图片包含 游戏机, 物体, 钟表, 画

描述已自动生成



实验总结报告

**WeChatterBot——**

**基于ChatterBot框架的**

**微信聊天机器人**

|  |  |
| --- | --- |
| 团队编号 | **E 组** |
| 团队成员 | **郭维泽 李书缘 吕江枫** |
|  | **王元玮 王云杰 张延钊** |

**Version 1.0.0**

**版本变更历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 备注 |
| 1.0.0 | 2020/06/08 | 郭维泽、吕江枫、李书缘、王云杰、王元玮、张延钊 | 初稿 |

**目 录**

[1 引言 1](#_Toc42792733)

[1.1 编写目的 1](#_Toc42792734)

[1.2 项目概述 1](#_Toc42792735)

[2 各个实验项目小组总结 2](#_Toc42792736)

[2.1 软件需求分析 2](#_Toc42792737)

[**2.1.1 实验内容** 2](#_Toc42792738)

[**2.1.2 实验效果** 2](#_Toc42792739)

[**2.1.3 实验成果汇总** 2](#_Toc42792740)

[**2.1.4 实验有效方法和问题分析** 2](#_Toc42792741)

[2.2 软件需求评审 2](#_Toc42792742)

[**2.2.1 实验内容** 2](#_Toc42792743)

[**2.2.2 实验效果** 2](#_Toc42792744)

[**2.2.3 实验成果汇总** 3](#_Toc42792745)

[**2.2.4 实验有效方法和问题分析** 3](#_Toc42792746)

[2.3 软件产品改进与展示 4](#_Toc42792747)

[**2.3.1 实验内容** 4](#_Toc42792748)

[**2.3.2 实验效果** 4](#_Toc42792749)

[**2.3.3 实验成果汇总** 4](#_Toc42792750)

[**2.3.4 实验有效方法和问题分析** 4](#_Toc42792751)

[2.4 软件测试 5](#_Toc42792752)

[**2.2.1 实验内容** 5](#_Toc42792753)

[**2.2.2 实验效果** 5](#_Toc42792754)

[**2.2.3 实验成果汇总** 5](#_Toc42792755)

[**2.2.4 实验有效方法和问题分析** 5](#_Toc42792756)

[2.5 软件测试评审 6](#_Toc42792757)

[**2.5.1 实验内容** 6](#_Toc42792758)

[**2.5.2 实验效果** 6](#_Toc42792759)

[**2.5.3 实验成果汇总** 6](#_Toc42792760)

[**2.5.4 实验有效方法和问题分析** 7](#_Toc42792761)

[2.6 软件进度计划与控制 7](#_Toc42792762)

[**2.2.1 实验内容** 7](#_Toc42792763)

[**2.2.2 实验效果** 7](#_Toc42792764)

[**2.2.3 实验成果汇总** 7](#_Toc42792765)

[**2.2.4 实验有效方法和问题分析** 9](#_Toc42792766)

[2.7 配置管理 10](#_Toc42792767)

[**2.2.1 实验内容** 10](#_Toc42792768)

[**2.2.2 实验效果** 10](#_Toc42792769)

[**2.2.3 实验成果汇总** 10](#_Toc42792770)

[**2.2.4 实验有效方法和问题分析** 10](#_Toc42792771)

[2.8 工作量估计与统计分析 10](#_Toc42792772)

[**2.8.1 实验内容** 10](#_Toc42792773)

[**2.8.2 实验效果** 10](#_Toc42792774)

[**2.8.3 实验有效方法和问题分析** 11](#_Toc42792775)

[**2.8.4 实验成果汇总** 11](#_Toc42792776)

[3 综合实验效果与问题分析 12](#_Toc42792777)

[4 小组成员贡献表 14](#_Toc42792778)

[5 结论和建议 16](#_Toc42792779)

# 1 引言

## 1.1 编写目的

本文档内容为WeChatterBot系统的进度控制分析，首先对本项目的具体项目计划及工作分配进行介绍，然后对已经发生了的计划变更及其影响因素进行分析，之后对实验中进度控制部分遇到的问题进行总结和分析。

## 1.2 项目概述

WeChatterBot是一个以微信作为用户交互界面的聊天机器人系统，使用ChatterBot开源框架，对其进行了性能改进并在其上接入了微信个人号，从而实现自动、快速、准确地响应用户输入的能力。任何人都可以通过WeChatterBot轻松创建聊天机器人并与用户进行对话。同时，WeChatterBot编码结构灵活，方便对其进行扩展再开发。

