面向工业应用的柔性可扩展的边缘计算软件系统

测试计划

版本 1.0

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 12日/1月/2022年 | 0.1 | 完成测试计划 | 芮召普 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 背景 4

1.3 范围 4

1.4 项目标识 5

2. 测试需求 6

3. 测试策略 7

3.1 测试类型 7

3.1.1 数据和数据库完整性测试 7

3.1.2 功能测试 7

3.1.3 业务周期测试 7

3.1.4 用户界面测试 8

3.1.5 性能评价 8

3.1.6 负载测试 9

3.1.7 强度测试 9

3.1.8 安全性和访问控制测试 10

3.1.9 配置测试 10

3.1.10 工具 12

4. 资源 13

4.1 角色 13

4.2 系统 15

5. 项目里程碑 16

可交付工件 17

5.1 测试模型 17

5.2 测试日志 17

5.3 缺陷报告 17

附录 A：项目任务 18

测试计划

# 简介

## 目的

面向工业应用的柔性可扩展的边缘计算软件系统的这一“测试计划”文档有助于实现以下目标：

• 确定现有项目的信息和应测试的软件构件。

• 列出推荐的测试需求。

• 推荐可采用的测试策略，并对这些策略加以说明。

• 确定所需的资源，并对测试的工作量进行估计。

• 列出测试项目的可交付元素。

## 背景

本项目以智能制造生产流程为背景，为了解决边缘计算与生产业务紧密绑定所带来的生产流程编排不灵活、异构设备接入困难等问题，本项目计划以提高边缘计算系统可扩展性、实现更智能的生产管理和更柔性的生产流程为目标，设计并研发面向工业应用的柔性可扩展的边缘计算软件系统，为工业制造企业提供高效、实时、便捷的生产流程控制与设备管理服务，并在边缘端提供数据API供上层应用接入，实现云端的以边缘端为基础的多级开发与应用。

本项目使用基于UML的面向对象方法作为软件开发方法。项目基于Spring Boot框架进行后端的开发，使用Vue.js框架开发前端，使用Python训练和执行机器学习模型。

本项目主要功能包括三部分，云端方面有对生产流程、对边缘端、对机器学习模型的管理；边缘端方面有边缘控制服务、数据转换、对设备的动态管理、机器学习推理；设备端对异构设备的多种协议与边缘控制服务连接。除了这三部分外还会兼顾易用性、可靠性、性能、可支持性、可兼容性等非功能特性，详细参考《软件需求规约文档》。

## 范围

单元测试：对各个模块的源代码的进行测试，保证各模块基本功能都能够正确地实现。

集成测试：将模块进行组合测试，保证所有功能都能够正确地实现。

系统测试：根据《需求规格说明书》对软件进行功能测试，对重点的模块进行性能测试，并结合可能的用户测试。

功能性测试：按照需求定义中的功能定义部分对系统实行的系统级别的测试。

非功能性测试：按照系统需求定义中的非功能定义部分（如系统的性能指标、可靠性指标等）对系统实行的系统级别的测试。

测试对象中将接收测试的功能包括服务的定义与部署，工艺流程的定义、解析与执行，消息通知、状态监控等 。

测试对象中将接收测试的非功能特性包括易用性、性能、可支持性等。

编写此文档的过程中假设测试环境有稳定的网络连接，且测试人员具备基本的软件测试技能。

可能会影响测试设计、开发或实施的约束、风险和意外事件：测试人员不具备专业的技能，在测试过程中可能出现疏漏和不严谨之处；由于设备和资源的约束，有些非功能性需求的测试不够精准；有些原计划需要进行测试的功能可能随着开发的进行被删去，这些功能将不再需要测试。

## 项目标识

下表列出了制定*测试计划*所用的文档，并标明了文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接受或已经过复审** | **作者或来源** | **备注** |
| 立项建议书 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 董彦君，张俸铭，芮召普，江嘉晋 |  |
| 软件需求规约 | 🗹 是 🞎 否 | 🗹 是 🞏 否 | 张俸铭，董彦君，芮召普，江嘉晋 |  |
| 软件开发计划 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 芮召普 |  |
| 软件架构文档 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 董彦君 |  |
| 原型 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | 董彦君，张俸铭，芮召普，江嘉晋 |  |

