<面向工业应用的柔性可扩展的边缘计算软件系统>

软件开发计划

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2021/10/18 | 1.0 | 软件开发计划初稿 | 芮召普 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[软件开发计划 5](#_Toc25423)

[1 简介 5](#_Toc29189)

[1.1 目的 5](#_Toc22914)

[1.2 范围 5](#_Toc16606)

[1.3 定义、首字母缩写词和缩略语 5](#_Toc26428)

[1.4 参考资料 5](#_Toc31997)

[1.5 概述 5](#_Toc1690)

[2 项目概述 6](#_Toc6354)

[2.1 项目的目的、规模和目标 6](#_Toc26092)

[2.2 假设与约束 6](#_Toc139)

[2.3 项目的可交付工件 6](#_Toc17244)

[3 项目组织 7](#_Toc14736)

[3.1 组织结构 7](#_Toc30448)

[3.2 对外联系 7](#_Toc2824)

[4 管理流程 7](#_Toc9608)

[4.1 项目估计 7](#_Toc24417)

[4.2 项目计划 7](#_Toc11452)

[4.2.1 阶段计划 7](#_Toc20266)

[4.2.2 迭代目标 9](#_Toc5837)

[4.2.3 发布版 9](#_Toc1996)

[4.2.4 项目时间表 9](#_Toc7945)

[4.2.5 项目资源分配 10](#_Toc5336)

[4.3 项目监测与控制 10](#_Toc21064)

[4.3.1 进度控制计划 10](#_Toc2316)

[4.3.2 预算控制计划 10](#_Toc29815)

[4.3.3 质量控制计划 11](#_Toc11325)

[4.3.4 报告计划 11](#_Toc23368)

[4.3.5 度量计划 11](#_Toc15810)

[4.4 收尾计划 11](#_Toc12151)

[5 技术流程计划 11](#_Toc5916)

[5.1 方法、工具和技巧 11](#_Toc20820)

[5.2 基础设施计划 11](#_Toc6376)

[5.3 产品验收计划 11](#_Toc17000)

[6 支持流程计划 12](#_Toc23651)

[6.1 评估计划 12](#_Toc17775)

[6.2 文档计划 12](#_Toc13830)

[6.3 质量保证计划 12](#_Toc12032)

[6.4 问题解决计划 12](#_Toc14850)

软件开发计划

# 简介

## 目的

为了保证项目组团队能够按时保质的完成项目目标，便于项目团队成员更好的了解项目情况，使项目工作展开的各个过程合理有序，有必要以文件化的形式，把对于在项目生命周期内工作的任务范围、各项工作的任务分解、项目团队的组织结构、各团队成员的工作分工、开发进度、风险对策等内容做出的安排，作为项目团队成员以及项目干系人之间的共识与约定、项目生命周期内的所有项目活动的行动基础、项目团队开展和检查项目工作的依据。

## 范围

本文档的内容覆盖了面向工业应用的柔性可扩展的边缘计算软件系统的开发活动以及参与这些活动的开发人员，并且可作为迭代计划的参考。

## 定义、首字母缩写词和缩略语

|  |  |  |
| --- | --- | --- |
| 定义 | 缩略语 | 含义 |
| 边缘计算软件系统 | ECSS | 在靠近数据源（设备端）采用网络、计算、存储、应用核心能力为一体的开发平台，就近提供有限的低延时服务，并接入云平台进行管理和计算的软件系统 |
| 业务流程管理标记 | BPMN | 一套图形化的流程建模标准，用于以业务流程模型详细说明各种业务流程 |
| 协议集成系统 | PIS | 将各类协议或规则集成到相互关联的、统一和协调的系统之中，使资源达到充分共享，实现集中、高效、便利的管理 |
| 工业异构设备 | IHD | 工业生产场景下应用不同通信协议、使用不同数据解析方法的高度异构化设备 |

## 参考资料

参考资料主要包括以下文档:

