**XX信息**

**软件开发项目技术管理规范**

|  |  |  |
| --- | --- | --- |
| 文件编号：  RK-S20170802 | 生效日期：  2017.8.20 | 受控编号： |
| 版次：Ver1.0 | 修改状态： |
| 编制： | 审核： | 批准： |

贵州XX信息科技有限公司

目录

[一、编写说明 3](#_Toc491267712)

[二、软件项目整体开发流程 4](#_Toc491267713)

[三、各阶段岗位职责与工作内容 5](#_Toc491267714)

[四、各阶段工作要求 8](#_Toc491267715)

[1．软件需求分析 8](#_Toc491267716)

[2 软件项目计划 12](#_Toc491267717)

[3 概要设计 16](#_Toc491267718)

[4 详细设计 19](#_Toc491267719)

[5 编码 23](#_Toc491267720)

[6 需求管理 24](#_Toc491267721)

[7 软件配置管理 26](#_Toc491267722)

[8 软件质量保证 27](#_Toc491267723)

[9 数据度量和分析 30](#_Toc491267724)

## 一、编写说明

为了把公司已经发布的软件开发过程规范有效地运作于产品开发活动中，把各种规范“逐步形成工程师的作业规范”，特制定本软件开发行为规范，以达到过程控制的目的。

与软件开发相关的所有人员，包括各级经理和工程师都必须遵守本软件开发行为规范。对违反规范的开发行为，必须按照有关管理规定进行处罚。

本软件开发行为规范的内容包括：软件需求分析、软件项目计划、概要设计、详细设计、编码、需求管理、配置管理、软件质量保证、数据度量和分析等。

本软件开发行为规范，采用以下的术语描述：

★ **规则**：在软件开发过程中强制必须遵守的行为规范。

★ **建议**：软件开发过程中必须加以考虑的行为规范。

★ **说明**：对此规则或建议进行必要的解释。

★ **示例**：对此规则或建议从正或反两个方面给出例子。

本软件开发过程行为规范由技术研发部负责解释和维护。

## 二、软件项目整体开发流程



## 三、各阶段岗位职责与工作内容

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 工作名称 | 负责人 | 参与人 | 审批人 | 工作内容 | 交付物 | 工作说明 |
| 1 | 立项管理 | 项目经理 | 售前经理 | 总经理 | 1.项目或产品建设内容；2.项目风险分析；3.明确后续工作；4.讨论解决方案。 | 1.风险分析报告； 2.如需进一步讲解，交付展示PPT; 3.如确定立项，交付立项报告及解决方案 4.立项后，确认开发经理 | 1.立项报告、解决方案提交到开发经理后，开始需求调研准备。 |
| 1.1 | 项目介绍 | 项目经理 | 总经理或售前经理 | 项目经理 | 系统或方案简介 | 无 |
| 2 | 需求分析 | 项目经理 | 售前经理、开发经理 | 总工程师 | 确认用户需求及功能边界 | 需求规格说明书 | 1.需求规格说明书由售前经理编制，提交开发经理后；开发经理开始开发计划编制 |
| 3 | 开发计划 | 开发经理 | 项目经理、售前经理 | 项目经理 | 1.确定开发工期； 2.明确开发人员。 3.开发计划交付甲方 | 项目开发计划书 | 开发经理完成计划编制，人员配置完成后，经项目经理提交客户审核通过，开发经理完成人员分工，开发业务启动 |
| 4 | 软件设计 | 开发经理 | 开发工程师 | 总工程师 | 1.数据库设计 2.概要设计 | 1.数据字典； 2.概要设计说明书 | 公司采用敏捷开发，开发经理需按通用模块-基础数据管理模块-业务管理模块-数据应用模块进行设计，区分无需设计的模块可直接进行开发 |
| 5 | 软件编码 | 开发经理 | 开发工程师、测试工程师 | 项目经理 | 1.完成软件编码； 2.完成详细设计说明书； 3.代码迭代及版本控制 | 1.软件代码及数据库 2.详细设计说明书 | 详细设计说明书由该功能的开发工程师编写 |
| 5.1 | 内部审核 | 开发经理 | 开发工程师 | 总工程师 | 1.审核数据库及代码是否按公司技术规范执行 |  | 采用定期抽样审核方式工作 |
| 5.2 | 版本控制 | 开发经理 |  | 总工程师 | 1.按公司要求进行代码迭代与版本控制； 2.完成代码备份 |  | 各研发组，可自行确认代码进行本地迭代方式，并定期将代码提交贵阳总部迭代、备份 |
| 5.3 | 静态质量审查 | 开发经理 |  | 总工程师 | 代码提交到SonarQube进行静态代码审核 | 代码静态质量审核报告及整改说明 | 进入动态测试环节前，必须提交静态质量报告 |
| 6 | 软件测试 | 测试经理 | 测试工程师、开发工程师 | 总工程师 | 完成软件测试 | 1.测试计划 2.功能测试报告（含测试用例） 3.压力测试报告 | 采用敏捷测试，测试经理根据开发进度，逐个模块跟进测试 |
| 6.1 | 试运行 | 测试经理 | 开发经理 | 项目经理 | 实际生产环境进行软件运行测试 | 1.软件试运行报告 | 取决于甲方是否提供试运行时间 |
| 7 | 软件部署 | 实施经理 | 项目经理、开发工程师 | 实施经理 | 在生产环境进行正式系统部署及投运 | 项目实施报告 |  |
| 8 | 验收交付 | 项目经理 | 实施工程师、售前工程师 | 总经理 | 完成项目验收并交付客户使用 | 验收报告 | 验收通过后，进行项目总结。开发组明确运维职责后，人员开始进入其他项目 |
| 9 | 项目运维 | 实施经理 | 项目经理 |  | 1.及时发现对项目运行期间的问题和客户新需求； 2.需求甄别，需及时更改的提交开发经理； 3.保持客户沟通 | 运维报告、需求更改说明书 |  |

