基于MPXJ的\*.mpp文件信息读写的JAVA桌面应用——Mini Project

**项目总结报告**

小组成员：

SY1506104 刘克瑞

SY1506106 詹鹏飞

SY1506114 陈阳

SY1506115 彭柯宾

目录

[1 概述 3](#_Toc454463905)

[2 实验重点 3](#_Toc454463906)

[2.1 软件项目计划 3](#_Toc454463907)

[2.2 软件需求分析 4](#_Toc454463908)

[2.3 软件需求评审 5](#_Toc454463909)

[2.4 软件产品改进与展示 5](#_Toc454463910)

[2.4.1 改进方案 6](#_Toc454463911)

[2.4.2 具体实现 7](#_Toc454463912)

[2.5 软件测试 8](#_Toc454463913)

[2.5.1测试需求定义和测试用例的设计 8](#_Toc454463914)

[2.5.2 测试结果 8](#_Toc454463915)

[2.5.3 发现问题 9](#_Toc454463916)

[2.6 软件测试需求评审 10](#_Toc454463917)

[2.7 软件进度计划与控制 12](#_Toc454463918)

[2.7.1 项目执行过程与特点分析 12](#_Toc454463919)

[2.7.2 计划变更及其影响因素分析 13](#_Toc454463920)

[2.7.3 工作量分配及其影响因素分析 15](#_Toc454463921)

[2.8 配置管理 16](#_Toc454463922)

[2.9 工作量估计与统计分析 16](#_Toc454463923)

[3 修订说明 16](#_Toc454463924)

[3.1 软件项目计划 16](#_Toc454463925)

[3.2 软件需求分析 17](#_Toc454463926)

[3.3 测试分求分析 19](#_Toc454463927)

[3.4 测试报告 19](#_Toc454463928)

[4 总结 19](#_Toc454463929)

[4.1 经验与启发 19](#_Toc454463930)

[4.2 有效方法 20](#_Toc454463931)

[5 建议 21](#_Toc454463932)

# 概述

本次软件工程综合实验过程在遵循软件工程开发过程的基础上，进行了适当的调整和创新，具体过程包括：软件项目计划阶段、软件需求分析阶段、软件需求评审阶段、软件产品改进与展示阶段、软件测试阶段、软件测试评审阶段。除此之外，还进行了软件进度计划与控制、工作量估计与统计分析、配置管理这些工作，贯穿在整个项目执行的过程之中。

在实验过程中，既完成了软件工程开发所要求的开发过程，又在开发过程中通过软件进度计划与控制、工作量估计与统计分析、配置管理这些工作对整个项目开发进行管理和控制，在保证项目按照计划执行的同时，还保留了项目执行和开发过程中的各项证据，进一步形成了证据链，使得项目的开发与执行更具有说服力。

# 实验重点

## 2.1 软件项目计划

在软件项目的计划阶段，主要进行了一些计划和预备的工作，例如确定了小组成员，确定了实验项目——Mini Project，根据小组成员的具体情况（能力、精力）确定了人员分工，根据课程的安排确定了项目计划，并依照项目计划撰写了项目计划书。项目计划和项目计划书的文档截图如图1所示。

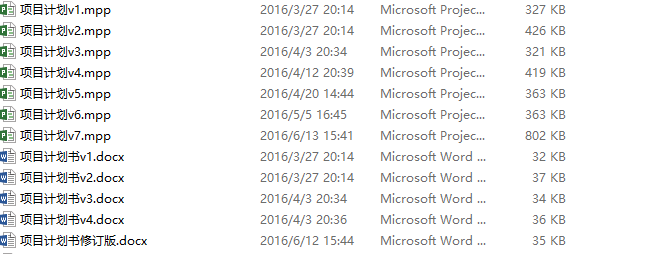


图1 项目计划和项目计划书的文档截图

## 2.2 软件需求分析

在软件需求分析阶段，首先根据所选软件的功能获取并说明本软件的功能性需求，之后对需求进行确认和验证，然后撰写需求规格说明书，最后对需求规格说明书进行修订。需求分析的用例图如图2所示。

