**软件开发计划书**

图书馆管理系统

**负责人：曾宇涛**

**文件版本**

**目录**

# 1 引言

## 1.1标识

文档标识号：A2016-10-05-00-20

文档标题：软件设计说明

项目/产品中文全称：图书管理系统

项目/产品英文全称：Library Management System

项目/产品英文简称：LMS\_1.0

项目/产品编码：201610050020

**1.2 系统概述**

本文档是对图书管理系统项目的相关计划书，该系统最为基本的功能是满足教师及学生自行注册，信息查询，图书借还、预约、延期等等相关操作并且为数据库系统管理员提供了对图书信息录入、编排、修改、删除等操作的接口。为了符合实际需求，并使系统更加人性化，我们将系统使用者分为了两种：读者和系统管理员。读者又分为教师和学生，这个两个角色有着各自的使用操作和不同权限，例如，教师可以申请借阅一些保护类书籍（借阅范围更大）、借阅时间也相对于学生有所延长（借阅允许时间延长），一次性在借书籍数量上限也有所放宽（借阅上限增大）；读者可以在系统中进行注册，能够修改查询自己的个人基本信息、当前借阅、借阅历史等，另外，读者可以查询图书馆馆藏书籍的各类信息（包括书籍的分类、版本、出版时间、作者、出版社、ISBN号、借阅情况、当前可借量）。系统管理员有系统最高权限，可以新增、删除数据库中书籍，修改，查询书籍信息，并且可以对读者进行相应管理。

本系统开发者：计算机学院软件工程课程小组20组。支持机构：北京航空航天大学计算机学院软件工程课程。

**1.3 文档概述**

为了保证项目团队按时保质保量地完成项目开发中的各项目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，故以文件的形式来对项目中的各项问题、安排加以描述，作为项目团队成员以及项目干系人之间的共识与约定，项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据

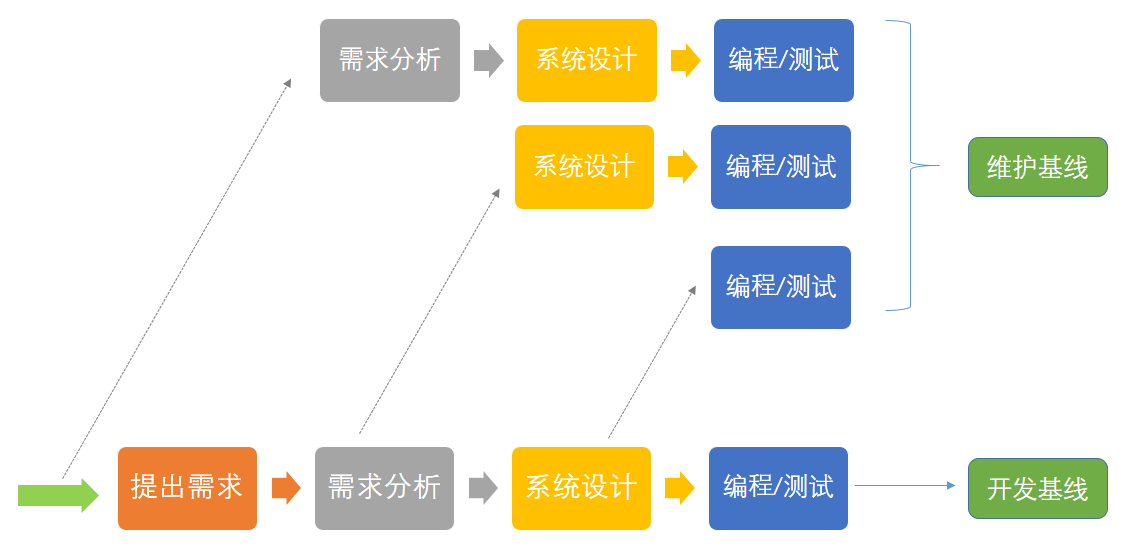
本文档编写于系统需求分析之前，是基于软件工程——软件项目管理课程部分出发，只从设计的角度来描述该系统的总体架构和基本设计思路、扩展，没有涉及到具体的设计内容，是图书管理系统软件开发设计的重要参考。在此基础之上，协同完成项目的开发，确保系统开发的进度与质量。

