

# 技 术 文 件

技术文件名称：测试工作规范

技术文件编号：

版 本：V1.0.0

共 15 页

(包括封面)

拟 制 邹年奎

审 核 \_\_\_\_\_

会 签 \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

标准化 \_\_\_\_\_

批 准 \_\_\_\_\_

成都国信安信息产业基地有限公司

修订记录

文件编号	版本号	拟制人/ 修改人	拟制/修改日期	更改理由	主要更改内容 (写要点即可)
	V1.0.0	邹年奎	2014-4-23	无	首次编写
<p>注 1：每次更改归档文件（指归档到事业部或公司档案室的文件）时，需填写此表。</p> <p>注 2：文件第一次归档时，“更改理由”、“主要更改内容”栏写“首次编写”。</p>					

# 1 概述

## 1.1 编写目的

该文档主要对测试团队测试活动进行了规范，使我们的产品测试活动更加科学有序的进行，旨在提高产品的测试质量。

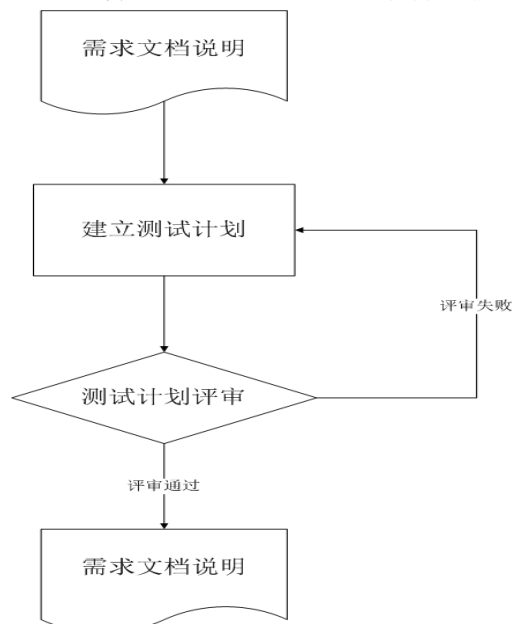
## 1.2 测试职责

测试是项目开发过程中的重要组成部分，肩负着如下责任：

1. 在项目的需求文档确立基线前对文档进行测试，并从用户体验和测试角度提出自己的看法。
2. 制定合理的测试计划，并与项目整体计划有机地整合在一起。
3. 科学的方法设计需求覆盖率高的测试用例。
4. 认真负责地实施测试工作，提交缺陷并协助开发重现与定位问题。
5. 针对缺陷进行跟踪与维护，及时反馈测试情况。
6. 提交项目测试报告，总结测试过程并持续优化。

# 2 制定测试计划

测试计划的目的是尽早的明确测试工作的内容（范围）、测试工作的方法以及测试工作所需要的各种资源，并把这些信息发布给所有涉及到测试工作的涉众，尽快将下一步测试工作需要考虑的问题和准备的条件落实下来。



测试计划主要包括测试需求及优先级、测试进度安排、测试资源情况、测试风险、测试策略（如功能测试、用户界面测试、性能测试等测试策略）、测试输出等内容。测试文档编写可按照《测试计划模版》撰写。

## 3 用例编写

### 3.1 设计策略

测试用例编写策略是指组织和编写有效的测试用例的方法和技巧。

策略从测试内容角度可以分为流程用例和功能点用例。其中流程用例指针对业务流程编写的测试用例，通常采用场景法。功能点用例指针对具体功能点编写的测试用例，可以采用等价类划分、边界值法、因果图等方法。

根据测试的策略又可以分为通过测试用例和失败测试用例，通过测试用例主要验证需求是否可以实现，一般采用等价类划分等测试方法。失败用例的编写主要为了尽可能多的发现缺陷，一般采用错误推测法、边界值分析法等测试方法。

