

软 件 工 程 实 验

**基于EOS平台的小众点评**

**软件项目计划书**

**编写人：20TeamD**

**编制时间：2020年3月17日**

变更记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 |
| 0.1 | 2020.3.16 | 刘新宇 | 添加项目概述 |
| 0.2 | 2020.3.16 | 叶明林 | 添加项目团队组织 |
| 0.3 | 2020.3.16 | 刘思远 | 添加项目简介 |
| 0.4 | 2020.3.17 | 许栋亮、黄雨鹤 | 添加项目进度计划 |
| 0.5 | 2020.3.17 | 陈瀚清 | 添加实施计划 |

**目 录**

[1. 引言 1](#_Toc35333153)

[1.1 背景介绍 1](#_Toc35333154)

[1.2 定义 2](#_Toc35333155)

[1.3 参考资料 2](#_Toc35333156)

[1.4 标准 2](#_Toc35333157)

[2. 项目概述 3](#_Toc35333158)

[2.1 项目目标 3](#_Toc35333159)

[2.2 工作内容 3](#_Toc35333160)

[2.3 应交付成果 4](#_Toc35333161)

[2.4 项目验收方式和依据 4](#_Toc35333162)

[3. 项目团队组织 5](#_Toc35333163)

[3.1 组织结构 5](#_Toc35333164)

[3.2 人员组成 5](#_Toc35333165)

[3.3 协作与沟通 5](#_Toc35333166)

[4. 实施计划 6](#_Toc35333167)

[4.1 工作任务分解与分析 6](#_Toc35333168)

[4.2 总体进度计划 7](#_Toc35333169)

[4.3关键问题 9](#_Toc35333170)

# 1. 引言

## 1.1 背景介绍

近年来，世界对比特币的态度起起落落，但作为比特币底层技术之一的区块链技术日益受到重视。在比特币形成过程中，区块是一个一个的存储单元，记录了一定时间内各个区块节点全部的交流信息。各个区块之间通过随机[散列](https://baike.baidu.com/item/%E6%95%A3%E5%88%97/9939480)(也称哈希算法)实现链接，后一个区块包含前一个区块的哈希值，随着信息交流的扩大，一个区块与一个区块相继接续，形成的结果就叫区块链。

2019年1月10日，国家互联网信息办公室发布《区块链信息服务管理规定》。2019年10月24日，在中央政治局第十八次集体学习时，习近平总书记强调，“把区块链作为核心技术自主创新的重要突破口”“加快推动区块链技术和产业创新发展”。“区块链”已走进大众视野，成为社会的关注焦点。

区块链技术是一种去中心化的分布式数据存储技术。而区块链技术则会涉及智能合约。智能合约是一种特殊协议，旨在提供、验证及执行合约。具体来说，智能合约是区块链被称之为“去中心化的”重要原因，它允许我们在不需要第三方的情况下，执行可追溯、不可逆转和安全的交易。在区块链上实现只能合约，可以解决对程序的信任问题。

目前流行的区块链智能合约平台有以太坊、EOSIO平台等。

EOSIO是一个由其创始团队block.one开发的一个平台软件，它构建一个区块链底层的技术架构，类似区块链中的操作系统，使得更多的开发者能够基于EOSIO平台快速方便的构建分布式应用软件(dapp)。EOS就像PC中的Windows，手机中的安卓和iOS，是区块链中的基础平台，其有非常大的扩展性，能够提供账户身份认证，数据库，异步通信等程序调用和并行计算，是一个以商用为目的的区块链平台，具有低延迟和免费交易费用，未来有望实现百万级tps（每秒处理交易数）。

## 1.2 定义

关于EOSIO与后续开发中的一些专业术语、缩略语如下表1所示。

**表 1 专业术语及缩略语**

|  |  |
| --- | --- |
| **术语/缩略语** | **描述** |
| DPOS | 股权证明算法，EOSIO中使用的共识算法。 |
| API | 一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码。 |

