<面向工业应用的柔性可扩展的边缘计算软件系统>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2021.11.1 | 1.0 | 对迭代一版本进行测试 | 芮召普 |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 4

4.1 需求覆盖率及缺陷分布 4

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

5.2 非功能性缺陷 5

6. 测试结论与建议 5

测试报告

# 简介

## 目的

该测试报告目的是记录面向工业应用的柔性可扩展的边缘计算软件系统项目的各方面测试环境、条件、数据、结果，并且对测试结果进行分析，并对项目进行改进。

## 范围

该测试报告影响到边缘计算软件系统项目前后端代码、测试用例文档、项目总结报告。

## 定义、首字母缩写词和缩略语

### 边缘计算软件系统

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。

## 参考资料

《面向工业应用的柔性可扩展的边缘计算软件系统需求规约》，《测试用例》。

## 概述

测试概要包含了测试的时间、地点、人员、测试方法、测试内容的简要说明。

测试环境描述了测试使用的软硬件环境及其配置。

测试结果及分析描述了测试的结果，并给出一些分析。

缺陷清单包含所有功能性缺陷和非功能性缺陷相关的信息。

测试结论与建议包含了对于项目测试的总结和对项目的改进建议。

# 测试概要

测试进行于11月20、21号，测试地点为软件大楼5314。人员为所有组员，包括董彦君、芮召普、江嘉晋、张俸铭。测试前期主要采用白盒测试、错误猜测、界面测试、控制流测试、基本路径测试，以确保操作逻辑、数据流的正确性和软件的可靠性；测试中期采用黑盒测试、变异测试、界面测试以及探索性测试，目的是增加软件可用性；测试后期采用集成测试，随机测试和潜在用户的即兴测试，以测试软件兼容性及其整体效果。

# 测试环境

实机测试使用了Ubuntu 18.04，64位，i5 7500H处理器。

电脑测试分别使用

1. Windows10，64位，i5 7200H处理器；
2. Windows10，64位，AMD R7-5800处理器；
3. Windows10，64位，i7-9750H 处理器；
4. MacOS 11 64位，i5-1038NG7处理器；

# 测试结果及分析

## 需求覆盖率及缺陷分布

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **边缘端设备数据管理** | 4 | 100% | 0 | 0% | 无 |
| **边缘端设备管理** | 4 | 100% | 0 | 0% | 无 |
| **边缘端状态管理** | 3 | 100% | 0 | 0% | 无 |
| **云端接收边缘端数据** | 1 | 100% | 0 | 0% | 无 |
| **云端边缘管理** | 4 | 100% | 0 | 0% | 无 |
| **功能项小计** | 16 | 100% | 0 | 0% | 无 |
| **非功能项** | **性能** | 4 | 100% | 0 | 0% | 无 |
| **可靠性** | 4 | 100% | 0 | 0% | 无 |
| **易用性** | 2 | 100% | 0 | 0% | 无 |
| **可测试性** | 1 | 100% | 0 | 0% | 无 |
| **可扩展性** | 1 | 100% | 0 | 0% | 无 |
| **可维护性** | 1 | 100% | 0 | 0% | 无 |
|  | **非功能项小计** | 13 | 100% | 0 | 0% | 无 |
|  | **总计** | 29 | 100% | 0 | 0% | 无 |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 0 | 0 | 0 | 0 | 0 |
| **占缺陷百分比** | 0% | 0% | 0% | 0% | 0% |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
|  |  |  |  |  |  |  |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

无

# 测试结论与建议

在本次迭代中实现的版本1软件，基于迭代1的界面原型和架构文档进行开发实现。总体功能点数量适中，开发较为顺利，因此测试过程中也没有发现明显的功能缺陷，使用较为流畅。