|  |
| --- |
| 17计科1班第2小组  日期： 2020/6/16 |

|  |
| --- |
|  |
| ssm的网上商城系统 |
| [质量管理说明书] |

目录

[系统质量管理说明书 3](#_Toc43102460)

[一、 软件质量 3](#_Toc43102461)

[1、软件质量定义 3](#_Toc43102462)

[2、软件质量特性 3](#_Toc43102463)

[3、软件质量模型 6](#_Toc43102464)

[4、软件的质量度量 6](#_Toc43102465)

[二、质量计划 7](#_Toc43102466)

[1、引言 7](#_Toc43102467)

[2、项目组织 7](#_Toc43102468)

[3、质量目标 9](#_Toc43102469)

[4、质量策略 10](#_Toc43102470)

[5、质量保证活动 10](#_Toc43102471)

[6、质量控制活动 12](#_Toc43102472)

[7、质量保证的报告途径 12](#_Toc43102473)

[8、记录的收集。维护和保存 12](#_Toc43102474)

[三、质量保证 13](#_Toc43102475)

[1、引言 13](#_Toc43102476)

[2、管理 13](#_Toc43102477)

[4、评审和检查 15](#_Toc43102478)

[四、质量控制 16](#_Toc43102479)

[1、 问题级别划分 17](#_Toc43102480)

[2、 各阶段质量目标和检查标准 18](#_Toc43102481)

# 系统质量管理说明书

## 软件质量

### 1、软件质量定义

软件质量就是“软件与明确地和隐含地定义的需求相一致的程度”。更具体地说，软件质量是软件与明确地叙述的功能和性能需求、文档中明确描述的开发标准以及任何专业开发的软件产品都应该具有的隐含特征相一致的程度。

从管理角度对软件质量进行度量， 可将影响软件质量的主要因素划分为六个部分特性(根据GB/T16260.1)功能性，可靠性，易用性，效率，维护性与可移植性。其中功能性包括适合性，准确性，互操作性，保密安全性，功能依从性;可靠性包括成熟性，容错性，易恢复性，可靠性依从性;易用性包括易学性，易理解性，易操作性，吸引性，易用性依从性;效率包括资源特性，时间特性，效率依从性;维护性包括易测试性，易改变性，稳定性，易分析性，维护性依从性;可移植性包括适应性，易安装性，共生性，易替换性，可移植性依从性的六个特征及若干子特征。

