**基于Torch平台的神经网络压缩**

**研究与应用**

**总结文档**

小组成员：

陈伟民

付强

曹进

李恬霖

**目录**

[1 概述 3](#_Toc485937131)

[2 实验过程分析 3](#_Toc485937132)

[2.1 软件项目计划 3](#_Toc485937133)

[2.2 软件需求分析 4](#_Toc485937134)

[2.3 软件需求评审 4](#_Toc485937135)

[2.4 软件产品改进与展示 8](#_Toc485937136)

[2.4.1 改进方案 8](#_Toc485937137)

[2.4.2 具体实现 8](#_Toc485937138)

[2.5 软件测试 9](#_Toc485937139)

[2.5.1 测试需求定义和测试用例的设计 9](#_Toc485937140)

[2.5.2 测试结果 10](#_Toc485937141)

[2.5.3 发现问题 10](#_Toc485937142)

[2.6 软件测试需求评审 11](#_Toc485937143)

[2.7 软件进度计划与控制 13](#_Toc485937144)

[2.7.1 项目执行过程与特点分析 13](#_Toc485937145)

[2.7.2 计划变更情况 14](#_Toc485937146)

[2.7.3 数据分析 15](#_Toc485937147)

[2.7.4 工作分配情况 16](#_Toc485937148)

[2.7.5 影响因素分析 17](#_Toc485937149)

[2.8 配置管理 18](#_Toc485937150)

[2.9 工作量估计与统计分析 18](#_Toc485937151)

[3 总结 19](#_Toc485937152)

[3.1 经验与启发 19](#_Toc485937153)

[3.2 有效方法 19](#_Toc485937154)

[4 建议 21](#_Toc485937155)

# 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。

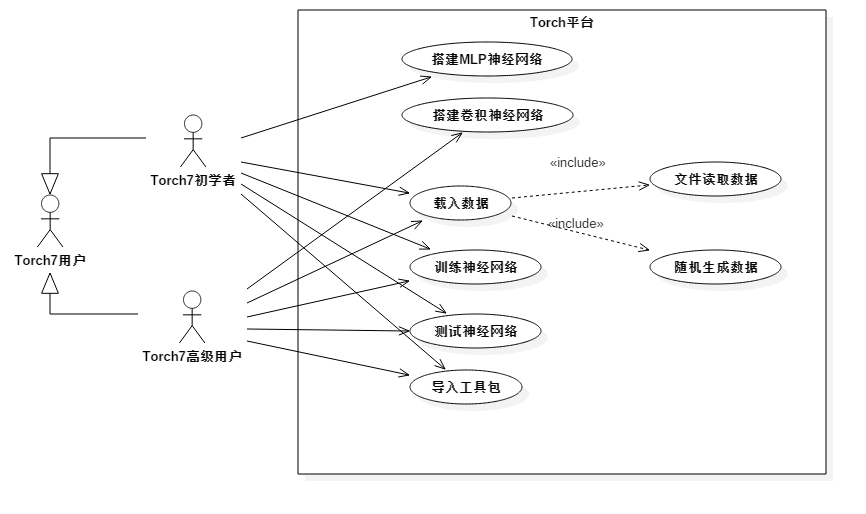
# 实验过程分析

# 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，包括确定了小组成员，确定了实验项目——Torch，并对Torch平台进行了初步调研，同时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《基于Torch平台的神经网络压缩研究与应用》项目计划书-V1.0.docx。

# 软件需求分析

在软件需求分析阶段，首先对所选软件Torch的功能进行分析，编制本软件的功能性需求和非功能需求，并对需求进行确认和验证，之后依照软件设计说明书和实际需求分析编制本软件的扩展需求，然后撰写《需求规格说明书》，最后根据老师和同学们的评审意见对《需求规格说明书》进行反复修订。需求分析的用例图如图2所示。



**图1 用例图**

# 软件需求评审

在软件需求评审阶段，主要根据我组自行制定的《软件需求评审检查单》对他组的《软件需求规格说明书》进行了网上评审和课堂评审，然后收集老师和其他组同学对我组《软件需求规格说明书》的多轮评审意见和建议，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。