## 四、各阶段工作要求

### 1．软件需求分析

**1-1：软件需求分析必须在产品需求规格的基础上进行，并保证完全实现产品需求规格的定义。**

**1-2：当产品的需求规格发生变更时，必须修订软件需求规格文档。软件需求规格的变更必须经过评审，并保存评审记录。**

**1-3：必须对软件需求规格文档进行正规检视。**

**1-4：软件需求分析过程活动结束前，必须经过评审，并保存评审记录。**

**1-5：在对软件需求规格文档的正规检视或评审时，必须检查软件需求规格文档中需求的清晰性、完备性、兼容性、一致性、正确性、可行性、易修改性、健壮性、易追溯性、易理解性、易测试性和可验证性、性能、功能、接口、数据、可维护性等内容。**

说明：参考建议1-1到1-16。

***1-1：采用以下检查表检查软件需求规格文档中需求的清晰性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 所有定义、实现方法是否清楚地表达了用户的原始要求？ |
| 2 | 在功能实现过程、方法和技术要求的描述上，是否没有背离了功能的实际要求？ |
| 3 | 是否没有不能理解或造成误解的描述 ？ |

***1-2：采用以下检查表检查软件需求规格文档中需求的完备性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 需求定义中是否包含了有关文件（指质量手册、质量计划以及其它有关文件）种所规定的需求定义所应该包含的所有内容？ |
| 2 | 需求定义是否包含了有关功能、性能、限制、目标、质量等方面的所有需求？ |
| 3 | 功能性需求是否覆盖了所有非正常情况的处理？ |
| 4 | 是否对各种操作模式（如正常、非正常、有干扰等）下的环境条件都作了规定？ |
| 5 | 是否对所有功能与时间因素有关的方面都作了考虑？ |
| 6 | 是否标识出了所有与时间因素有关的功能？它们的时间准则是否都说明了？时间准则的最大、最小执行时间是否都定义了？ |
| 7 | 是否标识并定义了在将来可能会变化的需求？ |
| 8 | 是否定义了系统所有的输入？ |
| 9 | 是否标识清楚了系统输入的来源？ |
| 10 | 是否标识出了系统的输出？ |
| 11 | 是否说明了系统输入、输出的类型？ |
| 12 | 是否说明了系统输入、输出的值域、单位、格式等？ |
| 13 | 是否说明了如何进行系统输入的合法性检查？ |
| 14 | 是否定义了系统输入、输出的精度？ |
| 15 | 是否定义了系统性能的各个方面？ |
| 16 | 在不同负载情况下，是否规定了系统的处理能力？ |
| 17 | 在不同情况下，是否规定了系统的响应时间？ |
| 18 | 是否充分定义了关于人机界面的需求？ |
| 19 | 是否对需求定义进行了可行性分析和相关文件（资料）是否已归档？ |
| 20 | 是否对影响需求实现的因素进行了调查，调查结果是否已归档？ |
| 21 | 是否有经济效益分析，分析结果是否已归档？ |
| 22 | 是否详细描述了有关硬件、软件、操作人员、操作过程等方面的安全性？ |
| 23 | 是否评估了本项目对用户、其它系统、环境的影响特性？ |
| 24 | 是否按完成时间、重要性对系统功能、外部接口、性能进行了优先排序？ |

***1-3：采用以下检查表检查软件需求规格文档中需求的兼容性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 界面需求是否使软硬件系统具有兼容性？ |
| 2 | 需求定义的文档是否满足项目文档编写标准？在矛盾时，是否有适当的标准可供选择？ |

***1-4：采用以下检查表检查软件需求规格文档中需求的一致性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 各个需求之间是否一致？是否有冲突和矛盾？ |
| 2 | 所规定的模型、算法和数值方法是否相容？ |
| 3 | 是否使用了标准的术语和定义形式？ |
| 4 | 需求是否与其软硬件操作环境相容？ |
| 5 | 是否说明了软件对其系统和环境的影响？ |
| 6 | 是否说明了环境对软件的影响？ |
| 7 | 所采用的技术是否与用户要求的技术一致？ |

***1-5：采用以下检查表检查软件需求规格文档中需求的正确性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 需求定义是否满足标准的要求？ |
| 2 | 算法和规则是否有科技文献或其它文献作为基础？ |
| 3 | 是否定义了对在错误、风险分析中所标识出的各种故障模式和错误类型所需的反应？ |
| 4 | 是否参照了有关的标准？ |
| 5 | 是否对每一个需求都给出了理由？理由是否充分？ |
| 6 | 对设计和实现的限制是否都有论证？ |

***1-6：采用以下检查表检查软件需求规格文档中需求的可行性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 需求定义是否使软件的设计、实现、操作和维护都可行？ |
| 2 | 所规定的模型、数值方法和算法是否对待解决问题合适？是否能够在相应的限制条件下实现？ |
| 3 | 是否能够达到关于质量的要求？ |

***1-7：采用以下检查表检查软件需求规格文档中需求的易修改性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 对需求定义的描述是否易于修改（如是否采用良好的结构和交叉引用表等）？ |
| 2 | 是否有冗余的信息？是否一个需求被定义了多次？ |

***1-8：采用以下检查表检查软件需求规格文档中需求的健壮性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否有容错的需求？ |

***1-9：采用以下检查表检查软件需求规格文档中需求的易追溯性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否可从上一阶段的文档中找到需求定义中的相应内容？ |
| 2 | 需求定义是否明确地表明前阶段中提出的有关需求和设计限制都已被覆盖了？ |
| 3 | 需求定义是否便于向后继开发阶段查找信息 |

***1-10：采用以下检查表检查软件需求规格文档中需求的易理解性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否每一个需求都只有一种解释？ |
| 2 | 功能性需求是否以模块方式描述的？是否明确地标识出了其功能？ |
| 3 | 是否有术语定义一览表？ |
| 4 | 是否使用了形式化或半形式化的语言？ |
| 5 | 语言是否有歧义性？ |
| 6 | 需求定义中是否只包含了必须的实现细节而不包含不必要的实现细节？是否过分细致了？ |
| 7 | 需求定义是否足够清楚和明确使其能够作为开发设计规约和功能性测试数据的基础？ |
| 8 | 需求定义的描述是否将对程序的需求和所提供的其它信息分离开来了？ |