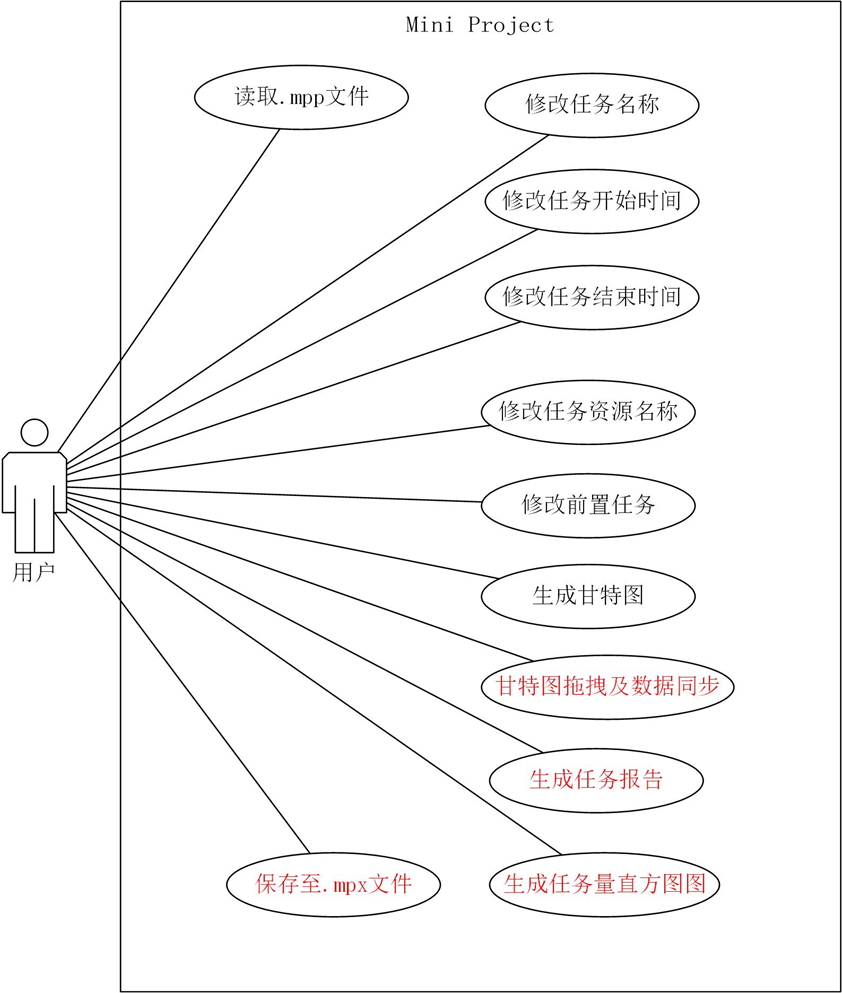


图2 用例图

## 2.3 软件需求评审

在软件需求评审阶段，主要根据软件需求分析规格说明书进行了网上评审和会议评审。其次收集老师和同学的评审意见和建议，撰写评审报告和问题清单，最后根据评审报告和问题清单修改和完善需求。需求评审的问题报告清单如表1所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 需求分析评审对象 | 评审意见 | 修改情况 |
| 1 | 软件名称 | 软件命名不够统一，定义不清晰 | 接受建议，已在v3版中修改，确定名称为MiniProject |
| 2 | 用例图 | 用户类型的分析和定义不够明确 | 接受建议，已在v3版中重新定义用户类型 |
| 3 | 提问环节 | 提前准备不足，没有针对性的建议 | 接受建议，在论坛上多多互动，提前阅读其他组的文档 |
| 4 | 项目计划 | 项目计划需要不断更新 | 接受建议，已在项目计划v4版中根据项目进度和实验安排进一步修改 |
| 5 | 引言软件需求分析理论部分 | 描述过于口语化 | 接受建议，已在v3版中进行修改 |
| 6 | 用户类型分析 | 用户类型与用例图不符 | 接受建议，已在最终版中删除用户类型分析一节 |
| 7 | 用例101 | 没有异常流的说明 | 接受建议，已在v3版中进行说明 |

**表 1 需求评审问题报告清单**

## 2.4 软件产品改进与展示

在软件产品改进与展示阶段，主要进行了如下的工作：改进思路的汇报和展示，根据老师和同学的建议对改进方案进行修改，设计改进方案，改进方案的落实和课堂展示，实施改进方案，改进方案实施情况汇报和展示，最后完成软件的改进与展示。