①对于业务流程比较重要的系统，首先要考虑用场景法编写流程用例，要求覆盖所有的基本流和备选流。其次需要编写功能点测试用例，要求覆盖所有的需求，保证需求的各个功能都能正常的实现。我们首先要考虑通过测试用例，来证明软件可以满足需求；再使用失败测试用例，来尽可能多的发现缺陷，保证软件具有一定的容错和安全能力。

②在测试用例的编写过程中需注意其详细程度，要注意执行此测试用例的人员。

③覆盖功能点不是指列出功能点，而是要写出功能点的各个方面，如果组合情况较多时可以采用等价类划分的方法。

### 3.2 测试用例设计原则

#### 1. 全面性

①应尽可能覆盖程序中的各种路径（要覆盖所有的需求，所有的流程和功能都应有相对应的测试用例）。

②应考虑存在跨年、跨月的数据。

③大量数据并发测试的准备。

#### 2. 正确性

①输入界面的数据应与测试文档所记录的数据一致。（测试执行时，要按照测试用例提供的数据执行）

②预期的测试结果应与需求设计产生的业务和结果吻合。（就是测试的预期结果不能违背或超出需求设计的要求）

#### 3. 符合正常业务惯例

①测试用例应符合用户实际工作业务流程。

（测试用例的设计要参考《需求规格说明书》和《概要设计说明书》）

②兼顾各种业务变化的可能。（设计的测试用例要有可变化性，即要定期维护）

#### 4. 系统性

①对于系统业务流程有一个完整、正确的说明，包括系统的各组织结构（子系统、模块）相互间的关系，对于相互间有联系的子系统的业务关系的描述一定要清晰、直观。

②模块业务流程要清晰描述各模块内部功能、它们相互间的联系；若有模块功能类似，应对其进行区分。

#### 5. 仿真性

人名、地名、电话号码等应具有模拟功能，符合一般的命名惯例。

#### 6. 可操作性

①测试用例中应尽可能的写清测试的内容和必要操作要求，以便测试人员能更好理解和测试操作。

②操作性的另外一方面是界面友好性，例如信息中的信息漏填，系统的提示是否明确（指出是具体哪项未填写），光标是否定位到相应输入项。

#### 7. 容错性（健壮性）

①容错性测试就是测试系统是否容易崩溃或瘫痪；

②程序能够接收正确数据输入并且产生正确（预期）的输出，输入非法数据（非法类型、不符合要求的数据、溢出数据等），程序应能给出提示并进行相应处理。把自己想象成一名对系统操作一点也不懂的客户，进行任意操作。

#### 8. 接口（连贯性）

①试各个模块相互间的协调和通信情况，数据输入输出的一致性和正确性。  
（这个在系统测试中是非常重要的环节，大多数比较严重的BUG往往是在模块间的接口和交互上的，一定要全面的测试它们之间的协调和通信）

②于系统业务流程来说，各个子系统之间是如何连接在一起，如果需要接口，各个子系统之间是否有正确的接口。

③若是依靠页面链接，页面链接是否正确；子系统间是否有制约（加锁、解锁），如果有制约，它们之间的关系要描述清楚。

④于模块业务流程来说，同级模块以及上下级模块是如何构成一个子系统，其内部功能接口是否连贯，接口间的交互是否正常。

#### 9. 数据库

数据库设计规范对软件系统的数据库结构、数据表及其之间的数据调用关系进行测试。（主要是测试数据是否正确、及时的导入/导出数据库，数据表及其之间的数据调用关系等，比如在界面显示的信息，是否与数据库中的信息一致。对信息的修改是否及时的导入数据库等。对于取消批改的信息，该

信息在数据库中的信息是否正确、及时回滚)

