NodeJS**项目总结报告**

小组成员：

李岳檑

谭伟良

温元帧

王春柳

目录

[1 概述 3](#_Toc485895565)

[2 实验重点 3](#_Toc485895566)

[2.1 实验准备 3](#_Toc485895567)

[2.2 软件需求分析 5](#_Toc485895568)

[2.3 软件需求评审 6](#_Toc485895569)

[2.4 软件产品改进与展示 8](#_Toc485895570)

[2.5 软件测试需求分析 10](#_Toc485895571)

[2.6 软件测试需求评审 12](#_Toc485895572)

[2.7 软件进度计划与控制 14](#_Toc485895573)

[2.8 配置管理 16](#_Toc485895574)

[2.9 工作量估计与统计分析 18](#_Toc485895575)

[3 实验规模的数据统计 18](#_Toc485895576)

[4 总结 20](#_Toc485895577)

[4.1 分析 20](#_Toc485895578)

[4.2 经验与启发 21](#_Toc485895579)

[4.3 有效方法 22](#_Toc485895580)

[4.3.1 解决非功能需求描述问题——质量属性 24](#_Toc485895581)

[5 建议 25](#_Toc485895582)

# 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，除去实验前的准备和实验总结，中间的实验一共分为八个阶段，分别对应是的是实验一到八，具体过程包括：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、测试需求与测试设计阶段、测试评审阶段、软件进度计划与控制、工作量估计与统计分析、配置管理，其中最后三个实验贯穿实验的全过程。

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 实验准备

#### 2.1.1 项目计划阶段

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——NodeJS，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。这其中还由于对项目的不了解和对工作的不明确，在确定项目为NodeJS之前，选择了难度大的Spark，在老师的指导和对实验加深理解之后才又更换为NodeJS.

#### 2.1.2 项目分工

**小组工作分配表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 性别 | 擅长 | 分工 | 职位 |
| 李岳檑 | 男 | Javascript，统筹兼顾，表达能力 | 项目计划、主持会议、需求和测试文档的审核及少量编写工作，改进与展示阶段文档和大部分代码工作、测试相关的工作，每周PPT制作，上台主讲 | 组长 |
| 谭伟良 | 男 | 做计划，学习能力强 | 项目计划，学习ms project，全程维护组内的mpp文件，分担分担一些代码和文档的工作，以及对C组的测试 | 组长 |
| 王春柳 | 女 | 文档能力以及对RUCM的使用 | 负责了大部分的文档和记录工作，每周的会议记录课堂记录也基本由她完成，负责了测试 | 组员 |
| 温元帧 | 女 | 文档能力以及代码工作 | 负责一些代码工作和文档工作，有时候也会完成课堂记录和会议记录 | 组员 |

### 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。从业务需求，改进与扩展设计，功能需求分析，非功能需求分析以及应用场景等方面进行了需求分析。在进行业务需求的时候，从Node最原始的需求出发，对Node创始人当初创造node的初衷进行分析来完成了业务需求的编写。在对非功能性需求描述遇到难题，无法使用RUCM进行描述，寻找到了一种新的软件工程描述方法——质量属性。需求分析所完成的任务如下所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验1：需求分析 | | | | |
| 需求文档规模  （字数） | 需求项数  （RUCM用例描述数量） | 用例图描述 | 版本更新次数 | 累计工时（含修改） |
| 6064 | 19 | 3 | 7 | 78 |

#### 非功能需求描述方法——质量属性

为了更好的描述非功能需求，我们用属性质量场景来描述非功能性需求的用例。描述时用到的一些术语解释如下：

 刺激源

刺激源是某个生成该刺激的实体。

 刺激

刺激是当刺激到达系统时需要考虑的条件。

 环境

刺激在某些条件下发生。当刺激发生时，系统可能处于过载或者正在运行等。

 制品

被刺激的制品，可以是整个系统，也可以是系统的一部分。

 响应

指刺激到达后所采取的行动。

 响应度量

当响应发生时，应能够以某种方式对其进行度量。

以兼容性为例，兼容性的质量属性描述为：

表2.1兼容性用例描述

|  |  |
| --- | --- |
| 场景 | 可能的值 |
| 源 | web开发人员 |
| 刺激 | 在不同操作系统上使用该系统的需求 |
| 制品 | 产出不同操作系统下的系统版本 |
| 环境 | 不同操作系统下 |
| 响应 | Node.js在操作系统和上层模块系统之间构建了一层libuv平台架构。在不同操作系统下，Node.js无需采用该软件准备的配置文件之外的规定就可以直接使用 |
| 响应度量 | 可移植的程度、在不同操作系统下需要更改的配置文件个数、除了配置文件之外是否需要其它文件 |

