当前测试对象的测试报告，对被测对象的缺陷进行分析，反映被测软件的质量，以便于项目组决定项目是否上线或者发布。

(1) 缺陷修复率统计

缺陷修复率=校验通过关闭缺陷数/总缺陷数

在 ALM 的“缺陷”中默认没有提供标识是否是校验的缺陷，所以需要测试工程师自定义一个字段来标识。

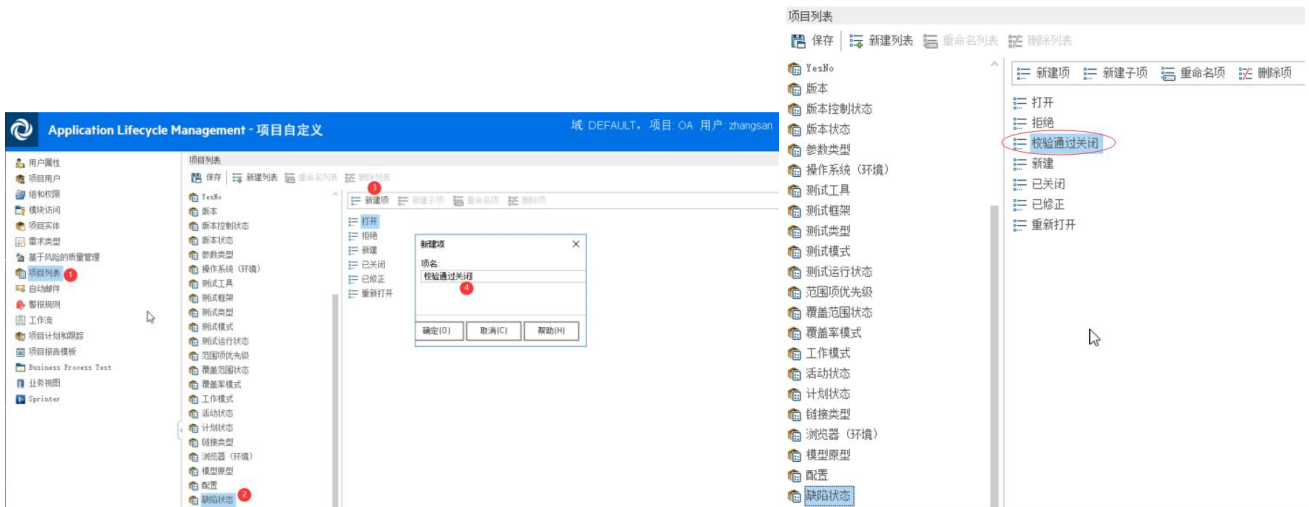
利用用户自定义字段维护方法，创建一个标识缺陷属于校验通过关闭的状态“**校验状态**”，以此字段来统计缺陷修复率。