# 2 各个实验项目小组总结

本章主要介绍整体的项目计划工作及任务分配情况，最后对项目每个阶段的工时进行统计和分析，也对每个成员的工时进行统计和分析。

## 2.1 软件需求分析

**2.1.1 实验内容**

实验一的内容为实践软件需求的分析、建模与验证方法，确定软件需求。本小组从业务需求、功能需求和非功能需求三个角度对WeChatterBot项目进行了需求分析。并结合RUCM、用例图等图表工具对需求进行了详细描述。

**2.1.2 实验效果**

通过软件需求分析，本小组确定了WeChatterBot项目的业务需求，从用户和维护人员角度详细分析了业务需求。之后，本项目通过用例图和RUCM的形式，描述了WeChatterBot的9个功能需求。同时，从可靠性、高效性等角度详细描述了WeChatterBot的非功能需求。最后，本项目对环境需求进行了约束，并通过类图的方式给出了系统设计方案。本小组通过软件需求分析，为之后的实验和开发打好了基础。

**2.1.3 实验成果汇总**

在软件需求分析实验中，本小组共完成8个版本的需求规格说明书，总结了9个功能需求，通过RUCM进行详细描述；总结了4类非功能需求，分别进行了描述。通过软件需求分析，明确了实验3中要实现的拓展功能，同时为之后的实验4软件测试提供需求上的参考。

**2.1.4 实验有效方法和问题分析**

我们总结实验有效方法如下：

1. 需求确定是系统开发中需要技术最少的一环，但是一旦出现问题，将在后续的开发工作中付出巨大代价。本组通过大约4周的时间（除6-8外所有实验中时间最长），完成了需求的确定，从而保证后续工作顺利进行。
2. 需求建模可以明确需求内容，如RUCM，质量属性场景，或可视化建模如用例图、类图。通过这些模型，可以让我们对每一项需求形成足够的认识和充足的讨论。
3. 学会砍需求的能力很重要，同时要结合实际情况确定需求（如响应时间的性能需求）。

需求分析阶段存在的问题：本组在需求分析阶段并没有出现严重问题，存在的小问题为软件接口定义不全。

## 2.2 软件需求评审

**2.2.1 实验内容**

软件需求评审的实验内容为：制作软件需求评审说明书评审表单和软件需求规格说明文档检查单。根据制作的检查单的相关要求对负责的组进行评审和复评审，同时对于其他组对本组的评审结果进行修正。

**2.2.2 实验效果**

在该实验中，成功的完成了软件需求评审说明书评审表单和软件需求规格说明文档检查单，同时在评审中，找出C组14个问题，找出D组12个问题，同时收到F组17个问题，接受15个，拒绝两个；收到G组23个问题，接受19个，拒绝4个；在复评审过程中，找出F组18个问题，其中15个轻微，3个一般；找出G组17个问题，其中11个轻微，6个一般；收到C组9个问题，其中5个轻微，3个一般，一个严重；部分接受4个，接受4个，拒绝一个。收到D组8个问题，其中5个轻微，2个一般，一个严重；接受7个，部分接受一个。

**2.2.3 实验成果汇总**

通过实验二，完成了软件需求评审说明书评审表单和软件需求规格说明文档检查单的制作，并根据其他组的评审结果，修改了我们组需求规格说明书中的若干问题。

在对其他组进行评审的过程中，我们组发现了一些关键问题：比如用例图中用例之间的关系表示错误、未标柱软件版本信息、用例图与需求的表述不一致等问题。针对相关问题我们与相关组进行协商和沟通，合力解决争议。

我们组在实验过程中也收到了其他组对于我们的需求规格报告的问题：比如对于用例之间的关系表述不准确、相关需求没有进行建模、对于并发要求描述的比较含糊等等问题。对于其他组提出的评审意见，我们小组都进行了充分的讨论，对一致同意的进行及时的采纳改正，对存在争议的进行及时的沟通交流。最终完善了本组的需求规格文档。

**2.2.4 实验有效方法和问题分析**

实验有效方法：在本实验中，我们首先确定了从完整性、一致性、准确性、规范性、易理解性、合理性和可追踪性来对其他组的需求规格文档进行评审，根据制定的相关要求制作出软件需求评审说明书评审表单和软件需求规格说明文档检查单。

实验问题分析：在实验过程中，对于评审说明书以及检查单的认知并不准确，在文件命名上存在问题。在汇报时老师针对该情况提出了我们在命名上的错误，针对命名问题我们进行了相关的改正。

