# **基于项目的案例教学系统QA计划**

需图标

|  |  |  |
| --- | --- | --- |
| 文件状态  【 】草稿  【 】正式发布  【√】正在修改 | 文件标识 | PRD2018-G16-QA计划 |
| 当前版本 | 0.1.0 |
| 作者 | 陈依伦、陈佳敏、徐毓茜、马益亮、吕煜杰 |
| 完成日期 | 2018-10-28 |

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 参与者 | 起止日期 | 备注 |
| 0.1.0 | 陈依伦、陈佳敏、徐毓茜、马益亮 、吕煜杰 | 2018-10-26至2018-10-28 | 对QA计划做出初步分析 |

**[[1]](#footnote-0)**

# 目录

[基于项目的案例教学系统QA计划 1](#_Toc2121)

[目录 3](#_Toc30302)

[一、 范围和参考文献 4](#_Toc12772)

[1.1范围 4](#_Toc18872)

[1.2参考文献 4](#_Toc20472)

[1.3定义 5](#_Toc3379)

[二、软件质量保证计划 5](#_Toc31234)

[2.1目的 5](#_Toc18627)

[2.2管理 5](#_Toc21735)

[2.2.1组织 5](#_Toc24435)

[2.2.2任务分配 5](#_Toc10842)

[2.2.3责任 6](#_Toc25525)

[2.3文件 7](#_Toc7267)

[2.3.1目的 7](#_Toc27464)

[2.3.2最低需求文档 7](#_Toc24122)

[2.4标准实现约定和指标 11](#_Toc18226)

[2.4.1目的 11](#_Toc25806)

[2.4.2标准 11](#_Toc26782)

[2.5审查和审查影响 12](#_Toc21681)

[2.5.1目的 12](#_Toc16802)

[2.5.2最低要求 12](#_Toc24643)

[2.6测试 14](#_Toc10549)

[2.7问题报告和纠正措施 15](#_Toc8271)

[2.8工具技术和方法 15](#_Toc17377)

[2.9代码控制 15](#_Toc633)

[2.10供应商控制 16](#_Toc16116)

[2.11记录收集维护和保存 16](#_Toc1068)

[2.12培训 16](#_Toc23096)

[2.13风险管理 16](#_Toc26090)

[[2]](#footnote-1)

# 范围和参考文献

## 1.1范围

该软件质量保证计划完成后及开始实施至项目完成为止，但此文档不应该被解释为禁止在添加额外的内容。应该对具体的软件项目进行评估，以确保覆盖范围的充分。

## 1.2参考文献

1. 百度百科. QA[EB/OL].

<https://baike.baidu.com/item/QA/476016?fr=aladdin#6.>[2018/10/26 1:00pm].

1. 百度百科. 讨论测试和QA的区别[EB/OL].

<https://blog.csdn.net/weixin_42538115/article/details/80871025.>[2018/10/26 1:10pm].

1. 百度百科. 测试开发之路--QA 的能力[EB/OL].

<https://blog.csdn.net/kkdd2013/article/details/53099396.>[2018/10/26 1:18pm].

CSDN博客. 敏捷开发，QA面临的10个挑战[EB/OL]. <https://blog.csdn.net/baynkbtg/article/details/52402727.>[2018/10/26 1:23pm].

1. 张海藩. 软件工程导论（第6版）[M]. 北京:清华大学出版社, 2018. 324-331

## 1.3定义

G16小组为陈依伦、陈佳敏、徐毓茜、马益亮 、吕煜杰所在的5人组。

# 二、软件质量保证计划

## 2.1目的

建立一套有计划，有系统的方法，来向管理层保证拟定出的标准、步骤、实践和方法能够正确地被所有项目所采用。使软件过程对于管理人员来说是可见的。通过对软件产品和活动进行评审和审计来验证软件是合乎标准的。这些将使软件项目满足机构方针的要求。

## 2.2管理

### 2.2.1组织

项目经理、业务分析师、需求开发人员、测试人员、原型设计师、会议记录师、配置管理员、CCB（变更控制委员会）、QA。

### 2.2.2任务分配

| **角色** | **人员** |
| --- | --- |
| 项目经理 | 陈依伦 |
| 配置管理员 | 陈依伦 |
| 业务分析师 | 陈依伦 |
| 会议记录员 | 陈依伦 |
| 原型设计 | 陈佳敏 |
| 测试 | 马益亮 |
| QA组织 | G16全员 |
| 需求开发 | G16全员 |
| CCB | G16全员 |

