ITiMe自动化监控平台（监控系统）项目测试方案

文档版本控制

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档版本号** | **日期** | **作者** | **审核人** | **说明** |
| V1.0 | 2022.3.28 | 武利娜 |  | 创建文档 |
|  |  |  |  |  |

目录

[1 引言 3](#_Toc99388256)

[1.1编写目的 3](#_Toc99388257)

[1.2文档读者对象 3](#_Toc99388258)

[1.3项目背景 3](#_Toc99388259)

[1.4测试目标 3](#_Toc99388260)

[1.5参考文档 4](#_Toc99388261)

[1.6术语 4](#_Toc99388262)

[2测试资源 6](#_Toc99388263)

[2.1测试环境 6](#_Toc99388264)

[2.2人员配置 6](#_Toc99388265)

[2.3测试工具 7](#_Toc99388266)

[3测试要求 7](#_Toc99388267)

[3.1总体计划 7](#_Toc99388268)

[3.2测试策略 8](#_Toc99388269)

[3.2.1单元测试 8](#_Toc99388270)

[3.2.2集成测试 8](#_Toc99388271)

[3.2.3系统测试 9](#_Toc99388272)

[3.3测试关注点 12](#_Toc99388273)

[3.3.1输入框 12](#_Toc99388274)

[3.3.2下拉列表 13](#_Toc99388275)

[3.3.3增加数据 13](#_Toc99388276)

[3.3.4修改数据 13](#_Toc99388277)

[3.3.5删除数据 13](#_Toc99388278)

[3.3.6查询数据 13](#_Toc99388279)

[3.3.7数据导入导出 14](#_Toc99388280)

[3.3.8数据接入与处理 14](#_Toc99388281)

[3.3.9其他 14](#_Toc99388282)

[3.4测试重点 14](#_Toc99388283)

[3.5测试入口准则 15](#_Toc99388284)

[3.6测试通过标准 15](#_Toc99388285)

[3.7测试用例 15](#_Toc99388286)

[3.7.1测试用例分类 15](#_Toc99388287)

[3.7.2测试用例评审 16](#_Toc99388288)

[4测试计划 16](#_Toc99388289)

[5测试环境要求 17](#_Toc99388290)

[6测试输出物 18](#_Toc99388291)

[6.1测试输出文档 18](#_Toc99388292)

[6.2测试计划 18](#_Toc99388293)

[6.3测试方案 18](#_Toc99388294)

[6.4测试用例 18](#_Toc99388295)

[6.5缺陷列表 18](#_Toc99388296)

[6.6测试报告 18](#_Toc99388297)

[7风险预估与应对 18](#_Toc99388298)

# 1 引言

## 1.1编写目的

为软件开发项目管理者、软件工程师、系统维护工程师、测试工程提供关于ITiMe自动化监控平台（监控系统）项目，系统整体功能和性能的测试指导.

## 1.2文档读者对象

本测试方案的读者对象为软件开发项目管理者、软件开发工程师、测试组、系统维护工程师。

## 1.3项目背景

项目名称：ITiMe自动化监控平台

简称：监控系统

开发单位：北京创联致信科技有限公司

为了提升IT系统可用性、可靠性，企业 IT 支持能力，基于ITIL开发最佳实践的IT系统综合管理软件。

## 1.4测试目标

在用户使用软件之前，尽可能的发现软件中存在的错误和不合理之处，排除软件中潜在的错误，最终把高质量的软件系统交付给用户。系统的测试覆盖范围包含：功能、用户界面（UI）测试、性能（响应时间）、安全性、兼容性。

## 1.5参考文档

|  |  |
| --- | --- |
| 名称 | 备注 |
| GBT 15532-2008 计算机软件测试规范 | 确定文档格式及设计内容 |
| GBT 9386-2008 计算机软件测试文档编制规范 | 确定文档格式及设计内容 |

