综合实验分析与总结

基于Flask的深度学习自动化部署系统

|  |  |  |  |
| --- | --- | --- | --- |
| 编写 | 许京爽，张利鹏，聂磊 | 日期 | 2020年6月11日 |
| 校对 | 聂磊 | 日期 | 2020年6月11日 |

北京航空航天大学 计算机学院

软件工程综合实验课 C组

二〇二〇年六月十一日

**修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 日期 | 修改章节 | 修改说明 | 修改人 | 审核人 |
| 1.0.0 | 2020.6.11 |  | 完成第一版 | 许京爽，张文斌，聂磊 | 张竹君 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1 项目概述 1](#_Toc42818666)

[2 实验一 软件需求分析小结 1](#_Toc42818667)

[2.1 软件需求分析说明 1](#_Toc42818668)

[2.2 数据统计 1](#_Toc42818669)

[2.3 数据分析与说明 2](#_Toc42818670)

[2.4 质量水平 2](#_Toc42818671)

[2.5 小结与建议 2](#_Toc42818672)

[3 实验二 软件需求评审小结 3](#_Toc42818673)

[3.1 软件需求评审说明 3](#_Toc42818674)

[3.2 数据统计 3](#_Toc42818675)

[3.3 数据分析与说明 3](#_Toc42818676)

[3.4 小结与有效方法建议 4](#_Toc42818677)

[4 实验三 软件设计与实现小结 4](#_Toc42818678)

[4.1 项目说明 4](#_Toc42818679)

[4.2 数据统计 4](#_Toc42818680)

[4.3 数据分析与说明 5](#_Toc42818681)

[4.4 小结与建议 5](#_Toc42818682)

[5 实验四 软件测试需求小结 5](#_Toc42818683)

[5.1 软件测试需求说明 5](#_Toc42818684)

[5.2 数据统计 6](#_Toc42818685)

[5.3 数据分析与说明 6](#_Toc42818686)

[5.4 小结及有效方法建议 7](#_Toc42818687)

[6 实验五 软件测试评审小结 7](#_Toc42818688)

[6.1 软件测试评审说明 7](#_Toc42818689)

[6.2 数据统计 7](#_Toc42818690)

[6.3 数据分析 8](#_Toc42818691)

[6.4 小结及有效方法建议 8](#_Toc42818692)

[7 实验六 项目进度计划与控制小结 9](#_Toc42818693)

[7.1 项目进度计划与控制说明 9](#_Toc42818694)

[7.2 数据统计 9](#_Toc42818695)

[7.3 数据分析与说明 9](#_Toc42818696)

[7.4 小结及有效方法建议 10](#_Toc42818697)

[8 实验七 软件配置管理小结 10](#_Toc42818698)

[8.1 配置管理 10](#_Toc42818699)

[8.2 数据统计 11](#_Toc42818700)

[8.2.1 提交次数随天数的变化情况 11](#_Toc42818701)

[8.2.2 一周内的提交变化情况 11](#_Toc42818702)

[8.2.3 各制品的提交次数 12](#_Toc42818703)

[8.3 版本控制方法 12](#_Toc42818704)

[8.3.1 Git版本标识 12](#_Toc42818705)

[8.3.2 优点 13](#_Toc42818706)

[8.3.3 缺点分析 13](#_Toc42818707)

[8.4 小结与建议 14](#_Toc42818708)

[9 实验八 工作量统计小结 14](#_Toc42818709)

[9.1 工作量统计说明 14](#_Toc42818710)

[9.2 数据统计 14](#_Toc42818711)

[9.3 数据分析 14](#_Toc42818712)

[9.4 小结及有效方法建议 15](#_Toc42818713)

[10 综合实验总结 15](#_Toc42818714)

[参考文献 15](#_Toc42818715)

基于Flask的深度学习自动化部署系统

综合实验分析与总结

**关键词：**深度学习，数据即服务（DaaS, Data as a Service）

**摘要：**本文描述了2020春季学期《软件工程综合实验》课程中，C组成员开发的深度学习自动化部署系统的综合实验分析与总结。

**专业术语清单：**

|  |  |
| --- | --- |
| **专业术语** | **中文解释** |
| SRS | SRS指软件需求规格说明书（Software Requirements Specification），是软件需求设计阶段的产物，描述了软件所需满足的需求。 |
| Python | Python是一种跨平台的计算机程序设计语言，具有面向对象和动态类型的特性。Python最初被设计用于编写自动化脚本，随着版本的不断更新和语言新功能的添加，越多被用于独立的、大型项目的开发。 |
| URL | 统一资源定位系统（Uniform Resource Locator）是因特网的万维网服务程序上用于指定信息位置的表示方法。 |
| 深度学习 | 深度学习是机器学习领域中一个新的研究方向，通过学习样本数据的内在规律和表示层次，让机器能够像人一样具有分析学习能力，能够识别文字、图像和声音等数据。 |
| 模型部署 | 将训练完毕的深度学习模型放置在特定硬件平台上，辅以合适的软件环境，使该模型能够被用户使用。 |
| HTTP | 超文本传输协议（英文全称：HyperText Transfer Protocol）是一种用于分布式、协作式和超媒体信息系统的应用层协议。HTTP是万维网的数据通信的基础。 |
| REST API | 表现层状态转换（英文全称：Representational State Transfer）是一种万维网软件架构风格，目的是便于不同软件/程序在网络中互相传递信息。表现层状态转换是根基于超文本传输协议之上而确定的一组约束和属性，是一种设计提供万维网络服务的软件构建风格。 |