### 2.2.3责任

| **角色** | **人员** | **职责** |
| --- | --- | --- |
| 项目经理 | 陈依伦 | 负责项目的整体规划和管理  负责项目计划的制定和维护  负责资源的分配和协调活动  负责项目的跟踪和管理  参与项目技术评审和阶段评审  对项目工作产品的最终质量负责 |
| 需求开发 | G16小组 | 负责项目的需求调研  负责编写需求规格说明书 |
| 原型设计 | 陈佳敏 | 负责产品原型的设计  负责产品界面的设计 |
| 开发人员 | G16小组 | 根据编码规范编写代码，并进行自测  进行系统集成  对项目的BUG进行跟踪修正 |
| 测试人员 | G16小组 | 负责制定测试计划  负责设计测试用例  准备测试数据、测试环境  执行测试，记录测试结果  编写测试总结报告 |
| 配置管理员 | 陈依伦 | 负责制定配置管理计划  建立与维护配置库  建立和发布基线  对配置库的状态进行跟踪和统计  负责配置变更的跟踪 |
| 会议记录员 | 陈依伦 | 对每周的会议进行记录 |
| QA | G16小组 | 负责制定质量保证计划  对项目进展、风险和问题进行跟踪和监控 |
| CCB | 助教小组 | 严格控制项目的版本及结构 |

## 2.3文件

### 2.3.1目的

编写最低需求文档。使软件过程对于管理人员来说是可见的。

### 2.3.2最低需求文档

#### 2.3.2.1软件需求规范

暂无

#### 2.3.2.2软件设计说明

本系统将几类用户的权限进行了以下划分：

1）管理员权限：

①案例管理——管理员能够查找和查看系统中存在的案例，并且对系 统中存在的案例进行冻结与删除。同时管理员还能够上传符合系统要求的案例。

②用户管理——管理员能够查找和查看系统中的某一用户的相应信 息，并对其进行相应的操作，如冻结、删除等。

③系统管理——包括对系统数据库的备份与恢复，查看系统的管理日 志、系统错误日志以进行系统维护等等功能。

2）普通用户权限：

正常状态下：

①案例查看——可以查看系统中存在的并且可以进行学习的案例，同时也能够对该案例进行讨论。也能够通过查看案例的详细信息获得从该案例中所建立的相关项目信息，或者自行建立一个基于该案例的项目。

②实例操作——实例是在系统中，由案例所建立的项目的统称。用户能够查看在系统中存在的并且可以进行参与的项目。用户还能够对某一项目进行申请或者退出等操作；建立该项目的用户则能够进行项目的关闭或者打开等操作。

③个人空间管理——管理自己在该系统中所注册的信息，并可以查看其他用户的留言或者给其他用户进行留言。

项目学习过程中：

⑴学生：

①角色扮演——能够在项目中扮演某个角色，以此了解和实践该角色在整个项目进程中所发挥的作用，从而获得项目的相应经验。

②个人任务查看——用户能够查看到自己所扮演的角色在项目整体过程中所需要完成的任务，并查看相应的任务描述。

③整体规划——用户能够看到整个项目中所设定的项目计划，以及每个子任务的具体信息。同时也能够查看项目的甘特图表以了解整个项目的进度计划。

④文档的上传与下载——用户能够上传自己完成的项目文档以供项目经理进行检查。同时用户也能够下载存在于标准案例中的相应文档以进行学习，找出自己的不足。

⑵教师：

①标准文档查看——教师与学生不同，教师从一开始就能够看到项目中所有的标准文档。教师能够通过阅读标准文档结合自己的项目经验给学生完成某个任务以更为合理的建议。

②任务提点——教师能够根据自己的经验对完成该项目中的某个任务所需要具备的知识与技能进行建议编写，也能够对参考书目进行推荐以达到学生在完成任务过程中事半功倍的效果。

③即时交流——教师能够通过即时通讯工具与正在项目中进行实践的学生进行交流，为学生指点江山。

#### 2.3.2.3软件验证和验证计划报告

详见《测试用例》

#### 2.3.2.4用户文档

详见《用户手册》

#### 2.3.2.5软件配置管理计划

软件项的标识基本按照《软件配置标识命名规则》进行。要通过标识能够确定软件项之间的相互联系。

1、 首先在服务器上建立一个目录，作为项目配置数据库。在此目录下按照每个项目组建一个分目录，项目组代码及项目组名构成目录名，然后在此项目组目录下按照所属每个项目建一个子目录，同一项目的开发文档存放在一个目录下，项目编号紧跟项目名就是目录名。在一个项目分目录下可按非受控文档与受控文档建立一级次目录，然后在一级次目录下按文档的不同类型建立二级次目录，使得所有开发文档能分门别类的组织存放，便于查询。目录结构可见下图的示例。

2、 项目子目录的受控文档一般只有项目经理和属于该项目的开发人员和配置管理员能够访问到。配置管理员负责分配访问权限，一般项目经理对该目录具有较大的权限——读取、添加和更改；一般开发人员只有读取的权限。