1）《面向工业应用的柔性可扩展的边缘计算软件系统-立项建议书\_v1.0》

2）《面向工业应用的柔性可扩展的边缘计算软件系统-软件需求规约文档\_v1.0》

3）《面向工业应用的柔性可扩展的边缘计算软件系统-软件架构文档\_v1.0》

4）《边缘计算参考架构3.0》白皮书

5）《软件工程原理》高等教育出版社，2013年2月第一版

## 概述

本文档后续将展开介绍此项目的项目概述、项目组织、管理流程、技术流程计划和支持流程计划。

# 项目概述

## 项目的目的、规模和目标

本项目基于“云—边—端”架构，设计并实现面向工业应用的柔性可扩展的边缘计算软件系统。通过在云端和工业生产线的设备之间加入边缘计算软件系统，快速接收并处理设备的数据上传，快速响应云端的流程下发，实施具体的指令调度，提供用户更加智能、高效的控制手段，实现用户对工业生产线更有效的管理。本项目旨在为工业应用提供柔性可扩展的生产线控制与管理服务，为异构设备提供可扩展的协议规则以接入本系统，为云端提供数据上传接口与流程下发接口以供云端完成数据接收与流程下发，实现对工业生产线的总体调配与灵活控制。系统为异构工业设备提供统一可扩展的方式接入，针对不同的生产线定义、配置、调度各异构设备完成生产过程，并结合实际生产情况对生产过程进行实时调整。系统还提供以机器学习算法为基础的扩展服务，提供数据分析服务接口。

## 假设与约束

1. 项目要求自立项起3个月内完成；
2. “云—边—端”架构的理念契合本系统的应用场景；
3. Java具有优秀的生态体系。

## 项目的可交付工件

|  |  |
| --- | --- |
| 可交付工件 | 预定交付日期 |
| 迭代一计划 | 2021.10.11 |
| 软件架构文档 | 2021.10.31 |
| 软件需求规约文档 | 2021.10.31 |
| 软件开发计划 | 2021.10.31 |
| 迭代一评估报告 | 2021.10.31 |
| 迭代二计划 | 2021.11.01 |
| 迭代二评估报告 | 2021.11.21 |
| 迭代三计划 | 2021.11.22 |
| 迭代三评估报告 | 2021.12.12 |
| 迭代四计划 | 2021.12.13 |
| 迭代四评估报告 | 2022.01.02 |
| 迭代五计划 | 2022.01.03 |
| 迭代五评估报告 | 2022.01.09 |
| 测试计划 | 2022.01.09 |
| 测试报告 | 2022.01.09 |
| 用户手册 | 2022.01.09 |
| 项目总结报告 | 2022.01.09 |
| 系统源代码 | 2022.01.09 |

# 项目组织

## 组织结构

本项目研发团队由董彦君担任项目经理，由张俸铭、江嘉晋和芮召普担任项目的开发及测试人员；本项目由SJTU董事会定期执行项目跟踪和管理，由蔡鸿明老师和沈备军老师担任董事会成员，由蔡鸿明老师担任项目总监，接收项目组的工作汇报，以确保项目按时交付。