### 2.4.1 改进方案

* 基本目标

1. MiniProject可以实现Microsoft Project的基本功能，包括：读取mpp文件、修改任务名称、修改任务模式、修改任务开始/结束时间、修改任务前置条件、修改任务的资源名称、生成甘特图。

2. 具有比较简洁、友好的用户界面。

* 具体方案

使用net.sf.mpxj工具来提取mpp文件中的文本内容，之后将文本内容导入格式为mpx的文件中，在mpx文件中进行内容修改和编辑。因为Microsoft Project可以支持mpp和mpx两种格式的文件，所以修改后的mpx文件可以在Microsoft Project下正常打开。

* 难点和问题

1、因为修改之后只能保存为mpx格式，因此只能间接的对原mpp文件进行修改。

2、读取内容时对mpp文件的格式有具体的要求和限制。

3、读取mpp文件时，需要根据对mpp文件的版本进行设计相应的读取模块。

4、界面设计

* 创新与改进

1、寻找更加优化的方法来实现对mpp文件的编辑

2、在实现基本功能之后，尝试增加一些新的功能，比如任务报告、甘特图拖拽与数据同步、任务量直方图等。

### 2.4.2 具体实现

* 生成report模块

1、Report模块是对已有的项目计划进行的一个文字性的总结,包括了项目的各个方面:日期规划,工作量,花费等功能。

2、以总结的方式来进行呈现,方便阅读。

3、主要功能:500行代码.绘制UI及其他100行左右。

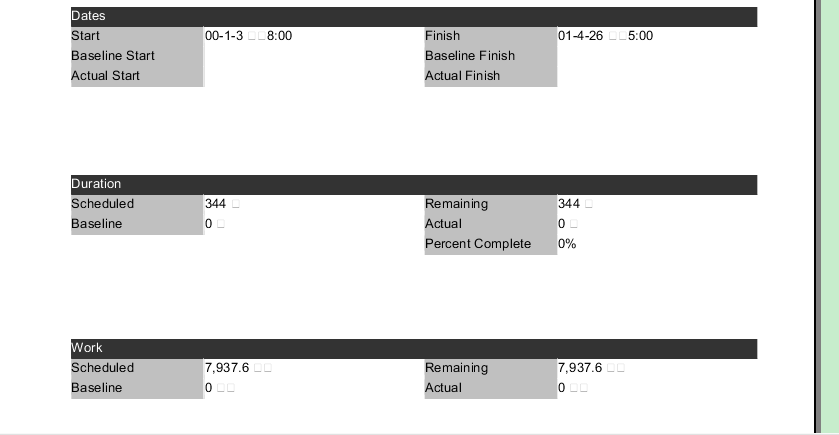


图3 “生成report”模块截图

* 生成工作量直方图

用图表的形式来反映每一个人在某一个日期的工作量。

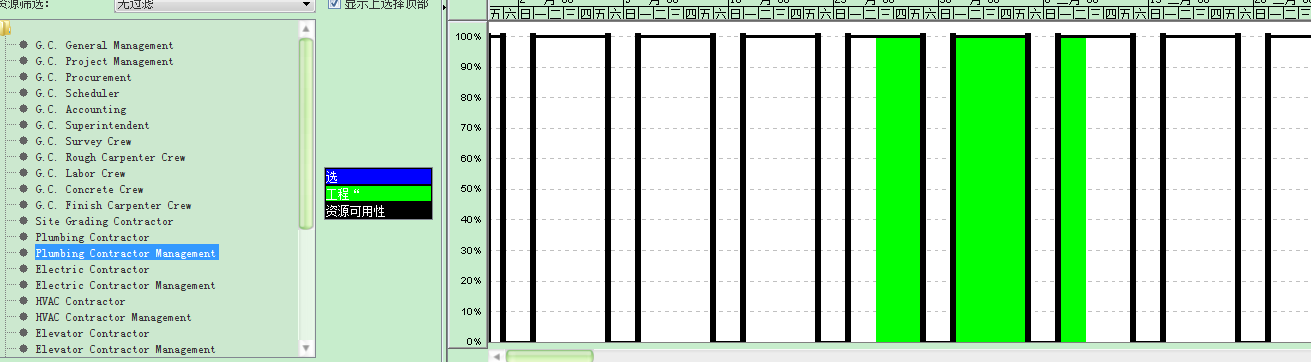


图4 “生成工作量直方图”模块截图

## 2.5 软件测试

在软件测试阶段，首先根据需求用例来定义测试的需求和设计测试用例，其次撰写测试需求规格说明书，然后根据测试需求规格说明书对软件进行测试，根据测试的结果撰写软件测试报告，最后对测试发现的问题进行修改和完善。