***1-11：采用以下检查表检查软件需求规格文档中需求的易测试性和可验证性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 需求是否可以验证（即是否可以检验软件是否满足了需求）？ |
| 2 | 是否对每一个需求都指定了验证过程？ |
| 3 | 数学函数的定义是否使用了精确定义的语法和语义符号？ |

***1-12：采用以下检查表检查软件需求规格文档中的性能需求描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
|  | 是否精确的描述了所有的性能需求和可容忍的性能降低程度？对每一个性能应包含两方面的内容： |
| 1 | a. 在最坏情况的执行结果 |
| 2 | b. 本性能失效后，对系统产生的影响 |

***1-13：采用以下检查表检查软件需求规格文档中功能需求描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否清楚、明确地描述了所有的功能？ |
| 2 | 所有已描述的功能是否是必须的？是否能满足任务书或系统目标的要求？ |

***1-14：采用以下检查表检查软件需求规格文档中的接口需求描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否清楚地定义了所有的接口？ |
| 3 | 所有接口是否必须？各接口间的关系是否一致、正确？ |

***1-15：采用以下检查表检查软件需求规格文档中的数据需求描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 在某异常数据（如条件、标志等）下，是否有真正没有考虑到的结果？ |
| 2 | 对异常数据产生的结果是否作了精确的描述？ |

***1-16：采用以下检查表检查软件需求规格文档中的可维护性需求描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 需求定义中是否包括了可行的系统维护方法？ |
| 2 | 软件系统间的关系是否是松耦合的（即能否保证在对某部分修改后，产生最小的连锁效应）？ |

### 2 软件项目计划

**2-1：软件项目计划必须以产品/软件的需求规格为基础。当发生需求更改时，必须修订软件开发计划。**

说明：软件项目计划必须依据需求规格进行制定。项目计划中的工作产品和工作任务应保证能完全实现需求规格的定义。当需求更改时，必须考虑需求更改的相关性，修订相应软件开发计划。

***2-1：制定软件项目计划的活动制定，必须遵守“软件项目计划规范”。***

***2-2：软件经理对软件项目计划的制定和结果负责。***

***2-3：软件经理和相关参与软件项目计划的制定和评审的人员，在参与计划制定之前必须经过软件工程和软件项目计划制定流程的培训。***

**2-2：对于软件项目计划中各项工作产品和工作任务，必须进行规模和工作量的软件估计，并在软件项目计划文档中记录估计的方法和估计数据。**

说明：参考建议2-4到2-8。

***2-4：可以使用PERT统计估计、专家判定平均法、经验类比估计、公式计算等方法，或以上方法的组合，进行软件估计。***

示例：PERT统计估计和经验类比估计的结合

PERT统计估计值 ＝ (最大估计＋4×期望估计＋最小估计〕/ 6

估计记录如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 工作产品任务 | 最大估计 | | 期望估计(根据经验类比获得) | | 最小估计 | | PERT估计 | |
|  | 规模 | 工作量 | 规模 | 工作量 | 规模 | 工作量 | 规模 | 工作量 |
| XX版本（增加XX特性〕话统模块概要设计 | 文档页数:45;增加、修改模块设计数目:12 | 12天 | 文档页数:42;增加、修改模块设计数目:10 | 10天 | 文档页数:30;增加、修改模块设计数目:5 | 5天 | 文档页数:41;增加、修改模块设计数目:10 | 9.5天 |