## 1.6术语

术语/定义，包括通用词语在本文档中专用解释。

|  |  |
| --- | --- |
| 术语/定义 | 说明 |
| 软件测试 | 软件测试就是利用手工测试和测试工具按照测试方案和流程对系统进行功能和性能测试，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方案可能出现的问题进行分析和评估。 |
| 系统测试 | 系统测试是通过与系统的需求规格作比较，发现软件与系统需求规格不相符合或与之矛盾的地方。它将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合起来，在实际运行环境下，对计算机系统进行的测试。 |
| 功能测试 | 黑盒测试是基于系统需求规格，在不知道系统或组件的内部结构的情况下进行的测试。通常又将黑盒测试叫做：基于规格的测试、输入输出测试、功能测试或数据驱动测试。是基于用户观点出发的测试。主要是验证功能是否符合需求，包括原定功能的校验、是否有冗余功能、遗漏功能。 |
| 接口测试 | 程序员对各个模块进行系统联调的测试，包含程序接口和程序外接口测试。这个测试，在单元测试阶段进行一部分工作，而大部分都是在集成测试阶段完成的。 |
| 压力测试 | 对系统不断施加压力的测试，是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。例如测试一个web站点在大量的负荷下，系统的响应何时会退化或失败。 |
| 性能测试 | 是交替进行负荷和强迫测试时常用的术语。性能测试关注的是系统的整体。它和通常说的说的强度、压力、负载测试有密切关系。所以压力和强度测试应该与性能测试一同进行。 |
| 安全性测试 | 主要是测试系统在没有授权的内部或者外部用户对系统进行攻击或者恶意破坏时如何进行处理，是否仍能保证数据的安全。 |
| 兼容性测试 | 指对所设计程序与硬件、软件之间的兼容性的测试。分为浏览器兼容测试 和分辨率兼容测试两类。一般来说，兼容性指能同时容纳多个方面，在计算机术语上兼容是指几个硬件之间、几个软件之间或是软硬件之间的相互配合程度。兼容性测试是指测试软件在特定的硬件平台上、不同的应用软件之间、不同的操作系统平台上、不同的网络等环境中是否能够很友好的运行的测试。  |
| 回归测试 | 是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他[代码](https://baike.baidu.com/item/%E4%BB%A3%E7%A0%81/86048)产生错误。回归测试作为[软件生命周期](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E7%94%9F%E5%91%BD%E5%91%A8%E6%9C%9F/861455)的一个组成部分，在整个[软件测试](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95/327953)过程中占有很大的工作量比重，[软件开发](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91/3448966)的各个阶段都会进行多次回归测试。在渐进和快速迭代开发中，新版本的连续发布使回归测试进行的更加频繁，而在[极端编程](https://baike.baidu.com/item/%E6%9E%81%E7%AB%AF%E7%BC%96%E7%A8%8B/7400429)方法中，更是要求每天都要进行若干次回归测试。 |

# 2测试资源

## 2.1测试环境

|  |  |  |  |
| --- | --- | --- | --- |
| 客户端 | 硬件 | 配置信息 | 数量 |
| 测试PC机 | Corei5，主频2.5GHZ，硬盘250G以上，内存16G | 1 |
| 软件分类 | 软件名称 | 版本 |
| 操作系统 | Windows | 11（64位） |
| 浏览器 | Google Chrome | 99.0.4844.7（正式版本） （64 位） |
| 服务器 | 硬件 | 配置信息 | 数量 |
|  |  |  |
| 软件分类 | 软件名称 | 版本 |
| 操作系统 | CentOS | 7 |
| Web中间件 | Tomcat |  |

## 2.2人员配置

|  |  |  |
| --- | --- | --- |
| 角色 | 人员 | 职责 |
| 项目经理 |  | * 提供技术指导并获取适当资源
* 负责整个项目的协调工作
 |
| 测试负责人 |  | * 编写测试方案、计划
* 项目测试的日常管理工作
* 监控测试工作，规避风险
* 编写系统测试报告等
 |
| 测试工程师 |  | * 编写和维护测试用例
* 执行测试并记录结果
* 缺陷跟踪
 |
| 前后端开发工程师 |  | * 对程序缺陷进行修改
* 程序新版本发布
* 对自己开发的功能进行自测
 |