### 2、软件质量特性

#### 2.1功能性

是指当软件在指定条件下使用，软件产品满足明确和隐含要求功能的能力。

##### 2.1.1子特性

（1）适合性：是指软件产品与指定的任务和用户目标提供一组合适的功能的能力。

（2）准确性：是指软件产品具有所需精确度的正确或相符的结果及效果的能力。

（3）互操作性：是指软件产品与一个或多个规定系统进行交互的能力。

（4）保密安全性：是指软件产品保护信息和数据的能力，以使未授权的人员或系统不能阅读或修改这些信息和数据，但不拒绝授权人员或系统对其的访问。

（5）功能依从性：是指软件产品依附与同功能性相关的标准、约定或法规以及类似规定的能力。

#### 2.2可靠性

在指定条件下使用时，软件产品维持规定的性能级别的能力。

##### 2.2.1子特性

（1）成熟性：是指软件产品避免因软件中错误发生而导致失效的能力。

（2）容错性：是指在软件发生故障或违反指定接口的情况下，软件产品维持规定的性能级别的能力。

（3）易恢复性：是指在失效发生的情况下，软件产品重建规定的性能级别并恢复受直接影响的数据的能力。

（4）可靠性依从性：是指软件产品依附与同可靠性相关的标准、约定或法规以及类似规定的能力。

#### 2.3易用性

是指在指定条件下使用时，软件产品被理解、学习、使用和吸引用户的能力。

##### 2.3.1子特性

（1）易理解性：是指软件产品使用户能理解软件产品是否合适以及如何能将软件用于特定的任务和使用环境的能力。

（2）易学性：是指软件产品使用户能学习它的能力。

（3）易操作性：是指软件产品使用户能操作和控制它的能力。

（4）吸引性：是指软件产品吸引用户的能力。

（5）易用性依从性：是指软件产品依附与同易用性相关的标准、约定、风格指南或法规以及类似规定的能力。

#### 2.4效率

是指在规定条件下，相对于所用资源的数量，软件产品可提供适当的性能的能力。

##### 2.4.1子特性

（1）时间特性：是指在规定条件下，软件产品执行其功能时，提供适当的响应时间和处理时间以及吞吐率的能力。

（2）资源利用性：是指在规定条件下，软件产品执行其功能时，提供合适的数量和类型的资源的能力。

（3）效率依从性：是指软件产品依附与同效率相关的标准或约定的能力。

#### 2.5维护性

是指软件产品可被修改的能力，修改可能包括修正，改进或软件适应环境、需求和功能规格说明中的变化。

##### 2.5.1子特性

（1）易分析性：是指软件产品诊断软件中的缺陷或失效原因，以及判定待修改的部分的能力。

（2）易改变性：是指软件产品使指定的修改可以被实现的能力。

（3）稳定性：是指软件产品避免由于软件修改而造成意外结果的能力。

（4）易测试性：是指软件产品使已修改软件能被确认的能力。

（5）维护性依从性：是指软件产品依附与同维护性相关的标准或约定的能力。

#### 2.6可移植性

是指软件产品从一种环境迁移到另一种环境的能力。

##### 2.2.1子特性

（1）适应性：是指软件产品无需采用有别于为考虑该软件的目的而准备的活动或手段，就可能适应不同的指定环境的能力。

（2）易安装性：是指软件产品在指定环境中被安装的能力。

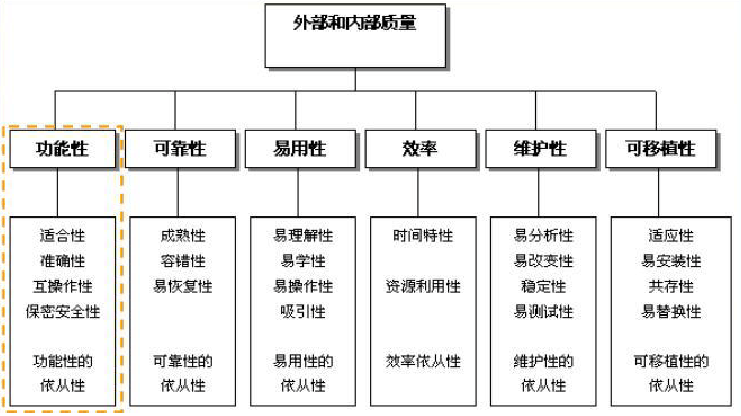
（3）共存性：是指软件产品在公共环境中同与其分享公共资源的其他独立软件共存的能力。

（4）易替换性：是指软件产品在环境相同、目的相同的情况下替代另一个指定软件产品的能力。

（5）可移植性依从性：是指软件产品依附与同可移植性相关的标准或约定的能力。

### 3、软件质量模型

该项目采用ISO质量模型



### 4、软件的质量度量

软件的质量度量是软件度量的一个子集，它侧重于产品，过程，和项目管理的质量环节。一般来说，软件质量度量与过程和产品联系比较紧密，而与项目管理度量的联系则没有那么紧密。

软件质量度量可以进一步分为最终产品质量度量和过程中质量度量。

## 二、质量计划

### 1、引言

#### 1.1编写目的

随着人们生活节奏的不断加快和Internet网的不断普及，网上购物成为一种新时尚，它不仅让人们在家就可以买到自己想要的商品，还可以节约大量时间，一般情况下还可以买到物美价廉的商品，由于网上购物不但方便，还节约时间金钱和劳动力，所以我们的系统为广大网上购物者提供操作简单，只需要在关键子栏目输入关键字就可以找到我们想要的商品，使用便捷的网上购物系统，有简单明了的用户登录界面，就可以把张杰中意的商品拖进购物车，还可以使用多种付款方式，所以需要一个便捷的系统吸引越来越多的人。

#### 1.2 文档范围

本文档是项目的软件的质量计划。

本文档使用对象为：

项目分析人员

程序设计人员

项目管理人员

#### 1.3参考资料

1. 软件工程与实践（第三版） 清华大学出版社
2. 软件工程课程项目参考案例
3. 网上查找的一些相关案例

### 2、项目组织

#### 2.1组织机构

在项目实施期间成立项目质量保证组织，该组织由质量保证人员和项目经理组成，项目经理负责质量监督工作及项目进展过程中各环节的质量把关，开发经理负责质量控制的工作，质量保证人员负责质量保证的工作。组织结构图1如下:

市场部

项目管理

配置管理

质量保证

软件开发

质量控制

设计

实施

#### 2.2职责

在本项目中，质量保证组织的职责如下：

##### 2.2.1项目的质量保证人员

质量保证人员的质量职责如下：

● 负责项目实施过程中对项目实施情况进行监督，包括对项目实施过程和工作产品进行监督检查；

● 实施项目组成员的质量保证培训；

● 制定质量保证计划；

● 按计划实施审计活动,依照质量保证计划执行评审/审计，并记录执行中发现的不符合项

● 对不符合问题提交不符合项报告，跟踪并验证纠正措施的执行情况

● 对项目内不能解决的不符合项问题向项目经理提交报告

● 向项目经理报告项目质量工作状况和质量度量结果

● 定期向项目组报告质量活动的结果

● 制订质量保证的过程改进计划，记录过程数据

##### 2.2.2项目经理

项目经理的质量职责如下：

● 评审质量计划；

● 与质量保证人员一起协商不符合项问题的纠正措施，并安排资源实施纠正措施；

● 定期或事件驱动的评审质量保证活动和结果

### 3、质量目标

根据软件工程国家标准文档的质量方针和质量目标，结合本项目特点,制定项目的总体质量目标：

（1）基于需求的测试覆盖率为100%；

（2）软件功能测试用例通过率不低于95％；

（3）每个阶段评审中发现的问题都已经解决或得到适当处理。

（4）产品发布时不存在严重及其以上的缺陷。

注：严重问题指导致系统或模块不能正常工作的问题。

结合以往的项目经验和软件工程国家标准文档的质量相应标准，制定质量标准如下表

表1：质量计划标准

|  |  |  |  |
| --- | --- | --- | --- |
| 项 目 | 具 体 描 述 | 计 划 | 实 际 |
| 缺陷排除率（缺陷数/页） | 需求检查 | 4 | 4 |
| 系统总体设计检查 | 2 | 2 |
| 缺陷排除率（缺陷数/KLOC） | 详细设计复核 | 30 | 30 |
| 详细设计检查 | 10 | 10 |
| 代码复核 | 65 | 65 |
| 代码检查 | 20 | 20 |
| 编译 | 20 | 20 |
| 单元测试 | 15 | 15 |
| 系统集成 | 5 | 5 |
| 系统测试 | 5 | 5 |

表1：质量计划标准表

### 4、质量策略

为了保证提交用户的产品是高质量, 实施过程中采取的质量保证措施包括：

（1）将质量贯彻到日常的项目进展过程中，

（2）应该特别注意项目工作产品质量的早期评审工作，无论是质量保证还是质量控制采取的策略都是早期预防和早期排除缺陷。

### 5、质量保证活动

质量保证的主要活动包括过程评审和产品审计。过程评审和产品审计的目的是为了确保在项目进展过程的各个阶段和各个方面采取各项措施来保证和提高提交给用户的产品质量。每一次过程评审和产品审计都应填写相应的报告或活动记录。

#### 5.1.产品审计

产品审计由质量保证人员来进行，检查项目产品是否达到质量目标。

质量保证人员对项目生存期中创建的工作产品可以有选择性的进行审计，以验证是否符合适当的标准，是否进行了质量检查。

表2便是质量审计一览表

|  |  |  |
| --- | --- | --- |
| 审计对象 | 审计阶段 | 参照的标准 |
| 软件项目计划 | 计划结束 | 软件工程国家标准文档 |
| 软件质量保证计划 | 计划结束 | 软件工程国家标准文档 |
| 总体设计文档 | 设计结束 | 软件工程国家标准文档和项目计划 |
| 详细设计文档 | 设计结束 | 软件工程国家标准文档和项目计划 |
| 数据库概念结构设计 | 设计结束 | 软件工程国家标准文档 |
| 系统功能模块设计 | 设计结束 | 软件工程国家标准文档 |
| 数据库逻辑结构设计 | 设计结束 | 软件工程国家标准文档 |
| 数据库物理结构设计 | 设计结束 | 软件工程国家标准文档 |
| 产品代码 | 每个阶段实施结束 | 软件工程国家标准文档 |
| 测试计划 | 设计结束 | 软件工程国家标准文档 |
| 测试报告 | 测试结束 | 软件工程国家标准文档 |
| 用户文档 | 测试结束 | 软件工程国家标准文档 |

表2：审计产品一览表

#### 5.2 过程评审

项目严格按照组织定义的软件过程进行开发，过程评审的具体依据参照软件工程国家标准文档的过程规范，保证项目中的所有过程活动都在实施范围内。在每次评审之后，要对评审结果做出明确的决策并形成评审记录。评审可采取文件传阅、评审会等形式。