### 2.5.1测试需求定义和测试用例的设计

**表2 需求用例和测试用例对照表**

|  |  |  |
| --- | --- | --- |
| 需求规格说明书 | | 测试用例 |
| 功能测试 | 用例101 读取\*.mpp文件 | 用例101 读取\*.mpp文件 |
| 用例102 修改任务名称 | 用例102 修改任务名称 |
| 用例103 修改任务开始时间 | 用例103 修改任务开始时间 |
| 用例104 修改任务结束时间 | 用例104 修改任务结束时间 |
| 用例105 修改任务的前置任务 | 用例105 修改前置任务 |
| 用例106 修改任务的资源名称 | 用例106 修改任务资源名称 |
| 用例107 生成任务甘特图 | 用例107 生成任务甘特图 |
| 用例108 拖拽甘特图，可同步到任务表中 | 用例108 拖拽甘特图，可同步到任务表中 |
| 用例109 生成任务量直方图 | 用例109 生成任务量直方图 |
| 用例110 生成任务报告 | 用例110 生成任务报告 |
| 用例111 保存至\*.mpx文件中 | 用例111 保存至\*.mpx文件中 |
| 用例112 另存为\*.mpx文件 |
| 非功能测试 | 用户界面部分 | 用例113 用户界面测试 |
| 性能需求 | 用例114 进行性能测试 |

### 2.5.2 测试结果

功能性测试

* + 101 读取\*.mpp文件 通过
  + 102 修改任务名称 通过
  + 103 修改任务开始时间 通过
  + 104 修改任务结束时间 通过
  + 105 修改前置任务 通过
  + 106 修改任务资源名称 通过
  + 107 生成任务甘特图 通过
  + 108 拖拽甘特图，同步到任务表中 通过
  + 109 生成任务量直方图 通过
  + 110 生成任务报告 通过
  + 111 保存\*.mpx文件 通过
  + 112 另存为\*.mpx文件 通过
  + 113 用户界面测试 通过

非功能性测试

* + 打开测试文件 2秒内显示内容 通过
  + 生成任务量直方图 2秒内响应 通过
  + 拖拽甘特图 2秒内任务表发生变化 通过
  + 保存文件 2秒内完成保存 通过

### 2.5.3 发现问题

1、测试用例101 打开\*.mpp文件中，读取非\*.mpp文件的时候无错误提示。

2、进行“新建”或是“保存”工作时弹出的确认窗口会有错字。

## 2.6 软件测试需求评审

在软件测试需求评审阶段，主要进行了软件测试需求自评，软件测试需求互评的工作，之后根据评审的意见和建议撰写测试需求自查问题清单和测试互评问题清单。测试自查问题清单和测试互评问题清单如表3、4所示。

**表3 测试自查问题清单**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 问题位置 | 问题说明 | 处理意见 |
| 1 | 文档修改记录 | 文档修改没有及时填写 | 已填写 |
| 2 | 3.测试策略 | 有些测试策略可能未用到 | 待进一步确定细化 |
| 3 | 4.测试用例 | 测试用例还不够完善 | 正在细化 |
| 4 | 2.3测试数据 | 需要增加章节，解释测试数据的获取来源 | 已添加 |
| 5 | 2.4测试通过准则 | 没有测试通过准则 | 已添加 |