项目管理员 zhangsan 登录，“自定义”：

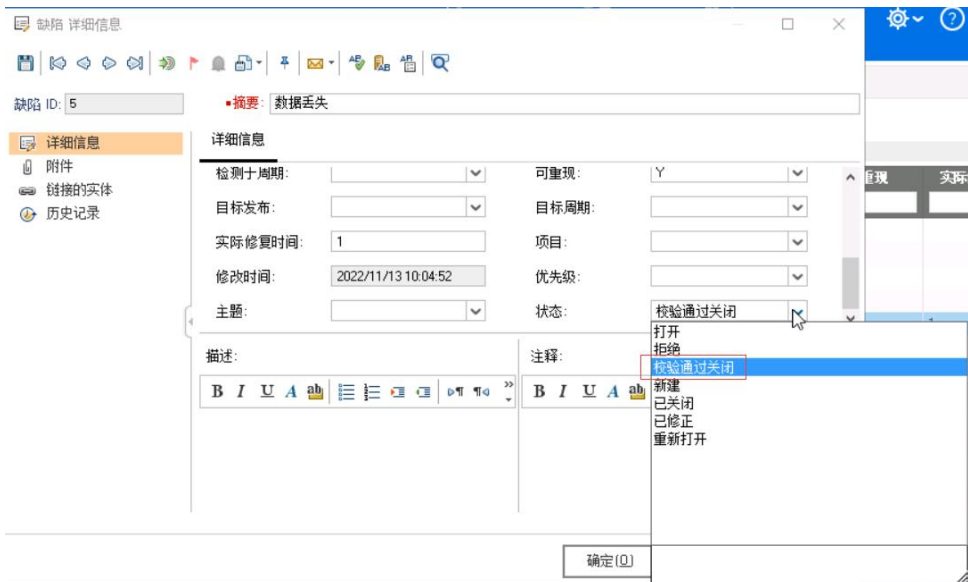


“项目列表”->“缺陷状态”，新建缺陷状态项“校验通过关闭”。

保存，新建成功，缺陷状态增加“校验通过关闭”项。

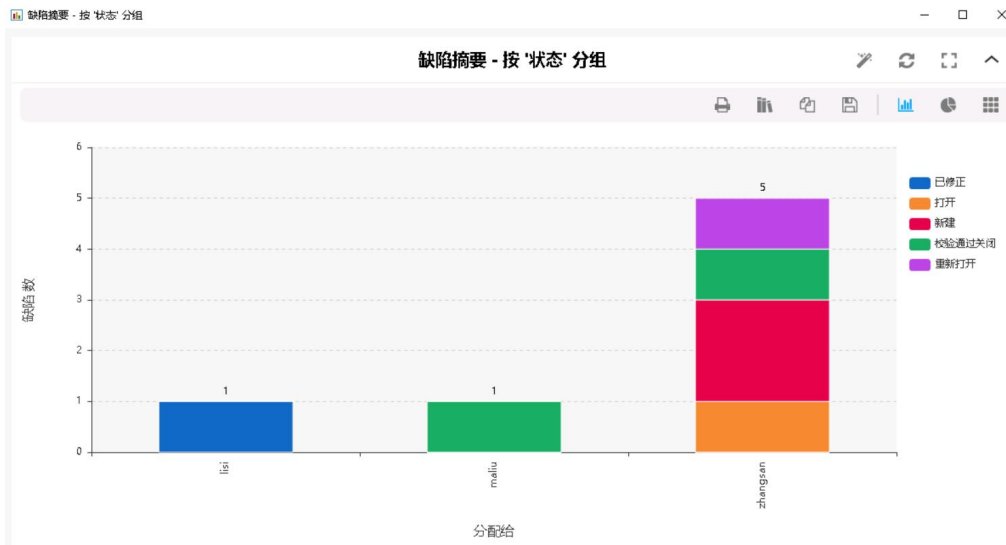


更新缺陷状态，将校验通过的缺陷状态置为“校验通过关闭”。

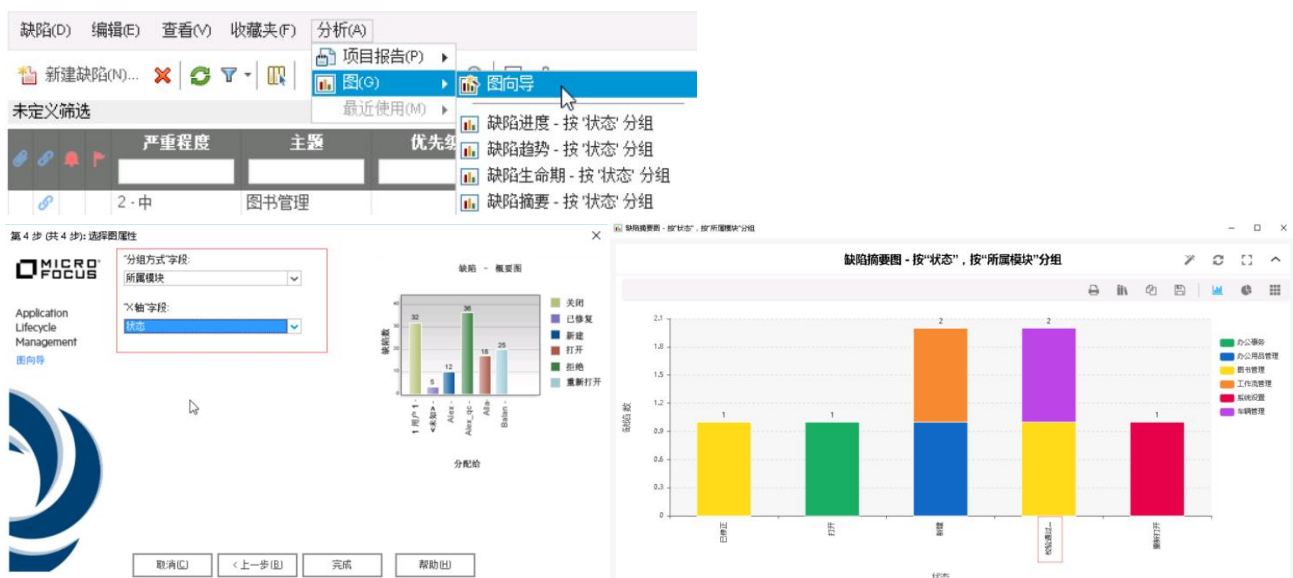


添加成功后，单击“缺陷”→“分析”→“图”→“缺陷概要-按‘状态’分组”命令：





根据缺陷所属模块，统计缺陷：分析-图-图向导。第4步，分组方式字段选择“所属模块”，X轴字段选择“状态”，完成，生成统计图。



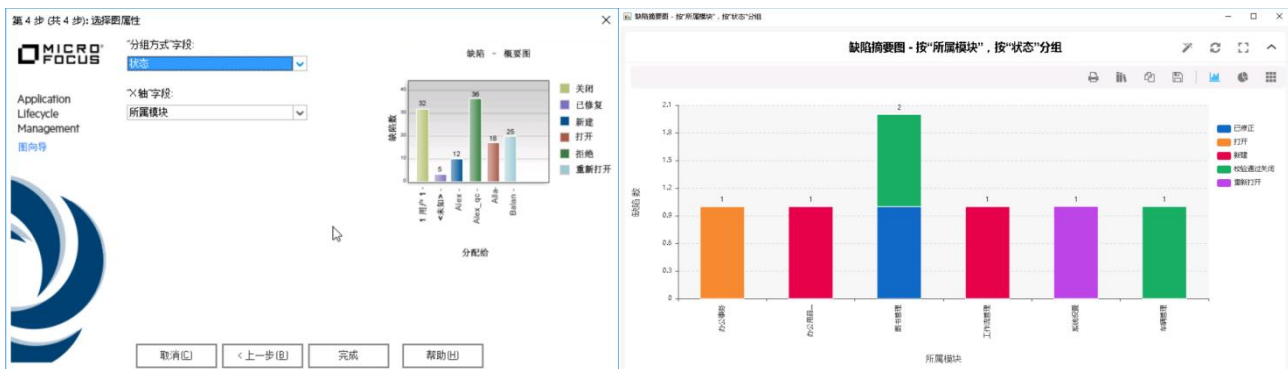
从图可以看出，“校验通过关闭”的缺陷有2个，当前缺陷的修复率就是：

缺陷修复率=校验通过关闭缺陷数（2个）/总缺陷数（7个） $\approx 28.6\%$

(2) 缺陷分布情况