3、 在项目开发的某一阶段结束时，通过了该阶段评审的这些开发文档交配置管理员保存到项目数据库，做为正式版本的第一版——1.0版本。

4、在以后的开发中，如果软件需要修改，那修改后的软件可用多级编号来表示新版本——1.1.0、1.2.0等加以区别标识。

5、 在各个评审阶段产生的所有评审报告和修改报告都要进行编号保存，编号与相应文档的编号要对应。

**对于较小的改动：**

1、在评审或测试后发现的问题由评审组组长或项目经理形成《软件问题报告单》，并通知配置管理员。

2、由配置管理员将需要修改的软件的备份从项目配置数据

库中检出，开发人员执行修改。

3、修改完毕后进行修改测试，编程错误累计到了一定的量或者测试时间已满一个月（从上一次入配置库后算起），凭〖源代码修改记录单〗及修改后的源代码，通知配置管理员，配置管理员确定测试报告的完备性，并在核对软件修改内容和修改人员填写的〖软件修改报告单〗或〖源代码修改记录单〗中的修改描述一致后，将文件登入项目配置数据库中，生成新版本。

4、配置管理员修改《软件配置状态表》和《软件变更记录表》，以使其他相关开发人员及时了解软件变化情况。

**对于较大的改动：**

1、 开发人员或用户提出影响较大的修改要求。（这是指要增加或删除某些功能或者是发现错误的阶段在造成错误的阶段的后面等。）

2、配置管理员在收到这类修改要求时，必须组织有项目经理以及开发人员参加的修改评审会，讨论修改的影响范围，修改的必要性、可行性以及修改方法、步骤和实施计划。

3、 在修改方案通过并经项目经理审核后，要由产品开发部经理签字批准。涉及重大技术方案的修改时，修改方案必须由总工程师或技术总监签字批准。以决断修改工作中各项活动的先后顺序及各自的完成日期，以保证整个开发工作按原定计划日期完成。

4、 配置管理员在接到修改批准——由项目经理或产品开发部经理或总工程师或技术总监签字同意的《软件问题报告单》后才可将需修改的软件的备份从项目数据库中检出，开发人员执行修改。

5、修改完毕后，交客户服务部进行测试和评审，测试和评审都通过后，交配置管理员处。

6、配置管理员检查测试报告和评审报告是否完备，核对〖软件修改报告单〗中的修改描述和修改后的软件是否相符。核查结果符合要求，配置管理员将修改后的软件登入项目数据库中，生成新版本。

7、配置管理员修改《软件配置状态表》和《软件变更记录表》，以使其他相关开发人员及时了解软件变化情况对受影响的软件做出相应的修改。

**配置状态报告：**

1、两份配置状态报告——〖软件配置状态表〗和〖软件变更记录表〗分别以电子表格的形式存放在项目分目录下，以便项目开发人员随时查询，了解软件的修改变化情况。

2、〖软件配置状态表〗由配置管理员负责填写，主要反映项目中各软件项的配置情况。开发人员通过查阅该表可及时全面的了解项目中软件项的配置使用情况。

3、〖软件变更记录表〗由配置管理员负责填写，主要记录

软件开发过程中所有的修改情况，该表以修改时间排序，以便开发人员及时了解软件项最新的变化。

**配置审核：**

为保证各项产品在技术上和管理上的完整性，总经理室在软件开发过程中的详细设计阶段和测试阶段完成时，对配置情况进行抽查。总经理室先提出要审核的内容和各项指标，逐项审核完成后要作好记录，形成《配置审核报告》。

## 2.4标准实现约定和指标

### 2.4.1目的

客观地评价了所执行的过程以及产生的工作产品和服务对适用的过程描述、标准以及流程的符合度。

### 2.4.2标准

采用IEEE软件工程标准：软件质量保证计划——IEEE Std 730—2001。

## 2.5审查和审查影响

### 2.5.1目的

检验软件项目是否满足机构方针的要求，预防缺陷发生，保证软件产品的质量和减小返工所带来的附加成本。

### 2.5.2最低要求

#### 2.5.2.1软件需求评审

在软件需求分析阶段结束后必须进行软件需求评审，以确保在软件需求(规格)说明中所规定的各项需求的合适性。

#### 2.5.2.2初步设计审查

在系统/子系统设计结束后必须进行系统/子系统设计的评审，并且自底向上，由小系统到大系统的顺序进行评审，以评价软件(结构)设计说明中所描述的软件设计在总体结构、外部接口、主要部件功能分配、全局数据结构以及各主要部件之间的接口等方面的合适性。