**表 1 需求评审问题报告清单**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **问题位置** | | **问题描述** | |
| 1 | 目录 | | 目录结构排版较单一，不够一目了然，建议将子章节向后错落开。 | |
| 2 | 3.3.2 | | 标题为“定义模型”，内容为“生成模型” | |
| 3 | 3.3.3 | | “2.训练方法。。。。。。”格式不太好 | |
| 4 | 1.3 | | “描述了与此次系统实施相关的硬件环境的一些要求；描述了与此系统实施相关的软件环境的要求”，后文并没有提到 | |
| 5 | 3.3 | | 图的标号最好改为，3-1等 | |
| 6 | 3.4 | | 图没有下标 | |
| 7 | 项目源码 | | 没有更新 | |
| 8 | 目录 | | 目录结构排版较单一，不够一目了然，建议将子章节向后错落开。 | |
| 9 | 全文 | | 没有页码。 | |
| 10 | 3.4 | | 可以对用例图进行一些文字描述 | |
| 11 | 全文 | | 术语和缩略语过少，只看到了缩略语，很多领域术语没有看到 | |
| 12 | 3.4.1读取数据 | | 用例图基本事件流第一步引入了未定义的用例“导入paths包” | |
| 13 | 3.4.1导入工具包 | | 不完整，缺少异常事件流 | |
| 14 | 需求文档v1.0(1)(1) | | 功能需求希望可以与用例进行匹配 | |
| 15 | 需求文档v1.0(1) | | 用例图的选择模型与搭建神经网的Include关系有待商榷 | |
| 16 | 需求文档v1.0(1) | | 读取数据的rucm没有4验证路径没有异常事件流 | |
| 17 | 需求文档v1.0(1) | | 图1 用例图中的“文件读取”看起来并不是一个用例，用例描述的是用户想要完成的业务。不应该用户读取文件，只是为了读取文件。。。如果改成读写文件，就好多了。 | |
| 18 | 需求文档v1.0(1) | | RUCM中考虑了很多异常，个人觉得还需要考虑中断异常和运行时异常，例如被用户关闭、或断电情况下不可抗的硬中断，Torch平台会给出什么反应。 | |
| 19 | 需求文档v1.0(1) | | 文档目录页分级目录没有缩进，不清晰，阅读有困难。 | |
| 20 | 需求文档v1.0(1) | | 用例“随机生成数据”中提到用户输入命令随机生成大量数据，与用例"输入命令"有重叠，两用例之间关系未明确。 | |
| 21 | 3需求分析 | | 没有业务需求 | |
| 22 | 3.4RUCM模型 | | 对RUCM图没有描述 | |
| 23 | 3.3需求识别 | | 本节中详细介绍了各个用例的具体步骤，是不是可以在用例图中也有相应的体现 | |
| 24 | 3.4RUCM模型 | | | “描述图1中的用例”，没有找到图1 | |
| 25 | 3.4.5随机生成数据 | | | 图标号应为3-9 | |
| 26 | 3.4.6测试网络 | | | 图标号应为3-10 | |
| 27 | 软件要求 | | | 没有具体指明软件版本要求，应该有最低版本限制 | |
| 28 | 鲁棒性 | | | 建议举例异常情况以及系统应急机制 | |
| 29 | 1.4 | | | 术语和缩略语过少，在文档中看到了很多专有词汇但是在术语表中都没有介绍 | |
| 30 | 3.4.1 | | | 用例图基本事件流第一步引入了未定义的用例“导入paths包” | |
| 31 | 3.4.3 | | | 用例图基本事件流第一步引入了未定义的用例“导入nn包” | |
| 32 | 4.1 | | | 硬件要求中的 CPU部分建议量化，说一下最低条件是什么，主流CPU到底主流到什么地步 | |
| 33 | 3.4.1 | | | 文件读取RUCM读取的后置条件是否应该为读取成功/失败，这样做用户友好性可能好一些 | |
| 34 | 3.4.4 | | | 训练网络RUCM 基本事件流"torch平台 Validates that 命令"后是否应该还有操作，否则怎么体现"训练网络"这一用例 | |
| 35 | 3.4.5 | | | 随机生成数据RUCM 基本事件流验证完命令后续无操作，其他RUCM也有类似问题，建议修改 | |
| 36 | 目录 | | | 字体大小和行距不同 | |
| 37 | 3.4.1 文件读取数据 | | | VALIDATES THAT用法不正确，该关键词表示，后面的命题一定要为真才可以继续运行，而“命令”不是一个命题 | |
| 38 | 3.4.2 导入工具包 | | | steps3 “且”后面的部分不是一个动作，可以作为步骤四使用验证语句。 | |