软件测试工作中，需要注意**缺陷群集现象**。当某个模块发现很多的缺陷时，那么就有可能在该模块发现更多的缺陷。缺陷分布情况主要描述项目中缺陷的分布位置，各个模块都有多少缺陷，严重度如何。根据缺陷的分布情况，找出项目问题比较严重的功能模块，有针对性地加强测试。

按缺陷所属模块进行统计：分析-图-图向导。第4步，分组方式字段选择“状态”，X轴字段选择“所属模块”，完成，生成统计图。

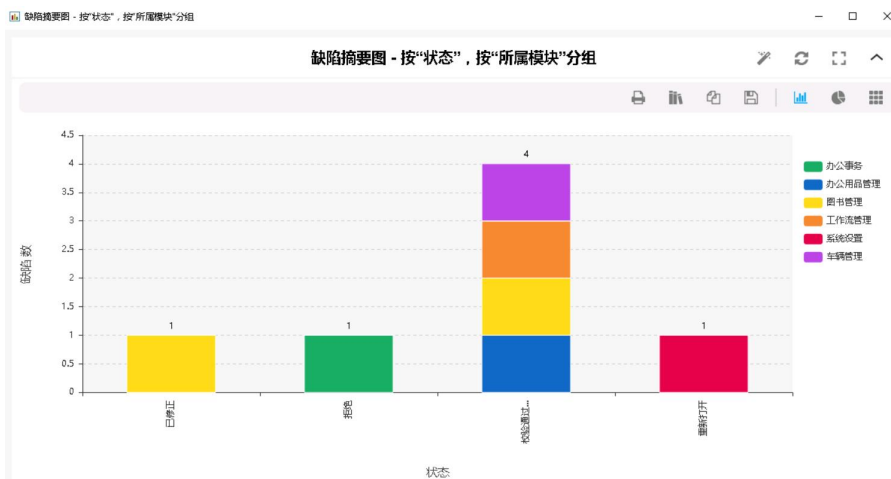


由图可以看出各个模块的缺陷数及当前缺陷所处状态。

(3) 当前遗留缺陷

在测试工作结束时，非关闭状态的缺陷，都需要分析原因。

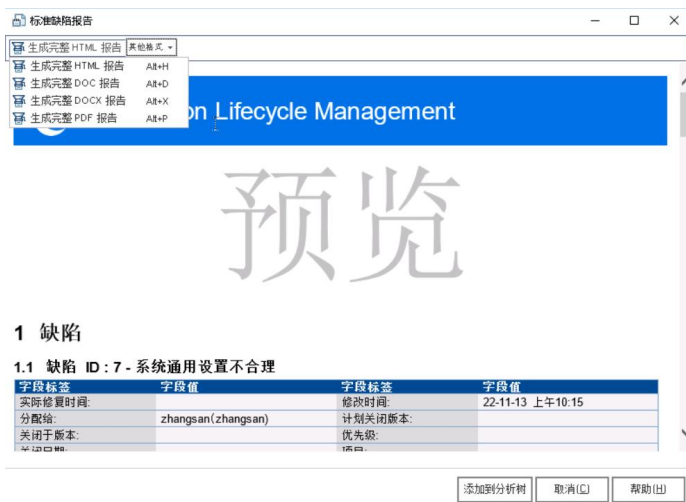
按状态，按所属模块分组，统计结果如下图。对于“已修正”的缺陷，应确认修复结果；“拒绝”缺陷应关闭；“重新打开”的缺陷需继续跟踪。



(4) 缺陷报告

分析-项目报告-标准缺陷报告，缺陷报告可以 HTML、DOC、PDF 格式导出。





输出的缺陷报告，包括缺陷解决情况及历史记录详情。



通过图形、文字描述最后输出项目质量评价。