## 1.3 参考资料

1. EOSIO官方文档<https://developers.eos.io/welcome/latest/overview/index>
2. EOSIO源代码<https://github.com/EOSIO/eos>

## 1.4 标准

参照如下标准：

* GB/T 13702-1992 计算机软件分类与代码
* GB/T 19003-2008 软件工程
* GB/T 5538-1995 软件工程标准分类法
* GB/T 9385-2008 计算机软件需求规格说明
* GB/T 5532-2008 计算机软件测试规范
* GB/T 18221-2000 信息技术程序设计语言
* GB/T 8567-2006 计算机软件文档编制规范

# 项目概述

## 2.1 项目目标

基于EOS平台开发一款类似大众点评的店铺评价/打分应用——小众点评。与现有大众点评应用不同的是，由于我们是基于EOS平台开发，因此小众点评是去中心化的，这一特性使得该应用可以最大化保证其评论的真实性、准确性。因为去中心化的特性可以避免商家与平台暗箱操作删除差评，也可以避免恶意水军刷好评、差评，因为区块链具有信息公开透明、全程留痕、不可伪造的特性。

## 2.2 工作内容

1.学习EOS平台提供的开发人员文档，了解平台基本原理，熟悉DApp开发流程。

2.调研当前大众点评应用的优缺点，分析其背后的深层原因，并在此基础上进行小众点评的需求分析。

3.利用EOS平台提供的SDK进行开发，开发过程中要注意组员之间的协作，以及版本迭代控制，并对开发过程进行记录。

4.开发完成后，结合需求分析文档对应用进行测试，注意测试覆盖的广度和深度，编纂测试报告并对测试出现问题的部分进行分析、改进及二次测试。

5.根据课程的整体安排和每个实验的具体要求，依照项目时间计划完成其他各类文档和报告。

## 2.3 应交付成果

项目应交付的产品如表2所示。

**表 2 应交付成果列表**

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **名称** | **交付时间** | **备注** |
| No.1 | 软件项目计划 | 2020/3/18 | 暂无 |
| No.2 | 软件需求规格说明书 | 待定 | 暂无 |
| No.3 | 软件需求分析报告 | 待定 | 暂无 |
| No.4 | 软件评审报告 | 待定 | 暂无 |
| No.5 | 软件问题报告 | 待定 | 暂无 |
| No.6 | 软件设计模型及文档 | 待定 | 暂无 |
| No.7 | 软件产品 | 待定 | 暂无 |
| No.8 | 被测软件执行结果分析报告 | 待定 | 暂无 |
| No.9 | 软件问题报告 | 待定 | 暂无 |
| No.10 | 工作日志 | 待定 | 暂无 |
| No.11 | 进度控制分析报告 | 待定 | 暂无 |
| No.12 | 变更与管理分析报告 | 待定 | 暂无 |
| No.13 | 实验分析报告 | 待定 | 暂无 |
| No.14 | 实验小组总结 | 待定 | 暂无 |
| No.15 | 个人总结 | 待定 | 共7份 |

## 2.4 项目验收方式和依据

1. 验收方式

项目验收方式采用组内自行测试并编写报告并在课堂进行组间评审，评审合格后验收成功。

1. 验收依据

验收依据“应交付成果”清单以及评审结果进行验收。

# 项目团队组织

## 3.1 组织结构

本项目团队的组成为一名组长和六名组员。组长与组员沟通协调，统筹规划，给组员分配任务；组员则根据组长的安排，及时完成任务。

## 3.2 人员组成

人员组成如表3所示。

**表 3 人员组成**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **学号** | **姓名** | **角色** |
| 1 | SY1906710 | 许栋亮 | 组长 |
| 2 | BY1906003 | 陈瀚清 | 组员 |
| 3 | BY1906025 | 林讯 | 组员 |
| 4 | SY1906119 | 刘新宇 | 组员 |
| 5 | SY1906511 | 叶明林 | 组员 |
| 6 | SY1906512 | 刘思远 | 组员 |
| 7 | SY1906708 | 黄雨鹤 | 组员 |

