

软件项目计划

Cealgull
(简化版)

版本 <1.1>

修订历史记录

日期	版本	说明	作者
19/06/2023	1.0	初始版本	吴逸洋
05/07/2023	1.1	界面原型迭代后更新	徐轲

目录

- 1.简介
 - 1.1目的
 - 1.2范围
 - 1.3定义、首字母缩写词和缩略语
 - 1.4参考资料
- 2.项目概述
 - 2.1项目的目的、规模和目标
 - 2.2假设与约束
 - 2.3项目的可交付成果

3.项目组织

4.项目计划

4.1风险分析

4.2方法和工具

4.3开发计划

4.4质量保证计划

4.5项目沟通计划

4.6培训计划

5.附录

软件项目计划

1. 简介

本文档**Cealgull**的软件项目计划文档，本文档旨在为**Cealgull**的开发分析项目风险，提供项目开发的目的，项目开发的具体时间，以及一些其他的计划开发规约。

目的

本软件项目计划旨在为开发人员提供项目时间表，软件的基本功能，软件栈的介绍，软件项目约束，软件技术的集中培训时间等有关该软件项目开发的具体事宜。

范围

该软件项目计划针对**Cealgull**项目提出，并且针对其开发人员给出了有关的项目规约以及项目风险界定。

定义、首字母缩写词和缩略语

1. 区块链 —— 借由**密码学**与**共识机制**等技术建立与储存庞大交易资料区块串链的**点对点**网路系统。
2. 去中心化 —— **去中心化**（英语：decentralization）是**互联网**发展过程中形成的**社会关系**形态和内容产生形态，是相对于“中心化”而言的新型网络内容生产过程。
3. 非具名化 —— 将用户信息与实际操作的ID分离仅用户保留自己的ID，分布节点无法了解用户的信息。

4. web3 —— **Web3**（也被称为**Web 3.0**，又写为**web3**是关于**万维网**发展的一个概念，主要与基于**区块链**的去中心化、**加密货币**以及**非同质化代币**有关。与区块链有关的web3概念是由**以太坊**联合创始人**Gavin Wood**于2014年提出，并于2021年受到**加密货币**爱好者、大型科技公司和**创业投资公司**的关注。

2. 项目概述

项目的目的、规模和目标

本项目旨在利用现有的web3的一系列新技术，来构建校内的非具名论坛，即在做到在基本的Jaccount或邮箱鉴权后实现隐私保护，活动无痕，用户自治，知识共享的一个主要平台，以保护客户的通信自由以及言论自由以及信息所有权。任何自然都可以下载服务器节点服务来对服务进行host，并在服务器本地保存一份channel账本进行归档，已做到在服务器崩溃时能够自动恢复。**本项目的最终开发目标通过区块链中的联盟链技术、p2p存储协议、通讯加密来校内学生提供搭建于去中心化存储之上的允许交易的、端到端加密的去中心化论坛、社交网络以及即时通讯应用。**

假设与约束

1. 项目预算 ---- 2000 RMB
2. 开发人员 ---- 四人
3. 设备 ---- 手机端
4. 时间表 --- 软件1.0版本需要于7月14号交付

项目的可交付成果

1. 后端的可部署的chaincode代码库
2. 前端的代码库
3. APP安装包
4. Cealgull peer后端部署文档

3. 项目组织

- 吴逸洋（组长） --- 负责区块链搭建，智能合约创建，分布式存储，以及服务部署。
- 徐轲 --- 负责chaincode维护，负责路由和服务维护
- 王浩丞 --- 负责前端界面设计以及移动端应用开发

- 任柏俊 -- 负责前端界面以及移动端应用开发

4. 项目计划

风险分析

优先级	风险名称	风险描述	风险缓解措施
高	需求风险	产品与实际需求不符合 <div><div>1.</div><div>界面和用户实际操作直觉不符合</div></div> <div><div>2.</div><div>无法形成可靠的安全性保障</div></div> <div><div>3.</div><div>功能实现与产品理念不符合</div></div>	进行需求调研 以及需求分析
高	技术风险	技术经验无法支撑产品开发 <div><div>1.</div><div>学习华为云CI/CD工具</div></div> <div><div>2.</div><div>学习前端App开发</div></div> <div><div>3.</div><div>学习邮箱验证鉴权</div></div> <div><div>4.</div><div>学习web3的基本理念</div></div> <div><div>5.</div><div>学习IOS/Android应用开发</div></div> <div><div>6.</div><div>学习Hyperledger Fabric智能合约的编写</div></div> <div><div>7.</div><div>学习IPFS协议</div></div> <div><div>8.</div><div>学习lokinet洋葱路由解析</div></div> <div><div>9.</div><div>学习hyperledger的节点部署</div></div> <div><div>10.</div><div>学习kubernetes的部署</div></div> <div><div>11.</div><div>学习orbitdb,couchdb</div></div>	进行技术培训 进行有关的技术攻坚
中	进度风险	产品开发无法赶上交付进度 <div><div>1.</div><div>项目开发滞后</div></div> <div><div>2.</div><div>返工风险</div></div>	指定详细迭代计划 完善Sprint任务板