期望估计值是根据XX版本的话统模块设计的数据获得。

***2-5：对某项工作产品和任务的软件，同时采用两种或以上的方法进行估计，以避免一种方法的偏差。***

***2-6：尽量采用历史经验数据进行软件估计。***

***2-7：参照“软件估计指导书”进行软件估计。***

***2-8：软件估计对应项目的任务分解结构进行。***

说明：软件估计对于项目的任务分解结构对应得越清晰、越细致，相应的估计越准确。

***2-9：在“软件项目计划”中必须包括项目管理活动的计划。***

***2-10：在“软件项目计划”中包括软件重用计划。包括重用软件部件的计划和开发可重用软件部件的计划。***

***2-11： 在“软件项目计划”包括人员的培训计划。***

说明：项目人员计划包括需要的人员类型、数量和技术等级的要求，相关人员的开始工作时间、工作周期、接受培训的计划等。

***2-12：对软件项目进行风险分析与评估。***

说明：可能存在的风险领域含：需求的不明确和变更、外部的限制与对外的依赖、人力资源的到位情况、人力资源的技术等级满足要求状况、技术问题等。

对风险的分析与评估实践包括：

从已知的情况推导出潜在风险；

对风险进行分析，得出：潜在风险可能引发的问题的影响、潜在风险发生的可能性大小、风险发生的时间段等；

排列风险的重点次序；

对风险记录成文件（属于软件项目计划中的一部分）；

风险经受风险影响人审核，并取得他的同意；

根据需要，在开发过程中对风险文档进行维护和修订。

**2-3：对应工作任务，制定项目的文档计划。**

**2-4：软件项目计划中应该包括正规检视活动计划、软件质量保证计划、软件配置管理计划。软件质量保证计划和软件配置管理计划可以和软件项目计划在同一份文档中，也可以分开为三份文档。**

说明：参考建议2-13。

***2-13:软件质量保证计划和软件配置管理计划作为独立的计划文档。***

***2-14：软件项目计划必须是整个项目开发过程的计划，包括测试。***

***2-15：测试经理对照整个开发计划建立软件验证与确认计划。软件验证与确认计划可作为独立的计划文档。***

**2-5：必须对项目工作进行分解，确定项目的工作任务，任务的责任人、资源要求、时间要求、项目的进度。**

**2-6：必须分析任务之间的依赖性，确定并明确标识项目的关键路径。**

**2-7：“软件项目计划”必须按照文档模板的要求编写。项目组可根据项目的实际情况，对文档模板中的内容进行裁减。项目组对文档模板内容的裁减必须得到上级管理部门（包括产品计划处、软件工程组SEPG）的审核批准。**

**2-8：软件项目计划必须经过评审。**

说明：参考建议2-16，。

***2-16：软件项目计划的评审采用以下检查表。***

|  |  |
| --- | --- |
| **序号** | **问题** |
| 1 | 软件项目计划是否完全反映（对应）“软件需求说明书”里的需求？ |
| 2 | 软件项目计划是否有开发方法的说明？ |
| 3 | 软件项目计划是否有资源需求的说明？ |
| 4 | 软件项目计划是否包含风险管理计划？ |
| 5 | 软件项目计划是否包含了版本发布的机制？ |
| 6 | 软件项目计划是否标识了所有必须的培训计划？ |
| 7 | 软件项目计划是否标识了所有内部和外部的传递关系？ |
| 8 | 软件项目计划是否标明了项目的依赖关系？ |
| 9 | 软件项目计划是否标明了角色和职责？ |
| 10 | 软件项目计划是否标明了汇报的机制？ |
| 11 | 软件项目计划是否说明了跟踪和监控机制？ |
| 12 | 软件项目计划是否包含“软件质量保证计划”和“软件配置管理计划”？ |
| 13 | 软件项目计划是否包含项目开发使用的工具？ |
| 14 | 软件项目计划是否包含项目的各里程碑的说明？ |
| 15 | 进度中是否标明了软件项目计划的关键路径？ |

***2-17：参加“软件项目计划”评审的人员，除软件经理和项目组人员外，必须有产品经理、上级管理部门（包括软件工程组SEPG）、SQA人员。***

***2-18：“软件项目计划”通过评审后，软件经理组织相关人员对任务进行承诺，签定工作任务书。***

**2-9：必须对“软件项目计划”进行配置管理，“软件项目计划”的更改必须经过评审。**

**2-10：在开发活动中，必须按照项目跟踪与监控计划和体制，对照“软件项目计划”，跟踪项目开发的实际结果和性能。**

**2-11：当实际结果和“软件项目计划”发生偏离时，必须进行分析，根据分析结果标明纠正措施。必要的情况下，要及时修订“软件项目计划”。**

**2-12：在软件项目跟踪监控活动中，必须定期进行总结和评审，撰写开发状态报告。**

***2-19：根据项目的特点，报告的周期可以为周、双周、月。***

**2-13：在软件开发各里程碑阶段结束前，必须进行阶段评审，对软件项目进行重估计，必要的情况下修订“软件项目计划”。**

***2-20：必须提供相应资源，包括工具和人员等，进行软件项目计划和项目跟踪监控活动。***

**2-14：在软件项目计划和项目跟踪监控过程活动中，必须进行数据度量和分析。**

说明：参见“9. 数据度量和分析”。

### 3 概要设计

**3-1：概要设计要以软件需求规格为基础，必须保证需要实现的需求规格已经被设计。**

**3-2：当需求规格发生变更时，必须修订相关概要设计文档。**

**3-3：在概要设计文档或需求管理文档中，必须记录、验证需求和概要设计的跟踪关系。**

说明：需求和概要设计的跟踪关系可参考建议3-1。

***3-1：采用需求、子系统、模块的跟踪矩阵表记录需求和概要设计的跟踪关系。***

**3-4：必须保证概要设计文档和代码的一致性。当发生设计更改时，必须修订相应设计文档。**

**3-5：必须对概要设计文档进行正规检视。**

**3-6：概要设计过程结束前，必须通过评审，并保存评审记录。**

**3-7：设计更改必须经过相关评审，并保存评审记录。**

**3-8：对概要设计文档的正规检视或评审，必须检查概要设计文档的清晰性、完备性、规范性、一致性、正确性、数据、功能性、接口、详细程度、可维护性、性能、可靠性、可测试性、可追溯性。**

说明：参考建议3-2。

***3-2：采用以下检查表检查概要设计文档的清晰性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 程序结构，包括数据流、控制流和接口的描述是否清楚？ |

***3-3：采用以下检查表检查概要设计文档的完备性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 设计目标是否定义？ |
| 2 | 需求规格评审中不完整的需求(TBD)是否都已经解决? |
| 3 | 如果以前定义的不完整的需求(TBD)发生了改变,本设计是否能够支持? |
| 4 | 是否对不完整需求(TBD)的影响进行了评估? |
| 5 | 对有可能不能实现的设计是否有风险管理计划? |
| 6 | 是否对设计模式进行了描述? |

***3-4：采用以下检查表检查概要设计文档的规范性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 文档是否符合公司模板和写作要求? |

***3-5：采用以下检查表检查概要设计文档的一致性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 2 | 程序、模块、函数、数据成员的名称是否保持一致？ |
| 3 | 设计是否反映了真正的操作环境?硬件环境?软件环境? |
| 4 | 对系统设计的多种可能的描述之间是否保持一致?(例如:静态结构的描述和动态描述) |

***3-6：采用以下检查表检查概要设计文档的正确性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 设计在计划、预算、技术上是否可行？ |
| 2 | 逻辑是否正确和完备？ |

***3-7：采用以下检查表检查概要设计文档的数据描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否对所有的数据成员,参数,对象进行了描述? |
| 2 | 是否所有需要的数据结构都进行了定义,或者定义了不需要的数据结构? |
| 3 | 是否所有的数据成员都进行了足够详细的描述? 数据成员的有效值区间是否定义? |
| 4 | 共享和存储数据的使用是否描述清楚? |

***3-8：采用以下检查表检查概要设计文档的功能性要求。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 模块的规格是否和软件需求文档中的功能需求和软件接口规格要求保持一致. |
| 2 | 是否给每个子模块确定了抽象算法? |
| 3 | 设计和算法是否能满足模块的所有需求? |