## 3.3 协作与沟通

由于疫情的影响，小组成员无法面对面交流，所以主要采取线上交流的方式进行沟通。平时主要在微信群中进行及时沟通，每周的重要会议主要通过微信群、腾讯会议等方式进行。

组内成员使用腾讯文档共同编辑相关的Word、Excel和PPT文件；使用北航云盘分享保存参考资料及项目文档；使用Git对项目的代码和文档进行版本管理，托管平台则使用课程指定的Github；使用Teambition进行项目进度管理、日程安排及工作量统计。

# 4. 实施计划

## 4.1 工作任务分解与分析

### 4.1.1熟悉EOS平台

根据现有EOS平台的相关教程，尽可能地熟悉如何基于EOS平台进行开发，熟悉如何编写EOS应用的测试用例，熟练掌握如何分别利用EOS平台提供的工具和代码库来运行程序和进行前端开发。大体了解EOS平台的运行机制。

### 4.1.2需求分析与评审

根据EOS平台所能提供的功能以及已有系统的不足，分析并确定项目的需求，分析需求优先级，编写软件需求规格说明书。团队将进行需求评审，编写软件需求相关的软件评审报告。

### 4.1.3软件设计与实现

根据软件需求规格说明书进行软件的概要设计和详细设计，设计模型并编写必要的设计文档。完成基于EOS平台的应用代码编写，生成可供展示的软件产品。

### 4.1.4软件测试与评审

定义测试需求并设计测试用例后，团队对需测试需求和测试用例进行评审，编写软件测试相关的软件评审报告。利用EOS平台提供的工具，编码测试用例，对开发出的软件产品其进行测试，对软件执行结果进行分析，编写软件执行结果分析报告，和软件问题报告。

### 4.1.5软件项目计划与监控

整个项目的开发过程中使用Teambition工具对项目任务分解、分配与调整，同时进行进度监控。在监控的同时，编写工作日志和进度控制分析报告。

### 4.1.6软件配置管理

整个项目的开发过程中制定配置管理计划，使用github工具进行软件变更控制和版本管理，同时进行配置管理分析，编写变更与管理分析报告。

### 4.1.7软件工程实验追踪与分析

计划时刻跟踪整个项目的开发过程，使用工作量估计方法和统计分析方法分析各项软件工程过程的执行情况和结果，对团队每个人任务工作量差异及其影响因素进行分析，编写实验分析报告。