# 1 项目概述

高效的深度学习模型需要通过实际部署才能在生产生活中发挥作用。常用的部署方式是将模型包装成Web服务，向外界提供基于HTTP协议的接口，从而提供各类预测功能和服务。对于算法研究人员而言，将深度学习模型部署为Web服务的学习成本较高，通常需要工程开发人员协助部署，这一定程度上降低了算法研发和实际部署的效率。本软件旨在通过自动化部署深度模型，降低算法研究人员实际部署模型的门槛，提高深度学习模型迭代研发和部署的效率。

项目采用基于Python语言的轻量级Web框架Flask和容器技术作为实际的模型部署工具，自动化部署系统包括了网站前端页面，网站后端模块和模型部署模块。用户通过Web界面完成模型的上传和部署，并在随后通过Web界面获得访问该模型的REST API。在完成模型的部署之后，用户可在需要使用该模型的场景下调用此REST API，把待识别的图片发送至服务器，服务器将通过HTTP响应返回图片的识别结果。

# 2 实验一 软件需求分析小结

## 2.1 软件需求分析说明

需求分析过程通过对项目预期功能的评估，在明确定义和用户需求的基础上，进一步确定软件的功能需求、性能需求，以及软件对运行环境的要求。需求分析过程以Flask开源框架及相关参考资料为输入，分析软件设计需求，结合软件工程综合实验具体要求，输出软件需求规格说明书。

## 2.2 数据统计

如下表所示统计软件需求分析过程中涉及到的软件需求条目，文档迭代过程等相关信息。

**表 2.1测试需求制品数据统计**

|  |  |  |  |
| --- | --- | --- | --- |
| 制品 | 字数 | 图表 | 需求个数 |
| 《软件需求规格说明书》 | 10353 | 7 | 24 |

**表2.2实验一制品目录表**

实验1：软件需求分析

├──C\_Flask\_3.26\_软件需求规格说明书\_ver1.0.0.docx

├──C\_Flask\_4.2\_软件需求规格说明书\_ver2.0.0.docx

├──C\_Flask\_4.2\_软件需求规格说明书\_ver2.0.1.docx

├──C\_Flask\_4.2\_软件需求规格说明书\_ver2.0.2.doc

├──C\_Flask\_4.2\_软件需求规格说明书\_ver2.0.3.docx

├──C\_Flask\_4.2\_软件需求规格说明书\_ver2.0.4.docx

├──C\_Flask\_4.9\_软件需求规格说明书\_ver2.1.0.docx

├──C\_Flask\_4.9\_软件需求规格说明书\_ver2.1.1.docx

├──C\_Flask\_4.9\_软件需求规格说明书\_ver2.1.2.docx

├──C\_Flask\_4.17\_软件需求规格说明书\_ver2.2.0.docx

├──C\_Flask\_5.9\_软件需求规格说明书\_ver2.3.0.docx

├──C\_Flask\_5.18\_软件需求规格说明书\_ver2.4.0.docx

└──C\_Flask\_5.18\_软件需求规格说明书\_ver2.4.1.docx

## 2.3 数据分析与说明

软件需求分析是本项目的第一个重要环节，其决定了接下来开发工作的重心和产品质量。软件需求分析文档是本实验修改次数最多的文档，从3月26日的初稿，再到5月18日的终稿，共经历了12次的迭代。终稿的字数达到10353字，共包含了24条软件需求。其中包括15条功能需求，2条性能需求，2条用户接口需求，1条软件接口需求，4条软件质量特征需求。

软件需求分析过程共有7名组员参与，涉及两次小组会议讨论，两周的充分设计和修改，并形成了供开发人员参考的一个相对完善的文档。在需求分析规格说明书中，需求设计人员围绕开发产品的功能需求、性能需求、接口需求、软件质量特征这四大方面进行了充分的说理论证，在充分明确Flask框架和项目规划的基础上，细致分析软件与外部交互过程的众多细节，形成了一个完整的文档结构。

## 2.4 质量水平

软件需求规格说明书的总体结构较为完备，涵盖了软件需求设计过程的主要考虑因素。在两次需求评审的过程中，文档撰写人员充分吸纳其他组成员的建议，对软件需求规格说明书进行了四次修改，消除了格式错误和语句错误，增添了部分需求描述细节，较好地提升了软件需求规格说明书的质量。