***3-9：采用以下检查表检查设计的接口描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否描述了接口的功能特征? |
| 2 | 接口是否便于查错? |
| 3 | 接口相互之间、和其他模块、和需求说明书及接口规格书保持一致？ |
| 4 | 对接口的数量和复杂度进行了有效的平衡，使接口数量控制在一个较小数量，每个接口具有可接受的复杂度？ |
| 5 | 是否所有的接口都能描述了必要的类型、数量、质量等信息？ |
| 6 | 操作界面是否考虑了用户（例如：提供准确、清晰、有用的提示信息）？ |

***3-10：采用以下检查表检查设计的详细程度。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否估计了每个子模块的规模（代码的行数）？是否可信？ |
| 2 | 是否考虑了足够数量及代表性的系统状态？ |
| 3 | 详细程度是否足够进行下一步的详细设计？ |

***3-11：采用以下检查表检查设计的可维护性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否模块化设计？ |
| 2 | 模块是否为高内聚、低耦合？ |

***3-12：采用以下检查表检查设计的性能。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否进行了性能模型分析？ |
| 2 | 是否描述了所有的性能参数？（例如：实时性能约束，存储空间，速度要求，磁盘I/O空间） |
| 3 | 进程是否有时间窗？（例如：需要“加锁”的标记，信号灯，某些代码执行时需要屏蔽中断）？ |
| 4 | 程序执行过程中的关键路径是否都被标识和经过分析？ |

***3-13：采用以下检查表检查设计的可靠性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 设计是否考虑了检错和恢复措施？（例如：输入检查） |
| 2 | 是否考虑了异常情况？ |
| 3 | 是否完全准确描述了所有的出错情况？ |
| 4 | 设计是否能够满足所有系统集成方面的要求？ |

***3-14：采用以下检查表检查设计的可测试性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 设计是否能够被实验、演示或检视以显示它满足了需求？ |
| 2 | 设计是否能够使用以前的测试代码，是否能够进行增量式的测试？ |

***3-15：采用以下检查表检查设计的可追溯性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否每一部分的设计都可以追溯到需求说明书，接口规格说明书、或其他产品文档？ |
| 2 | 是否所有的设计决策都可以追溯到财务分析？ |
| 3 | 对所继承下来的那些特别和不常用的特性对目前设计的影响是否进行了分析？ |
| 4 | 对所继承设计中已知的风险是否进行了定位和分析？ |

### 4 详细设计

**4-1：详细设计要以软件需求规格和概要设计为基础，必须保证需要实现的需求规格已经被设计，必须保证概要设计定义的所有模块已经被详细设计。**

**4-2：当需求规格或概要设计发生变更时，必须修订相关详细设计文档。**

**4-3：在详细设计文档或需求管理文档中，必须记录、验证需求、概要设计、详细设计的跟踪关系。**

说明：需求、概要设计、详细设计的跟踪关系可参考建议4-1。

***4-1：采用需求、子系统、模块、函数的跟踪矩阵表记录需求、概要设计、详细设计的跟踪关系。***

**4-4：必须保证详细设计文档和代码的一致性。当发生设计更改时，必须修订相应设计文档。**

**4-5：必须对重要的详细设计文档进行正规检视。**

说明：参考建议4-2。

***4-2：根据模块的复杂度、规模和在软件系统中的重要程度，选择重要的详细设计文档进行正规检视。在产品中，进行正规检视的详细设计文档比例要达到60%。***

**4-6：详细设计过程结束前，必须通过评审，并保存评审记录。**

**4-7：设计更改必须经过相关评审，并保存评审记录。**

**4-8：对详细设计文档的正规检视或评审，必须检查详细设计文档的清晰性、完备性、规范性、一致性、正确性、数据、功能性、接口、详细程度、可维护性、性能、可靠性、可测试性、可追溯性。**

说明：参考建议4-3。

***4-3：采用以下检查表检查详细设计文档的清晰性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否所有的单元和进程的设计目的都已文档化？ |
| 2 | 单元设计，包括数据流、控制流、接口描述是否清楚？ |
| 3 | 单元的整体功能是否描述清楚？ |

***4-4：采用以下检查表检查详细设计文档的完备性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否提供了所有程序单元的规格？ |
| 2 | 是否描述了所采用的设计标准？ |
| 3 | 是否确定了单元应用的算法？（例如：PDL） |
| 4 | 是否列出了单元的所有调用？ |
| 5 | 是否记录了设计继承的历史和已知的风险？ |

***4-5：采用以下检查表检查详细设计文档的规范性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 文档是否遵从了公司的标准？ |
| 2 | 单元设计是否使用了要求的方法和工具？ |

***4-6：采用以下检查表检查详细设计的一致性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 在单元和单元的接口中数据成员的名称是否保持一致？ |
| 2 | 所有接口之间，接口和接口规格书之间是否保持一致？ |
| 3 | 详细设计和概要设计文档是否能够完全描述“正在构建”的系统 |

***4-7：采用以下检查表检查详细设计的正确性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否有逻辑错误？ |
| 2 | 需要使用常量名称的地方是否有错误？ |
| 3 | 是否所有的条件都被处理？（>,=,< ,switch case)? |
| 4 | 分支所处的状态是否正确？ （逻辑没有搞反） |

***4-8：采用以下检查表检查详细设计的数据描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否所有声明的数据块都已经使用？ |
| 2 | 定位于单元的数据结构是否已经描述？ |
| 3 | 如果有对共享数据、文件的修改，对数据的访问是否按照正确的共享协议进行？（例如：通过信号灯同步进程） |
| 4 | 是否所有的逻辑单元、事件标记、同步标记都已经定义和初始化？ |
| 5 | 是否所有的变量、指针、常量都已经定义并初始化？ |