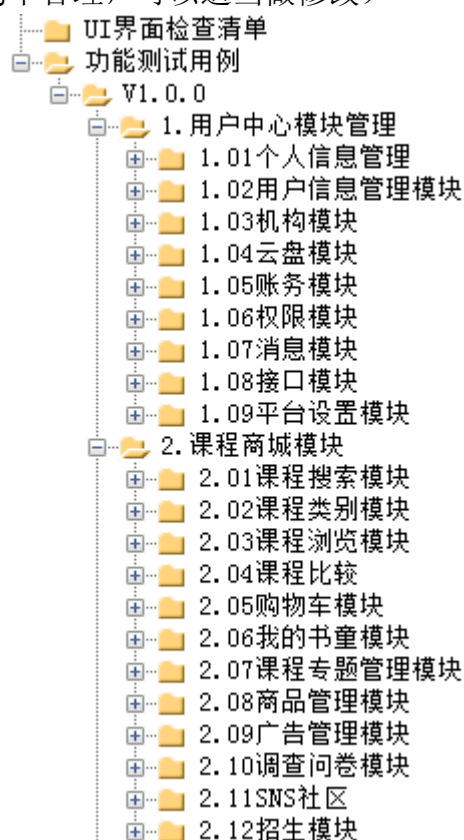
## 10. 可移植性

在不同操作系统及硬件配置情况下的运行性。(例: 把系统安装在 windows 2003/win7 的英文版、中文的操作系统下进行测试, 检验系统的所有功能是否能正常的运行, 系统性能是否稳定等)

## 3.3 QC 用例编写规范

测试用例要包括欲测试的功能、应输入的数据和预期的输出结果。测试数据应该选用少量、高效的测试数据进行尽可能完备的测试; 基本目标是: 设计一组发现某个错误或某类错误的测试数据。

1. 界面测试和功能测试用例分开编写。界面测试可以根据需求和行业规范列出 checklist, 不单独针对每个页面进行用例编写。
2. 如果需求是增量的, 应该对用例进行版本规划。起始编号为 V1.0.0, 具体版本号可以根据需求制定。
3. 根据需求文档结构视图, 采用树形结构进行对用例的编写。(如果需求模块结构不合理, 可以适当做修改)



4. 注意命名前面加上编号, 如果下级编号大于 10 的, 用“01”作为起始编号, 如: 2.01。(同样也适用于用例步骤编号)

5. 每条测试用例对应系统的一个流程，用例名称应该简洁易懂。
6. 每条测试用例的目的，测试等级和预置条件等信息统一填写到详细信息里。  
(测试等级与相应需求的优先级对应)

详细信息

设计步骤

测试脚本

附件

需求覆盖

链接的缺陷

\* 测试名称:

2.3.1用户购买学币课程，它自

创建日期:

2012/4/23

设计者:

邹年奎

状态:

Design

描述:

测试目的: 用户购买学币课程，它的一级、二级、三级上线提取佣金比例正确。

测试等级: 高

预置条件:

1、网络运行正常。

2、学云网测试环境运行正常。

3、推荐功能已经正确部署，运行正常。

4、存在四个测试用户user1、user2、user3，user4: user2为用户1的一级下线，user3为用户2的一级下线，user4为用户3的二级下线，user4账户中有足够的学币进行课程购买，且

7. 用例尽量根据实际情况，按照最高执行效率进行排版；测试用例中的每个步骤: 不能出现二义性，仅是一个可执行的步骤，每个步骤对应一条预期结果。

步骤名称	描述	预期结果
case 1	1、打开企业学习平台首页。 2、点击“免费注册”按钮。 3、在注册页面输入合法（满足产品需求中字段约束条件）的帐号、昵称、密码、确认密码、用户邮箱和用户手机，并勾选上“我已经阅读并同意《大家来学习注册服务协议》”，点击“注册”按钮。 4、登录注册邮箱，点击发送的帐号激活链接。 5、在首页，输入刚注册的帐号、密码，以及正确的验证码后，点击登录。	1、成功打开企业学习平台首页，页面展示正确。 2、跳转到用户注册页面。 3、注册成功，进入到邮箱注册激活页面。 4、登入到个人邮箱，点击链接后成功激活帐号。 5、成功登录，进入到企业学习平台云桌面。
case 2	1、打开企业学习平台首页。 2、点击“免费注册”按钮。 3、在注册页面输入合法（满足产品需求中字段约束条件）的帐号、昵称、密码、确认密码、用户邮箱和用户手机，并勾选上“我已经阅读并同意《大家来学习注册服务协议》”，点击“注册”按钮。 4、不激活邮箱，进入首页，输入刚注册的帐号、密码，以及正确的验证码后，点击登录。	1、成功打开企业学习平台首页，页面展示正确。 2、跳转到用户注册页面。 3、注册成功，进入到邮箱注册激活页面。 4、登录失败，提示帐号为激活，并跳转到邮箱激活页面。