在组间互评过程中，本组接收到的最多的反馈是文档可读性较差，缺乏清晰易读的图表。经过小组会议的讨论与分析，文档撰写人员认为这一问题是由于文档撰写过程中没有邀请组外人员阅读与点评，导致读者与撰写者的角度出现较大偏差，进而影响了读者的阅读体验。除此之外，文档格式规范性不足，描述不够准确也是常被提及的问题。

## 2.5 小结与建议

对于文档可读性较差的问题，首先需要从文档总体框架入手，加入更多的图表元素的设计，以满足读者对于可读性的需求。此外，在文档完成后邀请组外人员阅读与点评也是一个很好的方法，这可以帮助文档撰写者得到合理的反馈，进而改进文档，使其更容易被开发人员和测试人员理解，这也进一步地提升了整体项目的质量。

# 3 实验二 软件需求评审小结

## 3.1 软件需求评审说明

软件需求评审环节是对软件需求规格说明书质量的验证。本环节采用组间互评的方式，我们小组对A，B，D，E四个小组的软件需求规格说明书进行了评审，同时接受了这四组对我们的需求规格说明书的评审。

## 3.2 数据统计

**表3.1 软件需求评审数据统计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 检查项数量 | 组号 | 向被评审组提出的问题数 | 接收到的问题数 | 接受并修改的问题数 | 工时 |
| 20 | A | 17 | 15 | 14 | 21h |
| 20 | B | 13 | 8 | 8 | 18h |
| 20 | D | 11 | 18 | 17 | 22h |
| 20 | E | 9 | 14 | 13 | 16h |

**表3.2 实验二制品目录**

├──第一次评审

├──C\_对A组的需求评审表单\_ver1.0.0.docx

├──C\_对B组的需求评审表单\_ver1.0.0.docx

├──C\_对D组的评审意见反馈\_ver1.0.0.docx

└──C\_对E组的评审意见反馈\_ver1.0.0.docx

├──第二次评审

├──C\_对D组的需求评审表单\_ver1.0.0.docx

├──C\_对E组的需求评审表单\_ver1.0.0.docx

├──C\_对A组的评审意见反馈\_ver1.0.0.docx

└──C\_对B组的评审意见反馈\_ver1.0.0.docx

└──C\_Flask\_4.2\_需求评审表单设计\_ver1.0.1.docx

## 3.3 数据分析与说明

软件需求评审共进行了两个阶段，第一阶段是我们组对A和B组进行评审，接收D和E组对我们的评审。对A组提出了17个文档的问题，主要集中在文档格式不规范以及软件需求描述不清楚的问题，对B组提出了13个文档的问题，主要是表述方式口语化，缺乏专业性的问题。接收到了D和E组提出的32个问题，修改了30个，解释了2个。

第二阶段对D和E组进行评审，接收A和B组对我们的评审。向D组提出了11个文档上面的问题，主要是软件需求描述不清楚和文档格式问题，向E组提出了9个文档上面的问题，主要是语病和文档结构的问题。接收到了A和B组提出的23个问题，修改了19个，解释了4个，这些问题主要集中在格式和用语上面。

## 3.4 小结与有效方法建议

软件需求评审是整个实验的一个重要部分，本学期的课程中进行了两轮的组间网上互评审，在课上还接受了老师和其他同学的评审。本组的软件需求规格说明书经过同学和老师的评审，暴露出了很多的不足，帮助了我们进一步完善文档以及软件制品。

关于需求评审的有效方法，在软件需求评审及复评审过程中，建议组间增强沟通与协调，很多时候的分歧产生于相互之间缺乏沟通。在本次需求评审及复评审过程中，暴露出的一个问题就是组间相互不了解，导致部分的评审意见难以得到被评审组的认可。若组间沟通渠道通畅，很多问题在撰写评审意见之间就能得到解决，评审意见的被接收率也会有较大幅度的提升。

# 4 实验三 软件设计与实现小结

## 4.1 项目说明

本项目旨在通过自动化部署深度模型，降低算法研究人员实际部署模型的门槛，提高深度学习模型迭代研发和部署的效率。考虑到语言和部署效率的因素，项目将采用基于Python语言的轻量级Web框架flask和容器技术作为实际的模型部署工具。

## 4.2 数据统计

**表3.1**

|  |  |  |  |
| --- | --- | --- | --- |
| 制品 | 文字 | 图表 | 代码行数 |
| 《软件详细设计说明书》 | 12890 | 27 |  |
| 产品代码 |  |  | 4650 |

**表3.2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 制品 | 文字工时 | 图表工时 | 代码工时 | 学习成本 |
| 《软件详细设计说明书》 | 14h | 3.5 |  |  |
| 产品代码 |  |  | 120h | 39h |

实验3：软件设计与实现

├──C\_Flask\_4.11\_软件详细设计说明书\_ver1.0.0.docx

└──C\_Flask\_4.11\_软件详细设计说明书\_ver1.1.0.docx

## 4.3 数据分析与说明

在实验三的制品方面，小组主要产出了《软件详细设计说明书》和软件制品。表3.1是小组实验三的整体工作量统计，数据来源于小组各个成员提交的完成文档字数、图标个数以及编码数量。表3.2是完成制品的工时统计，数据来源于小组各个成员提交的工时报告单。实验三所涉及的《软件详细设计说明书》共迭代了两个版本。