| 39 | 文档 | | | 应当对系统的环境需求加以明确，便于清晰执行环境 | |
| 40 | 3.2.4 | | | 非功能性需求中的鲁棒性概念应当做以简要介绍。如果能举出一个体现鲁棒性的场景就更好 | |
| 41 | 3.6 RUCM模型 | | | 第一段最后一句话是“接下来将使用 RUCM 模型描述图标3-1中的用例”，实际上描述的是图表3-3。 | |
| 42 | 文章整体 | | | 格式不规范，大标题的字体偏小，小标题的字体偏大 | |
| 43 | 引用文档 | | | 没有举出适当数目的参考资料 | |
| 44 | 2章引用文档 | | | 注意缩进，对http链接进行格式刷操作，可防止不当的字间距出现 | |
| 45 | 3.5 需求识别 | | | RUCM中的主语与参与者不同 如3.6.1中的文件读取数据参与者是“Torch7”用户，但是基本事件流中出现的是“用户” | |
| 46 | RUCM | | | rucm中的后置条件应该与这一操作相关联而不是，是否可以具体化一些，而不是“平台正常运行” | |
| 47 | 3.4.1，3.4.2 | | | 可以对用例图中不存在的用例撰写RUCM吗 | |
| 48 | 3.2 | | | 功能性需求分析内容过少，没有讲清楚 | |
| 49 | 3.4.1 修改卷积层RUCM | | | 修改卷积层用例中有验证，但是没有分支流对其进行异常处理 | |
| 50 | 3.5.4 测试网络 | | | 没有说清楚具体如何测试，测试的输入、输出分别是什么？ | |
| 51 | 1.2系统概述 | | | 英文单词“Facebook、Twitter”前后不加空格 | |
| 52 | 1.2系统概述 | | | 第4段第2行中说到的“我们”没有具体指向 | |
| 53 | 1.3文档概述 | | | 文中提到“状态图”但是后文中没有出现 | |
| 54 | 1.3文档概述 | | | 内容描述用词不恰当，“此系统实施”没有上下文依据 | |
| 55 | 1.4术语和缩略语 | | | 第10条说明内容“以步长”描述存疑 | |
| 56 | 3.2功能性需求分析/3.2.1丰富的工具包 | | | 标点符号有误（“，”改成“、”）、文字使用错误 | |
| 57 | 3.2功能性需求分析/3.2.2模块化搭建神经网络 | | | 标点符号错误（“，”改成“。”）、段落叙述有语病 | |
| 58 | 3.3非功能性需求分析 | | | 建议将3.3节和3.4节的内容对调 | |
| 59 | 3.3非功能性需求分析/3.3.1高效性 | | | 标点符号错误（“，”改成“。”）、段落叙述有语病 | |
| 60 | 3.3非功能性需求分析/3.3.4鲁棒性 | | | 标点符号错误（“，”改成“。”）、段落叙述有语病 | |
| 61 | 3.4扩展需求/3.4.1修改卷积层RUCM | | | 图表名称命名不规范，图和表应加以区分（多处存在该问题） | |
| 62 | 3.4扩展需求/3.4.1修改卷积层RUCM | | | 用例的RUCM描述存在问题：1.dependency行应加入包含用例；2.Step4、7，增加Specific Alt. Flow；3.软中断未表述到位。其他用例RUCM描述有类似问题 | |
| 63 | 3.4扩展需求/3.4.1修改卷积层RUCM | | | 输入命令不合法的情况应具体说明 | |
| 64 | 3.4扩展需求/3.4.2修改线性层RUCM | | | 软中断处理持续时间很长的情况应具体说明 | |
| 65 | 3.5需求识别 | | | 用例图中的网格背景应该去掉 | |
| 66 | 3.5需求识别 | | | 用例图中的用例与本节子章节的具体说明内容不相符 | |
| 67 | 3.5需求识别 | | | 建议增加图表在段落中的对应内容描述（多处存在该问题） | |
| 68 | 3.6 RUCM模型 | | | 3.6.1章节名建议修改为“文件数据读取” | |
| 69 | 3.6 RUCM模型/3.6.1 | | | 用例的RUCM描述存在问题：include用例，命令和路径检查后，应当执行该名令 | |
| 70 | 3.6 RUCM模型/3.6.2导入工具包 | | | 用例的RUCM描述存在问题：Step5应为执行工具包导入命令（多处存在该问题） | |
| 71 | 3.6 RUCM模型 | | | 是否缺少一个用例的RUCM说明 | |
| 72 | 全文 | | | 针对实验三的需求不明确，也没有必要的细化说明 | |
| 73 | 2 | | | 引用文档使用了大章节，但是里面只有一句话很尴尬 | |