文档包括的部分有：系统概述、产品交付、工作概述、项目开发计划、项目开发具体细节、进度管理、项目组织和资源、项目预算管理、风险管理等等部分。

本文档仅限开发小组成员及指导老师间共享。

1.4 基线

系统采用瀑布流的开发方式，产品维护基线及开发基线如下：



**1.5 参考资料**

《人月神话》 FrederickP. Brooks. Jr 人民邮电出版社

《计算机软件文档编制规范》 姚淑珍 软件工程课程

《软件工程实践者的研究方法》 Roger S. Pressman 机械工业出版社

# 2 引用文件

由于软件开发计划书是第一个文档，所以无引用文件。

**3 交付产品**

**3.1 程序**

软件名称：图书管理系统

编程语言：Java

开发工具：Eclipse Mars

数据库软件：MySQL

环境支持：Windows 7/8/8.1/10或 Windows XP

功能：基本功能包括读者的注册、查阅修改个人信息、查询书籍信息、借阅/续借/归还书籍；系统管理员增加/删除/修改/查询书籍信息等。书籍信息包括：书名、版本、作者、出版日期、出版社、ISBN编号、书籍数量、书籍当前余量。

**3.2 文档**

用户操作手册：该手册详细介绍了系统的功能模块、性能分析、操作界面，但屏蔽了具体的实现细节，使得用户对该系统有基本的了解并能方便地使用对应的模块功能。另外，手册中还特别包含了操作方法的具体细节，使用系统时相应的注意事项和一些常见问题的解决方案。

软件维护手册：该手册介绍了系统的主要架构和实现流程，详细介绍了软件的模块之间的相应关联、模块功能的实现、系统对硬件和软件环境的具体要求、基本问题的处理方式、维护过程的要求和说明等等，保障软件的基本维护。

**3.3 服务**

为保证用户能够正确而方便地对系统进行使用，我们将会推出以下服务：

视频教程：以视频的形式将使用系统的具体操作方法和流程放在互联网上供人们下载观看教程，具体地了解操作的各个步骤。

技术咨询：在工作时间内可以向技术人员询问系统相关问题，并由其给出相应解答。

远程协助：当客户遇到难以处理且描述的问题，可以向技术人员申请远程支持，协助解决遇到的问题。

软件维护：用户使用过程中遇到Bug和系统缺陷，向技术人员反馈之后，将对系统进行相应的修复，并且为使用方提供系统补丁。

软件升级：当系统开发版本升级时，将向用户提供升级程序

**3.4 非移交产品**

* 系统源代码文件
* 系统相应配置文件
* 《软件开发计划书》——项目计划书
* 《软件需求规格说明书》——项目需求分析
* 《软件设计说明书》——软件设计框架说明
* 《软件测试说明书》——关于系统测试的设计说明书
* 《软件测试报告》——记录说明系统测试结果和相应结论
* 《软件用户手册》——系统基本模块、功能介绍及详细操作说明

**3.5 验收标准**

客户根据软件需求分析说明书中各项需求对系统功能逐项进行测试并且检查通过即认定为项目通过验收。

**3.6最迟交付期限**

项目最迟交付期限是2016年12月20号。

**4 所需工作概述**

**4.1 所开发系统的需求和约束**

系统需要满足包括但不限于图书馆日常基本场景功能的实现。其中包括读者和系统管理员两个角度的各项功能：读者的注册、查阅修改个人信息、查询书籍信息、借阅/续借/归还书籍功能；系统管理员的增加/删除/修改/查询书籍信息等功能。为使得系统管理信息更加清晰，规定书籍信息至少包括：书名、版本、作者、出版日期、出版社、ISBN编号、书籍数量、书籍当前余量等，并且可以提供书籍信息扩展的功能。

**4.2 对文档编制的需求和约束**

项目文档作为对项目的指导或总结（包括但不限于项目开发计划书），需要能够完整而准确地传达相应的信息，达到一定的质量要求，所以需要做到一些几点要求：

针对性：需要分清楚文档使用对象，针对不同类型、不同层次的读者，需要从他们的角度出发，编写适合其阅读和理解的文档，对文档的形式也需要针对不同读者的阅读习惯进行相应的修改。