方法和工具

- 开发方法 --- RUP 与 Sprint开发法
- 建模工具 --- Draw.io/Power Designer
- IDE --- vscode
- 测试工具 --- ghz/JMeter/locust
- 版本管理工具 --- git
- 项目管理工具 --- Github ci/cd

开发计划

迭代名称	起止时间	所缓解的风险	需完成的任务	需提交的成果
界面技术迭代	2023.06.19 - 2023.06.30	需求风险 技术风险	<ol style="list-style-type: none">1. 需求分析2. 制定迭代计划3. app界面搭建4. 构建基本论坛以及时间线界面5. 构建基本个人主页页面6. 完成校内邮箱验证7. 基本完成后端私有链的构建8. 代码评审 代码的自动化的 ci/cd Linter检查9. 界面测试10. 功能正确性测试	<ol style="list-style-type: none">1. APP的安装包原型2. 需求规约文档3. 架构部署文档4. Docker镜像原型
架构技术迭代	2023.07.01 - 2023.07.14	技术风险 需求风险	<ol style="list-style-type: none">1. 需求分析2. 制订迭代计划3. 制订架构文档4. 构建基本电子钱包管理功能5. 基本构建后端智能合约逻辑6. 完成论坛基本时间线、评论以及收发的基本逻辑	<ol style="list-style-type: none">1. 软件架构文档2. App安装包原型3. Peer/Orderer和CA的节点部署4. 智能合约ChainCode的具体设计5. API文档

			<ul style="list-style-type: none"> 7. 集成式CA以及Orderer架构设计 8. 针对智能合约以及CA签发的单元测试 9. 前端界面细化并进一步接入API 10. 正确性测试以及功能测试 11. 华为云部署 	
功能整合迭代	2023.07.15 - 2023.09.09	进度风险 需求风险 技术风险	<ul style="list-style-type: none"> 1. 需求分析 2. 分布式配置智能合约服务以及CA和Order 3. 完成社交网络的基本时间线及最新构建 4. 完成论坛板块的合约搭建 5. 完成后端区块链事件构建 6. 完善单元测试 7. 完善性能测试以及部署 8. 华为云部署以及外部VPS联动统一部署 9. 完善Chaincode智能合约服务器与其他CA服务器的部署 	<ul style="list-style-type: none"> 1. APP完整安装包 2. Peer/CA/Order Docker镜像 3. ChainCode源代码以及部署镜像 4. ChainCode部署服务 5. 部署文档 6. 软件架构文档 7. 软件使用说明书

质量保证计划

需求评审

- 每次需求更新时应于当前的技术栈匹配，评审需求是否无误，是否存在二义性，是否能够予以实现以及满足。

代码评审

- 每次的代码提交需对所涉及的具体模块，影响范围以及代码的具体作用进行论述。
- 代码评审时应当对照代码规范使用对应的linter来对代码进行静态检查
- 代码评审时需要通过ci/cd所设计的单元测试以及集成测试

单元测试

- 单元测试需要在每一个模块进行提交后展开
- 对于每一个模块的开发者 提交代码时都需要提交其模块所对应的单元测试代码
- 单元测试需要完成语句覆盖率要求，并提供测试报告。

集成测试

- 集成测试需要在一个子系统整体或者业务逻辑整体设计完毕后展开，需要利用事件流测试以及其他的黑盒测试的方法对整个子系统进行正确性测试。
- 集成测试也应当包含随机测试以及对子系统的压力测试，来测试子系统本身的承载能力

系统测试

- 系统测试则是在版本即将开发完毕后进行
- 应当统一使用自动化测试对系统的每一个接口进行并发测试来测定软件的稳定性。
- 系统测试应当软件交付前全部完成

项目沟通计划

- 每日例会安排 每日早上9点

培训计划

培训内容	预定日期
Git以及CI/CD使用	2023.06.21
移动端app技术选型	2023.06.23
Hyperledger智能合约编写与后端grpc交流	2023.06.25
前后端交互逻辑API交流	2023.06.28

5. 附录