8. 如果用例间存在关联的（如，前一个用例执行成功是后一个用例执行的前置条件），把影响后面执行的用例放在前面。
9. 针对每个测试点，建议常规用例（以边界值分析、等价类划分、正交分解法等编写的用例）放在前面，非常规用例（仅指错误猜测法编写的用例）放在后面。
10. 用例编写应采用书面语，文字语言简练易懂，避免出现错别字，病句或逻辑错误。
11. 涉及到增加、删除或修改等用例时，应该增加一条验证步骤。

1、打开学习平台首页。

2、在登录框中点击“忘记登录密码？”链接。

3、在申请密码找回页面，输入正确的帐号和相应的注册邮箱地址，然后提交。

4、登录注册邮箱，查看返回的随机密码。

5、再次进入到学习平台首页，输入正确的帐号和获得的随机密码，然后登录。

12. 用例应该动态维护，如在测试执行中发现新的测试点，应及时添加到用例中去。

## 4 测试执行

### 4.1 获取并部署测试版本

由开发组（部）发起测试申请，并给出测试包的获取地址。测试包应该包括：测试版本及内容说明、程序安装包、测试部署手册、数据库脚本等工件（具体内容应根据项目实施规范而定）。

获取安装包后检查所含工件是否完备、正确，不完备则根据情况打回；完备则根据部署手册进行测试环境的搭建。环境搭建注意事项：

1. 测试环境应尽可能接近真实的用户部署环境。
2. 测试环境中必须保证没有安装开发调试环境，避免影响最终测试结果。
3. 由测试人员依照部署手册来完成，不得由开发人员代为部署。
4. 部署手册中出现的问题也应一并提为BUG，并于下个版本回归测试。
5. 测试环境需设定登录口令，不允许开发人员对部署版本单独替换/修改。
6. 项目测试环境口令需汇总至测试组长（经理）进行统一管理。

### 4.2 测试执行原则

1. 测试前做好环境备份（包括数据库、程序源文件），以便测试出现故障时还原环境。
2. 要明确该次的测试目的，自己要测试内容以及对应的测试依据。
3. 测试首先要依据测试用例执行，然后在此后再进行错误推测法进行测试。
4. 把事实与推测分开，总是用实际的证据来证明你的推测。
5. 偶然性出现的 BUG 一定要反馈到缺陷中去，有图的应尽量截图，无法截图的应在再次出现该 BUG 时立即联系开发人员进行确认。
6. 测试执行应分优先级，应优先保证关键功能和测试内容的充分测试。（在时间不充分情况下，该原则尤为重要）