质量保证人员负责对项目过程进行监督，发现的问题和解决情况在每周的例会上通报，对没有解决的问题进行讨论，对不能解决的问题提交高级管理者处理。

每个周末，进行一次配置管理审核，确认配置管理工作是否正常进行。

根据质量保证体系和本项目的具体特点，确定项目执行过程如下：

（1）项目规划过程及产品标准

（2）需求分析过程及产品标准

（3）系统设计过程及产品标准

（4）详细设计过程及产品标准

（5）调试运行过程及产品标准

（6）代码走查过程及代码编写标准

（7）质量保证过程及其标准

### 6、质量控制活动

质量控制活动包括代码走查、单元测试、集成测试、环境测试等，由开发人负责，详见进度计划。编码人员在编写代码时要进行同步单元测试，单元测试要达到分支覆 盖，产品通过单元测试和编码检查后，应提交测试部进行集成测试、系统测试。测试部的测试应达到质量目标要求，软件发布时应达到测试通过准则的要求。

### 7、质量保证的报告途径

质量保证人员对每次审计活动发现的不符合项，应该和项目经理协商不符合项的纠正措施，及预定完成日期，若和项目经理存在意见分歧，科和小组人员讨论。同时不符合项在项目周例会中汇报。

对不符合项，质量保证人员要在预定完成日期内重新审计、验证不符合项的纠正情况，若超过预定完成日期1周仍然有没解决的不符合项，由小组人员讨论决定最后的措施。

质量保证人员有独立的汇报途径，日常的汇报途径如下：

● 发现的问题通知项目经理，协调纠正措施。

● 将项目组内不能协调的问题汇报给项目管理员，由项目管理员协调解决。

● 日常工作和过程数据要汇报给质量经理统一收集、统计。

### 8、记录的收集。维护和保存

项目组应当保留项目执行过程中形成的各类文档、各种记录、各级周报、对于项目中问题的处理也需要形成记录保存。每周由质量保证人员根据任务清单的审计任务进行审计活动，并收集各活动的过程数据。

## 三、质量保证

### 1、引言

#### 1.1目的

本计划的目的在于对所开发的软件规定各种必要的质量保证指施,以保证所交付的软件能够满足项目预定需求,能够满足本项目总体组制定的且经小组评审批准的该软件系统需求规格说明书中规定的各项具体需求。  
 软件开发项目组在开发软件系统所属的各个子系统(其中包括为本项目研发或选用的各种支持软件、组件)时,都应该执行本计划中的有关规定,但可根据各自的情况对本计划作适当的剪裁,以满足特定的质量保证要求,剪裁后的计划必须经项目组相关负责人批准。

#### 1.2 参考资料

GBT 11457 软件工程术语  
GB 867计算机软件开发规范

GB/T 12504计算机软件质量保证计划规范

### 2、管理

#### 2.1 机构

在本软件系统整个开发期间,必须成立软件质量管理小组负责质量保证工作。

软件质量保证组和项目负责人及小组成员必须检查和督促本计划的实施。系统的软件质量保证人员有权直接向小组成员报告该项目的软件质量状况。系统的软件质量保证人员应该根据对项目的具体要求,制订必要的规程和规定,以确保完全遵守本计划的所有要求。

#### 2.2 任务

软件质量保证工作涉及软件生存周期各阶段的活动,应该贯彻到日常的软件开发活动中,而且应该特别注意软件质量的早期评审工作。因此,对于所负责系统,要按照本计划的各项规定进行各项评审工作。软件质量保证小组要参加所有的评审与检查活动。评审与检查的目的是为了确保在软件开发工作的各个阶段和各个方面都认真采取各项措施来保证与提高软件的质量。在软件开发过程中,要进行如下几类评审与检查工作:  
 （1）阶段评审:在软件开发过程中,要定期地或阶段性地对某一开发阶段或某几个开发阶段的阶段产品进行评审。在软件及其所属各子系统的开发过程中,应该进行以下三次评审:第一次评审软件需求、概要设计、验证与确认方法;第二次评审详细设计、功能测试与演示,并对第一次评审结果复核;第三次是功能检查、物理检查和综合检查。