### 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。需求评审工作主要如下面两表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验2：需求评审（评审+复评审） | | | | | |
| 检查单中的检查项数量 | 互评审中给被评审组提出的问题数（评审+复评审） | 接收到的问题数（评审+复评审） | 老师的问题数（含各组存在的共性问题） | 接受并修改的问题数 | 累积工时（不含需求修改工时） |
| 9 | 103 | 55 | 32 |  | 68h |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 需求评审问题清单（被评审） | | | | | | |
| 编号 | 被评审版本号 | 问题位置 | 问题说明 | 问题分类 | 评审组 | 处理情况 |
| 1 | V1.1 | 6.1 | 没有说明具体的运行环境 | 内容遗漏 | E | 接受意见，将兼容性具体是在哪些操作系统上操作进行说明。 |
| 2 | V1.1 | 3.3 | 文件系统用例图与描述内容不一致 | 内容不一致 | E | 接受意见，将文件系统重新进行描述，用例图重新绘制 |
| 3 | V1.0 | 第四章 | 正文字体为5号，最好全文保持一致 | 格式问题 | F | 接受意见，因马虎导致第四章的字体大小与其他部分不一致，故修改为5号 |
| 4 | V1.3 | 第六章 | 每种需求建议多谢几句话进行描述 | 内容缺失 | F | 接受建议，因被评审人对评审项目的不熟悉，对一些需求项不理解，故应该将内容详细说明 |
| 5 | V2.0.1 | 全文 | 英文格式不统一 | 格式问题 | E | 接受意见，已全部修改为Times New Roman |
| 6 | V2.0.1 | 4.1 | 用例图用例为英文，rucm为中文 | 内容不一致 | E | 接受建议，重新绘制用例图，将用例图中用例与RUCM中的用例名统一 |
| 7 | V1.2 | 3.1.1 | 创建模块的前置条件是什么？异常情况如何处理？ | 内容缺失 | 老师 | 部分修改，创建模块的前置条件为空，异常情况没有说明，已补全 |
| 8 | V1.2 | 3.1.1 | 加载模块在rucm中有，但是用例图中没有 | 内容不一致 | 老师 | 接受建议，该问题出现在很多地方，已将所有由该问题相同的部分进行修改，着重修改用例图 |
| 9 | V1.2 | 5.1 | 兼容性缺少rucm用例图 | 内容缺失 | 老师 | 接受建议，该问题出现在非功能性需求中的四处，已将四个非功能性需求用表格的形式进行描述 |
| 10 | V1.2 | 7.1 | I/O密集型的具体场景是什么 | 内容缺失 | 老师 | 接受建议，将I/O密集型场景以具体的实例进行说明，以便于后期的测试。 |

### 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

主要分为两部分完成，一是核心的异步过载保护模块，这块主要涉及算法的修改，是对node处理异步调用的一个大的改进。二是基于异步过载保护模块的DNS查询系统。

核心模块是异步过载保护部分。异步过载保护即系统在处理多个并发任务时采取的一种保护措施。Node.js内部并没有这个模块，因此当任务的并发量过多时，会出现系统崩溃的现象。为此，我们设计了异步过载保护模块，用户可以自己设置并发量和队列的长度，当队列中任务已满时，系统将报错，以此来防止系统崩溃的现象发生。

DNS查询系统是一个利用域名可以查询IP的系统。该系统一般设置在服务器端，当并发查询数量过多时，往往需要异步过载保护。

#### 2.4.2 改进与扩展统计

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验3：改进与扩展 | | | | | | | |
| 设计实现 规模（字 数） | 其他模型 （类图、 顺序图、 状态图等， 不含非规 范的示意 图） | 其他模型 中包含的 元素累积 数（节点 和边的数 量） | 代码行数 | 问题数 | 修改的问 题数 | 版本更新 （最大） 次数 | 累积工时 数 |
| 2573 | 流程图 | 3个 | 1300 | 55 | 10 | 5 | 46h |
|  | 类图 | 1 |  |  |  |  |  |
|  | 顺序图 | 1 |  |  |  |  |  |
|  | RUCM图 | 3 |  |  |  |  |  |