同时，在评审过程中，有其他组提出，系统设计图应该用系统框架图而不是类图。关于系统设计采用什么图来描述的问题，我们组经过讨论认为采用类图可以更加直接的了解各个模块以及接口之间的关系。而且根据WeChatterBot的代码结构，用类图来表示是更加符合直观认识的。会出现该问题的原因在于其他组并未对阅读WeChatterBot的原始代码，而且系统设计采用系统框架图也是合理的，只不过在该项目中采用类图是更加优秀的选择。

## 2.3 软件产品改进与展示

**2.3.1 实验内容**

实验三要求实验小组实验1~2的基础上，设计并实现软件需求。根据本小组的软件需求说明书，我们将工作分为了WeChatterBot后端，WeChatterBot数据库部分，微信端，管理员维护端四个部分进行开发。

**2.3.2 实验效果**

在实验三的过程中，我们共完成了WeChatterBot后端，WeChatterBot数据库部分，微信端，管理员维护端共四个模块的开发工作。Chatterbot后端完成了logic adapter部分，search algorithm部分，tagging部分，trainer部分的开发与优化工作。数据库部分设计了新的数据库表单，完成了storage\_adapter\_new，sql\_storage\_new部分的开发工作。管理员维护端，完成了管理员维护前端界面，管理员后端与数据库交互部分的开发工作。微信端完成了微信公众号的注册，与get response，wechat 以及签名验证部分的开发工作。最终，我们通过Flask框架实现了后端平台，并部署于阿里云服务器上。

**2.3.3 实验成果汇总**

通过实验三，小组完成了微信聊天机器人的开发，达到了用户在微信公众号与机器人进行聊天的要求。在微信公众号服务方面：实现了签名验证功能，完成微信端服务器与项目后端服务器的连接；实现了被动回复功能，达到用户在聊天框发送文字时，机器人进行回复的效果。在ChatterBot方面：扩充了语料库与对话规则，完善了对多种数据库的支持。在管理员方面，完成了运营维护界面的开发，使得管理员可直接在可视化界面上对机器人的对话规则进行管理。代码编写完成后，我们组将该项目部署在阿里云上使其成功运行，同时注册了一个微信公众号用于WeChatterBot微信端的使用。

**2.3.4 实验有效方法和问题分析**

实验有效方法：在该实验中，我们首先基于WeChatterBot的源代码，并结合实验1-2中的需求分析，确定了需要添加的部分。在WeChatterBot部分，我们在数据库中添加了对话规则有关的表，并在存储部分添加对话规则有关的接口，同时还扩充语料库以及修改相关算法，使其可以支持中文且可在可观的时间内获得恢复。在维护人员方面，我们组完成了维护界面以及维护人员相关操作的开发，使得维护人员可以直接通过界面对后台使用的数据库进行增删改查。在微信方面，实现了签名验证功能，完成微信端服务器与项目后端服务器的连接；实现了被动回复功能，达到用户在聊天框发送文字时，机器人进行回复的效果。

实验问题分析：在该实验中，存在消息回复过慢导致微信端认为超时的问题。经过对于WeChatterBot代码的分析，最后发现是WeChatterBot的filter部分对于最后结果会进行序列化，从而费时。最后将filter中的序列化操作去除从而达到提高回复速度的特点。去除序列化之后，在维护人员进行数据操作时发现存在部分情况无法操作的情况，经过分析，得出的原因是数据库采用的是SQLAlchemy框架，若操作的时候不将其加入到session中则无法进行操作，针对该问题，在维护人员进行操作时，都会把相关的操作数据加入到session中，从而解决问题。

微信公众号的开发存在较大问题，有多个hotfix提交。在开发时没有对应完善的测试环境，开发测试尚不充分就进行了代码提交。而代码的一些涉及到微信API与HTTP请求的部往往难以通过GitHub上的代码review发现其中错误，造成在merge之后紧急fix bug的情况。最好应该是有一个小服务器搭建一个测试环境，用来进行微信这部分的测试工作，就能大大减少出问题的概率。

## 2.4 软件测试

**2.2.1 实验内容**

软件测试是软件开发中的必要组成部分，目的是尽可能检测出软件存在的质量问题。因此，我们精心设计了测试用例，科学制定测试计划，并使用黑盒白盒等多种方法对WeChatterBot系统进行充分的测试。在此过程中，我们积极进行交流，发现问题及时反馈，开放人员迅速改正，实现了良好的测试效果。