## 对外联系

无。

# 管理流程

## 项目估计

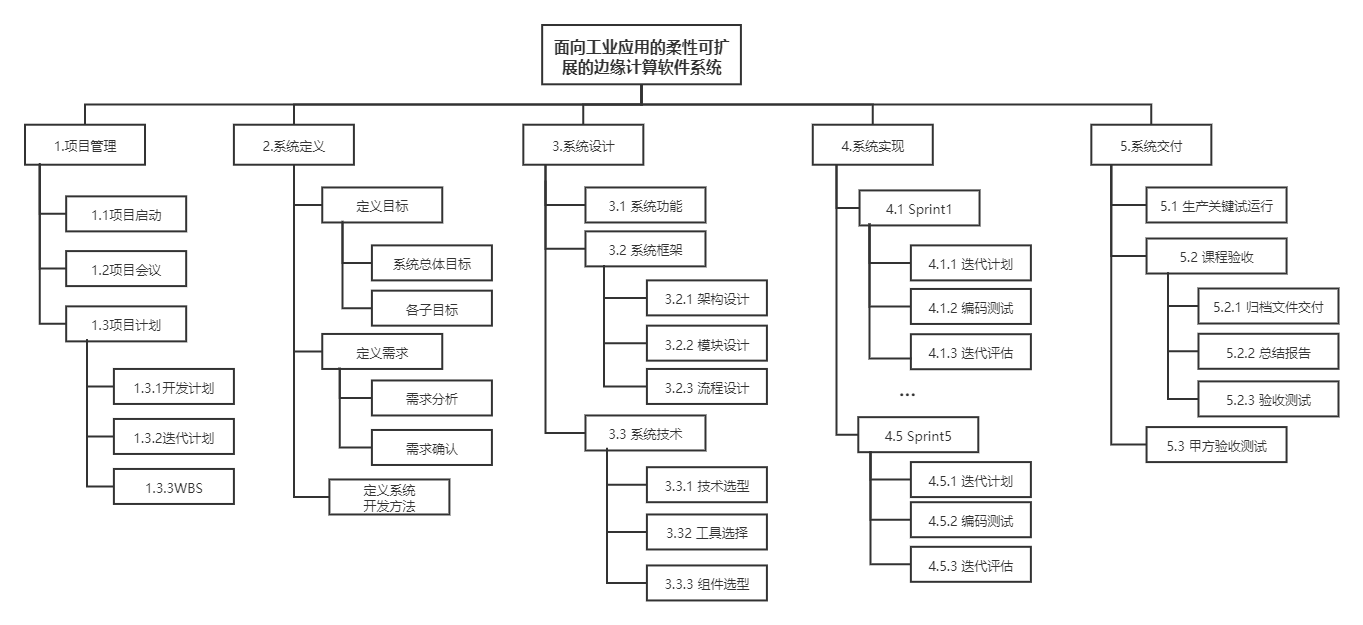
项目成本主要为人力时间成本，开发过程中不存在额外的项目费用。项目组成员都属于IST实验室成员，开发过程中拟采用开源的组件、技术框架以及免费的开发工具，使用实验室的服务器作为计算资源。

项目进度计划从2020年10月11日开始至2021年1月9日结束，共计13周、5个迭代。项目的开发内容以工业应用生产线为背景，重点设计“云—边—端”架构，为工业生产提供数据和控制服务。项目组成员对于“云—边—端”架构的设计与具体实现存在经验不足的情况，且架构整体复杂，实现过程中有潜在的技术风险；并且对于异构数据的转换以及流程控制、流程调度等具体功能实现也存在不确定的潜在风险。在项目的开发过程中，如果出现需求变更、难以解决的问题等情况，需要对项目进行重新估计，制订新的迭代计划，并重新确定需求优先级。

## 项目计划

### 阶段计划

1. WBS（系统实现中，具体的迭代内容见第2部分）



1. 迭代计划及时间分配

|  |  |  |
| --- | --- | --- |
| 迭代 | 时间 | 内容 |
| 迭代一 | 2021年10月11日~2021年10月31日 | 1. 分析并确定项目需求，完成《软件需求规约文档》 2. 调研相关需求的技术实现方案并设计基本软件架构，进行业务流程建模，初步完成《软件架构文档》，验证技术可行性 3. 配置项目环境 4. 设计界面原型并获得用户反馈，根据反馈对界面原型进行修改 5. 技术架构的实现与验证 |
| 迭代二 | 2021年11月1日~2021年11月21日 | 1. 实现高优先级核心功能模块及其前端用户界面的开发，核心功能有设备数字建模、数据转换、数据接收与发送、数据管理与存储、数据可视化 2. 对版本1进行测试，根据用户反馈进行修复与改进 |
| 迭代三 | 2021年11月22日~2021年12月12日 | 1. 在版本1的基础上实现下一版本的核心功能模块及其前端用户界面的开发，核心功能有生产流程控制、指令接收与下发、状态可视化与监控、状态反馈与响应 2. 对版本2功能模块进行测试，根据用户反馈进行修复与改进 |
| 迭代四 | 2021年12月13日~2022年1月2日 | 1. 在版本2的基础上实现下一版本的核心功能模块和非主要功能模块及其前端用户界面的开发，核心功能有机器学习推理、数据分析、权限管理，非主要功能有日志管理、运维管理 2. 对测试版本3功能模块进行测试，根据用户反馈进行修复与改进 |
| 迭代五 | 2022年1月3日  ~2022年1月9日 | 1. 在版本3的基础上完成项目总体的功能整合以及代码评审优化，在实际环境进行部署测试和验收测试，发布可交付版本 2. 完成《项目总结报告》等相关项目文档，整理交付成果 |