实验三的主要目标是完成软件制品的设计和开发，从数据统计中可以看出，小组主要的工作量集中在学习成本和软件开发这两块。由于小组开发成员对Flask等开发工具是第一次接触，所以各个成员都有一部分的工时花费在学习开发工具上面。此外，小组各个成员主要的工作量仍是集中在软件制品开发上面，从工时中也可以看出软件开发占了较大部分的工作量，这与实验三的目标是匹配的。

实验三的《软件详细设计说明书》共产生了两个版本。第一个版本是在动手开发之前，经过对软件制品的整体架构设计和功能模块划分，由各个开发组员分工完成的，主要的目的是体现出各个模块的功能以及约定接口。第二个版本是在软件开发完成后，由每个开发组员对自己负责的部分进行调整，主要是将开发过程中做的修改体现在文档中。

## 4.4 小结与建议

在软件设计与实现实验中，小组完成了第一个版本可用的软件制品，实现了需求文档中的各个功能需求，能够支持自动化部署深度学习模型。小组成员在开发工程中对软件开发的一些基本流程和一些基本工具都有了更深的理解。

一些不足主要是组内成员对开发工具不熟悉，导致学习使用和开发任务集中在了一周之内，造成时间上比较紧张。建议学习工作可以提前开展，以免对软件制品的开发造成阻碍。

# 5 实验四 软件测试需求小结

## 5.1 软件测试需求说明

本轮测试需求分析过程以软件需求规格说明书（SRS）及软件源代码为输入，分析软件测试需求，结合软件工程综合实验具体要求，输出测试需求规格说明书。主要进行单元测试、功能需求测试、性能需求测试、接口需求测试以及软件质量特征测试。

## 5.2 数据统计

如下表所示统计软件测试过程中涉及到的测试需求和报告、测试用例、测试数据等相关信息。

**表 5.1测试需求制品数据统计**

|  |  |  |  |
| --- | --- | --- | --- |
| 制品 | 字数 | 图表 | 工时 |
| 《软件测试需求规格说明书》 | 20570 | 50 | 18.5h |
| 《软件测试报告》 | 27934 | 44 | 5h |

**表5.2测试需求数据统计**

|  |  |  |
| --- | --- | --- |
| 测试类型 | 测试用例 | 覆盖率 |
| 单元测试 | 17 | 100% |
| 功能需求测试 | 15 | 100% |
| 性能需求测试 | 6 | 100% |
| 接口需求测试 | 2 | 100% |
| 软件质量特征测试 | 5 | 100% |

**表5.3实验四制品目录表**

├──单元测试报告

├──部署系统测试报告.html

└──门户系统测试报告.html

├──测试样例及结果

├──测试样例及结果

├──[cpkt\_model.zip](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/%E6%B5%8B%E8%AF%95%E6%A0%B7%E4%BE%8B%E5%8F%8A%E7%BB%93%E6%9E%9C/MNIST%E6%A8%A1%E5%9E%8B%E6%96%87%E4%BB%B6/cpkt_model.zip)

├──[mnist\_model.h5](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/%E6%B5%8B%E8%AF%95%E6%A0%B7%E4%BE%8B%E5%8F%8A%E7%BB%93%E6%9E%9C/MNIST%E6%A8%A1%E5%9E%8B%E6%96%87%E4%BB%B6/cpkt_model.zip)

├──[mnist\_model.pb](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/%E6%B5%8B%E8%AF%95%E6%A0%B7%E4%BE%8B%E5%8F%8A%E7%BB%93%E6%9E%9C/MNIST%E6%A8%A1%E5%9E%8B%E6%96%87%E4%BB%B6/cpkt_model.zip)

└──[torch\_model.zip](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/%E6%B5%8B%E8%AF%95%E6%A0%B7%E4%BE%8B%E5%8F%8A%E7%BB%93%E6%9E%9C/MNIST%E6%A8%A1%E5%9E%8B%E6%96%87%E4%BB%B6/cpkt_model.zip)

├──测试需求规格说明书历史版本

├──C\_Flask\_5.20\_测试需求规格说明书\_ver1.3.0.docx

├──[C\_Flask\_5.20\_软件测试报告\_ver1.1.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/C_Flask_5.20_%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E6%8A%A5%E5%91%8A_ver1.1.0.docx)

├──[C\_Flask\_5.20\_软件缺陷修复报告\_ver1.1.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/C_Flask_5.20_%E8%BD%AF%E4%BB%B6%E7%BC%BA%E9%99%B7%E4%BF%AE%E5%A4%8D%E6%8A%A5%E5%91%8A_ver1.1.0.docx)

├──[C\_Flask\_5.24\_回归测试报告\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/C_Flask_5.24_%E5%9B%9E%E5%BD%92%E6%B5%8B%E8%AF%95%E6%8A%A5%E5%91%8A_ver1.0.0.docx)