**2.2.2 实验效果**

在实验三的过程中，我们首先从功能需求和非功能需求两个方面明确了WeChatterBot系统需要进行测试的需求点以及测试工具需求，并针对测试需求项制定了相应的测试方案，得到测试需求规格说明书。我们的选择的测试方法分为黑盒测试和白盒测试。其中，黑盒测试覆盖了所有的功能点，使用了等价类划分的方法，将功能需求划分为5类测试用例，将非功能需求划分为3类测试用例。而白盒测试由对应的开发人员进行。

然后我们根据制定的测试需求点，对WeChatterBot系统进行实际测试，得到软件问题报告。在黑盒测试过程中，共发现了4处功能需求问题，2处非功能需求的问题，均在测试后及时改正。

**2.2.3 实验成果汇总**

通过实验四，我们得到了测试需求规格说明书和软件问题报告。其中，测试需求规格说明书共42页，详细说明了测试的目的、范围、计划、评价标准等。软件问题报告共15页，记录了最终回归测试的结果，所有的黑盒测试点均通过。

最关键的是，通过本次测试，我们找到了许多在开发中疏忽的或是理解有误的问题，对我们改进和完善系统有极大的帮助。同时，经过本次测试，我们学习使用了测试工具，熟悉了测试过程中的一些要点，并提高了软件测试和文档撰写的能力。

**2.2.4 实验有效方法和问题分析**

实验有效方法：在黑盒测试部分遇到的主要问题是如何选取等价类划分的标准。由于WeChatterBot是一个聊天机器人系统，核心功能仅有聊天一项，且难以划分。经过小组讨论后，我们决定按照用户输入句子的语义类别进行划分。在设计测试用例时，我们充分考虑了可能的用户输入，但是仍有遗漏。经过讨论交流，我们进一步完善了测试用例。

实验问题分析：在该实验中，由于我们的测试水平有限，必然有个别遗漏的软件需求没有覆盖，造成软件仍有bug未解决。

## 2.5 软件测试评审

**2.5.1 实验内容**

软件测试评审实验内容为：制作软件测试需求评审说明书评审表单和软件测试需求规格说明文档检查单。根据制作的检查单的相关要求对负责的组进行评审和复评审，同时对于其他组对本组的评审结果进行修正。

**2.5.2 实验效果**

在该实验中，小组完成了软件测试需求评审说明书评审表单和软件测试需求规格说明文档检查单。

在初评审工作中，找出A组29个问题，其中18个轻微、11个中等；找出B组20个问题，其中9个轻微、11个中等。同时收到H组13个问题，其中6个轻微、6个中等、1个严重，接受9个，拒绝4个；收到I组8个问题，其中4个轻微、4个中等，全部接受；

在复评审过程中，找出H组15个问题，其中8个轻微，7个一般；找出I组4个问题，全部为轻微；收到A组23个问题，其中13个轻微、7个中等、3个严重，全部接受。收到B组6个问题，全部为轻微；接受5个，拒绝1个。

**2.5.3 实验成果汇总**

通过实验五，小组完成了对A、B、H、I四个小组的测试评审，并根据这四个小组对我组的评审意见，完善了本组的测试需求规格说明书，改正了一些软件问题。

在对他组的测试评审工作中，小组找出了一些关键问题，如：没有测试数据导致测试根本无法复现，异常情况触发条件不明，黑白盒测试分类错误，测试用例前提约束设置不当等。并就评审发现的问题与被评审组进行沟通，合力解决争议。

在本组收到的测试评审意见中，也得到了宝贵的建议。如：没有从添加规则然后测试对应规则这样完整的响应测试，对异常输入没有测试，管理员维护界面的一些异常响应不正确，公众号缺少新手使用指引等。对于他组提出的评审建议，我们小组都进行了充分的讨论，对一致同意的进行及时的采纳改正，对存在争议的进行及时的沟通交流。最终完善了本组的测试文档与软件项目。

**2.5.4 实验有效方法和问题分析**

实验有效方法：在本实验中，我们从软件测试需求文档与软件项目测试两方面进行评审。在文档评审上，我们首先确定了从完整性、一致性、准确性、规范性来对其他组的需求规格文档进行评审，根据制定的相关要求制作出软件测试需求评审说明书评审表单和软件测试需求规格说明文档检查单。在软件测试上，我们会动手重现被评审组的测试用例，考察测试用例的可复现性。

实验问题分析：测试评审工作中，本组被多个小组均提出了同一评审意见即关于机器人回答的质量这一问题。对于机器人给出的回复，若回复不合理即认为这是出现了错误。但其实如何评判人工智能回复的质量这一问题目前尚无定论。虽然人们都本能的希望机器人的回复应当是合乎逻辑的，但其实在本项目中，只要机器人的回复是符合对话规则的，那么就说明软件并没有出错，只不过是该对话规则可能并不太符合逻辑。分析出现该争议的原因，可能是本组在需求分析时对于机器人回复质量的需求定义不够明确，评价指标过于个人化。