***4-9：采用以下检查表检查详细设计的功能性要求。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 设计是否使用了指定的算法？ |
| 2 | 设计是否能够满足需求和目的？ |

***4-10：采用以下检查表检查详细设计的接口描述。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 参数表是否在数量、类型和顺序上保持一致？ |
| 2 | 是否所有的输入输出都已经正确定义并检查过？ |
| 3 | 所传递参数的顺序是否描述清楚？ |
| 4 | 参数传递的机制是否确定？ |
| 5 | 通过接口传递的常量和变量是否与单元设计的相同？（例如，函数中定义的常量不能在所调用的子过程中被修改） |
| 6 | 传入、传出函数的参数，控制标记是否都已经描述清楚。 |
| 7 | 是否以度量单位描述了参数的值区间，准确性和精度。 |

***4-11：采用以下检查表检查详细设计的详细程度。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 代码和文档间的展开率是否小于10：1？ |
| 2 | 对模块的所有需求都已经定义？ |
| 3 | 详细程度是否足够开发和维护代码？ |

***4-12：采用以下检查表检查详细设计的可维护性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 单元是否是高内聚和低外部耦合？（例如：单元的改变不会在内部出现不可预见的影响，同时对其他单元的影响最小？ |
| 2 | 是否这种设计是复杂度最小的设计？ |
| 3 | 开始部分的描述是否符合公司的要求？（例如：目的，作者，环境，非标准特性，开发历史，输入输出参数，使用的文件，数据结构，引用此单元的其他单元，注释。 |

***4-13：采用以下检查表检查详细设计的性能。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 进程是否有时间窗？ |
| 2 | 是否所有的时间和空间的限制都已明确？ |

***4-14：采用以下检查表检查详细设计的可靠性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 初始化时是否使用了默认值，是否正确？ |
| 2 | 访问内存时是否进行了边界检查，以保证地址正确？（队列，数据结构，指针，等等） |
| 3 | 对输入、输出、接口和结果是否进行了错误检查？ |
| 4 | 对所有错误情况都安排了有意义的消息反馈？ |
| 5 | 特殊情况下的返回码是否和文档中定义的全局返回码一致？ |
| 6 | 是否考虑了异常情况？ |

***4-15：采用以下检查表检查详细设计的可测试性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否每个单元都可以被测试、演示、分析或者检视，以确认满足需求。 |
| 2 | 设计中是否包括辅助测试的检查点？（例如：条件编译代码、断言等） |
| 3 | 是否所有的逻辑都是可测的? |
| 4 | 是否描述了本单元的测试驱动模块，测试用例集，测试结果? |

***4-16：采用以下检查表检查详细设计的可追溯性。***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 是否每一部分的设计都可以追溯到需求？ |
| 2 | 是否每一个设计决策都可以追溯到效益分析？ |
| 3 | 是否所有的设计决策都可以追溯到成本/效益分析？ |
| 4 | 是不是描述了每个单元的详细需求？ |
| 5 | 单元需求是否能够追溯到软件规格文档（SSD-1）？软件规格文档是否能够跟踪到单元需求？ |
| 6 | 是否有到代码的引用或者包括代码本身？ |

### 5 编码

**5-1：编码必须以设计文档为基础，必须保证所有的设计都被编码实现。当设计发生变更时，必须修改相关代码。**

**5-2：必须保证设计文档和代码的一致性。当代码的修改已经造成设计更改时，必须修订相应设计文档。**

**5-3：必须对重要的代码进行正规检视。**

说明：参考建议5-1。

***5-1：根据模块、函数/单元/进程的复杂度、规模和在软件系统中的重要程度，选择重要的代码进行正规检视。在产品中，进行正规检视的代码比例要达到40%。***

**5-4：在代码已经基线化后，对代码的更改必须通过评审，并保存评审记录。**

**5-5：代码必须遵守相关的XX信息JAVA编程规范。**

**5-6：对代码的正规检视和评审，必须依照”XX信息JAVA编程规范”的规定检查编程规范程度，并对代码符合情况进行考量。**

**5-7：项目编码完成后，必须提交Sonarqube平台进行静态质量检测。**

### 6 需求管理

**6-1：产品项目必须安排人员负责需求管理的职责。**

说明：职责参见建议6-1。

***6-1：需求管理的职责至少应包括以下内容：***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 在产品项目整个生存周期内，管理系统需求和它们的分配，并对其建立文档。 |
| 2 | 实现对系统需求及其分配的更改。 |

**6-2：必须建立文档标识分配到软件中的产品系统需求。**

说明：文档的内容参见建议6-2。

***6-2： 标识分配到软件中的产品系统需求的文档至少应包含以下内容:***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 影响和确定软件项目活动的非技术性需求（即：协议、条件、合同条款等）。 |
| 2 | 对软件的技术需求。 |
| 3 | 用于确认软件产品满足分配需求的验收标准。 |

**6-3：相关人员必须接受需求管理活动方面的培训。**

说明：参见建议6-3**。**

***6-3： 培训至少包括以下内容:***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 项目所使用的方法、标准、规程 |
| 2 | 应用领域的知识 |

**6-4：必须对对经过评审和批准的需求文档进行管理和控制。**

说明：参见建议6-4。

***6-4：对经过评审和批准的需求至少应采用以下方法进行管理和控制：***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 在配置管理计划（SCMP）中将需求文档定义为CI。 |
| 2 | 对需求文档进行配置管理。 |
| 3 | 相应的参考文档进行变更/维护。 |

**6-5：必须对需求变更采用严格的变更控制流程控制。**

说明：参见建议6-5。

***6-5：变更控制流程至少应包含以下内容：***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 对变化的影响进行评估 |
| 2 | 经过CCB组织的评审 |
| 3 | 通知受影响的组和个人 |
| 4 | 跟踪解决该问题，直到关闭 |

**6-6：必须在开发过程中对需求进行跟踪。**

说明：参见建议6-6。

***6-6：需求跟踪活动至少应包括以下内容：***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 按照公司模板制定《需求跟踪说明书》 |
| 2 | 跟踪需求状态的变化 |
| 3 | 需求的跟踪和分配经过评审 |