### 2.5 软件测试需求分析

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。我们经过讨论以及对时间任务的合理预估，最后决定由

#### 2.5.1测试需求定义和测试用例的设计

**测试用例与测试需求对应表**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能性测试 | 测试用例名称 | 测试用例标识 | 对应需求文档的需求 |
| 创建模块 | NO.201 | 需求文档V1.1.0的2.1.2节 |
| 加载模块 | NO.202 | 需求文档v1.1.0的2.2.2节 |
| 创建包 | NO.203 | 需求文档v1.1.0的2.3.2节 |
| 加载包 | NO.204 | 需求文档v1.1.0的2.4.2节 |
| 管理包 | NO.205 | 需求文档v1.1.0的2.5.2节 |
| 构建TCP服务器 | NO.206 | 需求文档v1.1.0的2.6.2节 |
| 构建UDP服务器 | NO.207 | 需求文档v1.1.0的2.7.2节 |
| 构建HTTP服务器 | NO.208 | 需求文档v1.1.0的2.8.2节 |
| 构建WebSocket服务器 | NO.209 | 需求文档v1.1.0的2.9.2节 |
| 删除文件测试 | NO.210 | 需求文档v1.1.0的2.10.2节 |
| 读取文件测试 | NO.211 | 需求文档v1.1.0的2.11.2节 |
| 文件重命名测试 | NO.212 | 需求文档v1.1.0的2.12.2节 |
| 文件写入测试 | NO.213 | 需求文档v1.1.0的2.13.2节 |
| 创建目录测试 | NO.214 | 需求文档v1.1.0的2.14.2节 |
| 删除目录测试 | No.215 | 需求文档v1.1.0的2.15.2节 |
| 非功能性测试 | 兼容性测试 | NO.301 | 需求文档v1.1.0的3.1.2节 |
| 高效性测试 | NO.302 | 需求文档v1.1.0的3.2.2节 |
| 容错性测试 | NO.303 | 需求文档v1.1.0的3.3.2节 |
| 可扩展性测试 | NO.304 | 需求文档v1.1.0的3.4.2节 |

#### 2.5.2 测试计划

鉴于时间等条件所限，我们计划对DNS系统进行功能测试，而对其中的核心模块——异步过载保护模块进行单元测试和语句覆盖。

由于node.js的扩展模块由李岳檑、温元祯、谭伟良进行编写。DNS查询系统的编写也是由上述三位人员进行编写。在组内自测的环节，我们由没有参与编码工作的王春柳同学进行功能性测试，以便更好的发现编码人员容易忽视的错误，谭伟良同学进行功能性测试的辅助工作。在单元测试的环节，由温元祯、李岳檑同学采用mocha测试框架，进行单元测试，并进行语句覆盖测试。

#### 2.5.2 测试结果

单元测试

异步过载保护模块单元测试

|  |  |  |
| --- | --- | --- |
| 单元测试编号 | 测试用例 | 测试结果 |
| 1 | 《测试需求规格说明书v1.5》表4.1 | Pass |
| 2 | 《测试需求规格说明书v1.5》表4.2 | Pass |
| 3 | 《测试需求规格说明书v1.5》表4.3 | Pass |
| 4 | 《测试需求规格说明书v1.5》表4.4 | Pass |
| 5 | 《测试需求规格说明书v1.5》表4.5 | Pass |
| 6 | 《测试需求规格说明书v1.5》表4.6 | Pass |
| 7 | 《测试需求规格说明书v1.5》表4.7 | Pass |
| 8 | 《测试需求规格说明书v1.5》表4.8 | Pass |
| 9 | 《测试需求规格说明书v1.5》表4.9 | Pass |
| 10 | 《测试需求规格说明书v1.5》表4.10 | Pass |
| 11 | 《测试需求规格说明书v1.5》表4.11 | Pass |
| 12 | 《测试需求规格说明书v1.5》表4.12 | Pass |