## 2.3测试工具

|  |  |  |
| --- | --- | --- |
| 工具类型 | 工具名称 | 备注 |
| 用例管理工具 | Excel |  |
| 缺陷管理工具 | Excel、禅道 |  |
| 接口测试工具 | Postman |  |
| 性能测试工具 | Jmeter |  |

# 3测试要求

## 3.1总体计划

该系统测试策略有功能测试、用户界面（UI）测试、兼容性测试、安全性测试和性能测试，功能测试要覆盖系统中每个功能。在功能测试时既要输入正确的数据，测试功能是否满足，也要对每个功能中的每个数据输入域输入错误的数据，测试系统的健壮性。用户界面（UI）测试核实各个窗口风格（包括颜色、字体、提示信息、图标、title等）都要与需求保持一致，或符合可接受标准，保证用户界面的友好性、易操作性，而且符合用户操作习惯。兼容性测试核实系统在不同软件和硬件配置中运行稳定。性能测试往往针对软件的一部分功能，进行专项测试。执行完一组工作后，及时检查是否已达到预定目标，是否已执行完该过程所有的步骤等，如实际情况与计划出入较大，应及时调整计划。

系统的功能测试选用的是手工测试，运用黑盒测试中的等价类划分、边界值分析、错误推测、因果图法设计测试用例。系统界面（UI）测试包括：易用性测试、规范性测试、合理性测试、美观与协调性测试、快捷方式组合测试。系统的安全性、兼容性也采用手工测试。性能测试采用白盒测试。

## 3.2测试策略

### 3.2.1单元测试

首先按照系统、子系统和模块进行划分，但最终的单元必须是功能模块，或面向对象过程中的若干个类。单元测试是对功能模块进行正确性检验的测试工作，也是后续测试基础。目的是在于发现各模块内部可能存在的各种差错，因此需要从程序的内部结构出发设计测试用例，着重考虑以下五个方面：

* 模块接口：对所测模块的数据流进行测试。
* 局部数据结构：检查不正确或不一致的数据类型说明、使用尚未赋值或尚未初始化的变量、错误的初始值或缺省值。
* 路径：虽然不可能做到穷举测试，但要设计测试用例查找由于不正确的计算（包括算发错误、表达式的符号表示不正确、运算精度不够等）、不正确的比较或不正常的控制流（包括不同数据类型量的相互比较、不适当的修改了循环变量、错误的活不可能的循环终止条件等）而导致的错误。
* 错误处理：检查模块有没有对预见错误的条件设计比较完善的错误处理功能，保证其逻辑上的正确性。
* 边界：注意设计数据流、控制流中刚好等于、大于或小于确定的比较值的用例。

### 3.2.2集成测试

* 集成测试也叫组装测试或联合测试。通常在单元测试的基础上需要将所有的模块按照设计要求组装系统，这时需要考虑的问题如下：
* 把各个模块连接起来的时候，穿越模块接口的数据是否会丢失。
* 一个模块的功能是否会对另一个模块的功能产生不利的影响。
* 各个子功能组合起来，是否达到预期要求的父功能。
* 全局数据结构是否有问题。
* 单元模块的误差积累起来，是否会放大，从而达到不能接受的程度。我们在组装的时候可以参考采用一次性组装方式或者增值式组装方式。

### 3.2.3系统测试

系统测试目的是在于验证软件的功能和性能及其他特性是否与用户的要求一致，主要包含以下测试内容：

#### 3.2.3.1功能测试

验证系统功能是否符合其需求规格说明书，核实系统功能上是否完整，没有冗余和遗漏功能。

|  |  |
| --- | --- |
| 测试范围 | ITiMe自动化监控平台（监控系统）：验证数据精确度、数据类型、业务等相关方面的正确性 |
| 测试目标 | 核实所有功能均已正常实现，即是否与需求一致 |
| 工具与方法 | 手工测试 |
| 开始标准 | 开发阶段对应的功能完成并且测试用例设计完成 |
| 完成标准 | 测试用例覆盖率100%，且没有严重级别为1、2级bug |
| 测试重点与优先级 | 根据实际需求而定 |