准确性：文档中对于项目的功能、架构等等的描述应该做到准确无歧义，应该保证用户在使用时不会因此造成阅读障碍。

完整性：对系统的描述应该保证完整，能够让读者在阅读之后对系统有总体上完整清楚的了解。

直观性：可以采用多种形式来对文档进行编写，例如，采用概念图、E-R图、表格等等，使得文档更加直观易懂，降低读者的阅读门槛，并提高阅读文档的效率。

灵活性：针对项目的种类和项目的不同部分，文档的规模和复杂度都应该有所不同，在编写文档时，需要根据相应的实际情况来对文档进行适当的修改。

**4.3项目进度和资源需求**

项目针对实际场景和系统的基本功能实现方式划分不同模块进行开发，同时根据不同的模块的任务量和资源需求进行合理安排模块开发的时限和资源分配，保证能够使得各模块协调有序开展，同时能够满足项目中对于功能质量、进度安排、资源分配的要求。

**4.4其他需求和约束**

**4.4.1 对系统设计原则的约束**

为保证系统设计的合理性，在设计时应该满足以下原则：

先逻辑后物理的原则：图书管理系统是被严格的划分，采用结构化的方法来研究，在综合的基础上进行论证和分析，从而探索和实践的系统逻辑方案，是最终能够为用户解决问题的系统。系统设计应该首先关注需要“做什么”，然后根据系统的分析结果和逻辑程序，进入实施阶段的设计和物理系统。

用户参与原则：图书管理系统是用户直接参与管理的信息系统，是人机交互频繁的系统，会直接参与到辅助和协调工作的任务中去，并最终将成为各阶段建设系统工作的一部分。该系统的用户是学校师生及相关部分图书管理工作者，本系统建设的直接目的就是满足他们在图书管理活动中的信息需求与功能需求。在本系统的开发建设中，要把用户的特点和使用上的习惯与要求放在考虑范畴之内，以调节人机关系和提高人机系统的一体化水平为衡量标准。

“自顶向下”的原则：“自顶向下”的原则需要始终在本系统的开发过程中被

贯彻执行，在系统分析、系统设计与系统实施等各个阶段中都应该能充分体现。一整套的系统分析、设计、开发的完成是建立在以系统的总体目标和功能作为标准，逐级分解，逐级细化的基础上的。这样建立起来的系统，保证了三方面的平衡：结构的合理化，总体与各功能模块能相互协调，总体目标和总体功能的实现具有保障。

**4.4.2 对系统设计/功能的其他约束**

**可扩展性**

可扩展性是指系统确保可持续的增长来满足复杂的业务需求和兼容需求，系统模型中的动态变化刚开始通常表现只是很小的性能/时间开销，但是随着指数增长，需求指数也相应的增长。这种增长的速度是非常之快，它不仅表现在支持用户的数量上，而且也表现在性能和用户的复杂性与整合服务上。提高图书管理系统的功能，是一个渐进的过程，用户的要求和图书管理技术的发展是不断完善的。因此，可扩展性在这里是特别重要的。

**可靠性**

在规定的时间内系统对用户的请求做出响应，并且对响应的质量有所保证，同时保证任意合法用户能够在任意合法时间通过校园网或者是任意合法局域网对系统进行访问获取或修改对应信息这是对系统可靠性的一些重要标准。因此系统的设计必须与硬件、软件支持环境协调，以确保系统的稳定运行。

**易于使用**

由于图书管理系统是人机交互十分频繁的系统，因此保证系统简单、清晰、易懂是对系统质量的一个重要考量部分。用户界面友好、使用简单、操作方便，系统才能更好地协助图书馆进行相关工作的开展。

**可维护性**

系统在保证可扩展性的同时应该能够保证系统的可维护性。系统设计应采取充分考虑了系统的硬件和软件的可维护性，采用可扩展、易维护的硬件系统，同时应用软件系统应该采用模块化设计，系统所提供的服务模块/功能应该清晰、明确，满足在业务量增长和硬件、软件升级换代的情况下系统仍旧能够合理有效地工作，维护人员仍旧能够简单便捷地进行日常维护。

**安全性**

对系统管理员的授权、认证应该保证安全，认证方式应该尽可能多样，出现权限泄露迹象时应该及时预警，如管理员账号异地登录提醒等，，保证管理员权限的安全此外，系统必须留下记录或跟踪记录每一个重要的操作和处理，以便有效地管理相关的权利和责任。同时在设计之时应该保证普通用户和管理员用户权限没有重叠，不同权限用户之间的权限分明，保证不出现普通用户因为系统漏洞而获得高权限的情况。