DNS查询系统功能测试

DNS查询系统功能测试数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试类型 | 输入内容 | 正确输出内容 | 实际输出 | 通过测试 |
| 正向测试  （ipv4） | baidu.com | 220.181.57.217 | 220.181.57.217 | 是 |
| csu.edu.cn | 222.247.53.90 | 222.247.53.90 | 是 |
| cfpa.org.cn | 101.37.43.107 | 请输入正确域名 | 否 |
| lonten.net | 103.238.225.146 | 请输入正确域名 | 否 |
| 正向测试  （ipv6） | www.youtube.com | 2404:6800:4008:803::200e | 2404:6800:4008:803::200e | 是 |
| buaabt.cn | 2001:da8:203:502:bdf8:170e:4ddf:6366 | 2001:da8:203:502:bdf8:170e:4ddf:6366 | 是 |
| facebook.com | 2a03:2880:f102:83:face:b00c:0:25de | 请输入正确域名 | 否 |
| http://buaabt.cn | 2001:da8:203:502:bdf8:170e:4ddf:6366 | 2001:da8:203:502:bdf8:170e:4ddf:6366 | 是 |
| 反向测试 | 123456. | 请输入正确域名 | 请输入正确域名 | 是 |
| abcdef/ | 请输入正确域名 | 请输入正确域名 | 是 |
| 1a2b3c | 请输入正确域名 | 请输入正确域名 | 是 |
| \*~！1a2 | 请输入正确域名 | 请输入正确域名 | 是 |

#### 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互评的工作，之后根据评审的意见和建议修改测试需求规格书，还对前面两组进行了测试需求规格书的评审工作，又学习了C组的项目 scrapy, 在C组 的指导帮助下完成了对C组的测试。主要数据如下。

测试需求评审

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试需求评审（评审+复评审） | | | | | |
| 检查单中的检查项数量 | 互评审中给被评审组提出的问题数（评审+复评审） | 接收到的问题数（评审+复评审） | 老师的问题数（含各组存在的共性问题） | 接受并修改的问题数 | 累积工时（不含需求修改工时） |
| 15 | 43 | 36 | 8 |  | 20h |

测试互评问题清单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 被评文档版本 | 问题位置 | 问题说明 | 问题分类 | 评审组 | 处理意见 |
| 1 | 1.3 | 全文 | 可考虑对篇章中出现的术语进行统一的解释。 | 完整性 | F | 接受建议，在文档最后添加了术语描述参照表。 |
| 2 | 1.3 | 2.5.2测试用例 | 只描述了“安装包”测试用例，而省略了其他用例，是否合适？ | 规范性 | E | 接受建议，在2.5.2中添加了类似测试用例说明。 |
| 3 | 1.3 | 3.3.1 | 容错性应当是运行时的软件或硬件故障，仅仅通过错误代码去测试是否合理？ | 规范性 | E | 由于硬件故障无法认为制造，而软件故障如需要的软件没有安装这种情况在测试时是有测试的。 |
| 4 | 1.5 | 封面 | 封面的日期在第二页；日期编号没有更新。 | 规范性 | F | 接受建议，并做了修改。 |
| 5 | 1.5 | 2.1.2  测试用例 | Steps中测试员代码编写与系统操作间的时序关系是否合适？测试员编写是否应当在前面？ | 一致性 | E | 测试员启动Node.js和mocha系统更方便代码的编写。此系统是Node.js，是代码运行的基础，因此需要先启动系统。 |
| 6 | 1.5 | 2.4.1  加载包测试用例 | 是否可以通过测试加载系统提供的包来排除包创建失败的可能情况。 | 准确性 | E | 首先没有加载系统，加载模块是一个函数，不能提供包。测试加载包是对加载函数进行测试。可以用加载函数来判断包是否能创建成功。 |
| 7 | 1.5 | 2.10节 | 2.10.1中描述说有同步和异步，下面的测试用例貌似没有说明是同步还是异步 | 一致性 | E | 接受建议，添加上说明是对异步函数的测试。因为Node.js本身一直在强调异步的重要性，因此默认是异步方式。 |
| 8 | 1.6 | 2.8.1等处. | 正文中有中英文逗号混用情况；“四个部分, ，一部分”出现了两个逗号 | 规范性 | F | 接受建议，对混用及标点符号的错误进行了修改。 |