├──[C\_Flask\_5.24\_软件快速使用指南.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/C_Flask_5.24_%E8%BD%AF%E4%BB%B6%E5%BF%AB%E9%80%9F%E4%BD%BF%E7%94%A8%E6%8C%87%E5%8D%97.docx)

└──[C\_Flask\_5.28\_测试需求规格说明书\_ver1.4.0.do](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C4%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/C_Flask_5.28_%E6%B5%8B%E8%AF%95%E9%9C%80%E6%B1%82%E8%A7%84%E6%A0%BC%E8%AF%B4%E6%98%8E%E4%B9%A6_ver1.4.0.docx)cx

## 5.3 数据分析与说明

表5.1来自于对实验四制品的统计以及小组成员的工时统计。表5.2来自于对测试需求说明书的归纳统计。表5.3是实验四制品的清单。

实验四主要是依照软件需求规格说明书，从多个方面设计测试需求。从工作量和工时统计中可以看出，小组在实验四中的主要工作集中在设计测试需求规格说明书，此外还对实验三的软件制品，依照测试需求说明书进行了测试，产生了软件测试报告。此外，为了测试的严谨和完整，小组还完成了缺陷修复报告、回归测试报告和软件使用指南。小组各个成员的工作量和工时较为平均。

测试需求规格说明书迭代7个版本，共46个测试用例，分为了单元测试、功能需求测试、性能需求测试、接口需求测试以及软件质量特征测试5部分，各部分的测试覆盖率均为100%。软件测试报告是按照测试需求规格说明书开展测试工作后形成的报告，共迭代了3个版本，软件制品经过修改后测试用例全部通过。

## 5.4 小结及有效方法建议

软件测试时软件工程中至关重要的一步，没有经过完整测试流程的软件制品是不可靠的，是无法投入实际使用的。经过老师课上的讲解，小组从单元测试、功能需求测试、性能需求测试、接口需求测试以及软件质量特征测试5个方面分别设计测试用例，每个部分都确保测试覆盖性达到100%。经过实际测试后，发现了软件制品的缺陷，经过修改后可以通过测试。

有效方法建议方面，对软件的测试也要分主次，也要按优先级排列测试任务。首先要保证软件的功能是完好的，接着是性能测试，接口测试以及质量测试。但仍要保证的是测试工作要完整，上述的五个方面都是必须的。

# 6 实验五 软件测试评审小结

## 6.1 软件测试评审说明

软件测试互评审是对软件测试可靠性的第三方验证。我们小组对F，G，H和I的软件制品进行了测试并对测试需求规格说明书进行了评审，同时接受了这四组对我们测试工作的评审。

## 6.2 数据统计

**表5.1 软件测试评审数据统计**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 检查项数量 | 组号 | 向被评审组提出的问题数 | 接收到的问题数 | 接受并修改的问题数 | 评审报告字数 | 工时 |
| 16 | H | 29 | 40 | 36 | 2873 | 21h |
| 16 | I | 18 | 22 | 22 | 670 | 18h |
| 16 | G | 29 | 19 | 18 | 2718 | 22h |
| 16 | F | 15 | 8 | 7 | 809 | 16h |

**表5.2 实验五制品目录**

├──第一次评审

├──各成员评审

├──[C\_对F组测试评审意见反馈\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%B8%80%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E5%AF%B9F%E7%BB%84%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81%E5%8F%8D%E9%A6%88_ver1.0.0.docx)

├──[C\_对G组测试评审意见反馈\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%B8%80%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E5%AF%B9G%E7%BB%84%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81%E5%8F%8D%E9%A6%88_ver1.0.0.docx)

├──C\_汇总\_对H组的软件测试文档评审表单\_ver1.0.0.docx

├──C\_汇总\_对H组的软件问题报告\_ver1.0.0.docx

├──C\_汇总\_对I组的软件测试文档评审表单\_ver1.0.0.docx

└──C\_汇总\_对I组的软件问题报告\_ver1.0.0.docx

├──第二次评审

├──各成员评审

├──[C\_对H组文档测试评审意见反馈\_ver1.1.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%BA%8C%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E5%AF%B9H%E7%BB%84%E6%96%87%E6%A1%A3%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81%E5%8F%8D%E9%A6%88_ver1.1.0.docx)

├──[C\_对H组软件测试评审意见反馈\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%BA%8C%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E5%AF%B9H%E7%BB%84%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81%E5%8F%8D%E9%A6%88_ver1.0.0.docx)

├──[C\_对I组文档测试评审意见反馈\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%BA%8C%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E5%AF%B9I%E7%BB%84%E6%96%87%E6%A1%A3%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81%E5%8F%8D%E9%A6%88_ver1.0.0.docx)

├──[C\_对I组软件测试评审意见反馈\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%BA%8C%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E5%AF%B9I%E7%BB%84%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81%E5%8F%8D%E9%A6%88_ver1.0.0.docx)