### 迭代目标

|  |  |
| --- | --- |
| 迭代 | 目标 |
| 迭代一 | 1. 完成技术架构原型与界面原型的开发。 2. 完成需求与架构相关的开发文档。 |
| 迭代二 | 1. 开发并测试版本version1（设备数字建模、数据转换、数据接收与发送、数据管理与存储、数据可视化）。 2. 发布版本version1。 |
| 迭代三 | 1. 开发并测试版本version2（生产流程控制、指令接收与下发、状态可视化与监控、状态反馈与响应）。 2. 发布版本version2。 |
| 迭代四 | 1. 开发并测试版本version3（核心功能有机器学习推理、数据分析、权限管理，非主要功能有日志管理、运维管理）。 2. 发布版本version3。 |
| 迭代五 | 1. 发布可交付版本version4。 2. 软件文档交付。 |

### 发布版

|  |  |  |
| --- | --- | --- |
| 版本号 | 发布时间 | 类型 |
| 0.1 | 迭代二（2021年11月21日） | 测试版 |
| 0.2 | 迭代三（2021年12月12日） | 测试版 |
| 0.3 | 迭代四（2022年01月02日） | 测试版 |
| 1.0 | 迭代五（2022年01月09日） | 正式版 |

### 项目时间表

|  |  |
| --- | --- |
| 活动 | 预定日期 |
| 迭代一完成 | 2021年10月31日 |
| 迭代二完成 | 2021年11月21日 |
| 测试版version1发布 | 2021年11月21日 |
| 中期演示 | 2021年12月02日 |
| 迭代三完成 | 2021年12月12日 |
| 测试版version2发布 | 2021年12月12日 |
| 迭代四完成 | 2022年01月02日 |
| 测试版version3发布 | 2022年01月02日 |
| 迭代五完成 | 2022年01月09日 |
| 正式版发布 | 2022年01月09日 |
| 验收汇报 | 2022年01月09日 |

### 项目资源分配

#### 人员配备计划

项目组由4位成员组成，其中由董彦君担任项目经理。项目的开发测试团队包括董彦君、张俸铭、江嘉晋和芮召普。本系统的实现主要包含云端的前后端开发和边缘端的前后端开发，其中云端的前端开发需要1人完成，云端的后端开发需要2人完成，边缘端的前端开发需要1人完成，边缘端的后端开发需要2人完成。

在项目开展之际，项目组成员共同负责项目立项、需求调研、需求分析等工作；项目进行过程中，项目成员共同编制项目开发计划，识别项目风险，由组长董彦君总体控制项目开发过程和项目质量，并负责项目开发工作的管理和人员工作的分工以及项目资料的收集、整理、归档和保存，为项目组成员提供足够的技术支持和资源支持。

#### 资源获取计划

项目组的成员均来自上海交通大学软件学院信息系统实验室（IST）。

## 项目监测与控制

### 进度控制计划

本项目采用Scrum迭代式的增量软件开发过程，使用GitHub作为代码托管和研发协作平台，保证项目的高效进展。项目计划从2021年10月11日开始至2022年1月9日结束，共计13周、5个迭代。在每个迭代开始前组织迭代计划会，确定本轮迭代的具体工作内容，并编写《迭代计划》，并组织每日例会汇报工作进度，探讨并解决遇到的问题；在每个迭代的结束之时，组织会议对本次迭代进行评估，并编写《迭代评估报告》。

对于项目过程中可能出现的如需求变更等严重问题，应组织会议进行评估，有必要时需调整项目总体开发计划以保证项目开发过程的正常完成。

### 预算控制计划

项目组的开发人员均为上海交通大学软件学院信息系统实验室成员，不存在额外的人员成本；开发过程选用开源的工具及框架，使用的服务器资源为实验室现有资源，资源成本在可控预期范围内。