**表4 测试互评问题清单**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | 问题位置 | 问题说明 | 问题分类 | 评审组 | 处理意见 |
| 1 | 封面 | 第一页的题目，可以调整一下，把“桌”和“面”字合在一起，分开有点难看 | 格式问题 | E | 已修改 |
| 2 | 目录 | 第三章内容目录漏了 | 格式问题 | F | 已修改 |
| 3 | 目录 | 目录页中，“目录”2字是否可以居中 | 格式问题 | E | 已修改 |
| 4 | 文档修改记录 | 文档修改记录也应该包括文件生成的信息 | 内容遗漏 | F | 已修改 |
| 5 | 文档修改记录 | 文档修改记录为空表格 | 内容遗漏 | E | 已修改 |
| 6 | 1.1 | 编写目的：“对于进行软件的测试，需要拥有一个全面精确有预见性的设计来保证软件项目的顺利研发”  有点不通顺 | 用语问题 | E | 已修改 |
| 7 | 1.2.1 | 功能需求："如果有错误输入数据时的响应等。"格式不对吧，是否少了标点符号 | 格式问题 | E | 已修改 |
| 8 | 1.2.1 | 功能需求：“性能需求：静态量化、动态量化等。”，静态量化，和动态量化，是啥意思？有区别吗？ | 用语问题 | E | 已删除 |
| 9 | 1.2.1 | 功能需求最后一行有错别字，“用户借口部分” | 错别字 | E | 已修改 |
| 10 | 1.3 | 用语定义是否要补充 | 内容遗漏 | F | 已补充 |
| 11 | 1.3 | Mpxj是什么意思 | 用语问题  内容遗漏 | E | 已补充，见用语定义 |
| 12 | 1.5 | 参考文献：信息应该更加详细，方便别人参考 | 内容遗漏 | F | 已细化 |
| 13 | 2.1 | 百度超链接没去除 | 格式问题 | F | 已修改 |
| 14 | 2.1 | 项目描述：1，2，3没有句号结束 | 格式问题 | E | 已修改 |
| 15 | 2.2 | 主机的操作系统已经屏蔽了硬件差异，为什么还有硬件配置的要求 | 内容多余 | F | 该部分已删除 |
| 16 | 2.2.1 | 目标及架构是不是都能测试到 | 测试问题 | F | 无法覆盖全部，该部分已删除 |
| 17 | 2.2.2 | 主机中所有系统能否检测到（而且名字是否有错） | 测试问题 | F | 该部分已删除 |
| 18 | 3.1&3.3 | 3.1和3.3感觉有些重复 | 内容多余 | F | 已修改 |
| 19 | 3.3 | 集成测试：表格中的“技术”是指什么？是集成测试采用的技术吗？ | 用语问题 | E | 是的 |
| 20 | 3.5 | 用户界面测试与功能没看出区别。（鼠标键盘都是外设） | 测试问题 | F | 用户界面测试段可删除，待定 |
| 21 | 3.6 | 性能测试的具体量化预估 | 测试问题 | F | 已量化 |
| 22 | 3.7&3.8 | 针对需求测试不需要添加配置测试、负载测试这块内容，与测试环境相关 | 内容多余 | F | 已删除 |
| 23 | 测试用例 | Test Oracle应该描述结果的预测，不应该添加是否 | 测试问题 | F | 已修改 |
| 24 | 测试用例 | 有\*8问题中类似问题 | ？ | F | ？ |
| 25 | 测试用例 | 除了功能测试外，其余的测试策略所涉及的测试都没有在用例中显示 | 测试问题 | F | 已添加部分用例，如：“用户界面测试”、“进行性能测试” |
| 26 | 测试用例 | 测试用例描述前面的表格是否应该有个标识符 | 测试问题 | F | ？ |
| 27 | 测试用例 | 表格的名字应该居中 | 格式问题 | E | 已修改 |
| 28 | 测试表格 | 文档中的测试表格内容填写不全 | 内容遗漏 | E | 已修改 |
| 29 | 用例101 | 读取\*.mpp文件  Tester 是否应该具体的测试人员？而不是笼统的 "测试人员"。 | 用语问题 | E | 我们的测试人员只有一种类型，所以统称为“测试人员” |
| 30 | 用例102 | 缺少测试前提或者说是Test Setup应该要补充 | 内容遗漏 | F | 测试前提已写到“打开mini project程序  系统读取.mpp文件无异常“ |
| 31 | 用例109 | 用例109 拖拽甘特图，同步到任务表中  “拖拽甘特图使起始时刻提前与现在时刻”，有错别字吧？与？ | 错别字 | E | 已修改 |
| 32 | 用例112 | 另存为\*.mpx文件  “在另存为目录中多出指定名称的mpx文件”，一句不通顺。 | 用语问题 | E | 已修改 |