**6-7：在需求管理活动中必须建立相关度量记录。**

说明：参见建议6-7

***6-7：对需求活动的度量至少应包含以下内容：***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 需求的数量 |
| 2 | 需求的状态 |
| 3 | 需求的类型 |
| 4 | 需求的更改次数 |

**6-8：需求管理活动和其文档必须接受上级管理部门、产品项目经理、SQA的评审。**

### 7 软件配置管理

**7-1：产品项目要任命配置管理的人员和组织，在整个配置管理活动中明确他们的职责。**

说明：参考建议7-1。

***7-1：参照《软件配置管理规范》和《软件配置管理指导书》，任命SCM组织。***

**7-2：产品项目必须制定软件配置管理计划（SCMP)，指导整个配置管理活动。**

说明：参考建议7-2。

***7-2：项目经理根据《配置管理计划（模板）》，负责制定配置管理计划。***

**7-3：软件配置管理计划必须包括如下的内容：**

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 对各阶段应受控的配置项进行选择、分类、标识。 |
| 2 | 定义配置项(CI)的命名惯例 |
| 3 | 定义版本号命名方案 |
| 4 | 制定培训计划 |
| 5 | 定义相关SCM流程 |
| 6 | 制定相应配置评审计划和方法 |

**7-4：软件配置管理计划必须经过由开发人员、产品项目经理、SQA参加的评审，并获得批准，并基线化。**

**7-5：软件配置管理计划和软件项目开发计划必须同步变更。**

**7-6：问题跟踪要有一套流程支持，该流程要包括问题的描述，分类，评估，设计，实现，验证，归档的整个生命过程。**

**7-7：变更申请要有一套流程支持，该流程要保证该变更申请（针对已基线化的配置项）有一个初始化，分类，设计，评估，分派，实现，验证，归档的整个过程。**

**7-8：每个版本有一个符合规范的版本描述文档。**

**7-9：必须定义流程指导配置状态发布。**

说明：参考建议7-3。

***7-3：在配置管理计划中描述配置状态发布的周期，内容和模板。***

**7-10：配置项(CI)的变更和配置管理活动的运行状态通知到相关的部门组织和个人。**

**7-11：定期对变更申请(CR)的处理情况进行统计并将统计和分析结果进行发布，发布内容至少包括：单位时间内处理的CRs数量，CRs分布统计表，CRs流通量统计表，CRs状态分布统计表等。**

说明：参考建议7-4。

***7-4：建议正常情况2周发布一次，更改频繁时是1周，更改较少时是3周***

**7-12：建立可以体现开发版本和基线版本两种不同受控程度的配置库系统**

说明：参考建议7-5。

***7-5：建议使用SCM工具的分支功能实现不同类型的版本控制***

**7-13：制定一个基线化流程指导建立基线。**

说明：参考建议7-6。

***7-6：建议在配置管理计划中对流程进行描述，该流程要保证基线化过程中的物理配置审计（PCA〕,功能配置审计（FCA〕,SQA评审和审计等过程。***

**7-14：内外的发布必须只能来自基线库。**

**7-15：产品项目经理、SQA要定期对SCM的活动和其文档进行评审/检查，输出评审/检查结果，制定并实施改进措施**

**7-16：相关SCM评审要制定相应的Checklist进行指导，评审要有记录。**

### 8 软件质量保证

**8-1：产品项目组要有相关的SQA人员和组织，并开展SQA活动。**

**8-2: 产品项目SQA的组织活动必须通过如下检查。**

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 产品项目是否建立一个独立的、能够支持那些要求独立性活动的SQA组织？对所有项目，SQA功能是否到位？ |
| 2 | SQA组是否有一个向产品组之上的管理者、管理部门报告的渠道？ |
| 3 | 是否为组织进行SQA活动提供足够的资源和费用？ |
| 4 | SQA组的成员是否接受了培训以完成他们的SQA活动？ |
| 5 | 项目的软件相关成员是否接受了有关SQA组任务、职责、权利等的相关培训？ |
| 6 | 上级管理部门是否对产品项目的SQA活动及其结果进行了定期评审？ |
| 7 | 产品项目经理是否定期和事件驱动地参与评审SQA活动? |
| 8 | SQA组活动及其工作产品是否接受了SQA组之外的专家进行的定期评审？ |
| 9 | 项目组是否制定一个执行SQA活动的计划SQAP。如制定了SQA计划，计划的制订是否按照已文档化的组织的SQA规程和SQA计划模版执行？ |

**8-3: 产品项目必须有SQA计划，SQA计划必须通过如下检查。**

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 制定SQA计划的活动是否按照公司的相关规范进行？ 如果存在偏差，是否形成了偏差文档，并得到研究技术管理处的批准？ |
| 2 | SQA计划是否符合公司规范中SQA计划模板的要求？ 如果存在偏差，是否形成了偏差文档，并得到研究技术管理处的批准？ |
| 3 | SQA活动是否按照SQA计划进行？ |
| 4 | SQA计划是否经过计划中涉及的相关组和个人的评审，并得到SQA经理、产品项目经理的批准？ |
| 5 | SQA计划和软件项目计划是否在项目的里程碑处进行了修改，修改是否得到批准？SQA计划和软件项目开发计划是否同步变更？ |

**8-4：SQA必须对产品软件开发过程进行过程审计。**

说明：参考建议8-1。

***8-1：要对以下的过程进行审计：需求分析过程、软件概要设计过程、软件详细设计过程、软件测试过程、版本发布过程、配置管理过程、变更控制过程、需求管理过程。***

**8-5：SQA的过程审计必须通过如下的检查。**

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 产品项目是否明确定义了各种软件活动过程？定义的活动过程是否经过SQA和相关管理部门的批准？ |
| 2 | 软件过程审计是否按照公司制订的软件过程审计规程执行？ |
| 3 | SQA是否对每一个软件活动过程提交了过程审计报告？ |
| 4 | 是否提交了过程不符合项报告？ |
| 5 | SQA的过程审计结果是否通过适当的渠道报告给适当的管理者？ |