### 质量控制计划

在开发过程中，项目组通过代码审查来检查源代码与编码标准的符合性以及代码质量；通过单元测试对每个功能点进行验证和检查；通过集成测试确保模块内或前后端的正常运作；制订详细且全面的测试用例进行系统测试，根据测试结果对系统进行改进，以达到系统所需的功能需求和非功能需求；对各类相关文档进行评审、改进，直至达到质量要求。

### 报告计划

每个迭代开始时，编写迭代计划；每个迭代结束后，编写迭代评估报告；迭代五将编写测试计划、测试报告和项目总结报告。

### 度量计划

每个迭代结束之时开展度量活动，以评估项目组成员的工作量和项目进度，提高项目质量和掌握缺陷分布。度量项包括工作量、进度、问题和测试BUG。工作量以人日为单位，以完成的功能点来度量；进度以天为单位，由进度偏差来衡量，进度偏差的计算方式为 (|实际时间-计划时间|) / 计划时间；问题用于了解项目中发生的问题个数、类型、严重程度及分布，将以个数为单位记录归档；测试BUG以个数为单位，用缺陷密度来衡量，其计算方式为 (评审缺陷数+测试缺陷数) / 规模；

## 收尾计划

项目经理将开发过程中出现的经验教训总结归档，将整理好的文档分发给项目组成员，完成项目总结报告向上级领导和验收人员汇报项目总结。项目收尾后，项目组成员将继续投入IST实验室的其他工作，具体的分组情况和工作内容待定。项目开展过程中产生的文档、源代码、可执行程序、配置文件等均需要整理归档，并提交课程作业以及在实验室服务器上存档。

# 技术流程计划

## 方法、工具和技巧

项目的技术标准包括：

1）根据《UML参考手册》对系统的业务、用例和架构进行建模。

2）根据Ant Design提出的“自然且确定”的设计风格，进行用户界面设计。

3）根据《Google Java编程风格规范》和《Airbnb JavaScript风格指南》来规范编程风格。

4）根据《测试计划》对测试过程进行指导。

5）根据RUP的文档模板来编写各种文档。

## 基础设施计划

信息系统搭建的基础设施由IST实验室提供。

## 产品验收计划

在课程结束之际，将由项目组成员和沈备军老师进行产品验收。

# 支持流程计划

## 评估计划

每个迭代的进行过程中，对代码质量、文档内容和系统功能进行评估。评估原则遵循独立性原则、客观性原则、科学性原则、替代性原则和预期性原则。评估方法包括通过代码审查评估代码质量，文档复审评估文档质量，通过功能、进度的检查来评估项目的质量。

## 文档计划

项目主要产生的文档有《立项建议书》、《软件需求规约文档》、《软件架构文档》、《软件开发计划》、《迭代计划》、《迭代评估报告》、《测试计划》、《测试报告》、《用户手册》、《项目总结报告》。

项目开始之前，编写《立项建议书》，并根据修改意见确定《立项建议书》最终版；

每个迭代的开始和结束时，以《立项建议书》为参考，编写《迭代计划》和《迭代评估报告》；

迭代一完成《软件架构文档》、《软件开发计划》、《需求规约文档》，根据修改意见在后续迭代中修改完善文档内容；

迭代五编写《测试计划》、《用户手册》和《项目总结报告》，根据测试结果编写《测试报告》。

## 质量保证计划

项目组计划使用Scrum迭代式的增量软件开发方法，使用GitHub 作为研发协作平台，保证项目执行的质量。开发过程中，项目组计划通过代码审查来保证源代码的质量；通过单元测试保证各功能点的质量，通过集成测试保证各模块的质量，通过系统测试保证功能需求和非功能需求的质量，通过文档复审保证文档的质量。

## 问题解决计划

对于开发过程中出现的问题，由项目组成员在日常例会上提出并记录，对问题的难易程度和优先等级进行分类，从而制定对应的解决方案；对于需求变更或其他重要问题，将组织会议与指导教师进行探讨，记录问题并确定解决方案，对相关内容进行更改。