#### 3.2.3.2用户界面（UI）测试

核实用户与软件之间的交互。UI测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。另外，UI测试还可以确保UI中的对象按照预期的方式运行，并符合公司或行业的标准。

|  |  |
| --- | --- |
| 测试范围 | 1. 导航、链接、cookie、页面结构包括菜单、背景、颜色、字体、按钮名称、title、提示信息的一致性等
2. 友好性、可操作性（易用性）
3. 界面初始化数据加载是否与数据库一致等
 |
| 测试目标 | 核实各个窗口风格（包括颜色、字体、提示信息、图标、title等）都与需求保持一致，或符合可接受标准，能够保证用户界面的友好性、易操作性，而且符合用户操作习惯 |
| 工具与方法 | 手工测试、目测 |
| 开始标准 | 界面开发完成 |
| 完成标准 | UI符合可接受标准，能够保证用户界面的友好性、易操作性，而且符合用户操作习惯 |
| 测试重点与优先级 | 根据实际需求而定 |

#### 3.2.3.3性能测试：

对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。性能测试的目标核实性能需求是否都已满足。实施和执行性能测试的目的是将测试对象的性能行为当作条件（例如工作量或硬件配置）的一种函数来进行测试和微调。

|  |  |
| --- | --- |
| 测试范围 | 多用户长时间在线操作时性能方面的测试 |
| 测试目标 | 核实系统在大流量的数据与用户操作时软件性能的稳定性，不造成系统崩溃或相关的异常现象 |
| 工具与方法 | 手工测试、jmeter |
| 开始标准 | 界面开发完成 |
| 完成标准 | 系统满足用户需求的性能要求 |
| 测试重点与优先级 | 根据实际需求而定 |

#### 3.2.3.4兼容性测试

测试软件在不同平台上使用的兼容性。

|  |  |
| --- | --- |
| 测试范围 | 1. 使用不同版本的浏览器、分辨率、操作系统分别进行测试
2. 不同操作系统、浏览器、分辨率和各种运行软件等各种条件的组合测试
 |
| 测试目标 | 核实系统在不同软件和硬件配置中运行稳定 |
| 工具与方法 | 手工测试 |
| 开始标准 | 界面开发完成 |
| 完成标准 | 在各种不同版本不同类型浏览器、操作系统或者其组合下均能正常实现其功能(此测试根据开发提供的依据决定测试的范围） |
| 测试重点与优先级 | 根据实际需求而定 |

#### 3.2.3.5安全性测试

|  |  |
| --- | --- |
| 测试范围 | 1. 用户、管理员的密码安全
2. 权限
 |
| 测试目标 | 1. 用户、管理员的密码管理
2. 应用程序级别的安全性：核实用户只能操作其权限内能操作的功能
3. 系统级别的安全性：核实只有具备系统访问权限的用户才能访问系统
 |
| 工具与方法 | 手工测试 |
| 开始标准 | 功能测试完成 |
| 完成标准 | 执行各种非法操作无安全漏洞且系统使用正常 |
| 测试重点与优先级 | 根据实际需求而定 |

#### 3.2.3.6回归测试：

|  |  |
| --- | --- |
| 测试范围 | 所有功能、用户界面、兼容性、安全性等测试类型 |
| 测试目标 | 核实执行所有测试类型后功能、性能、等均达到用户需求所要求的标准 |
| 工具与方法 | 手工测试 |
| 开始标准 | 每当被测试的软件或其开发环境改变时，在每个核实的测试阶段上进行回归测试 |
| 完成标准 | 95% 的测试用例执行通过并通过系统测试 |
| 测试重点与优先级 | 根据实际需求而定 |