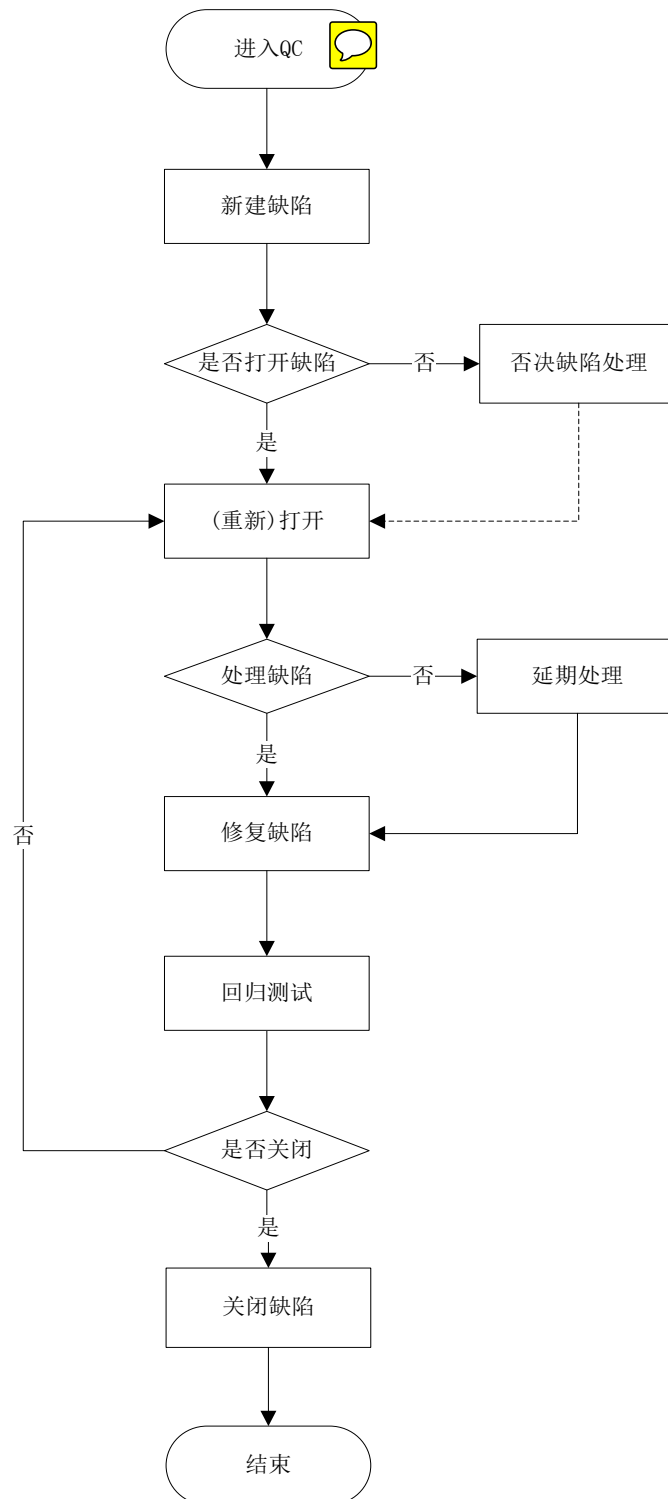
### 4.3 测试验收准则

1. 需求中定义的所有功能已实现；
2. 测试覆盖率已达到系统需求的 95%以上；
3. 测试中所发现的缺陷和错误已经 100%定位；
4. 所有要求的测试用例和测试程序已经 100%被执行；
5. 所有要求的高优先级测试用例和测试程序都已经被执行一次，并且所有严重级错误得到 100%的修改。