## 2.6 软件进度计划与控制

**2.2.1 实验内容**

软件进度计划与控制部分的主要工作是对整个软件项目进行进度控制与任务协调。在实验过程中，我们使用Microsoft Project工具对软件项目进行计划和监控，同时安排专人为该项目的工作量统计工作进行计算。软件项目进度控制实验中可以拆分成以下实验内容，即：任务分解与分配、任务完成标准、任务进度计划、进度监控、任务细化与调整和小组协同。其中，除了小组协同之外的实验内容我们通过Microsoft Project进行，小组协同的方法通过腾讯会议以例会的形式进行网上交流。实验最后产出进度控制分析报告。

**2.2.2 实验效果**

从下面实验的燃尽图中可以看到，我们组的工时实际进展与基线的计划进展有三次产生了比较大的分歧，其中之一就是在项目的需求分析阶段（3.28左右），此时由于我们对项目的初步把握不足，实际的项目进展有一定的延期现象。第二就是在软件开发的期间（4.25左右），因为我们没有在初期对项目的开发情况做出准确的估计，导致在真正开发的时候我们发现项目实际要做的比我们预想的要多一点，在4.25这个时间点来到之际，我们的项目有微小的工时延期。第三则是在软件测试和测试评审期间（5.9左右），因为我们组测试所占工时在一开始计划的时候就相对较少，所以我们只花了一周的时间就完成了所有的测试工作，相对于基线计划工时的完成稍有提前。

整体来说，小组本次实验效果并不算完美，因为在项目初期曾有多次因为对软件进度计划考虑不全导致的基线变动，但是在软件开发情况逐渐平稳后，小组的软件进度变得容易控制，甚至有一部分的超前完成。实验的效果还是可以接受的。

**2.2.3 实验成果汇总**

首先对项目的甘特图进行介绍，我们项目的甘特图采用Microsoft Project软件进行绘制，上方是时间轴，下方的任务区长度代表具体执行的任务。出于篇幅的限制，我们这里不会将所有的工作分解结构和甘特图进行一一对应并放在文档中，具体的甘特图可以查看我们组的“实验进度安排.mpp”项目文件。

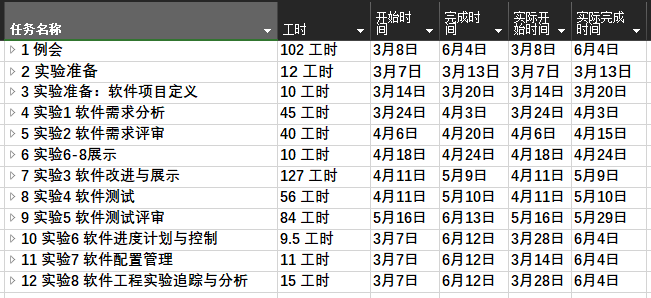


图1各个实验任务分配

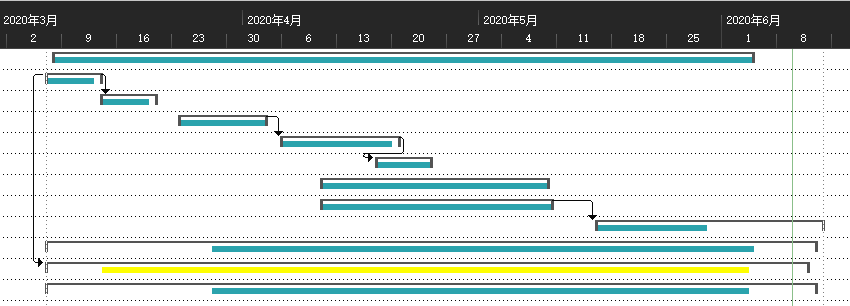


图2项目甘特图

在我们的甘特图中，可以看到，实验6-8和例会是贯穿整个实验项目生命周期的。并且实验3的软件改进与展示是与实验4的软件测试覆盖机型的，这是因为实验4的时间有限，我们在开发阶段就要求负责测试的同学进行测试用例的编写。

本项目实际工时521.5工时，工时完成情况100%。对于项目阶段工时的分析如图3所示，从图中可以看到，我们小组的大部分工时分配都着重在实验3、实验4和实验5上面，每周也有大部分时间放在例会中，由于我们在例会上能够完成当周大部分的任务，所以例会的工时较高也是可以预见的。