5 实施整个软件开发活动的计划

5.1 软件开发过程

以下为软件开发各阶段相应安排：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阶段 | 子阶段 | 任务 | 产出 | 验证要求 |
| 项目启动阶段 | 了解项目 | 1、确定项目 2、了解项目相关知识 |  |  |
| 设定计划 | 1、对项目内容进行探讨 2、编写项目计划书 | 《软件开发计划书》 | 评审 |
| 需求分析阶段 | 需求获取 | 1、目标人群问卷调查 2、参照同类系统需求描述 | 系统需求初步调研 |  |
| 需求分析 | 1、需求识别与建模 2、需求分析与综合 3、完成SRS文档(系统需求规格文档)撰写 4、对需求模型和系统需求规格文档进行评审和验证 | 1、系统需求模型 2、SRS文档 | 评审 |
| 系统设计阶段 | 概要设计 | 1、对需求模型进行分析并给出设计思路 2、完成概要设计文档 | 概要设计文档 |  |
| 数据设计 | 1、分析数据特点，归纳数据信息，对接底层数据库 2、归纳得到数据字典 3、数据库底层设计 | 数据字典 |  |
| 详细设计 | 1、软硬件结合完成系统细节设计 2、完成软件结构设计说明书 | 软件结构设计说明书 | 评审 |
| 编程实现阶段 | 各模块实现 | 1、实现读者模块的各项功能 2、实现系统管理员模块各项功能 3、进行相关测试 | 各模块功能源代码 |  |
| 界面UI设计 | 1、编写前台用户界面 2、编写系统后台管理员界面 3、进行相关测试 | 界面源代码 |  |
| 软件集成 | 1、将两大功能模块与数据库集成 2、集成界面、数据库、模块 3、进行相关测试 | 系统Alpha版本 | 集成测试 |
| 系统交付阶段 | 系统测试 | 1、系统安装配置 2、按照《软件测试文件》进行相关测试 3、进行相关调整 | 《软件测试说明文档》 | 评审 |
| 正式交付 | 1、提交项目源代码 2、提交相关文档 3、提交用户手册 | 1、源代码 2、文档 3、用户手册 |  |
| 项目关闭阶段 | 总结 | 1、完成项目总结分析 2、撰写项目总结 | 项目总结报告 |  |
| 维护 | 1、为用户提供技术支持 2、准备后续的系统升级 |  |  |

5.2 软件开发总体计划

5.2.1 软件开发方法

系统的开发过程中将使用面向对象的编程方法，开发模式将采用瀑布流的方式。严格按照这种方式保证软件能够准时保质保量地顺利开发完成。

系统的开发将分为制定计划、需求分析、系统设计、编写程序、软件测试和运行维护等部分，在《项目计划书》、《需求规格说明书》、《软件设计说明书》、《软件测试说明书》等文档的基础上进行系统的开发，并且按照产品开发基线注册有序进行。

5.2.2 处理关键性需求

5.2.2.1 安全性保证

安全性保证建立在系统设计的正确性和完备性上。为保证正确性上，需要系统可能存在的一些漏洞进行严格的审核（比如是否存在书籍信息/个人信息泄露、用户之间权限是否分明，是否能够保证权限隔离、权限不发生泄露，是否存在通过系统漏洞提权的可能性）。在此基础之上，才能确保系统能够正确地用于日常使用。

5.2.2.2 保密性保证

在不同环境中对于系统的保密性要求不同，由于学校图书管理系统常常会被整合到学校系统中，故对于设计保密性较强的部分应该进行特别处理，从而适应相关环境中的使用。但是由于我们的图书管理系统对于保密性的要求不是很高，故一般性的约束即可。

5.2.2.3 私密性保证

保密性需求是建立在不同用户之间信息相对隔绝的基础之上的。为了满足不同用户之间的隔绝，就需要保证用户登录时验证过的完整性和不可越过。系统管理员因为可以进行相应信息的管理，但是也需要做到不能将信息泄露。

5.2.3 计算机硬件资源利用

在预算范围内使用计算机硬件资源如服务器也需要保证能够做到不浪费，尽量提高硬件资源的利用率。在项目不同部分需要根据实际情况对其进行合理的分配，不能因为这个而引起不必要的麻烦。

5.2.4 需方评审途径

在软件开发过程中，需方可以进行项目进度的督促，也能够了解相应开发进度和开发内容；但是对于需方也有一定的限制，需要保证开发程序源代码不能带出，不能泄露开发实现核心细节。不够由于本项目没有很严的要求，所以此点可以适当放宽。