阶段评审工作要组织专门的评审小组，原则上由项目总体小组成员或特邀专家担任评审组长,评审小组成员应该包括项目所有成员、质量保证人员。

（2）日常检查:在软件的工程化开发过程中,各子系统应该填写项目进展报表,即软件进展报表表头、软件阶段进度表、软件阶段产品完成情况表、等表格。项目其他人员通过项目进展季报表发现有关软件质量的问题。  
 （3）软件验收:必须组织专门的验收小组对软件系统及其所属各个子系统进行验收。验收工作应该满足项目的需求,质量管理小组验收内容应包括文档验收、程序验收、演示、验收测试与测试结果等几项工作。

#### 3、文档 该章给出了在软件开发过程各阶段需要编制的文档名称及其要求,并且规定了评审文档质量的通用的度量准则。

#### 3.1 基本文档

为了确保软件的实现满足认可的需求规格说明书中规定的各项需求,软件开发项目组至少应该编写以下方面内容的文档:  
（1）软件需求规格说明书(SRS);  
（2）软件设计说明书(SDD),对一些规模较大或复杂性较高的项目,应该把本文档分成概要设计说明书(PDD)与详细设计说明书(DDD)两个文档;  
（3）软件测试计划（STP);

（4）软件测试报告(STR);

#### 3.2 文档质量的度量准则

文档是软件的重要组成部分,是软件生存周期各个不同阶段的产品描述。验证和确认就是要检查各阶段文档的合适性。评审文档质量的度量准则有以下六条:

（1）完备性:所有承担软件开发任务的项目,都必须按照GB 8567 (是国家标准局的指南文档,名称叫《计算机软件产品开发文件编制指南》)的规定编制相应的文档,以保证在开发阶段结束时其文档是齐全的  
 （2）正确性:在软件开发各个阶段所编写的文档的内容,必须真实地  
反映该的段的工作且与该阶段的需求相一致

（3）简明性:在软件开发各个阶段所编写的各种文档的语言表达应该清晰、准确简练,适合各种文档的特定读者。

（4）可追踪性:在软件开发各个阶段所编写的各种文档应该具有良好的可追踪性。文档的可追踪性包括纵向可追踪性与横向可追踪性两个方面。前者是指在不同文档的相关内容之间相互检索的难易程度;后者是指确定同一文档某一内容在本文档中的涉及范围的难易程度。

（5）自说明性:在软件开发各个阶段所编写的各种文档应该具有较好的自说明性。文档的自说明性是指在软件开发各个阶段中的不同文档能独立表达该软件其相应阶段的阶段产品的能力。

（6）规范性:在软件开发各个阶段所编写的各种文档应该具有良好的规范性。文档的规范性是指文档的封面、大纲、术语的含义以及图示符号等符合有关规范的规定。

### 4、评审和检查

该部分具体规定了应该进行的阶段评审、阶段评审的内容和评审时间要求。对新开发的或正在开发的各个子系统,都要按照GB 8566(计算机软件开发规范)的规定认真进行定期的或阶段性的各项评审工作。就整个软件开发过程而言,至少要进行软件需求评审、概要设计评审,详细设计评审、软件验证和确认评审、功能检查、物理检查综合检查以及管理评审等八个方面的评审和检查工作。下面给出每次评审应该进行的工作。

#### 4.1 第一次评审

第一次评审会对软件需求，概要设计。

1. 软件需求评审（SRR）应确保在软件需求规格说明书中规定的各项需求的合理性。
2. 概要设计评审（PDR）应评价软件设计说明书中的软件概要实际的技术合适性。

#### 4.2 第二次评审

第二次评审会要对详细设计、功能测试与演示进行评审,并对第一次评审结果进行复核。如果在软件开发过程中发现需要修改第一次评审结果,则应按照《软件配置管理计划》的规定处理。

1. 详细设计评审(DDR)应确定软件设计说明书中的详细设计在满足软件需求规格说明书中的需求方面的可接受性。
2. 编程格式评审应确保所有编码采用规定的工作语言,能在规定的运行环境中运行,并且符合GB 8566中提倡的编程风格。在满足这  
   些要求之后，方可进行测工作。
3. 测试工作评审应对所有的程序单元进行静态分析,检查其程序结构(即模块和函数的调用关系和调用序列)和变量使用是否正确。在通过静态分析后,再进行结构测试和功能测试。

#### 4.3 第三次评审

第三次评审会要进行功能检查、物理检查和综合检查。这些评审会应在集成测试阶段结束后进行。

1. 功能检查(FA)应验证所开发的软件已经满足在软件需求规格说明书中规定的所有需求。
2. 物理检查(PA)应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。
3. 综合检查(CA)应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的一致性、功能需求和测试描述的一致性。