老师批改测试需求说明书v1.3问题清单

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 问题位置 | 问题说明 | 处理意见 |
| 1 | 2.1 | 在需求规格说明书中没有这一项需求，是包含在“创建模块”需求项中的。建议：修改需求规格说明书，将创建和加载分成两个需求项。其他类似的需求项，也应这样分解开。 | 接受意见，将需求说明书中的创建和加载模块分开。 |
| 2 | 2.5 | 只是管理包的安装测试？ | 这里是由于管理包的意思不够明确。在术语描述表中添加了说明。管理包包括安装包。对管理包的其它类型的测试也已添加 |
|  |  |  |  |

### 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。

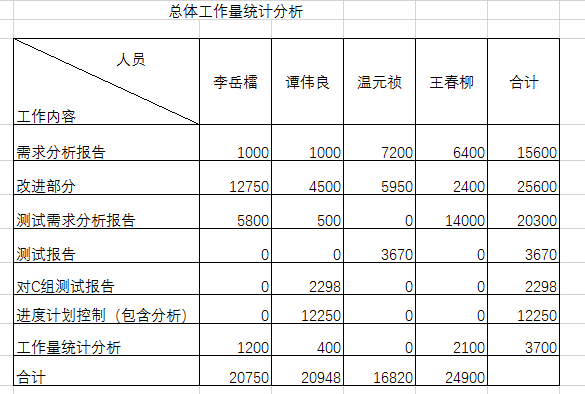
本项目在执行过程中既完成了软件工程开发所要求的开发过程，又在开发过程中通过软件进度计划与控制、工作量估计与统计分析、配置管理这些工作对整个项目开发进行管理和控制，在保证项目按照计划执行的同时，还保留了项目执行和开发过程中的各项证据，进一步形成了证据链，使得项目的开发与执行更具有说服力。

#### 2.7.2 工作量分配及其影响因素分析

##### 2.7.2.1 工作分配情况

工作分配的具体情况如表6所示。

**表6 小组工作统计表**

****

##### 2.7.2.2 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

本次实验主要的配置管理工具是github。Github为我们对数据的统计分析提供了高效清晰的数据。实验过程中需要提交的文档，以及对文档的修改记录，改进代码，测试代码都在github上有所体现，这对组员之间的交流和共享也很有帮助。

在整个配置管理的过程中，首先我们改进了目录结构，目录结构改进后，现在的目录比之前的目录更清晰明了，也方便了组员的上传，也不会再向之前一样有人在群里问该上传到哪里，也没有发生过评审组找不到我们组的文档的事情。

其次是对版本号命名的规范，之前没有明确的规定，版本号基本是看修改人的想法，每次在原基础上增加小数点后第一位，第二位都有存在，再我们进行了规范后，之后的文档版本号都清晰明了。

#### 2.8.1 文档分类组织结构

如图2.8.1所示，依据本次软件工程综合实验八个子实验的任务要求建立八个子文件夹，用于存放各子实验阶段对应的产出物，依据实验阶段对文档进行组织分类。此外，附件1存放会议记录，附件2存放参考资料，附件3存放我们组队别的组的评审意见，附件4存放PPT。增加一个旧版内容备份文件夹，存放旧版的目录及文件。此目录比之前的目录更清晰明了，也方便了组员的上传，也不会再向之前一样有人在群里问该上传到哪里。