## 2.7 软件进度计划与控制

### 2.7.1 项目执行过程与特点分析

本项目的执行过程在遵循软件工程开发过程的基础上，进行了适当的调整和创新，具体过程包括：软件项目计划阶段、软件需求分析阶段、软件需求评审阶段、软件产品改进与展示阶段、软件测试阶段、软件测试评审阶段。除此之外，还进行了软件进度计划与控制、工作量估计与统计分析、配置管理这些工作，贯穿在整个项目执行的过程之中。

本项目在执行过程中既完成了软件工程开发所要求的开发过程，又在开发过程中通过软件进度计划与控制、工作量估计与统计分析、配置管理这些工作对整个项目开发进行管理和控制，在保证项目按照计划执行的同时，还保留了项目执行和开发过程中的各项证据，进一步形成了证据链，使得项目的开发与执行更具有说服力。

### 2.7.2 计划变更及其影响因素分析

#### 2.7.2.1 计划变更情况

本项目计划变更的具体情况如表5所示。

**表5 项目计划变更情况**

|  |  |  |  |
| --- | --- | --- | --- |
| 项目计划版本号 | 变更情况 | 变更原因 | 时间 |
| V1 | 初步确定项目计划，无变更 | 无 | 20160320 |
| V2 | 修改了资源分配的情况 | 老师提出资源分配需要细化 | 20160322 |
| V3 | 修改了需求评审部分的计划，延后了需求评审的时间 | 老师在课程上安排的变动对项目进度的影响 | 20160329 |
| V4 | 修改了项目计划表，增加了实际工时、实际开始时间和实际完成时间，表明了和更新了项目进度 | 老师根据通过对项目计划表的审查，提出了改进和完善意见 | 20160403 |
| 更新了项目进度，完成了需求分析复评审任务 | 需求分析复评审结束，项目计划需要与实际项目进度保持一致 | 20160412 |
| V5 | 修改了需求分析到改进与展示阶段的计划方案，更新了项目进度 | 实际的安排情况与项目计划不符 | 20160420 |
| V6 | 更新了项目展示与改进部分的进度 | 项目展示与改进完成，项目计划与实际情况需要保持一致 | 20160505 |
| V7 | 更新了项目计划的测试部分 | 软件测试部分完成，项目计划与实际情况需要保持一致 | 20160526 |
| 添加了基线，对工时进行了完善和修改 | 项目的工时部分出现问题，没有设定基线部分 | 20160607 |

#### 2.7.2.2 数据分析

从上表统计的数据可以看出，V1是项目计划初稿，根据整个课程的课时初步进行项目计划，因此无变更情况和变更原因；V2是在V1的基础上，根据老师在课堂上提出资源分配的问题进行完善和修改的；V3是因为老师在课程上安排的变动，将需求评审向后延长了一周，因此需要更改项目计划；V4是根据老师从项目计划审查之后提出的有关工时、实际开始时间、实际完成时间的问题在V3的基础上对项目计划进行修改，并且根据项目的实施进展情况对项目计划进度进行更新，更新到需求评审阶段；V5是因为需求分析阶段到更新与展示的阶段的计划与实际情况不相符，因此对V4进行了修改和完善；V6是在V5的基础根据项目实际进展情况对项目计划进度进行了更新。V7在V6的基础上更新了测试部分的计划进度，并且对项目的工时和基线的部分进行了完善和修改。

项目计划一共有7个版本，版本数过多，像V5和V6一些细小的更新和改动没有必要重新生成一个版本。影响项目计划变更的主要因素除了老师课程上的变动之外，就是实际项目进展不顺利造成的实际项目进度与项目计划不相符。进展不顺利主要是因为小组成员在项目执行过程中执行力不足。

### 2.7.3 工作量分配及其影响因素分析

#### 2.7.3.1 工作分配情况

工作分配的具体情况如表6所示。

**表6 小组工作分配表**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 性别 | 分工 | 职位 |
| 陈阳 | 男 | 需求分析阶段的文档工作、测试阶段文档和测试工作 | 组员 |
| 刘克瑞 | 男 | 项目计划、主持会议、需求分析阶段的文档工作、需求评审文档工作、改进与展示阶段文档和少量代码工作、项目统计分析、配置管理、工作量统计与分析 | 组长 |
| 彭柯宾 | 男 | 需求分析阶段文档工作、改进与展示阶段主要代码工作和文档工作、会议记录 | 组员 |
| 詹鹏飞 | 男 | 需求分析阶段的文档工作、测试阶段文档和测试工作 | 组员 |