### 4.1.8其他

项目的开发过程中，如果遇到困难或疑问，团队将及时通过微信或腾讯会议进行沟通讨论。讨论结束后，编写会议记录。

## 4.2 总体进度计划

|  |  |  |  |
| --- | --- | --- | --- |
| 任务名称 | 计划工期 | 计划开始时间 | 计划完成时间 |
| **项目确定以及介绍** | **3 个工作日** | **2020年3月8日** | **2020年3月11日** |
| 项目调研 | 8 工时 | 2020年3月8日 | 2020年3月8日 |
| 选题会议 | 21 工时 | 2020年3月10日 | 2020年3月10日 |
| PPT制作 | 3 工时 | 2020年3月11日 | 2020年3月11日 |
| **项目计划书** | **3 个工作日** | **2020年3月16日** | **2020年3月18日** |
| 项目计划会议 | 21 工时 | 2020年3月16日 | 2020年3月16日 |
| 项目简介 | 3 工时 | 2020年3月16日 | 2020年3月16日 |
| 项目概述 | 3 工时 | 2020年3月16日 | 2020年3月16日 |
| 项目团队组织 | 3 工时 | 2020年3月17日 | 2020年3月17日 |
| 实施计划 | 3 工时 | 2020年3月17日 | 2020年3月17日 |
| 计划书修改与整合 | 14 工时 | 2020年3月17日 | 2020年3月17日 |
| teambition学习与使用 | 14 工时 | 2020年3月18日 | 2020年3月18日 |
| **软件需求分析** | **7 个工作日** | **2020年3月19日** | **2020年3月26日** |
| 需求分析会议 | 21 工时 | 2020年3月19日 | 2020年3月19日 |
| 软件模块分析 | 4 个工作日 | 2020年3月19日 | 2020年3月21日 |
| 测试需求分析 | 4 个工作日 | 2020年3月22日 | 2020年3月23日 |
| 软工规范分析 | 2 个工作日 | 2020年3月24日 | 2020年3月24日 |
| 扩展功能分析 | 4 个工作日 | 2020年3月25日 | 2020年3月26日 |
| 需求分析会议 | 21 工时 | 2020年3月26日 | 2020年3月26日 |
| **软件需求评审** | **9 个工作日** | **2020年3月28日** | **2020年4月5日** |
| 需求评审会议 | 21 工时 | 2020年3月28日 | 2020年3月28日 |
| 修改软件模块需求 | 4 个工作日 | 2020年3月28日 | 2020年3月29日 |
| 修改测试需求 | 4 个工作日 | 2020年3月30日 | 2020年3月31日 |
| 修改软工规范需求 | 4 个工作日 | 2020年4月1日 | 2020年4月2日 |
| 修改扩展功能需求 | 4 个工作日 | 2020年4月3日 | 2020年4月4日 |
| 需求评审会议 | 21 工时 | 2020年4月5日 | 2020年4月5日 |
| **软件需求复评审** | **4 个工作日** | **2020年4月6日** | **2020年4月9日** |
| 需求复审会议 | 21 工时 | 2020年4月6日 | 2020年4月6日 |
| 确定软件模块需求 | 2 个工作日 | 2020年4月6日 | 2020年4月6日 |
| 确定测试需求 | 2 个工作日 | 2020年4月7日 | 2020年4月7日 |
| 确定软工规范需求 | 2 个工作日 | 2020年4月8日 | 2020年4月8日 |
| 确定扩展功能需求 | 2 个工作日 | 2020年4月9日 | 2020年4月9日 |
| 需求复审会议 | 21 工时 | 2020年4月9日 | 2020年4月9日 |
| **项目进度展示(一)** | **30个工作日** | **2020年3月8日** | **2020年4月16日** |
| 工作日志 | 30 个工作日 | 2020年 3月8日 | 2020年 4月16日 |
| 进度控制分析报告 | 30 个工作日 | 2020年 3月8日 | 2020年 4月16日 |
| 项目部署 | 30 个工作日 | 2020年 3月8日 | 2020年 4月16日 |
| 实验分析报告 | 30 个工作日 | 2020年 3月8日 | 2020年 4月16日 |
| 工作量估计 | 30 个工作日 | 2020年 3月8日 | 2020年 4月16日 |
| **软件设计与实现** | **17 个工作日** | **2020年4月18日** | **2020年5月7日** |
| 软**件**设计会议 | 21 工时 | 2020年4月18日 | 2020年4月18日 |
| 软件设计 | 9 个工作日 | 2020年4月18日 | 2020年4月20日 |
| 实现分工会议 | 21 工时 | 2020年4月21日 | 2020年4月21日 |
| 软件实现（一） | 21个工作日 | 2020年4月22日 | 2020年4月28日 |
| 问题讨论会议 | 21 工时 | 2020年4月29日 | 2020年4月29日 |
| 软件实现（二） | 21个工作日 | 2020年4月30日 | 2020年5月6日 |
| 软件展示 | 21 工时 | 2020年5月7日 | 2020年5月7日 |
| **软件测试分析** | **6 个工作日** | **2020年5月9日** | **2020年5月14日** |
| 软件测试会议 | 21 工时 | 2020年5月9日 | 2020年5月9日 |
| 测试需求分析 | 10 个工作日 | 2020年5月10日 | 2020年5月14日 |
| **软件测试评审** | **5个工作日** | **2020年5月16日** | **2020年5月21日** |
| 测试评审会议 | 21 工时 | 2020年5月16日 | 2020年5月16日 |
| 修改测试需求 | 8 个工作日 | 2020年5月17日 | 2020年5月21日 |
| **软件测试复评审** | **5 个工作日** | **2020年5月23日** | **2020年5月28日** |
| 测试复评审会议 | 21 工时 | 2020年5月23日 | 2020年5月23日 |
| 修改测试需求 | 10 个工作日 | 2020年5月24日 | 2020年5月28日 |
| **软件演示与测评** | **5 个工作日** | **2020年5月30日** | **2020年6月4日** |
| 软件演示会议 | 21 工时 | 2020年5月30日 | 2020年5月30日 |
| 软件演示与测评 | 10个工作日 | 2020年5月31日 | 2020年6月4日 |
| **项目进度展示(二)** | **38 个工作日** | **2020年4月17日** | **2020年6月11日** |
| 工作日志 | 38 个工作日 | 2020年 4月17日 | 2020年 6月11日 |
| 进度控制分析报告 | 38 个工作日 | 2020年 4月17日 | 2020年 6月11日 |
| 项目部署 | 38 个工作日 | 2020年 4月17日 | 2020年 6月11日 |
| 实验分析报告 | 38 个工作日 | 2020年 4月17日 | 2020年 6月11日 |
| 工作量估计 | 38 个工作日 | 2020年 4月17日 | 2020年 6月11日 |
| **实验分析与总结** | **6个工作日** | **2020年6月13日** | **2020年6月18日** |
| 会议 | 21 工时 | 2020年6月13日 | 2020年6月13日 |
| 综合实验分析 | 4 个工作日 | 2020年6月13日 | 2020年6月18日 |
| 综合实验总结 | 4 个工作日 | 2020年6月13日 | 2020年6月18日 |
| **每周组会** | **12 个工作日** | **2020年3月16日** | **2020年6月18日** |