**图2.8.1 文档分类组织结构**

#### 2.8.2 命名方法

本次软件工程综合实验文档命名由三部分组成：

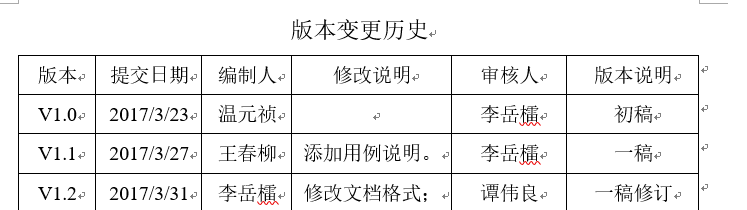
第一部分： 配置项名称。如需求规格说明书、评审意见单、项目计划规格说明书等。

第二部分：迭代号/版本号。初始版本号为V1.0，经评审后的改进版本每次加0. 1如V1.1，V1.2；

#### 2.8.3 更新说明

每次文档版本更新时，需要附上版本更新记录，记录中的字段依次为：版本号、提交日期、编制人、修改说明、审核人以及主要版本说明，版本变更记录表如表2.8.3所示。

**表2.8.3 版本变更记录表**



### 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1.0.docx”。

# 实验规模的数据统计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验1：需求分析 | | | | |
| 需求文档规模  （字数） | 需求项数  （RUCM用例描述数量） | 用例图描述 | 版本更新次数 | 累计工时（含修改） |
| 6064 | 19 | 3 | 7 | 78 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验2：需求评审（评审+复评审） | | | | | |
| 检查单中的检查项数量 | 互评审中给被评审组提出的问题数（评审+复评审） | 接收到的问题数（评审+复评审） | 老师的问题数（含各组存在的共性问题） | 接受并修改的问题数 | 累积工时（不含需求修改工时） |
| 9 | 103 | 55 | 32 |  | 68h |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验3：改进与扩展 | | | | | | | |
| 设计实现 规模（字 数） | 其他模型 （类图、 顺序图、 状态图等， 不含非规 范的示意 图） | 其他模型 中包含的 元素累积 数（节点 和边的数 量） | 代码行数 | 问题数 | 修改的问 题数 | 版本更新 （最大） 次数 | 累积工时 数 |
| 2573 | 流程图 | 3个 | 1300 | 55 | 10 | 5 | 46h |
|  | 类图 | 1 |  |  |  |  |  |
|  | 顺序图 | 1 |  |  |  |  |  |
|  | RUCM图 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验5软件测试评审（评审+复评审） | | | | | |
| 检查单中的检查项数量 | 互评审中给被评审组提出的问题数（评审+复评审） | 接收到的问题数（评审+复评审） | 老师的问题数（含各组存在的共性问题） | 接受并修改的问题数 | 累积工时（不含需求修改工时） |
| 15 | 43 | 36 | 8 |  | 20h |

|  |
| --- |
| 实验6：项目计划 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目计划书、小组会议记录和周记等累积字数 | 项目计划表（MPP）中分解的任务项数（含组合任务） | 项目计划耗费的实际工时数 | 项目计划（MPP）更新次数 | 项目的累积实际工时数 | 项目计划书、小组会议记录和周记等累积字数 |
| 34659 | 126 | 37.5 | 18 | 985.12 | 34659 |

|  |  |
| --- | --- |
| 实验7：配置管理 | 实验8：统计分析 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本更新此 数（提交次 数） | 配置管理报 告等累积字 数 | 累积工时 | 统计分析项 数 | 统计分析报 告字数 | 累积工时 |
| 217 | 5888 | 30h左右 | 8 | 4394 | 13小时 |

# 总结

## 4.1 分析

项目刚开始的时候，组内对题目的确定都有些草率，还经历了换题目的风波，其他分工基本是采取自由自愿的方式。由于采取的是自由自愿的方式，大家基本都是挑选了自己所擅长的部分，整个项目周期也大体延续了这个分工。具体的分工是由组长李岳檑负责了代码和PPT的制作，王春柳去学习RUCM工具，谭伟良学习使用MS project工具并负责以后的mpp维护，温元帧负责补充一些遗留的任务。

在需求分析阶段，王春柳学会了RUCM，所以由王春柳来完成用例的RUCM图，温元帧之前有过软件工程的经验，对UML图有一定的了解，她完成了用例图的绘制。而组长李岳檑由于对Node的学习和理解比较深刻，由他来在会议上提出需求。整个的需求文档所有人都参与了编写，都对需求有了较好的了解。

到了改进与展示阶段，由于mpp文件耗时也比较多，谭伟良就依然负责mpp文件的更新以及一些琐碎的小任务，王春柳依旧是文档的主力，所以由组长李岳檑负责整个改进与展示的部分，温元帧协助。。

在改进与展示阶段之后，是软件的测试阶段。由于我们组要进行单元测试，所以依然由擅长代码的组长李岳檑来编写单元测试用例，至于功能测试则有没有编写代码的王春柳负责，更容易问题，事实也确实如此，王春柳在测试时发现了不少编写代码未曾想到的问题，比如域名前的www。在测试评审中，各评审组提出了很多的建议和问题，都很有帮助，E组对我们的易用性提出的建议也是我们没有考虑到的。