#### 2.7.3.2 影响因素分析

影响上表小组各组员的工作分配情况的最主要因素是小组成员的特长。比如小组成员陈阳和詹鹏飞擅长文档工作，因此他们负责需求阶段和测试阶段文档的编写；彭柯宾擅长编码，因此主要负责项目改进与展示阶段的代码编写；刘克瑞擅长计划、总结和分析，因此担任组长一职，负责一切有关总结计划和分析的任务。除此之外的影响因素包括个人的时间和精力、工作的积极性以及态度等。

## 2.8 配置管理

配置管理阶段的工作主要是运用Github对项目文档进行本变更控制和管理，目前小组在Github上一共commit了127次。然后根据Github提供的数据撰写配置管理总结报告。具体的报告可以参考文档“配置管理总结修订版”。

## 2.9 工作量估计与统计分析

在工作量估计与统计分析阶段，主要进行的工作有：统计分析实验设计、统计收集实验数据、数据分析和撰写工作量估计与统计分析报告。这部分的工作量较大，涉及的范围较广，难度较高。具体的内容可以参考文档“工作量估计与统计分析修订版”。

# 修订说明

## 3.1 软件项目计划

|  |  |  |  |
| --- | --- | --- | --- |
| 项目计划版本号 | 变更情况 | 变更原因 | 时间 |
| V1 | 初步确定项目计划，无变更 | 无 | 20160320 |
| V2 | 修改了资源分配的情况 | 老师提出资源分配需要细化 | 20160322 |
| V3 | 修改了需求评审部分的计划，延后了需求评审的时间 | 老师在课程上安排的变动对项目进度的影响 | 20160329 |
| V4 | 修改了项目计划表，增加了实际工时、实际开始时间和实际完成时间，表明了和更新了项目进度 | 老师根据通过对项目计划表的审查，提出了改进和完善意见 | 20160403 |
| 更新了项目进度，完成了需求分析复评审任务 | 需求分析复评审结束，项目计划需要与实际项目进度保持一致 | 20160412 |
| V5 | 修改了需求分析到改进与展示阶段的计划方案，更新了项目进度 | 实际的安排情况与项目计划不符 | 20160420 |
| V6 | 更新了项目展示与改进部分的进度 | 项目展示与改进完成，项目计划与实际情况需要保持一致 | 20160505 |
| V7 | 更新了项目计划的测试部分 | 软件测试部分完成，项目计划与实际情况需要保持一致 | 20160526 |
| 添加了基线，对工时进行了完善和修改 | 项目的工时部分出现问题，没有设定基线部分 | 20160607 |

## 3.2 软件需求分析

| 修改日期 | 版本 | 修改页码、章节、条款 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2016.4.4 | 2.0 | 需求说明书副标题  2.2用户类型分析  3.2用例说明 | 根据课上老师提出的问题进行了修改，主要包括确定并添加软件名称，重新界定了用户类型。 | 陈阳 |
| 2016.4.8 | 3.0 | 需求说明书标题,  1.2需求分析理论,  2.1产品信息描述  3.1用例图标题  3.2用例说明101  3.2用例说明自动计划及手动计划 | 主要根据课上老师提出的问题进行了修改  在用例说明101中插入若未能打开.mpp文件的解决方法  在用例说明后插入自动计划及手动计划的注释。 | 詹鹏飞 |
| 2016.4.12 | 4.0 | 2.2用户类型分析 | 删除了用户类型分析一节 | 刘克瑞 |
| 2016.5.03 | 5.0 | 3.2 用例说明  6 问题及解决方案 | 增加的每个用例的用例场景，使得需求具体化。  对存需求中存在的问题和相应的解决方案进行的描述 | 刘克瑞 |
| 2016.5.26 | 6.0 | 3.1 用例模型  3.2 用例说明 | 修改了用例图  删除了用例“修改任务模式”、“设置提醒时间”  增加了用例“生成任务报告”  修改了用例ID | 刘克瑞 |
|  |  |  |  |  |