├──C\_汇总\_对F组的软件测试文档评审表单\_ver1.0.0.docx

├──C\_汇总\_对F组的软件问题报告表单\_ver1.0.0.docx

├──C\_汇总\_对G组的软件测试文档评审表单\_ver1.0.0.docx

└──[C\_汇总\_对G组的软件问题报告\_ver1.0.0.docx](https://github.com/bhsei/20_C/blob/master/%E5%AE%9E%E9%AA%8C5%EF%BC%9A%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95%E8%AF%84%E5%AE%A1/%E7%AC%AC%E4%BA%8C%E6%AC%A1%E8%AF%84%E5%AE%A1/C_%E6%B1%87%E6%80%BB_%E5%AF%B9G%E7%BB%84%E7%9A%84%E8%BD%AF%E4%BB%B6%E9%97%AE%E9%A2%98%E6%8A%A5%E5%91%8A_ver1.0.0.docx)

├──C\_Flask\_5.18\_软件测试需求规格说明书检查单\_ver1.0.0.docx

├──C\_Flask\_5.18\_软件测试需求评审表单设计\_ver1.1.0.docx

├──C\_Flask\_5.18\_软件问题报告表单设计\_ver1.0.0.docx

└──C\_Flask\_5.18\_软件问题报告表单设计\_ver1.2.0.docx

## 6.3 数据分析

软件测试评审共进行了两个阶段，第一阶段是我们组对H和I组进行评审，接收F和G组对我们的评审。对H组提出了24个文档方面的问题，主要集中在文档格式不规范以及测试用例描述不清楚的问题，对H组提出了5个软件问题，主要是前端方面的显示问题。对I组提出了18个文档方面的问题，主要是测试用例描述模糊以及少量的格式问题，I组的软件测试未发现问题。接收到了F和G组提出的27个问题，修改了25个，解释了2个。

第二阶段对F和G组进行评审，接收H和I组对我们的评审。向F组提出了13个文档上面的问题，主要是测试用例描述不清楚和文档格式问题，向F组提出了2个软件问题。向G组提出了23个文档上面的问题，主要是测试用例不完善的问题，向G组提出了6个软件问题，包括了软件可靠性和易用性的问题。接收到了H和I组提出的62个问题，接收了58个，这些问题主要集中在格式和用语上面。

## 6.4 小结及有效方法建议

软件测试评审是整个实验的一个重要部分，本学期的课程中进行了两轮的互评审，在课上还接受了老师和其他同学的评审。我们本以为完善的文档和软件制品经过同学和老师的评审和测试，暴露出了很多的不足，帮助了我们进一步完善文档以及软件制品。

有效方法建议。在软件测试评审及复评审过程中，建议前期多多了解前两组同学的项目。在本次测试评审及复评审过程中，确实感觉到前期缺乏了解造成测试评审过程中的无法深入，只能透过表面文字，在其他小组协助下才能顺利完成软件测试评审工作。

# 7 实验六 项目进度计划与控制小结

## 7.1 项目进度计划与控制说明

本次实验六分为进度计划与进度控制两个部分，进度计划部分在撰写项目计划书的时候同步进行，进度计划的目的在于初步划定项目各个阶段，并初步确认各个组员的任务分配情况，以便明确项目开发的整体思路。在项目开发的进程中，由组员使用开源软件Gantt Project进行项目进度控制，初始项目计划被导入Gantt Project的管理文件中，并随着每周的实验进行更新和调整。

## 7.2 数据统计

**表 7.1项目计划与控制制品数据统计**

|  |  |  |  |
| --- | --- | --- | --- |
| 制品 | 字数 | 图表 | 记录条数 |
| 《项目进度安排》 | 351 | 0 | 49 |
| 《项目进度燃尽图》 | 211 | 2 | - |
| 《进度计划与控制报告》 | 3060 | 3 | - |

**表7.2实验六制品目录表**

├──C\_Flask\_6.11\_项目进度安排\_ver1.3.0.gan

├──C\_Flask\_6.11\_项目进度燃尽图\_ver1.1.0

└──C\_Flask\_6.11\_进度计划与控制报告\_ver1.1.0

## 7.3 数据分析与说明

项目计划与控制环节贯穿了软件工程综合实验的全过程，其产物包括《项目进度安排》、《项目进度燃尽图》、《进度计划与控制报告》。其中，《项目进度安排》指的是进度管理软件生成的Gantt Project文件，该文件详细记录了本次软件工程综合实验的49个子任务的任务名称、起止时间节点、人员分配情况、任务之间的依赖关系。《项目进度燃尽图》包含了两个子图，它们分别是项目的剩余任务数量燃尽图和剩余累计工时燃尽图，项目的剩余任务数量燃尽图描述了项目的实际剩余任务数量与基线剩余任务数量的对比情况，项目的剩余累计工时燃尽图描述了项目的实际剩余累计工时与基线剩余累计工时的对比情况。