在项目开发的重点，即软件开发、软件测试和测试评审中，我们总共投入了超过250个工时，占整个实验中的大部分，一方面说明我们对软件开发和测试的侧重非常高，另一方面也说明我们小组的软件为了完成需求需要编写很多的项目代码。

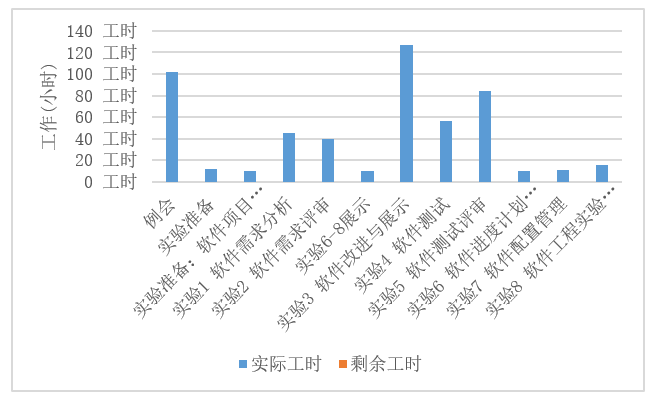


图3 项目各个阶段工时统计

图4 项目燃尽图

**2.2.4 实验有效方法和问题分析**

从本实验中总结的有效方法如下：

问题：刚开始对项目整体没有太大概念，无法做到详细的工作分解，会导致项目的计划预计不准确。

有效方法，通过例会的形式与小组成员进行讨论，可以以工作日的时间长度来对项目某个阶段可能安排的任务进行分解，最后在任务发生前进行再分解。

项目计划的主要问题集中在三点：1.项目管理人员对项目管理以及管理软件本身的生疏；2.项目管理人员对整个软件工程中项目的生命周期没有准确把握；3.文档内容方面准备的欠缺，没有准备模板文档，也就不知道产出文档应有的内容

## 2.7 配置管理

**2.2.1 实验内容**

配置管理（Configuration Management，CM）是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理实验要求实验小组对实验过程的成果（文档和代码）进行有效管理，以保证项目质量。

**2.2.2 实验效果**

本小组主要使用GitHub作为项目的配置管理工具，来记录软件产品的演化过程，同时完成实验文档的提交。由于本次软件工程综合实验主要以各阶段实验过程文档作为实验结果，而GitHub不适合文档的配置管理，因此，本小组采用OneDrive对实验中的各阶段文档进行配置管理，在OneDrive中编制完成可提交的实验文档后，再统一提交至GitHub。

通过配置管理，本小组有效且高效地管理了实验过程的文档和代码文件。

**2.2.3 实验成果汇总**

在项目开始时，本小组完成了配置管理计划\_V1.1.0，用于规定配置管理方案，让组员在实验过程中遵守。

在项目结束时，本小组完成了配置管理分析报告\_V2.0.0，总结了各实验文档的版本情况和各代码项目的管理情况，

**2.2.4 实验有效方法和问题分析**

本实验总结的有效方法如下：通过制定有效的配置管理方案可以提高实验过程效率同时保证实验成果质量。

总结的主要问题如下：

1. 文档版本管理中存在小修订未记录的情况。
2. 主项目存在个别提交信息不规范的情况。
3. 代码项目存在个别使用force-push的情况。

## 2.8 工作量估计与统计分析

**2.8.1 实验内容**

软件工程实验追踪与分析部分的主要工作是通过对实验项目的跟踪，分析各项软件工程实验过程的执行情况和结果，比如工作量差异及其原因分析等。实验输入为各项软件实验过程统计数据，输出文档为工作量统计分析报告。

**2.8.2 实验效果**

实验八用于统计基于基于ChatterBot框架的微信聊天机器人(WeChatterBot)在每个实验阶段各组员的工作量。通过对实验项目的跟踪，分析了每个实验过程的执行情况和结果，对比各组员的工作量差异及原因分析，最后给出整体的分析与总结。

**2.8.3 实验有效方法和问题分析**

问题：初次设计工作量统计方法时，统计方法为每个成员自主汇报工作量，缺少客观评价过程，相对设计不合理。

有效方法：后来经过讨论，确定了开会讨论制定计划工时，工作完成后汇报实际工时->小组内成员对其工作进行打分确定最后工作量的模式。通过这种方式，我们以每次工作的用时为基础统计工作量，得到了小组内成员的一致认同。