# 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

|  |  |
| --- | --- |
| 功能性需求 | 边缘端设备数据管理 |
| 边缘端设备管理 |
| 边缘端状态管理 |
| 云端接收边缘端数据 |
| 云端边缘管理 |
| 云端工艺流程配置 |
| 云端工艺流程下发 |
| 边缘端工艺流程接收与设备绑定 |
| 边缘端流程执行 |
| 边缘端设备状态可视化 |
| 云端机器学习模型管理 |
| 云端机器学习模型训练 |
| 云端机器学习模型下发 |
| 边缘端机器学习模型管理 |
| 边缘端机器学习模型绑定设备 |
| 边缘端机器学习模型执行 |
| 边缘端机器学习模型反馈 |
| 云端用户管理 |
| 非功能性需求 | 性能 |
| 可靠性 |
| 易用性 |
| 可测试性 |
| 可扩展性 |
| 可维护性 |
| 可兼容性 |

# 测试策略

## 测试类型

### 数据和数据库完整性测试

|  |  |
| --- | --- |
| 测试目标： | 确保数据库访问方法和进程正常运行，数据不会遭到损坏。 |
| 方法： | • 调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据或对数据的请求。  • 检查数据库，确保数据已按预期的方式填充，并且所有数据库事件都按正常方式出现；或者检查所返回的数据，确保为正当的理由检索到了正确的数据。 |
| 完成标准： | 所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。 |
| 需考虑的特殊事项： | • 进程应该以手工方式调用。  • 应使用小型或最小的数据库（其中的记录数很有限）来使所有无法接受的事件具有更大的可见性。 |

### 功能测试

|  |  |
| --- | --- |
| 测试目标： | 确保测试对象的功能正常，其中包括微服务的定义与部署，工艺流程的定义、解析与执行，消息通知、状态监控等功能。 |
| 方法： | 利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。 |
| 完成标准： | • 所计划的测试已全部执行。  • 所发现的缺陷已全部解决。 |
| 需考虑的特殊事项： | 无 |

### 业务周期测试

|  |  |
| --- | --- |
| 测试目标 | 确保测试对象及后台进程都按照所要求的业务模型和时间表正确运行。 |
| 方法： | 通过执行以下活动，测试将模拟若干个业务周期：  • 将修改或增强对测试对象进行的功能测试，以增加每项功能的执行次数，从而在指定的时段内模拟若干个不同的用户。  • 将使用有效的和无效的日期或时段来执行所有与时间或日期相关的功能。  • 将在适当的时候执行或启动所有周期性出现的功能。  • 在测试中还将使用有效的和无效的数据，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。 |
| 完成标准： | • 所计划的测试已全部执行。  • 所发现的缺陷已全部解决。 |
| 需考虑的特殊事项： | • 系统日期和事件可能需要特殊的支持活动。  • 需要通过业务模型来确定相应的测试需求和测试过程。 |

### 用户界面测试

|  |  |
| --- | --- |
| 测试目标 | 核实以下内容：  • 通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法（Tab 健、鼠标移动和快捷键）的使用。  • 窗口的对象和特征（例如：菜单、大小、位置、状态和中心）都符合标准。 |
| 方法： | 为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。 |
| 完成标准： | 证实各个窗口都与基准版本保持一致，或符合可接受标准。 |
| 需考虑的特殊事项： | 并不是所有定制或第三方对象的特征都可访问。 |

### 性能评价

|  |  |
| --- | --- |
| 测试目标： | 核实所指定的事务或业务功能在以下情况下的性能行为：  • 正常的预期工作量。  • 预期的最繁重工作量。 |
| 方法： | • 使用为功能或业务周期测试制定的测试过程。  • 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项事务的迭代次数。  • 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为基准），并在多台客户机（虚拟的或实际的客户机，请参见下面的“需考虑的特殊事项”）上重复。 |
| 完成标准： | • 单个事务或单个用户：在每个事务所预期或要求的时间范围内成功地完成测试脚本，没有发生任何故障。  • 多个事务或多个用户：在可接受的时间范围内成功地完成测试脚本，没有发生任何故障。 |
| 需考虑的特殊事项： | 综合的性能测试还包括在服务器上添加后台工作量。  可采用多种方法来执行此操作，其中包括：  • 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客户机。 此负载可通过“远程终端仿真”(Remote Terminal Emulation) 工具来实现。 此技术还可用于在网络中加载“流量”。  • 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载。  性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。 |