## 5 BUG 处理流程

5.1 BUG 处理流程图如下：



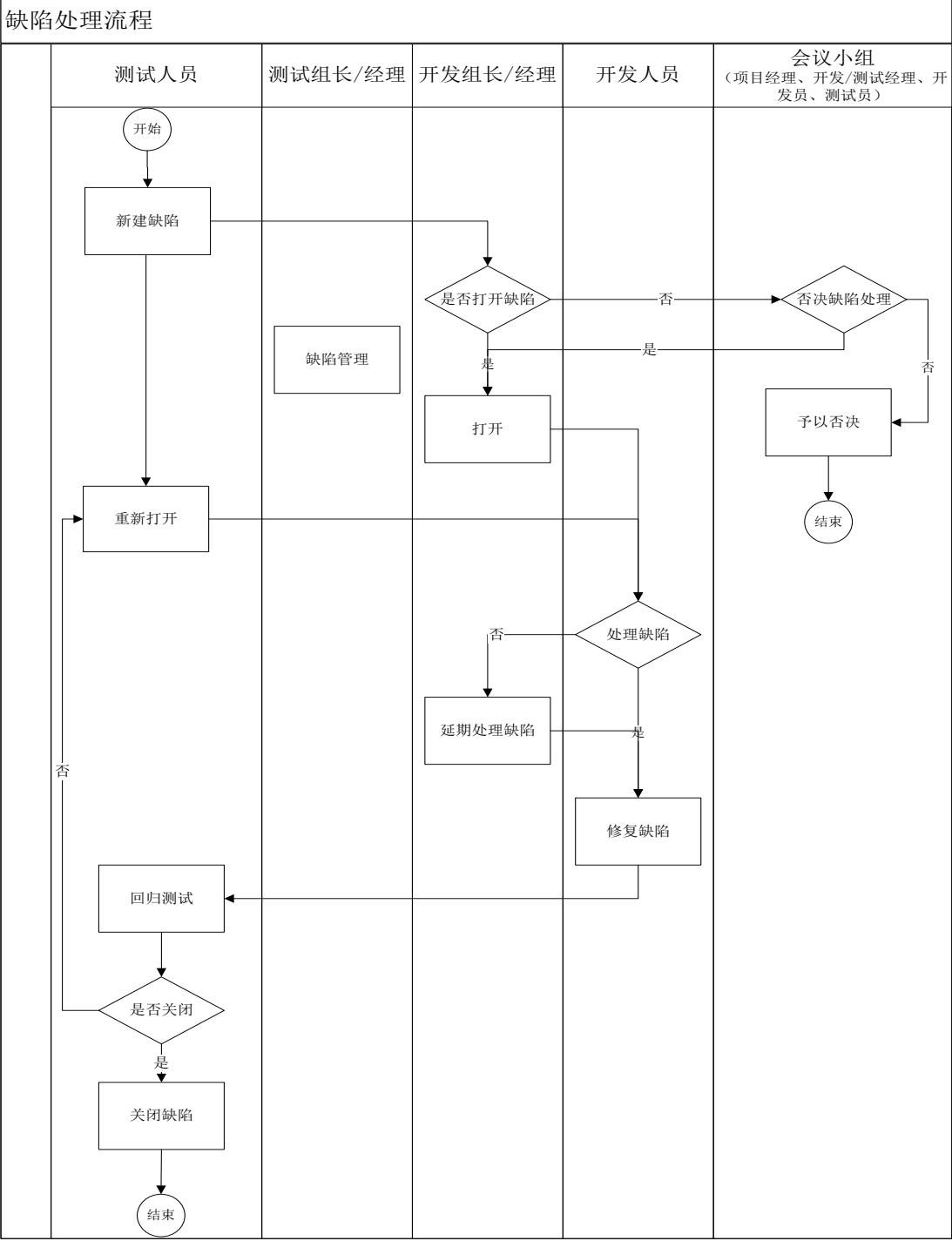
1. 缺陷处理流程图中判定说明：

- 1) 是否打开缺陷：开发组长/经理查阅缺陷，确认为缺陷后，指定优先级、估计修复日期再指派给相关开发人员；如果确认为不是缺陷的，注释中说明理由，予以否决。
- 2) 处理缺陷：开发处理缺陷；如果缺陷短期内进行修复存在困难，且该缺陷对于功能实现影响不大的，应该给开发组长/经理说明情况，让开发组长/经理与缺陷相关人员协调后延期处理该缺陷，并在注释中说明理由，估计修复日期和指明计划关闭版本。
- 3) 是否关闭：测试人员对回归通过的缺陷进行关闭；否则重新打开缺陷。并在注释中说明重新打开理由。

2. 缺陷处理流程图中流程说明：

- 1) 新建缺陷：测试人员（其他人员）根据缺陷填写说明，新建缺陷。
- 2) 否决缺陷处理：对已否决的缺陷，最后由测试发起会议（形式可以根据情况而定），找到缺陷相关人员进行确认。如果确认为是无效的缺陷，保持“已否决”状态，否则重新打开缺陷，并指派给相关处理人员。
- 3) （重新）打开：开发人员应该处理自己手上“打开”和“重新打开”的缺陷。
- 4) 延期处理：开发组长/经理根据情况，对缺陷进行延期处理。
- 5) 修复缺陷：开发人员处理完缺陷后，把缺陷状态改为“已修复”状态。并通知测试人员进行回归。
- 6) 回归测试：测试人员对已经修复的缺陷进行回归。
- 7) 关闭缺陷：测试人员回归测试通过后，对缺陷进行关闭。