## 3.3 测试分求分析

| 修改日期 | 版本 | 修改页码、章节、条款 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2016.5.11 | 初稿 | 全文 | 编写 | 詹鹏飞、陈阳 |
| 2016.5.19 | 修改版v1.0 | 1.1编写目的  1.3用语定义  1.5参考文献  2.3测试数据  2.4测试通过准则  3.1单元测试  3.5性能测试 | 根据E组和F组的意见，进行了大面积修改，详见批注 | 陈阳 |
| 2016.5.19 | 修改版v2.0 | 目录  1.2.1功能需求  测试用例 | 根据E组和F组的意见，进行了第二次大面积修改，详见批注 | 詹鹏飞 |
| 2016.5.26 | 修改版  v3.0 | 测试用例部分 | 添加了需求与测试用例对应表。  测试用例“生成任务燃烬图” | 詹鹏飞 |

## 3.4 测试报告

| 修改日期 | 版本 | 修改页码、章节、条款 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2016.5.25 | 1.0 | 全部 | 初始编写 | 陈阳 |
| 2016.5.26 | 2.0 | 3.1功能测试  4.覆盖分析  5缺陷统计与分析 | 完善 | 陈阳 |

# 总结

## 4.1 经验与启发

在整个项目执行过程中，最开始的项目计划和人员的分工很重要，尤其是人员分工的确定，可以很大程度上提升项目执行的效率。本组在项目初期人员分工上没有明确的分工，导致经常性的任务堆积在一个人身上或者是没有人去完成任务，使得项目执行过程中多了很多不必要的麻烦。

其次，每周的组会和周报告一定要按计划进行。组会和周报告相当于是每周的总结，可以及时的知道小组项目的真实情况，发现问题并且及时的解决。从另一边面来说能够帮助小组顺利的按照项目计划进行项目的执行。

项目进度的控制也很重要，通过对项目计划进度控制能够清楚地从项目整体来判断当前阶段的情况，以便之后更好地进行项目计划的执行。

## 4.2 有效方法

在整个项目的过程中，每个阶段都会遇到一些的问题。再通过小组成员的努力和学习之后大部分都得到了较好的解决。总结如下表所示：

**实验有效方法归纳表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 实验 | 遇到的问题 | 解决方法 | 结果 | 人员 |
| 1 | 软件需求分析 | 如何确定软件的需求 | 由软件的功能出发，倒推需求 | 较好的完成了需求分析 | 刘克瑞、彭柯宾、陈阳、詹鹏飞 |
| 2 | 软件需求评审 | 由于不了解其他组的项目所用的相关软件和技术而在评审时提不出问题 | 查找相关的资料，或者从软件工程的规范出发来进行评审 | 较好的完成了需求评审 | 刘克瑞 |
| 3 | 改进与展示 | 如何确定改进部分的功能 | 从用户的角度来思考，更多人性化的考虑 | 确定了较好的改进方案，但是在实现过程中没能顺利的实现 | 刘克瑞、彭柯宾 |
| 4 | 测试需求分析 | 如何确定测试用例 | 从需求用例入手，与需求用例一一对应或者更为细化 | 较好的实现了测试需求分析 | 陈阳、詹鹏飞 |
| 5 | 测试评审 | 从什么角度评审和提问 | 从文档规范、测试方法、测试用例的细化考虑 | 较好的完成了测试评审，所提出的问题和建议大都被采纳 | 刘克瑞、彭柯宾、陈阳、詹鹏飞 |
| 6 | 进度计划与控制 | 由于对所用软件不熟悉导致进度计划与控制不顺利 | 多多了解所使用的软件，向有经验者进行求助 | 在实验的后期进行了补救 | 刘克瑞 |
| 7 | 配置管理 | 如何保留变更的证据 | 每次变更都进行相应的记录，并用适当的工具（GitHub）辅助工作 | 较好的完成了配置管理的工作 | 刘克瑞 |
| 8 | 工作量估计与统计分析 | 如何对数据进行统计和分析 | 在平时的项目进展过程中进行相关数据的记录，数据、分析和结论一一对应，形成证据链，更有说服力 | 由于前期数据收集上的失误，导致数据收集不完整，证据链不足 | 刘克瑞 |

# 5 建议

课程设计比较新颖，理论和实践相互交叉，而且除了编程之外，还有一些评审和讨论的环节，让学生更加地了解软件工程过程中的各个阶段，并且亲身的体验，非常有意义。建议可以将课时的周数压缩一下，16周的时间战线太长，学生的积极性会随着时间的增加而减弱。