**8-6：SQA必须参与项目的技术评审活动。**

说明：参考建议8-2，8-3。

***8-2：SQA必须参与项目的技术评审活动包括：需求评审、系统设计评审、概要设计评审、详细设计评审等 。***

***8-3：SQA在技术评审过程应检查：***

|  |  |
| --- | --- |
| 序号 | 问题 |
| 1 | 技术评审的方法对被评审的软件工作产品是合适的？ |
| 2 | 技术评审的过程是按照公司制订的技术评审过程规程执行的吗？ |
| 3 | 技术评审的结果是否相应的评审规程的要求形成了报告？ |
| 4 | 技术评审的报告，报告给SQA人员了吗？ |
| 5 | SQA人员对技术评审的结果进行分析了吗？ |

**8-7：SQA人员必须定期生成SQA活动的报告。**

说明：参考建议8-4。

***8-4：对SQA报告的检查包括：***

|  |  |
| --- | --- |
| 序号 | 内容 |
| 1 | 是否报告各种软件工作产品的评审记录？ |
| 2 | 报告的评审记录是否符合公司规范的要求？ |
| 3 | 是否有软件过程审计的审计报告？ |
| 4 | 是否把报告送交给上级管理部门、技术管理处、产品项目经理吗？ |
| 5 | 是否有软件过程分析和质量报告？ |

**8-8：产品项目的SQA人员必须制定一个实施SQA工作的月度计划、季度计划，和年度计划。计划必须得到上级SQA经理的评审和批准。**

**8-9：SQA经理应当每月定期地与其下属SQA人员，就其工作的月度计划、季度计划，和年度计划进行协商沟通。**

**8-10：SQA经理应当对其下属的SQA人员的SQA活动的实际完成情况与计划进行监督和管理。对其管辖的SQA人员的SQA活动进行定期地（至少每月一次)审核。**

**8-11：当SQA人员遇到不能在产品项目组内部协商解决的质量问题时，必须将该问题上报给产品/业务部管理者和其上级SQA经理。**

## 9 数据度量和分析

**9-1：每个项目要记录软件过程数据。**

***9-1:建议每个项目建立一个软件过程数据库。***

说明：该数据库可以采用电子表格或NOTES流等形式。

**9-2：项目软件过程数据库得到管理且存取访问受控。**

说明：敏感数据要受到保护。

**9-3：必须对软件规模进行度量。**

说明：按书面规程估计工作产品的规模和更改规模，对所有主要软件工作产品和活动都作估计，软件工作产品能分解到能满足估计所需要的粒度。可由一组同行或专家审核估计值。软件规模度量的例子如下：

功能点数目

代码行数

文档页数

需求数目等的估计值与实际值

**9-4：必须对软件复杂度进行度量。**

**9-5：对工作量进行度量**

说明：按书面规程导出工作量估计，要进行软件工作量度量如下：

工作量在软件生存周期的分布的估计值与实际值

各软件工作产品的工作量的估计值与实际值

独立管理的作业或阶段的工作量的估计值与实际值

SQA工作量的估计值与实际值

SCM工作量的估计值与实际值

准备同行评审的工作量的估计值与实际值

参加同行评审的工作量的估计值与实际值

用于过程评估的工作量的估计值与实际值

用于过程制定和改进的工作量的估计值与实际值

管理项目的工作量的估计值与实际值

项目计划和执行跟踪监控的工作量

重计划工作的工作量

对每个需求的更改提议进行分析的工作量

软件工程组支持其它工程组花费的工作量

其它工程组支持软件工程组花费的工作量

因同行评审返工的工作量，等等。

**9-6：对人员配置进行度量。**

说明：在软件生存周期上的人员分布的计划值与实际值，并记录所有影响承诺的人员变更。

**9-7：对软件进度进行度量。**

说明：按书面规程导出进度估计，应进行度量的例子有：

里程碑完成进度的估计值与实际值

各种活动日期的估计值与实际值

完成工作的进度估计值与实际值

SQA里程碑进度的估计值与实际值

SQA活动完成工作的进度的估计值与实际值

SCM里程碑进度的估计值与实际值

SCM活动完成工作的进度的估计值与实际值

项目计划里程碑的估计值与实际值

过程开发的进度里程碑

过程维护的里程碑

风险管理的进度

**9-8：对生产率进行度量。**

说明：实际生产率与目标生产率比较。

**9-9：对同行评审进行度量。**

说明：对同行评审度量的例子包括：

同行评审的覆盖率

同行评审中发现的问题数目、分类、分布

同行评审的效率

**9-10：对测试进行度量**

说明：对测试度量的例子包括：

测试中发现的问题数目、分类、分布

测试的效率

测试覆盖率

**9-11：对软件质量进行度量**

说明：反映软件质量的度量数据主要有：

软件需求中的缺陷数目

软件代码中的缺陷数目

软件产品中的缺陷数目

软件产品中的缺陷类型

问题存在时间长度，等。

**9-12：对软件稳定度进行度量**

说明：与软件稳定度相关的度量的例子有：

需求变更的频率

缺陷变更的频率

问题解决效率

***9-2：对风险进行度量。***

说明：风险度量的例子有：

风险的优先级（初始的和被修订的）

风险发生的可能性

风险可能造成的损失及不利影响

未预期的不利影响数目

风险管理所需要的人力资源（人员和工具）

***9-3：对软件重用进行度量***

说明：重用度量的例子有：

需求重用数量的估计值与实际值

设计重用数量的估计值与实际值

代码重用数量的估计值与实际值

测试用例重用数量的估计值与实际值

测试规程重用数量的估计值与实际值

***9-4：按照《软件过程数据库规范》中的相关数据进行度量和分析。***

**9-13：产品项目必须定期和阶段性地对过程程数据进行度量和分析，用以分析开发状况，提出产品开发质量和效率的改进建议，并为过程改进提供依据。**