5.2 为了说明各个角色在缺陷处理流程中的职责，据测试流程所画泳道图如下：



如果上面判定和流程中，某一方存在异议的，应及时反馈上级。然后上级根据缺陷优先级、实际情况等，找恰当的时间发起会议（或其它）的方式找到缺陷相关人员进行沟通、协调和处理。

## 5.3 QC 中 BUG 填写说明

1. “摘要”，用简单明了的语句说明白这个BUG，相当于BUG的中心语句。

2. 详细信息填写规范：

1) “分配给”，选择这个BUG所属模块是属于哪个研发人员，并把问题指派给他（如没有特别说明，则直接提交给该开发负责人）。

2) “缺陷类别”，分为9种

**BUG-功能**——功能上的缺陷，如按钮没响应，充值不成功，需求上提到的功能没实现等。

**BUG-样式**——页面样式的缺陷，如界面颜色、字号、排版、图片大小与所需求不符等。

**功能建议**——新增加或需要改进的功能性建议。

**UI建议**——对我们网页的布局、设计、色彩、交互、按钮、动静态效果、字体、文本框、表情、图片等有关视觉效果和操作便利性方面的建议。

**BUG-性能**——并发量、数据量、压缩率以及响应时间存在缺陷，如页面响应时间过长。

**BUG-需求**——流程功能不合理，但满足需求，问题出在需求设计不合理导致的问题。

**BUG-安全**——影响系统安全的缺陷，比如会导致SQL注入，XSS跨站攻击的安全隐患。

**文字描述**——包括错别字、错别句、用词不当、描述与实际功能不符等。

**易用性**——不易见、不易学、不易用以及用户体验性差的问题。

3) “缺陷主题”，选择你提交的BUG所属那个模块。

4) “项目”选择提交BUG时测试系统的影响版本。

5) “严重程度”，分为5个等级

**1-低 (trivial/minor)：**

- ① UI 控件不符合界面规范。
- ② 影响UI友好性。
- ③ 用户不频繁使用的功能易用性差。

**2-中 (normal)：**

- ① 用户需求未实现（不影响用户完成业务、用户使用不频繁）。

注：用户执行删除操作时系统应弹出确认提示将固定视为用户需求，无删除确认提示的缺陷归属本类。

- ② 用户需求实现错误（不影响用户完成业务、用户使用不频繁）。
- ③ 用户操作过程中系统出现异常报错，但不影响系统功能的使用。
- ④ 用户使用不频繁的功能，响应时间超出忍耐限度。

注：忍耐限度根据实际软件系统的特点而定。

- ⑤ UI 上存在错误引导用户的信息。
- ⑥ UI 上信息缺失、无法显示完整或出现乱码从而给用户造成疑惑的。
- ⑦ 用户频繁使用的功能易用性差（操作起来麻烦、复杂、效率低）。