# 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。

# 改进方案

对Torch的卷积层和全连接层（线性层）的计算进行修改。我们设计了权重二值化的算法，在原有实现的基础上，增加了投影矩阵，对二值化后不同的求导方案做了设计，并设计了基于受训练的投影矩阵的算法。

# 具体实现

|  |  |  |
| --- | --- | --- |
|  | **Algo 1. 随机P** | **Algo 2.训练P** |
| 更新变量 | W, b | W, b, P |
| 修改文件 | Linear.lua  SpatialConvolution.lua  SpatialConvolutionMM.c | Module.lua  Linear.lua  SpatialConvolution.lua  SpatialConvolutionMM.c |
| 激活函数 | y=x  y=Htanh(x) | y=x  y=Htanh(x) |
| 进度 | Complete | Complete |

# 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。

# 测试需求定义和测试用例的设计

**表2 测试用例和需求对照表**

|  |  |  |
| --- | --- | --- |
| **测试用例名称** | **测试用例标识** | **对应需求文档的需求** |
| Linux下安装测试 | TC001 | 需求文档v2.2中的4运行要求 |
| Mac OS下安装测试 | TC002 | 需求文档v2.2中的4运行要求 |
| 导入工具包 | 用例TC401 | 需求文档v2.2中的3.6.2节 |
| 文件数据读取 | 用例TC402 | 需求文档v2.2中的3.6.1节 |
| 搭建MLP神经网络 | 用例TC403 | 需求文档v2.2中的3.6.3节 |
| 搭建卷积神经网络 | 用例TC404 | 需求文档v2.2中的3.6.4节 |
| 训练神经网络模型 | 用例TC405 | 需求文档v2.2中的3.6.5节 |
| 测试神经网络模型 | 用例TC406 | 需求文档v2.2中的3.6.7节 |
| 压缩MLP神经网络 | 用例TC501 | 需求文档v2.2中的3.3.2节 |
| 压缩卷积神经网络 | 用例TC502 | 需求文档v2.2中的3.3节 |
| 高效性 | 用例TC601 | 需求文档v2.2中的3.4.1节 |
| 用户友好性 | 用例TC602 | 需求文档v2.2中的3.4.2节 |
| 可修改性 | 用例TC603 | 需求文档v2.2中的3.4.3节 |
| 鲁棒性 | 用例TC604 | 需求文档v2.2中的3.4.4节 |

# 测试结果

**表3 测试结果表**

|  |  |  |
| --- | --- | --- |
| **测试用例名称** | **测试结果** | **测试结论** |
| Linux下安装测试TC001 | 安装并启动成功 | 通过 |
| Mac OS下安装测试TC002 | 安装并启动成功 | 通过 |
| 导入工具包用例TC401 | 平台有相应输出或者相应报错 | 通过 |
| 文件数据读取用例TC402 | Torch平台输出按照格式输出的数据与文件中的数据一致 | 通过 |
| 搭建MLP神经网络TC403 | Torch平台输出与设计的网络结构一致 | 通过 |
| 搭建卷积神经网络TC404 | Torch平台输出与设计的网络结构一致 | 通过 |
| 训练神经网络模型TC405 | Torch平台输出模型训练过程中，每次迭代的输出误差以及迭代次数 | 通过 |
| 测试神经网络模型TC406 | 得到网络对所求问题的预测准确率 | 通过 |
| 压缩MLP神经网络TC501 | 得到网络对所求问题的预测准确率 | 通过 |
| 压缩卷积神经网络TC502 | 得到网络对所求问题的预测准确率和时间 | 通过 |
| 高效性TC601 | Torch平台计算效率优于Tensorflow平台 | 通过 |
| 用户友好性TC602 | 平台正常启动运行且网络搭建简单，运行成功。 | 通过 |
| 可修改性TC603 | 合理修改代码，代码可运行，网络修改成功。 | 通过 |
| 鲁棒性TC604 | 输入错误指令平台提示命令不合法 | 通过 |