## 3.3测试关注点

### 3.3.1输入框

* 检测空数据
* 检测过长数据（超出空间本身的长度和数据库中该字段所允许的长度）
* 检测特殊字符，尤其是数据库中不允许的字符，回车字符、空格字符等
* 检测字符类型，比如应该输入数字的文本框输入英文字
* 中文字符的处理
* 对于日期时间型数据，检查格式正确性以及时间日期的合理性。比如开始时间不能晚于结束时间等

### 3.3.2下拉列表

* 列表数据是否正确、完整
* 下拉列表与其他空间的联动关系
* 是否允许多选

### 3.3.3增加数据

* 数据个数的上限
* 重复数据处理，尤其是键值的重复
* 相关表格的更新
* 检查多次使用back剪的情况：在有back的地方，back回到原页面，再back，重复多次，看是否会出错

### 3.3.4修改数据

* 不能破坏数据库的关联和完整
* 重复数据处理，尤其是键值的重复
* 修改登录用户本身信息时对系统的影响
* 修改正在使用的数据
* 检查多次使用back剪的情况：在有back的地方，back回到原页面，再back，重复多次，看是否会出错

### 3.3.5删除数据

* 不能破坏数据库的管理和完整
* 删除正在使用的数据
* 删除登录用户本身

### 3.3.6查询数据

* 多条件组合查询的正确性
* 多次连续查询正确性

### 3.3.7数据导入导出

* 导入数据格式要求不应太苛刻，提示明确
* 导出数据不应乱码

### 3.3.8数据接入与处理

* 数据接入方式是否全不能用，数据是否能正确接入
* 数据处理方式是否全部可行
* 数据的动态监测是否正确无误

### 3.3.9其他

* 对网络故障的提示
* 多点登录
* 内存使用情况
* 压力测试，系统承受能力，多用户同时登录使用

## 3.4测试重点

测试重点主要从以下几个方面考虑，针对测试重点，在用例的辩编写与评审、人员安排、测试轮次、缺陷解决要求等方面都应高于其他部分。

* 需求中，优先级高的重点功能或用户常用的功能。
* 开发过程中，重点关注的模块、功能及特性（此项通过交易的代码修改量等内容确定，由项目经理提提供）。
* 相关领导的关注点和意见。
* 以往版本中常见的问题。

注：此项内容由项目经理配合进行确认。

## 3.5测试入口准则

* 在提交测试组进行系统测试前，开发人员需要通过自测
* 测试组接收测试，且通过冒烟测试，即可开始进行系统测试

## 3.6测试通过标准

* 系统无业务逻辑错误和1、2级缺陷，经确定的所有缺陷都已得到商定的解决结果且没有发现新的缺陷
* 设计的测试用例全部执行完成，由于其他因素导致未能执行的用例有相应的记录
* 测试覆盖率100%， 测试用例通过率100%

注：缺陷严重等级说明

1级：严重影响系统运行的错误

2级：功能方面一般缺陷，影响系统公运行

3级：界面布局不美观或轻型错误

4级：不影响运行的错别字等

5级：合理化建议

## 3.7测试用例

### 3.7.1测试用例分类

* 测试用例与测试类型对应：功能测试用例、用户界面测试用例及性能测试用例
* 重点用例通过通过用例中的优先级进行标记
	+ 1. 关键业务正常流测试
		2. 功能点详细测试
		3. 交互测试：主要测试界面、易用性等内容
		4. 异常测试