6 实施详细软件开发活动的计划

6.1 项目计划和监督

6.1.1软件开发计划：

|  |  |  |
| --- | --- | --- |
| 任务名称 | 任务内容 | 任务起止时间 |
| 项目开发计划书撰写  （项目启动） | 描述项目系统的总体开发计划，保证项目团队按时保质保量地完成项目开发中的各项目标；对项目中的各项问题、安排加以描述，作为项目团队成员以及项目干系人之间的共识与约定；作为项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据。 | 2016.09.22-2016.10.10 |
| 需求规格说明书撰写  （需求分析） | 分析使用者对系统的使用要求及对应的功能具体划分，对系统功能需求、性能需求和其它非功能性需求进行详细的描述，并以可验证的方式做出完全、精确陈述，为之后的系统设计提供基础，同时也为后面编写测试样例和系统测试提供依据。 | 2016.10.11-2016.10.25 |
| 结构设计说明书撰写  （系统设计） | 确定系统的总体设计、总体结构、模块设计、接口设计，并给出详细设计供软件编程人员参考，用于指导各构件的实施、集成及测试；通过使用多种视图来从不同角度描述本系统的各个主要方面，以满足图书杂志采购和借阅系统的相关涉众对本系统的不同关注焦点和需求；记录并表述了系统架构的设计人员对系统构架方面做出的重要决策；将本文档中所涉及的所有术语、缩略语进行详细的定义（数据字典）。 | 2016.10.26-2016.11.10 |
| 软件测试说明书撰写  （测试计划） | 为了在软件投入生产性运行之前，尽可能多地发现软件的错误。文档将围绕这几个方面出发，设计相关的测试方法/样例，并且记录预期结果供开发人员参考：  依据需求分析中的各项需求逐项分析是否完成相应功能的开发；  测试系统中的各个功能模块是否满足用户要求，并测试是否存在bug，是否达到预期；  测试系统是否能够使系统进行快速的改进和升级；  测试系统是否满足性能和效率的要求；  重复测试验证系统的可靠性和对软硬件环境的适应性。 | 2016.11.11-2016.11.25 |
| 程序编写、调试和系统测试（系统实现） | 以项目计划书、需求分析规格说明书、结构设计说明书、软件测试说明书这些文档为基础，设计编写相应程序和调试工作，采用模块化编程和边开发边测试的基本模式，对每个模块都安排专人进行单独测试，系统联调及系统测试，对系统处理逻辑、例外处理能力、容错能力等进行大规模的测试，对发现的问题进行彻底纠正。 | 2016.11.26-2016.12.20 |
| 项目交付及总结  （系统交付） | 对系统源代码及相关文档进行交付并作出相关总结 | 2016.12.20-2016.12.20 |

6.1.1软件开发计划

6.1.2CSCI测试计划

CSCI是计算机软件配置项(Computer Software Configuration Item)，对于软件配置项的测试主要针对系统在不同软件环境下安装配置时可能出现的问题。

|  |  |  |
| --- | --- | --- |
| 系统 | MySQL | 是否能够正常运行 |
| Win7 | 5.6 |  |
| Win8 | 5.6 |  |
| Win8.1 | 5.6 |  |
| Win10 | 5.6 |  |
| Win7 | 5.7 |  |
| Win8 | 5.7 |  |
| Win8.1 | 5.7 |  |
| Win10 | 5.7 |  |

6.1.3系统测试计划

系统测试分以下四步进行：单元测试、集成测试、系统测试、验收测试。由于验收测试是需方共同参与下进行，故在此仅仅进行前三项的测试计划。但是要注意这几项测试的区别：

单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。因为单元测试需要知道内部程序设计和编码的细节知识，一般应由程序员而非测试员来完成，往往需要开发测试驱动模块和桩模块来辅助完成单元测试。

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。根据集成测试计划，一边将模块或其他软件单位组合成越来越大的系统，一边运行该系统，以分析所组成的系统是否正确，各组成部分是否合拍。

系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求。系统测试应该按照测试计划进行，其输入、输出和其他动态运行行为应该与软件规约进行对比。