在以上两个文档的基础上，负责进度控制的组员撰写了《进度计划与控制报告》，该报告长度为3060字，描述了项目进度计划与控制的过程，分析本学期项目计划的制定，进度控制的流程，并在此基础上总结项目的执行情况，阐述优点与不足。

## 7.4 小结及有效方法建议

回顾本学期软件工程综合实验的项目进展情况，可以看出全学期的项目进度与预期相近，小幅度的进度波动处在可控范围之内，能够通过临时增加工作强度来弥补缺失的进度。总体而言，本项目能够按照预期计划顺利完成。结合实验八的任务工作量统计数据，可认为进度计划环节的任务分配合理。项目成员每个人的工时较为平均，与进度计划时的工作量意愿相符。

分析进度控制过程取得较好成效的原因，首先是项目计划的制定策略和依据较为完善。项目规划阶段充分地分析软件工程综合实验的任务要求， 在此基础上结合项目开发经验，各位组员共同商讨，得出了合理的进度计划。其次，项目进行过程中的计划调整能及时跟进。在每周任务进行的过程中，负责进度跟踪的同学能够依据实际情况，动态调整任务计划，及时将当前进度通知各位组员，提示进度缓慢的组员提高效率。每周五的进度确认和每周六的组内会议确保了整体小组对当前进度的了解，组内会议通过群策群力的方式改进本组的项目规划过程。

此外，也应当看到，本组在项目计划过程中，部分进度的计划不够充分，导致工作量峰值的出现。例如本次实验开发阶段的实际任务强度大于预期，使得开发人员在此段时间的任务压力偏大。未来需要在任务规划阶段更加充分地考虑可能的影响因素，结合实际开发经验判断工作量，并适当高估工作量，以留下缓冲的空间，避免临时出现的意外情况影响项目工期。

# 8 实验七 软件配置管理小结

## 8.1 配置管理

在软件开发过程中，从项目的建立到项目最终完善，其版本变更是必须的，而在变更过程中如果没有一套统一的变更控制标准，那么对于团队开发项目进行过程中，很容易造成混乱。因此，软件配置管理(Configuration Management，CM)需要贯穿于整个软件工程实验，在软件研发过程中，采用有效且统一的方法进行软件变更控制和版本管理。

在软件研发过程中，每周对当周在GitHub上的提交数据进行整理，分别以提交的实验制品、所属的实验单元、提交的成员进行统计。最终进行数据的汇总与分析。

## 8.2 数据统计

**表 8.1 实验七数据统计**

|  |  |  |
| --- | --- | --- |
| 实验七：配置管理 | | |
| 版本更新次数 | 配置管理报告累计字数 | 累计工时 |
| 4 | 12341 | 13.1工时 |

### 8.2.1 提交次数随天数的变化情况

卡通人物

描述已自动生成

**图 8.1 提交次数随天数的变化情况**

从图中可以看出， commit提交占据了三、四、五月份，提交次数较多的周分别是04/05~04/12（实验2软件需求评审），05/17~05/24（实验5软件测试评审），对于需求评审和测试评审的两周里，提交的次数比较多，符合实际情况。

### 8.2.2 一周内的提交变化情况

图片包含 文字, 地图, 游戏机

描述已自动生成

**图 8.2 一周内的提交变化情况**

### 8.2.3 各制品的提交次数

手机屏幕截图

描述已自动生成

**图 8.3 各制品的提交次数**

其中commit次数较多的制品包括需求规格说明书、软件问题报告表单、测试需求规格说明书、测试需求评审单等。

## 8.3 版本控制方法

### 8.3.1 Git版本标识

本组的版本标识共分为两类：

1. 提交备注

对于每个单独的成员commit时的附加备注，格式为m.d\_姓名\_提交概况\_具体内容描述

例如3.22\_许京爽\_配置管理计划书\_初稿完成

其中m.d为日期，姓名是提交人员的姓名，提交内容概况为子项目名称，具体内容描述为此次进行了什么修改或者变更。如果是非单人合作的项目提交，需要在姓名处填写所有参与人员的姓名。

(2) 提交文档

对于该软件过程中所有的所有文档对象，其规范命名方式如下：

C\_Flask\_m.d\_X\_verx.y.z.ext

例如：C\_Flask\_3.22\_配置管理计划书\_ver0.1.0.docx

因为该组为软件工程实验C组，所以以C开头，项目基于Flask框架，第二个关键词为Flask，m.d表示完成该文档的时间，m为月，d为日，以今天为例就是3.22，X为该文档的全称，比如配置管理计划书，verx.y.z为该文档版本号，其中ver表示version版本，x为主版本号，y为次版本号，z为修订号，均为非负整数，且禁止在数字前方补零，