### 负载测试

|  |  |
| --- | --- |
| 测试目标： | 核实所指定的事务在不同的工作量条件下的性能行为时间。 |
| 方法： | • 使用为功能或业务周期测试制定的测试。  • 通过修改数据文件来增加事务数量，或通过修改测试来增加每项事务发生的次数。 |
| 完成标准： | 多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有发生任何故障。 |
| 需考虑的特殊事项： | • 负载测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  • 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。 |

### 强度测试

|  |  |
| --- | --- |
| 测试目标： | 核实测试对象能够在以下强度条件下正常运行，不会出现任何错误：  • 服务器上几乎没有或根本没有可用的内存（RAM 和 DASD）。  • 连接或模拟了最大实际数量的客户机。  • 多个用户对相同的数据/账户执行相同的事务。  • 最繁重的事务量或最差的事务组合（请参见上面的“性能测试”）。  注： 强度测试的目标还可表述为确定和记录那些使系统无法继续正常运行的情况或条件。  客户机的强度测试在“配置测试”的第 3.1.9 节中进行了说明。 |
| 方法： | • 使用为性能评价或负载测试制定的测试。  • 要对有限的资源进行测试，就应该在一台计算机上运行测试，而且应该减少或限制服务器上的 RAM 和 DASD。  • 对于其他强度测试，应该使用多台客户机来运行相同的测试或互补的测试，以产生最繁重的事务量或最差的事务组合。 |
| 完成标准： | 所计划的测试已全部执行，并且在达到或超出指定的系统限制时没有出现任何软件故障，或者导致系统出现故障的条件并不在指定的条件范围之内。 |
| 需考虑的特殊事项： | • 如果要增加网络工作强度，可能会需要使用网络工具来给网络加载消息或信息包。  • 应该暂时减少用于系统的 DASD，以限制数据库可用空间的增 长。  • 使多个客户机对相同的记录或数据账户同时进行的访问达到同步。 |

### 安全性和访问控制测试

|  |  |
| --- | --- |
| 测试目标： | * 应用程序级别的安全性：核实角色只能访问其所属用户类型已被授权使用的那些功能或数据。 * 系统级别的安全性：核实只有具备系统和应用程序访问权限的角色才能访问系统和应用程序。 |
| 方法： | * 应用程序级别的安全性：确定并列出各用户角色及其被授权使用的功能或数据。   • 为各用户角色创建测试，并通过创建各用户角色所特有的事务来核实其权限。  • 修改用户角色并为相同的用户重新运行测试。对于每种用户角色，确保正确地提供或拒绝了这些附加的功能或数据。   * 系统级别的访问：请参见下面的“需考虑的特殊事项” |
| 完成标准： | 各种已知的角色类型都可访问相应的功能或数据，而且所有事务都按照预期的方式运行，并在先前的应用程序功能测试中运行了所有的事务。 |
| 测试目标： | * 应用程序级别的安全性：核实角色只能访问其所属用户类型已被授权使用的那些功能或数据。 * 系统级别的安全性：核实只有具备系统和应用程序访问权限的角色才能访问系统和应用程序。 |

### 配置测试

|  |  |
| --- | --- |
| 测试目标： | 核实测试对象可在要求的硬件和软件配置中正常运行。 |
| 方法： | • 使用功能测试脚本。  • 在测试过程中或在测试开始之前，打开各种与非测试对象相关的软件（例如 Microsoft 应用程序：Excel 和Word），然后将其关闭。  • 执行所选的事务，以模拟主角与测试对象软件和非测试对象软件之间的交互。  • 重复上述步骤，尽量减少客户机工作站上的常规可用内存。 |
| 完成标准： | 对于测试对象软件和非测试对象软件的各种组合，所有事务都成功完成，没有出现任何故障。 |
| 需考虑的特殊事项： | 无 |