**2.8.4 实验成果汇总**

最终，我们完成了对实验准备阶段和对实验1-8的工作量统计与分析，记录了小组内每个成员在每个实验中的工作量统计，并且统计了小组成员的总工作量。对每一个试验阶段的工作，统计了每位组员的具体工作内容以及工时，绘制了贡献度饼图，并在最后对小组内每位成员的贡献度进行了汇总。根据统计结果，最终小组内每位成员的总工时占比范围为14.8%-16.7%，相对较为平均，证明了我们的任务分配与工作量统计方法均较为合理。

# 3 综合实验效果与问题分析

介绍完实验1-8的具体内容和问题分析之后，本章我们对整个软件工程实验本身进行简单的总结和问题分析。在本次的软件工程实验中，我们从3月6日开始，到6月初一共经过了16周的具体工作时间。其中虽然包括了一周的五一假期，但是我们的进度仍然按照安排有序地推进，最后完成了WeChatterBot这个聊天机器人的项目。

从上文对实验1-8的总体分析可以看出，虽然在实验的进度方面我们曾有几次把控不好，但是对应于每一个实验，我们都在预期的时间内完成了实验预计的产出，也做足了实验应有的内容。可以说，在整体实验的完成度上，我们小组的情况是非常完美的。

但是实验的进展却并非是一帆风顺的，我们整个实验的进展大概分为三个阶段：准备期，发展期和收尾期。

准备期指的是在实验1开始之前，我们有大约两周的实验准备阶段，从实验的选题再到人员的分工和任务的分解，是为了整个实验的进行打下基础的阶段。这一阶段的效果并不理想，主要体现在我们花费了第一周的时间却并没有商量出一个较为理想的题目和分工，经过了第二周甚至是第三周的部分时间，我们才磨合成功，进入实验的正轨。

这一问题的主要原因，我们认为主要有两条。其一，我们对于软件工程实验本身的理解并不到位，虽然我们组的有些同学在本科阶段曾经选修过软件工程，但是对软件工程的理解仍然不够深入，没有能在实验的初期把握整个实验的可能进展，同时准备好符合格式要求的文档。没有在初期对整个实验的可能发展进行把控，这就意味着初期对实验整体的计划、安排，在后面都是有较大改动的风险，甚至推倒重来的。对于一个开发和测试任务很紧张的项目来说，等到了项目的中期再进行计划的修改是不可接受的。而没有准备好符合实验要求的文档则意味着，软件工程的核心——项目的计划和文档的管理会因此产生很严重的问题，尤其是在整个项目以瀑布形式进行的情境下。项目的管理者都不知道接下来的项目会如何发展，应当从哪些方面入手，生成的文档应该包括哪些内容，这样的项目做起来一定是非常吃力的。当然，从好处来讲，这也是这门课的一大成果，毕竟在校的试错成本非常低。

接下来再来谈一谈项目的发展期，发展期包括了从需求分析到软件测试的全部过程。因为我们经过了开始两周的磨合，能够在软件工程这条老路上坚定的走了下去。在这个阶段，我们小组的进度是非常平稳的，即使经过了五一一周的放假（实际上我们小组的进度仍然在被推进），和老师途中两次对项目安排的较大调整，我们也准时准点地完成了项目的开发和测试工作。

这一阶段能够进行的如此顺利，主要归功于组员之间的协力配合，以及单独每位成员的能力和责任感。在我们小组的任务进行中，从来没有一位同学发生掉队、或是偷懒的情况，大家甚至会主动参与到项目的推进过程中来，飞快地完成自身的任务，并帮助那些因为项目初期分配工时不妥导致可能要出现延后的任务。组里的六个人通常是一次开会只需要两小时的时间，就完成了一周内大部分的任务，这是非常值得肯定和赞赏的。

但是到了项目的最后，也就是收尾期，项目的进展曾一度变得缓慢起来。这一阶段主要包括从实验6-8到最后的总结部分。对于实验6-8来说，项目计划决定了任务的走向，配置管理确保了项目的产出，工作量分析公平且公正地对待了每一位项目组员的劳动成果，都是整个项目的重要组成部分。但是在最后的总结部分其实是有许多问题存在的，其一便是项目计划的多次更改导致基线反复变更，并由于项目管理人员的生疏，以及项目管理软件可能本身体量较大，导致项目计划过于混乱，只能大致反映出一周内任务的完成情况，却不能较准确地显示出实际与基线的差距。虽然这一问题基本上所有的组都遇到了，但从项目管理的角度来讲，仍是一次不成熟的实践，不成熟的计划会导致不准确的任务，不准确的任务就无法在给定时间内完成准确的需求，这就给这个项目的进行带来了很大的阻力。同时，也就影响了配置管理和工作量的分析，这就导致最后项目需要频繁地讨论才能完成原本预计中早就该做完的工作。