**3-高 (major)：**

- ① 用户需求未实现（影响到用户完成业务）。
- ② 用户需求实现错误（影响到用户完成业务）。

③ 用户使用频繁的功能，响应时间超出忍耐限度（不影响其它功能）。

#### 4-非常高（critical）：

① 用户体验性非常差，会导致“大量”用户投诉的。

②重要功能基本实现，但不稳定：一些边界条件下操作会导致报错、文件操作异常、通讯异常、数据丢失或破坏等错误。

#### 5-紧急（blocker）：

① 重要功能模块未完成或未按照需求完成。

② 后台数据受损或丢失。

③ 导致被测软件响应明显很慢（假死）、死机、非法退出、崩溃。

④ 与“钱”沾边的，如充值、购买课程后不能使用、不购买课程也能使用课程等。

### 3. 缺陷“描述”规范

1) 指明当时出这个BUG的现场环境，示例如下：

测试服务器：www.ixueyun.com

浏览器：IE9、360浏览器（兼容模式）

2) 指明BUG所属模块或页面的路径，示例如下：

URL：http://www.ixueyun.com/forum （如果URL不能描述清楚的，应该用“云桌面->我的账户->我的信息”类似路径标明）

3) 把BUG产生的步骤一步一步写清楚，可以用以下方法写。（如果一句话就可以说明的BUG，就不必要分步骤了）

示例：

BUG重现步骤：

1、。。。

2、。。。

3、。。。

4、。。。

测试结果：。。。。。。

期望结果：。。。。。。

4) 尽可能可以通过上传截图或附件，进行简单明了的说明BUG存在，也可作为BUG证据。（谨记“有图有真相”）

注意：

①关于优化建议方面、一目了然的BUG，根据实际情况，可以简化以上的一些步骤。

②BUG描述简洁明了，条理清晰，使开发人员能据此复现定位该BUG。

③BUG应该为客观的描述语句，且应避免出现错别字，语病，歧义等。（避免过多主观看法：如我认为，我觉得，我想…如果有更好的想法应该提成建议）

## 5.4 BUG 验证/关闭问题说明

1. 当BUG状态变为“已修复”，根据回归清单或测试申请（由开发提供）进行回归测试，如果回归测试后该问题被解决，则关闭该BUG，并在注释中填写如下信息：

验证通过：是

验证日期：。。。

2. 如果回归测试验证不通过，则“重新打开”该BUG，并在注释中填写明情况。
3. 如果出现“延期处理”、“已否决”的BUG，首先查明原因，如果与研发不能达成一致的需要及时向上级反馈。

## 5.5 关于研发人员处理 BUG

1. 当研发人员接到一个BUG后，应该填写“估计修复工时”、“估计修复日期”。
2. 研发人员解决BUG写的注释一定要有以下几点：
  - 1) 说清楚BUG产生的原因。
  - 2) 简单说明BUG处理的方法或过程。
  - 3) 并在bug中注明实际修复工时，实际修复日期。
3. 研发人员否决或延期处理bug，需要注明原因。

## 5.6 BUG 优先级定义

紧急—系统正常业务流程无法通过，必须马上修改。

非常高—系统主要功能实现错误，或与用户需求实现出现偏差。

高—系统次要功能错误，但是不影响主功能实现及继续测试。

中—系统一般类错误，但是不影响测试，需在最终发布前修改的。

低—问题对系统影响很小，可以暂不修改。

## 6 测试交付

### 6.1 需求点测试通过标准

①所有需求功能点/项中无遗留等级“高”及以上BUG。

②一级需求点中总计遗留bug $\leq 5$  例如：遗留Bug=5，若存在“中”级BUG，则“中”级BUG $\leq 2$ ；若无“中”级BUG，则“低”级BUG $\leq 5$ 。

### 6.2 测试通过标准

①需求中定义的所有功能已实现。

②所有要求的测试用例和测试程序已经 100%被执行。

③测试覆盖率已达到系统需求的 95%以上。

④测试中所发现的缺陷和错误已经 100%定位。

⑤缺陷等级为“高”或以上的 BUG 100%得到解决，中、低（一般严重或轻微严重问题）等级 BUG 95%以上得到解决。其它 BUG 若不能解决应给出不能解决或不计划解决的原因。

### 6.3 测试结果编写

当一轮测试完成后，测试负责人应编写测试结果。

测试报告内容主要包括：测试版本号，测试内容，需求点总个数，新建 bug 总数、回归 bug 数、回归 bug 通过数、需求点通过数、需求点通过率、bug 分布情况，以及测试结论。

然后以邮件方式发送给项目经理、开发经理，并抄送给项目组成员。

### 6.4 测试报告编写

当一个版本测试完成后，测试负责人应该根据《测试报告模版》编写测试报告。说明测试情况并下测试结论，然后以邮件方式发送项目经理，抄送给项目组全体人员。