## 4.3关键问题

本计划的风险因素如表4.1所示。经过分析，影响本计划完成的关键问题有：

1.需求定义不明确，导致项目开发的结果与预想不同

2.对EOS平台陌生或开发经验欠缺导致项目质量难以保证

3.团队缺少沟通导致项目各模块集成存在问题

**表4.1 项目风险因素**

|  |  |  |  |
| --- | --- | --- | --- |
| **风险排序** | **风险项名称** | **风险描述** | **风险解决方案** |
| 1 | 需求定义不明确 | 在项目需求分析阶段对需求描述不清，开发人员的理解与需求描述者的想法间存在差异 | 在整个项目的开发阶段，一旦发现需求描述不明确的地方，通过沟通达成一致 |
| 2 | 对EOS平台不熟悉 | 开发人员多数对EOS平台的开发和测试不熟悉，给项目开发带来困难 | 对EOS平台使用方式进行培训 |
| 3 | 经验欠缺 | 开发人员使用C++进行应用开发的经验不足 | 结对编程或培训 |
| 4 | 团队沟通不足 | 开发人员之间由于缺少沟通，导致各自的想法不同，给软件集成带来困难 | 有问题及时通过微信沟通，代码编写规范事先商量 |

# 5. 组内互评

## 5.1 评分标准

如公式4.1.2所示, 为了让组内成员在平时落实好工作，小组打算每个实验结束后评一次分，其中客观分为f1 , 主观分为f2 。每个实验满分为10分，8个实验满分一共80分。

考虑到每周评分的片面性，组内决定在8个实验结束后，增加一个组内主观评分。计为T2 总分10分，相应公式如4.1.1所示。

关于f1的计算如公式4.1.3所示，根据成员工时计算，为了让f1 控制在10分，将每个成员的工时/最大工时\*10.

组内互评得分采用以下方法：去掉一个最高分、一个最低分，然后再取平均分。