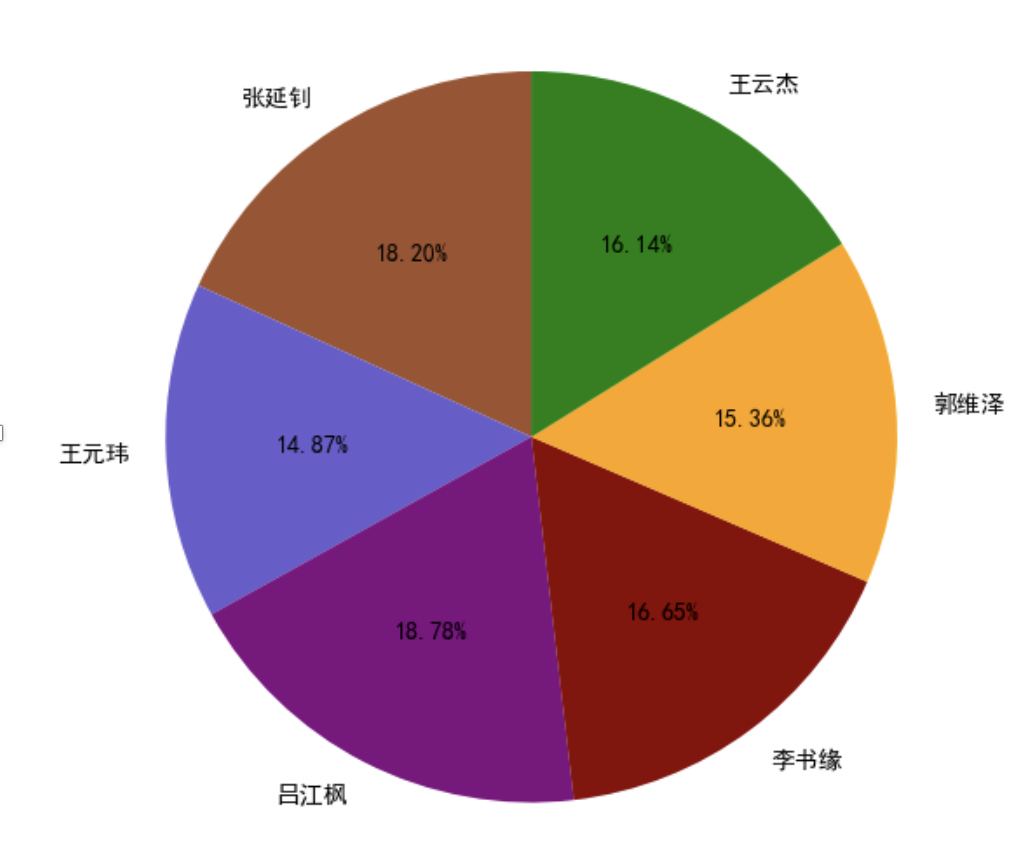
虽然从严格的角度来说，这不是一次完美的软件工程实验，但是完成项目的每一个人也经过了认真的参与，将自身的优势和精力都发挥在本次的项目中，从而完成了这样一个流程和产出都相对完善的综合实验。从各种角度上来说，对于我们这都是一次成功的实验。或许我们并不会在将来的工作中扮演本次实验中的角色，但是一起合作的经历与交流的经验却是胜任任何一个岗位应该积累的东西。反思问题，改善自我，不仅是项目，人生也值得如此对待。

# 4 小组成员贡献表

综合实验1-8，各组员总工时统计如下：

|  |  |
| --- | --- |
| 姓名 | 工作量 |
| 王元玮 | 94.1152 |
| 郭维泽 | 97.21651 |
| 王云杰 | 102.1533 |
| 李书缘 | 105.3812 |
| 吕江枫 | 118.8624 |
| 张延钊 | 115.1914 |

各组员总贡献率如下图所示。



由上图可以看出，总的来说，小组内每位成员的工作量分布，大致相同，说明我们的任务安排相对合理。各个实验阶段的工作量统计与分析，请参考《E\_工作量统计与分析报告1.0》

# 5 结论和建议

本部分主要介绍项目的非功能测试的测试工具需求、测试需求项的列表以及具体的测试用例设计。非功能测试的测试主要与需求分析说明书之中的非功能需求部分相对应，由于1.3节测试工作内容和范围所述，本测试只关注性能需求中的高效性、可靠性部分，具体测试用例和测试见第4.3节。