进度计划控制是由谭伟良负责。在进度计划控制过程中，总体还是按照进度计划在推进，除了有一次刘老师临时将需求复评审（2）和实验六七八（1）调换顺序外，我们在改进的时候也因为对工作难度的错误的估计以及对node的不熟悉导致没有按时完成计划，这一点是我们需要改进的。

配置管理和统计分析部分的工作由小组组长李岳檑负责，分析的工作都由李岳檑负责，而统计的工作需要耐心和细心，所以交由王春柳同学负责。

## 4.2 经验与启发

项目的分工很重要，不仅仅是依据组员的擅长方面来分工，还应该兼顾到工作量和工作难度两方面，我们组开始的时候没有这方面的经验，所以每周工作的安排都不太合理，也导致了会有组员晚上加班来完成任务的情况。

其次，每周的工作日志也很重要，是我们完成mpp文件以及下周任务调整的依据，在项目的前半段我们没有要求每周每人必须上传工作日志，因此mpp文件的实际工时都由大家估计，在更新文件的 时候有时候自己也记不清用了多久，导致了mpp文件的不准确，在后来加上提交工作日志的要求后mpp文件更加准确合理，对我们后续的任务分配也有了一定的参考作用。

## 4.3 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 |
| 1 | 确定项目 | 刚开始没有充分了解就选择了一个难度较大的项目 | 经过老师提醒，我们又进行了充分的调研 | 在和老师沟通后，我们及时更换了题目 |
| 2 | 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 |
| 2 | 软件需求评审 | 对其他组项目的不了解加上自身对老师要求的不了解导致提不出什么问题 | 去咨询相关组，在微信上交流，仔细看老师对格式等方面的要求 | 较好的完成了需求评审 |
| 3 | 改进与展示 | 如果选择改进的部分，以及代码量不足的问题 | 去搜索网上的对 node的评论，寻找已有的第三方模块是在对node的哪方面进行改进 | 找到了一个很好的方向，改进了node的异步过载保护，并为其添加了应用场景来增加代码量 |
| 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 |
| 5 | 测试评审 | 对C组项目不了解，无法进行测试 | 学习理解scrapy，在C组的指导下完成测试 | 较好的完成了测试评审，所提出的问题包括分布式下效率不高的问题也都很好 |
| 6 | 进度计划与控制 | 对ms project的不熟悉导致项目前半个阶段一直有各种错误 | 根据老师的要求去学习修改 | 在后半段mpp文件明显出错变少 |
| 7 | 配置管理 | 目录不合理以及版本号不规范 | 重新修改了目录，在需求完成后专门制定了版本号的规范要求 | 重新修改目录后比之前更清晰，找文档也更容易了，后续的文档版本号也变的规范 |

### 4.3.1 解决非功能需求描述问题——质量属性

。在对非功能性需求描述遇到难题，无法使用RUCM进行描述，寻找到了一种新的软件工程描述方法——质量属性。描述时用到的一些术语解释如下：

 刺激源

刺激源是某个生成该刺激的实体。

 刺激

刺激是当刺激到达系统时需要考虑的条件。

 环境

刺激在某些条件下发生。当刺激发生时，系统可能处于过载或者正在运行等。

 制品

被刺激的制品，可以是整个系统，也可以是系统的一部分。

 响应

指刺激到达后所采取的行动。

 响应度量

当响应发生时，应能够以某种方式对其进行度量。

以兼容性为例，兼容性的质量属性描述为：

兼容性用例描述

|  |  |
| --- | --- |
| 场景 | 可能的值 |
| 源 | web开发人员 |
| 刺激 | 在不同操作系统上使用该系统的需求 |
| 制品 | 产出不同操作系统下的系统版本 |
| 环境 | 不同操作系统下 |
| 响应 | Node.js在操作系统和上层模块系统之间构建了一层libuv平台架构。在不同操作系统下，Node.js无需采用该软件准备的配置文件之外的规定就可以直接使用 |
| 响应度量 | 可移植的程度、在不同操作系统下需要更改的配置文件个数、除了配置文件之外是否需要其它文件 |

# 5 建议

在课程开始的阶段，大家对课程理解包括要做什么基本都不清楚，建议老师多推荐一些往届的优秀资源，并在课堂上以此为例多讲一些。