6.1.4软件安装计划

6.1.5软件移交计划

6.1.6跟踪和更新计划，包括评审管理的时间间隔

6.2 建立软件开发环境

建立、控制、维护软件开发环境所遵循的方法。

6.2.1软件工程环境

硬件：

内存：4G及以上

硬盘：500G以上；

CD－ROM：32倍速以上；

网络： 一台服务器

一台工作站

使用TCP/IP协议的局域网

软件：

操作系统为Windows 10，使用集成开发工具Eclipse，编程语言为Java，数据库采用MySQL，项目运行环境为JDK1.8

6.2.2软件测试环境

6.2.3软件开发库

6.2.4软件开发文档

6.2.5非交付软件

6.3 系统需求分析

6.3.1用户输入分析

6.3.2运行概念

6.3.3系统需求

6.4 系统设计

6.4.1系统级设计决策

6.4.2系统体系结构设计

6.5 软件需求分析

需求分析中要遵循的方法

6.6 软件设计

软件设计中所遵循的方法。

6.6.1CSCI级设计决策

6.6.2CSCI体系结构设计

6.6.3CSCI详细设计

6.7 软件实现和配置项测试

软件实现和配置项测试中要遵循的方法。

6.7.1软件实现

6.7.2配置项测试准备

6.7.3配置项测试执行

6.7.4修改和再测试

6.7.5配置项测试结果分析与记录

6.8 配置项集成和测试

配置项集成和测试中要遵循的方法。

6.8.1配置项集成和测试准备

6.8.2配置项集成和测试执行

6.8.3修改和再测试

6.8.4配置项集成和测试结果分析与记录

6.9CSCI 合格性测试

CSCI合格性测试中要遵循的方法。

6.9.1CSCI合格性测试的独立性

6.9.2在目标计算机系统(或模拟的环境)上测试

6.9.3CSCI合格性测试准备

6.9.4CSCI合格性测试演练

6.9.5CSCI合格性测试执行

6.9.6修改和再测试

6.9.7CSCI合格性测试结果分析与记录

6.10CSCI/HWCI 集成和测试

CSCI/HWCI集成和测试中要遵循的方法

6.10.1CSCI/HWCI集成和测试准备

6.10.2CSCI/HWCI集成和测试执行

6.10.3修改和再测试

6.10.4CSCI/HWCI集成和测试结果分析与记录

6.11 系统合格性测试

系统合格性测试中要遵循的方法

6.11.1系统合格性测试的独立性

6.11.2在目标计算机系统(或模拟的环境)上测试

6.11.3系统合格性测试准备

6.11.4系统合格性测试演练

6.11.5系统合格性测试执行

6.11.6修改和再测试

6.11.7系统合格性测试结果分析与记录

6.12 软件使用准备

软件应用准备中要遵循的方法

6.12.1可执行软件的准备

6.12.2用户现场的版本说明的准备

6.12.3用户手册的准备

6.12.4在用户现场安装

6.13 软件移交准备

软件移交准备要遵循的方法

6.13.1可执行软件的准备

6.13.2源文件准备

6.13.3支持现场的版本说明的准备

6.13.4“已完成”的CSCI设计和其他的软件支持信息的

准备

6.13.5系统设计说明的更新

6.13.6支持手册准备

6.13.7到指定支持现场的移交

6.14 软件配置管理

软件配置管理中要遵循的方法

6.14.1配置标识

6.14.2配置控制

6.14.3配置状态统计

6.14.4配置审核

6.14.5发行管理和交付

6.15 软件产品评估

软件产品评估中要遵循的方法

6.15.1中间阶段的和最终的软件产品评估

6.15.2软件产品评估记录(包括所记录的具体条目)

6.15.3软件产品评估的独立性

6.16 软件质量保证

软件质量保证中要遵循的方法

6.16.1软件质量保证评估

6.16.2软件质量保证记录、包括所记录的具体条目

6.16.3软件质量保证的独立性

6.17 问题解决过程( ( 更正活动) )

软件更正活动中要遵循的方法

6.17.1问题/变更报告

6.17.2更正活动系统

6.18 联合评审( ( 联合技术评审和联合管理评审) )

进行联合技术评审和联合管理评审要遵循的方法。

6.18.1联合技术评审包括----组建议的评审

6.18.2联合管理评审包括----组建议的评审

6.19 文档编制

文档编制要遵循的方法

6.20 其他软件开发活动

进行其他软件开发活动要遵循的方法

6.20.1风险管理，包括已知的风险和相应的对策

6.20.2软件管理指标，包括要使用的指标

6.20.3保密性和私密性

6.20.4分承包方管理

6.20.5与软件独立验证与确认(IV&V)机构的接口

6.20.6和有关开发方的协调

6.20.7项目过程的改进

6.20.8计划中未提及的其他活动