### 3.7.2测试用例评审

* 组内评审：测试组内部对测试用例进行交叉评审
* 组外评审：项目组相关人员（或由项目经历或部门经历指定），对重点用例进行评审

# 4测试计划

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目里程碑 | 任务名称 | 开始时间 | 结束时间 | 输出物 | 执行人员 | 备注 |
| 制定测试方案 | 编写测试方案 | 2022.3.22 | 2022.3.28 | 测试方案 | 武利娜 |  |
| 设计测试 | 测试用例编写 | 2022.3.29 | 2022.4.19 | 测试用例 | 武利娜 |  |
| 测试用例评审 | 2022.4.20 | 2022.4.20 | 测试用例评审记录表 | 武利娜 |  |
| 执行测试 | 第一轮 | 2022.4.21 | 2022.5.11 | 缺陷记录 | 武利娜 |  |
| 第二轮 | 2022.5.12 | 2022.5.25 | 缺陷记录 | 武利娜 |  |
| 第三轮 | 2022.5.26 | 2022.6.1 | 缺陷记录 | 武利娜 |  |
| 评估测试 | 测试总结 | 2022.6.2 | 2022.6.2 | 测试报告 | 武利娜 |  |

注：轮次测试的具体内容会根据各子系统开发进度做适当调整。

其他注意事项：

* 测试工程师根据测试用例进行测试，并将测试中发现的bug，记录到禅道中
* 开发工程师对bug进行修改，并说明bug产生的原因及阶段
* 如果对需要修改的bug意见不统一，则需要项目经理、开发、测试开会商定
* 第二轮系统测试开始，测试工程师首先对第一轮测试中遗留的问题进行回归验证，即验证上一轮发现的bug是否已经全部解决。回归测试完成后，测试工程师再根据测试用例，开展新一轮的系统测试工作。
* 第三轮测试，结合核心系统进行测试，同时加强对业务系统中重点交易的测试

# 5测试环境要求

本次测试建议在测试环境下进行，如果在生产环境中进行测试，可能会带来以下影响：

5.1软件在测试过程中，测试工程师可能有意或者无意的给被测软件产生一些垃圾数据，由于被测软件正在使用，而这些垃圾数据可能会影响被测软件功能的正常显示、可能会影响被测软件的数据查询、统计的准确性。

5.2软件在进行功能测试时，测试工程师可能会设计一个或多个功能方面的安全测试用例，如果被测软件对这方面处理的不好，测试过程中，可能会导致系统崩溃、死机等现象，因而影响系统的正常运行。

5.3软件在进行功能测试时，还可能出现一些不能预知的问题，同样会影响系统的正常运行。

所以，本次测试建议部署独立的测试环境，避免对生产环境带来风险。

# 6测试输出物

## 6.1测试输出文档

## 6.2测试计划

## 6.3测试方案

## 6.4测试用例

## 6.5缺陷列表

## 6.6测试报告

# 7风险预估与应对

下表列出了在项目测试工作中存在的各种风险的假定，需要考虑项目测试过程中可能发生的具体事务，分别分析并加以应对。

|  |  |  |  |
| --- | --- | --- | --- |
| 风险类型 | 风险内容 | 处理优先级 | 备注 |
| 时间计划 |  |  |  |
| 人员风险 | 测试人员不足 | 1 | 持续关注求职平台，早日找到合适的测试人员 |
| 资源协调 |  |  |  |
| 插入事务 |  |  |  |
| 任务超预期 |  |  |  |

注：各个风险类型解释如下：

* 时间计划：关键里程碑无法匹配的延期风险
* 人员风险：测试人员和需配合方的人员变动导致的工作任务无法按计划完成或者完成质量无法保证的风险，包括新人风险、人员变化、投入不足、投入质量不高等
* 资源协调：包括所需资源不能如期到位，或者资源质量低于预期等风险。比如测试工具开发的风险、各个阶段交付物质量风险等
* 插入事务：包括临时插入优先级高的任务，打乱原有计划等风险
* 任务超预期：实际执行时工作复杂程度、结果的质量同预期不符所带来的风险。属于不可预期风险，只能待出现时及时合理地调整

风险分为可预期的和不可预期的，对于可预期的风险，可以要求资源，制定提前的应对措施。但是对于不可预期的风险，只能待出现时，充分考虑各方因素，及时调整。所以，对于可预期的风险，需要的能力的充分的预估，对于不可预期的风险，需要的是及时察觉并调整应对。