# 发现问题

1、系统的非功能性需求在大部分情况下均按照预想的情况执行，但在用例604的第四种场景，用户强行Kill当前程序运行进程时出现了偏差，我们期望系统会保存现有数据状态，并在下次启动系统时自动恢复但是实际运行中再次打开Torch平台时系统并未恢复意外关闭前的数据状态。这是Torch平台本身设计存在的问题，我们无法修改，只能在之后的使用中尽量避免这种情况的发生。

2、下表为MLP网络在不同压缩率下的准确率表，可以看出压缩比越高，其压缩后的准确率总体反而会出现越高的情况，不符合压缩比越高，其准确率越低的预期，我们无法做出合理的解释，虽然存在一定的随机性，但仍然可以认为在压缩算法存在不合理的地方，还有改进的空间。

**表4 MLP网络准确率表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **压缩倍数** | **8** | **12** | **16** | **24** | **32** | **48** | **64** |
| 未压缩 | 0.960 | | | | | | |
| 已压缩 | 0.941 | 0.9496 | 0.9437 | 0.9477 | 0.9493 | 0.9449 | 0.948 |

# 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规格说明书检查单编写，软件测试需求互评的工作，之后撰写了测试需求评审意见汇总清单。软件测试需求规格说明书检查单和测试需求评审意见汇总清单如下表所示。

**表5 测试规格说明书检查单**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **类别** | **检查项说明** | **检查要点** |
| 1 | 完整性 | 测试项和测试用例是否完整，无缺漏？ | 对于软件的需求来说，测试是否足够，是否覆盖了所有需求 |
| 2 | 完整性 | 测试项和测试用例是否有对应表，即是否有测试项覆盖表和测试用例覆盖表。 | 测试项是否跟需求项对应  测试用例是否跟需求用例对应 |
| 3 | 完整性 | 是否对所有的不易理解的项都做了充分的解释和说明 |  |
| 4 | 一致性 | 概念和术语定义和使用一致，统一规范、无歧义 |  |
| 5 | 规范性 | 测试用例描述是否符合规范 | 是否采用RTCM模型或者其他自定义的能够清晰描述用例的模型描述 |
| 6 | 规范性 | 图、表等是否规范 | 相应的题注是否规范 |
| 7 | 规范性 | 文档章节和目录是否符合规范,概述的顺序与目录顺序是否一致 |  |
| 8 | 规范性 | 单词拼写和大小写是否正确 |  |
| 9 | 规范性 | 字体是否统一和规范 | 正文中中文应该使用宋体5号，英文应当是Time New Rome 5号 |
| 10 | 规范性 | 标点符号是否正确使用 |  |
| 11 | 规范性 | 段落缩进是否规范 | 中文的段落缩进一般都是2字符 |
| 12 | 规范性 | 各级标题字号大小是否规范 | 一般上级标题比下级标题的字号大 |
| 13 | 准确性 | 测试项和测试用例描述是否准确 | 测试项分解清晰合理  测试项定义和描述准确 |
| 14 | 准确性 | 文字描述是否符合中文或英文语法，是否通顺，用词是否准确，无歧义 |  |
| 15 | 准确性 | RTCM图描述是否准确符合逻辑 | RTCM图的名称、简要概述、前置条件和约束、测试步骤、评价准则等是否准确合理 |