### 工具

此项目将使用以下工具：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 厂商/自行研制 | 版本 |
| 用于功能性测试的 ASQ 工具 | JMeter | Apache | 5.4 |
| 用于性能测试的 ASQ 工具 | JMeter | Apache | 5.4 |
| 项目管理 | Jira | Atlassian | 8.13 |
| DBMS 工具 | MySQL | Oracle | 8.0.27 |

# 资源

## 角色

下表列出了在此项目的人员配备方面所作的各种假定。

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 角色 | 推荐的最少资源  （所分配的专职角色数量） | 具体职责或注释 |
| 测试经理，  测试项目经理 | 1 | 进行管理监督。  职责：   * 提供技术指导 * 获取适当的资源 * 提供管理报告 |
| 测试设计员 | 1 | 确定测试用例、确定测试用例的优先级并实施测试用例。  职责：   * 生成测试计划 * 生成测试模型 * 评估测试工作的有效性 |
| 测试员 | 1 | 执行测试。  职责：   * 执行测试 * 记录结果 * 从错误中恢复 * 记录变更请求 |
| 测试系统管理员 | 1 | 确保测试环境和资产得到管理和维护。  职责：   * 管理测试系统 * 授予和管理角色对测试系统的访问权 |
| 数据库管理员 | 1 | 确保测试数据（数据库）环境和资产得到管理和维护。  职责：   * 管理测试数据（数据库） |
| 设计员 | 1 | 确定并定义测试类的操作、属性和关联。  职责：   * 确定并定义测试类 * 确定并定义测试包 |
| 实施员 | 1 | 实施测试类和测试包，并对它们进行单元测试。  职责：   * 创建在测试模型中实施的测试类和测试包 |

## 系统

下表列出了测试项目所需的系统资源。

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 | mysql://localhost:3306/ecss |
| —网络或子网 | 192.168.0.1 |
| —服务器名服务器名 | 无 |
| —数据库名 | Ecss |
| 客户端测试 PC | Window10, MacOS |
| —包括特殊的配置需求 | 无 |
| 测试存储库 | mysql://localhost:3306/ecss |
| —网络或子网 | 192.168.0.1 |
| —服务器名服务器名 | 无 |
| 测试开发 PC | Window, MacOS |

# 项目里程碑

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 | 4小时 | 2021/10/25 | 2021/10/26 |
| 设计测试 | 4小时 | 2021/11/26 | 2021/11/30 |
| 实施测试 | 6小时 | 2022/1/4 | 2022/1/5 |
| 执行测试 | 2小时 | 2022/1/5 | 2022/1/6 |
| 评估测试 | 2小时 | 2022/1/9 | 2022/1/8 |

# 可交付工件

1. 《测试计划》

## 测试模型

[本节确定将要通过测试模型创建并分发的报告。测试模型中的这些工件应该用 ASQ 工具来创建或引用。]

## 测试日志

[说明用来记录和报告测试结果和测试状态的方法和工具。]

## 缺陷报告

[本节确定用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具。]

# 附录 A：项目任务

以下是一些与测试有关的任务：

• 制定测试计划

* + 确定测试需求
  + 评估风险
  + 制定测试策略
  + 确定测试资源
  + 创建时间表
  + 生成测试计划

• 设计测试

- 准备工作量分析文档

- 确定并说明测试用例

- 确定并结构化测试过程

- 复审和评估测试覆盖

• 实施测试

* + 记录或通过编程创建测试脚本
  + 确定设计与实施模型中的测试专用功能
  + 建立外部数据集

• 执行测试

- 执行测试过程

- 评估测试的执行情况

- 恢复暂停的测试

- 核实结果

- 调查意外结果

- 记录缺陷

• 评估测试

- 评估测试用例覆盖

- 评估代码覆盖

- 分析缺陷

- 确定是否达到了测试完成标准与成功标准