主版本号更新规则：必须在有任何不兼容的修改被加入时递增。每当主版本号递增时，次版本号和修订号必须归零。

次版本号更新规则：必须在有向下兼容的新功能出现时递增。也可以在内部程序有大量新功能或改进被加入时递增。每当次版本号递增时，修订号必须归零。

修订号更新规则：必须在只做了向下兼容的修正时才递增。这里的修正指的是针对不正确结果而进行的内部修改。

主版本号为零（0.y.z）的软件处于开发初始阶段，此时一切都可能随时被改变。而当主版本号为1（1.0.0）时，一般代表一个文档的初稿完成版本。

每次修改时，需要更新版本号，保留原版本文档。

### 8.3.2 优点

1. 较为准确的行为记录

在所有的commit过程中，我们小组都按照约定进行备注，因此也能够每周对于提交的内容、动作和所属实验进行统计。

1. 较为完善的项目描述

我们小组的GitHub主页中的目录导览直截了当的对于每个实验制品所属的文件进行说明，对于需要进行查找的实验内容快捷方便。

1. 统计精确

采用人工统计的方式，在一些可能存在疑虑的地方，比如重复提交、项目描述不准确的时候，可以直接与提交的成员进行沟通，确保每一条统计的数据精确无误。

### 8.3.3 缺点分析

1. 提交与文件不能完全对应

我们在提交文档的时候，会发现文档的变更记录与提交的记录数据不是完全一致，那是因为对于某一个文档的修改，比如需求规格说明书，我们组员之间彼此对于存在的问题进行了更正，可能会累积多次变更后，才进行commit，因此会出现变更记录比实际提交数据多的情况。

## 8.4 小结与建议

在配置管理期间，由于组员均对Git工具和Github网站有较长的使用时间，因此在实验中，没有发生文档覆盖、无法解决冲突的情况。

采用了规范化的注释使得提交显得较为清晰，并且在提交期间，大多数提交均能够按照约定提交，能够在后期的数据收集与统计。

# 9 实验八 工作量统计小结

## 9.1 工作量统计说明

本实验通过对实验项目的跟踪和记录，统计出每个成员的工作量和情况，分析成员工作量差异及其原因。为后续的任务分配、调整、控制和最终的确定成员贡献度提供支持。

## 9.2 数据统计

**表 9.1 实验八数据统计**

|  |  |  |  |
| --- | --- | --- | --- |
| 迭代版本数： | 统计分析项数： | 报告字数： | 累计工时 |
| 5 | 16 | 9252 | 25.3 |

## 9.3 数据分析

实验八的工作量统计对实验项目从实验一到实验八的实验过程中，小组成员所产生的工作记录以及制品数据等做了较为详尽的收集，分析和总结，得到了各个实验的工作量数据以及各个实验小组各成员的工作量情况，最后，根据最先设计好的工作量评定计算方式，汇总得出总的项目的工作量数据和小组各成员的工作量占比情况。

在实验八的统计过程中，我们组采用的方式是每周对于当周的实验制品数据进行收集，分别收集工时和具体的数值。

每周根据当前的实验进度，按照属于当前周需要进行收集的数据项制作表格，分发给每位成员，在每次课后进行数据回收，再进行汇总统计。

每次数据有一名同学负责收集，没有采用在线表格的方式，这样每位成员只能填写自己的实际数据值和自己的实际工时，不会出现因为看了别人填写的数据，而更改自己的数据的情况。可以说明每次的数据都是比较真实的，也更加具有分析意义。

## 9.4 小结及有效方法建议

工作量统计是软件项目开发中的重要一环，能够帮助项目团队主管了解当前的软件开发情况，最重要的是能够决定着项目内各成员的贡献率分配和项目的总工时估计，这对调节控制软件开发成本，以及之后的类似项目报价而言非常重要。

有效方法建议：工作量和贡献率之间关系不确定。需要注意的是，虽然实验八采用的数据是真实而且可靠的，数据能在一定程度上反映各成员的贡献率，但是由于小组各成员的分工不同，任务和人的特性有所区别。所以在计算最后的贡献率时，小组需要达成共识，给出一致认可的计算权重，从而得到和工作量一直匹配的贡献率占比。

# 10 综合实验总结

本学期的软件工程综合实验持续时间为15周。经过一个学期的付出，本组成员齐心协力，完成了软件工程综合实验的任务。软件需求分析过程与需求评审过程相互补充，帮助我们学会如何评估软件的预期需求，如何撰写合适的需求描述文档。软件改进过程锻炼了组员之间的协作能力，增强了各位组员的规范性意识。软件测试与测试评审过程教会了我们正确设计恰当的软件测试用例，以及保障软件需求得到满足。进度控制、配置管理、工作量统计与分析这三个环节贯穿于整个实验，合理的项目计划对项目的有序顺利完成提供了很大的帮助，配置管理实验让我们小组的协助更方便和科学，可以分析我们整个项目期间的修改情况。工作量统计与分析环节记录了各位组员在这个项目中的贡献程度，为贡献率的评判提供了依据。总体而言，本组完成了软件工程综合实验的课程要求，各个实验阶段均帮助我们更好地学会了工程化开发的方法。

# 参考文献

1. 邹欣. 构建之法[M]. 北京.人民邮电出版社,2015:264-296
2. 李大友. 软件工程方法[M]. 机械工业出版社, 1996.