#### 2.5.2.3关键设计审查

在软件设计结束后必须进行软件设计的评审，以评价软件(结构)设计说明中所描述的软件设计，在功能、算法和过程描述等方面的合适性。

#### 2.5.2.4软件验证及验证计划审查

在制订软件验证与确认计划之后要对它进行评审，以评价软件验证与确认计划中所规定的验证与确认方法的合适性与完整性。

#### 2.5.2.5功能审查

在软件发行前，要对软件进行功能检查，以确认已经满足在软件需求规格说明中规定的所有需求。

#### 2.5.2.6物理审查

在验收软件前，要对软件进行物理检查，以验证程序和文档已经一致并已做好了交付的准备。

#### 2.5.2.7进程内的审查

在软件制作时，要规律性的对其进行抽查，以保证软件在正确的道路上研发，避免早起的错误导致的翻盘重做。

#### 2.5.2.8管理审查

1. 组长全面负责有关软件质量保证的各项工作；
2. 发起人负责有关阶段评审、项目进展报表检查以及软件验收

准备等三方面工作中的质量保证工作；

1. 配置管理人员负责有关软件配置变动的质量保证活动；
2. 各子项目负责人负责测试复查和文档的规范化检查工作；
3. 用户代表负责反映用户的质量要求，并协助检查各类人员对

软件质量保证计划的执行情况。

#### 2.5.2.9软件配置管理计划审查

必须编制有关软件配置管理的条款，或单独制订文档。在这些条款或文档中，必须规定用于标识软件产品、控制和实现软件的修改、记录和报告修改实现的状态以及评审和检查配置管理工作等四方面的活动。还必须规定用以维护和存储软件受控版本的方法和设施；必须规定对所发现的软件问题进行报告、追踪和解决的步骤，并指出实现报告、追踪和解决软件问题的机构及其职责。

#### 2.5.2.10事后剖析审查

在软件验收时，要允许用户对所要验收的软件进行设计抽样的综合检查，以验证代码和设计文档的一致性、接口规格说明之间的一致性(硬件和软件)、设计实现和功能需求之间的一致性、功能需求和测试描述的一致性。

## 2.6测试

本系统测试工作由测试人员与开发人员共同来完成，主要采用下列方法进行系统的测试：

黑盒测试：着重于测试软件系统的外部特性；根据系统的设计要求，每一项功能都要进行逐个测试，检查其是否达到了预期的要求，是否能正确地接受输入，是否能正确地输出结果。

白盒测试：由于软件的所有源代码都要由项目组成员编写，对其内部的逻辑规则和数据流程，都要进行测试，以检查其代码编写是否符合设计要求。

集成测试：在所有模块都通过了单元测试后，将各个模块组装在一起，进行组装测试，用于发现与接口相联系的问题。在通过组装测试后，将经过单元测试的模块组装成一个符合设计要求的软件结构。

系统测试：在本项目通过了以上的测试步骤后，与其它系统元素（如硬件服务器、网络系统等）进行集成测试和系统级的确认测试，将各种可能的缺陷完全排除掉，从根本上保证系统的长期稳定运行。

在测试之前需要完成测试计划，建立测试环境，创建硬件设备，需要基于功能说明文档编写测试用例。

在评审功能测试工作时，确保软件符合用户规格说明书中所列的需求，不仅要运行准备的测试用例，而且要允许运行任务委托单位或用户、评审人员选定的采样用例。执行测试用例的结果需要记录，而且需要记录随后对任何缺陷的修改。

## 2.7问题报告和纠正措施

问题报告经整理后形成文档统一交付给项目经理并由项目经理决定纠正与否，纠正时间及方案。

## 2.8工具技术和方法

暂无

## 2.9代码控制

开发过程中采用JAVA语言编写程序，每一个版本的代码都进行存档控制以便于回档重塑以及分支修改。最终代码经多次审核后方可提交。

## 2.10供应商控制

获取供应商在最初阶段提出的大致要求并为实现需求实施相应计划，并在后续阶段不断与供应商接触以确定需求的正确性以及获取新的需求，并以此为保证作品质量的一环。

## 2.11记录收集维护和保存

所有过程文件包过历史版本与最新版本都应保存至项目经理电脑与GIT远程库，进行两端同时保存及维护。每次会议都应有会议记录员记录会议的主要内容与任务的分配以便于事后追究。

## 2.12培训

关于Axure RP软件的使用还需要界面原型的设计者去进一步的学习这个软件的使用。

## 2.13风险管理

对无法参与工作的员工，项目经理进行合理的安排分解工作。对已丢失的文档尽可能还原，如不行则以慎重保存后续文档为主。

1. [↑](#footnote-ref-0)
2. [↑](#footnote-ref-1)