**表7 测试需求评审意见汇总清单**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **问题位置** | **问题描述** | **处理结果** |
| 1 | 文档结构 | 可能讲测试分为平台本身和扩展功能两部分，然后每部分按照功能测试和非功能测试会更清晰 | 已修改 |
| 2 | 2.2 | 若对测试工具介绍的话，建议加上使用此工具测试的优势 | 已修改 |
| 3 | 4.1概述 | 第一段少了个句号 | 已修改 |
| 4 | 3.安装测试 | 需求中没有对应项？应该对应到“4. 运行要求 | 已修改 |
| 5 | 4.2导入工具包 | 对异常处理的测试：对应的需求项中，还有“Specific Alt. Flow”等？ | 已修改 |
| 6 | 6.1.1测试策略描述 | 具体选择？ | 已修改 |
| 7 | 6.3.1测试策略描述 | 采用什么方式？扩展接口？ | 已修改 |
| 8 | 6.3.2 | 表格中具体修改见第5章？ | 已修改 |
| 9 | 6.4.2测试用例描述 | 用户友好性和鲁棒性含义完全不同！ | 已修改 |
| 10 | 6.4.2测试用例描述 | 下面是4种测试，而不是连续的测试步骤！ | 已修改 |
| 11 | 表3-1 | 安装git、安装Torch没有说明版本号 | 已修改 |
| 12 | 功能测试 | 大部分测试用例没有预期结果，也没有约束输入 | 已修改 |
| 13 | 测试方案 | 目标是单元测试而不仅仅是功能测试，因此每个步骤（每个方法）应用多组测试用例 | 未修改，已有测试用例具有代表性 |
| 14 | 可修改性 | 可修改性指的是修改的难易程度，而不是能不能修改，因此对可修改性测试的用例好像存在一点问题 | 未修改，通过直接修改源代码并运行来判断修改的难易程度 |
| 15 | 全文 | 所有测试用例在表格前是否可以用文字先简单介绍下该用例，然后测试步骤是否可以再详细点 | 已修改 |
| 16 | 全文 | 没有测试结果 | 已修改 |
| 17 | 全文 | 测试用例中只有步骤没有数据 | 已修改 |
| 18 | 6.4 鲁棒性 | 测试用例介绍太泛,如“大规模”“其他程序”应该适当细化 | 已修改 |

# 软件进度计划与控制

# 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，我们还进行了软件进度计划与控制、工作量估计与统计分析、配置管理等工作，这些工作贯穿于整个项目的执行过程。

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。

# 计划变更情况

我们对本项目的计划变更情况进行了分析，具体情况如表所示。

**表8 项目计划变更情况表**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目计划文件** | **变更情况** | **变更原因** | **时间** |
| 软工实验项目.mpp | 初步确定项目计划，无变更 | 无 | 2017-03-31 |
| 软工实验项目0406.mpp | 增加了贯穿项目始终的软件进度计划与控制、配置管理 | 根据老师指导，需要添加软件进度计划与控制和配置管理 | 2017-04-06 |
| 软工实验项目0407.mpp | 细化了资源分配的情况 | 老师在课堂上指出资源分配需要细化 | 2017-04-07- |
| 软工实验项目0414.mpp | 修改实际工时 | 老师分析了实际工时的计算方法 | 2017-04-14 |
| 软工实验项目0421.mpp | 更新项目需求评审实验进度 | 项目需求评审实验完成，项目计划与实际情况需要保持一致 | 2017-04-21 |
| 软工实验项目0428.mpp  软工实验项目0505.mpp  软工实验项目0512.mpp | 更新软件产品改进实验进度  调整实际工时、计划工时、实际工期、计划工期、资源顺序 | 软件产品实验完成，项目计划与实际情况需要保持一致  老师指出需要调整实际工时、计划工时、计划工期、实际工期和资源的顺序 | 2017-04-28  2017-05-05  2017-05-12 |
| 软工实验项目0519.mpp | 更新软件测试分析实验进度 | 软件测试分析实验完成，项目计划与实际情况需要保持一致 | 2017-05-19 |
| 软工实验项目0526.mpp  软工实验项目0602.mpp | 更新软件测试与软件测试复评审实验进度 | 软件测试与软件测试复评审实验完成，项目计划与实际情况需要保持一致 | 2017-05-26  2017-06-22 |
| 软工实验项目0609.mpp | 更新软件演示与测评实验进度 | 软件演示与测试实验完成，项目计划与实际情况需要保持一致 | 2017-06-09 |
| 软工实验项目0616.mpp | 更新综合实验分析实验进度 | 项目综合实验分析完成，项目计划与实际情况需要保持一致 | 2017-06-16 |
| 软工实验项目0623.mpp | 更新综合实验总结实验进度 | 综合实验总结完成，项目计划与实际情况需要保持一致 | 2017-06-23 |