## 四、质量控制

质量控制活动包括代码走查、单元测试、集成测试、环境测试等,由开发入负责,详见进度计划。编码人员在编写代码时要进行同步单元测试,单元测试要达到分支覆盖,产品通过单元测试和编码检查后,应提交给测试部门进行集成测试、系统测试。测试部的测试应达到质量目标要求,软件发布时应达到测试通过准则的要求。

### 问题级别划分

#### 1.1设计阶段

|  |  |
| --- | --- |
| 级别 | 说明 |
| 严重 | 1.无法实现的设计  2.设计内有矛盾和冲突  3.设计中有严重影响下一阶段工作的遗留问题  4.重要需求设计错误 |
| 中等 | 1.一般需求设计错误  2.总体和概要设计中有影响下一阶段工作的遗留问题 |
| 轻微 | 1.表述繁琐，不易于理解  2.详细设计中有影响下一阶段工作的遗留问题 |

#### 1.2开发阶段

|  |  |
| --- | --- |
| 级别 | 说明 |
| 严重 | 1.提交的代码无法进行集成测试  2.重要需求设计实现错误 |
| 中等 | 1.一般需求设计实现错误  2.多处没有遵循编码规范 |
| 轻微 | 1.个别之处没有遵循编码规范  2.其他错误 |

#### 1.3测试阶段

|  |  |  |
| --- | --- | --- |
| 缺陷级别编号 | 级别 | 说明 |
| 1 | 严重 | 将会导致整个产品开发失败，系统崩溃，系统挂起，产品不能发版 |
| 2 | 对系统功能有普遍影响，业务完成方面的严重错误 |
| 3 | 中等 | 数据库或表不存在，参数出错，程序异常出错等 |
| 4 | 数据未校验，语言错误 |
| 5 | 轻微 | 小缺陷，如拼写错误，顺序错误等 |

### 各阶段质量目标和检查标准

#### 2.1需求分析阶段

需求分析阶段，对各交付物的质量目标如下：

|  |  |  |
| --- | --- | --- |
| 评审对象 | 评审项目 | 质量目标 |
| 需求规格说明书 | 准确性 | 1.重要需求功能点描述正确率>98%  2.一般需求功能点描述正确率>95% |
| 完整性 | 1. 重要需求功能点描述遗漏为0  2. 一般需求功能点描述正确率<5% |
| 按时性 | 需求规格说明书提交延迟时间<10% |

#### 2.2设计阶段

对设计阶段质量目标如下：

|  |  |  |
| --- | --- | --- |
| 评审对象 | 评审项目 | 质量目标 |
| 设计文档 | 准确性 | 1.级别为“严重”的遗留问题为0  2.级别为“中等”以上的遗留问题<5%  3．级别为“轻微”以上的遗留问题<10% |
| 完整性 | 1. 对重要需求功能点的覆盖率为100%  2. 对一般需求功能点的覆盖率>95%  3.接口遗漏为0 |
| 按时性 | 设计文档提交延迟时间<10% |

#### 2.3开发阶段

开发阶段质量目标如下：

|  |  |  |
| --- | --- | --- |
| 评审对象 | 评审项目 | 质量目标 |
| 编码 | 完整性 | 1. 对重要需求功能点的覆盖率为100%  2. 对一般需求功能点的覆盖率>95%  3.严重的功能问题为0 |
| 代码走读/检视 | 1.“严重”算法错误0  2.“中等”算法错误<5个  3.“严重”编码缺陷0  4.“中等”编码缺陷<5个 |

|  |  |  |
| --- | --- | --- |
| 单元测试 | 完全性 | 有效代码的覆盖率>80% |
| 内存操作 | 无内存遗漏 |
| 界面 | 统一性 | 界面风格统一，无明显错误 |

#### 2.4测试阶段

测试阶段质量目标如下：

|  |  |  |
| --- | --- | --- |
| 评审对象 | 评审项目 | 质量目标 |
| 集成测试 | 完全性 | 完成所有的集成测试用例 |
|  | 准确性 | 1.遗留的严重缺陷比率<2个  2.遗留的中和低缺陷比率<2个 |
| 系统测试 | 完全性 | 1.每个需求被>2个测试用例覆盖  2.要求覆盖常见隐蔽需求 |
| 准确性 | 1.严重缺陷=0  2. 遗留的陷比率<0.35个 |
|  | 稳定性 | 正常负载下连续运行7天无故障 |
| 用户文档 | 完整性 | 1.严重功能遗漏=0  2.一般功能遗漏<5% |