# 数据分析

我们按照每周任务，对我们的项目计划进行分析，共更新了14版的mpp文件。前5版的mpp文件变化较大，从上表统计的数据可以看出，软工实验项目.mpp是项目计划初稿，根据整个课程的课时初步进行项目计划，软工实验项目0406.mpp在软工实验项目.mpp基础上添加了贯穿项目始终的软件进度计划与控制、配置管理实验进度；在软工实验项目0407.mpp中，根据老师的指导细化了资源分配情况；根据老师课程分析实际工时的计算方法，我们在软工实验项目0414.mpp中，重新填写了任务的实际工时，软工实验项目0421.mpp中更新了项目需求评审实验进度；后9版的mpp文件主要根据实验完成情况，为保证项目计划与实际情况一致进行更新。

项目计划按照每周任务分为14个版本，版本数过多，如软工实验项目0428.mpp、软工实验项目0505.mpp、软工实验项目0512.mpp可以合并成一个版本，均属于软件产品改进阶段的实验进度。影响项目计划变更的主要因素分为两个方面，一是老师指导我们修改项目计划中不合理的地方，二是我们为了更清楚展现我们的项目计划，重新按照每周实验任务对项目的进度进行记录。其中实验进度与实验计划不相符的情况，是由于我们最初不熟悉记录mpp文件的方法，以及小组成员在项目执行过程中执行力不足造成的。

# 工作分配情况

工作分配的具体情况如表所示。

**表9 工作分配表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **性别** | **擅长** | **分工** | **职位** |
| 陈伟民 | 男 | 语言组织能力、学习能力 | 项目计划、主持会议、需求分析阶段的文档工作、改进与展示阶段文档和代码工作、测试阶段文档和测试工作、项目统计分析 | 组长 |
| 曹进 | 男 | 做计划、语言组织能力、语言表达能力、总结反思 | 项目计划、需求分析阶段的文档工作、需求评审文档工作、改进与展示阶段代码工作、工作量统计与分析 | 组员 |
| 付强 | 男 | 文档工作、代码能力、配置管理工作 | 需求分析阶段文档工作、改进与展示阶段代码工作、测试阶段文档和测试工作、会议记录、配置管理 | 组员 |
| 李恬霖 | 男 | 学习能力、代码能力 | 需求分析阶段的文档工作、改进与展示阶段代码工作、测试阶段文档和测试工作、工作量统计与分析 | 组员 |

# 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。在实验的各项工作工作要求中，每名成员均承担一部分文档的撰写和编码的工作；陈伟民和李恬霖主要负责算法的研究和测试文档的撰写；付强和曹进主要负责需求文档的撰写，并协同管理GitHub。

本小组的工作中，各项工作的分配在比较均匀的同时，也很大程度上充分利用了成员的各项能力，其中陈伟民负责了大量的ppt制作与展示工作，曹进承担了mpp编写的工作，付强进行了github的配置管理，李恬霖查阅了大量算法资料，并编写了相关文档。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。

# 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变更信息管理，目前小组在Github上一共commit了147次。然后根据Github提供的数据和文档版本变更详细情况撰写《变更与管理分析报告》。具体报告内容可以参考文档《变更与管理分析报告v2.0》。

**表10 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 | 陈伟民、付强、曹进、李恬霖 |
| 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问不充分，建议不专业 | 尽量查找相关资料对他组项目进行专业评审，并多从软件工程的规范性角度来进行评审 | 较好的完成了需求评审 | 陈伟民、付强、曹进。、李恬霖 |
| 7 | 配置管理 | 如何使用Github进行版本管理？ | 查阅资料学习使用Github，每次使用Github变更都进行规范的记录 | 较好的完成了配置管理的工作 | 付强 |

# 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稿G组2.docx》。

# 总结

# 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。

# 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 | 付强、陈伟民、曹进、李恬霖 |
| 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 | 曹进、李恬霖 |
| 3 | 改进与展示 | 如何确定改进部分的功能，如何制定适合短时开发的算法 | 简化需求，耐心推导 | 确定了较好的改进方案，并顺利的实现 | 付强、陈伟民 |
| 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 | 付强、陈伟民 |
| 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 | 付强、李恬霖 |
| 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、论坛进行求助 | 在实验的后期进行了补救 | 陈伟民、曹进 |
| 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 | 付强、曹进 |
| 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 | 曹进、李恬霖 |

# 建议

课程注重实践，整个过程彼此间联系较强，除了编程之外，还有评审和课堂讨论的环节，让学生切身了解软件工程过程中的各个阶段，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。