Online-bug-detection-engine-for-java-programs

测试计划文档 V1.0

2009-6-30

上海交通大学软件学院

hp

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **版本** | **描述** | **作者** |
| 2009-7-29 | v1.0 | 初始版本 | 季尚松 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[目录 2](#_Toc236932240)

[1 简介 4](#_Toc236932241)

[1.1 目的 4](#_Toc236932242)

[1.2 背景 4](#_Toc236932243)

[1.3 范围 4](#_Toc236932244)

[1.4 项目标识 4](#_Toc236932245)

[2 测试需求 5](#_Toc236932246)

[2.1 数据库测试 5](#_Toc236932247)

[2.2 功能测试 5](#_Toc236932248)

[2.3 业务周期测试 5](#_Toc236932249)

[2.4 用户界面测试 5](#_Toc236932250)

[2.5 性能评价 5](#_Toc236932251)

[2.6 负载测试 6](#_Toc236932252)

[2.7 强度测试 6](#_Toc236932253)

[2.8 容量测试 6](#_Toc236932254)

[2.9 安全与访问控制测试 6](#_Toc236932255)

[2.10 故障转移与恢复测试 6](#_Toc236932256)

[2.11 配置测试 6](#_Toc236932257)

[2.12 安装测试 6](#_Toc236932258)

[3 测试策略 7](#_Toc236932259)

[3.1 测试类型 7](#_Toc236932260)

[3.1.1 数据与数据库完整性测试 7](#_Toc236932261)

[3.1.2 功能测试 7](#_Toc236932262)

[3.1.3 用户界面测试 7](#_Toc236932263)

[3.1.4 性能评价 8](#_Toc236932264)

[3.1.5 负载测试 8](#_Toc236932265)

[3.1.6 强度测试 9](#_Toc236932266)

[3.1.7 容量测试 9](#_Toc236932267)

[3.1.8 安全性和访问控制测试 9](#_Toc236932268)

[3.1.9 配置测试 10](#_Toc236932269)

[3.2 工具 10](#_Toc236932270)

[4 资源 10](#_Toc236932271)

[4.1 角色 10](#_Toc236932272)

[4.2 功能摘要 10](#_Toc236932273)

[4.3 假设与依赖关系 10](#_Toc236932274)

[4.4 成本与定价 11](#_Toc236932275)

[5 产品特性 11](#_Toc236932276)

[5.1 用户管理 11](#_Toc236932277)

[5.2 用户文件管理 11](#_Toc236932278)

[5.3 生成报告 11](#_Toc236932279)

[5.4 生成JAVA字节码文件 11](#_Toc236932280)

[6 约束条件 11](#_Toc236932281)

[7 质量范围 11](#_Toc236932282)

[7.1 可靠性 11](#_Toc236932283)

[7.2 易用性 12](#_Toc236932284)

[7.3 效率 12](#_Toc236932285)

[7.4 可维护性 12](#_Toc236932286)

[7.5 可移植性 12](#_Toc236932287)

[8 优先级 12](#_Toc236932288)

[9 其他产品需求 12](#_Toc236932289)

[9.1 适用标准 12](#_Toc236932290)

[9.2 环境需求 12](#_Toc236932291)

[10 文档需求 12](#_Toc236932292)

[10.1 用户手册 13](#_Toc236932293)

[10.2 联机帮助 13](#_Toc236932294)

[10.3 自述文件 13](#_Toc236932295)

[10.4 包装标签 13](#_Toc236932296)

# 简介

## 目的

<OnlineDebug在线纠错引擎>（下称OnlineDebug引擎） 的这一“测试计划”文档有助于实现以下目标：

* 确定<OnlineDeubg引擎>的信息和应测试的软件构件。
* 列出推荐的测试需求（高层次）。
* 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< OnlineDeubg引擎>的可交付元素。

## 背景

* 此次测试对象即OnlineDebug引擎。
* 该项目用于提供一个在线的Java程序分析纠错工具在标准化菜场的选址过程中对选址规划策略进行分析的系统。用户通过不同的方式输入所要进行纠错的文件或项目后,系统将会缺陷错误报告并将缺陷进行统计。
* 本测试计划将针对OnlineDebug引擎的主要功能以及特性而制定。

## 范围

* 在单元测试阶段，测试人员将会对源代码中的PMD以及Findbugs部分和Utils包中的部分类进行测试.这些代码完成了系统的基本功能,包括上传源代码编译,上传RAR文件的解压, Report的生成,XML的解析等.该阶段测试为功能测试.
* 在集成测试中，将对该系统的主要功能，即对上传的文件进行进行测试，将会设计一定数量的测试用例以保证系统的正确性.本阶段测试为功能与性能测试的结合。
* 在安装测试中，将会用不同的浏览器使用该系统,以验证系统能否在各种浏览器运行.本阶段测试为性能测试.

## 项目标识

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |
| 需求规约 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |
| 功能性规约 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |
| 项目计划 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |
| 设计规约 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |
| 原型 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |
| 用户手册 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |
| 数据模型或数据流 | 🗹 是 🞏 否 | 🗹 是 🞏 否 | Baldur开发小组 | 无 |

# 测试需求

## 数据库测试

验证系统能够正确地连接到OnlineDebug数据库。

验证能够对数据库进行正确的增删改查操作。

## 功能测试

验证用户能够正确的注册。

验证注册用户能够正确的登录。

验证用户能够正确地通过将粘贴至TextArea中提交并进行纠错。

验证用户能够正确地通过上传选中的单个文件进行纠错。

验证注册用户能够正确地新建项目，通过上传整个项目源文件的RAR压缩包。

验证系统能够正确地对注册用户上传的RAR压缩包进行解压。

验证系统能够正确地对新建的项目进行纠错。

验证系统能够正确地对纠错后的结果进行解析并生成符合格式报告。

验证注册用户能够正确地修改自己纠错的设置。

验证系统能够在进行纠错的同时进行对缺陷的统计。

验证用户能够正确地查看已经记录于数据库的缺陷统计数据。

验证用户能够通过统计数据上上的链接点击查看缺陷的结善。

## 业务周期测试

无

## 用户界面测试

对于所有用例，验证所有的UI元素对于用户来说都简洁易懂。

验证所有的UI元素风格一致。

## 性能评价

验证OnlineDebug系统的响应时间。

验证系统在本地访问下的响应时间

验证系统在部署至服务器上后通过internet访问的响应时间。

## 负载测试

验证系统在10个用户同时进行单一文件纠错时的反应。

验证系统在50个用户同时进行单一文件纠错时的反应。

验证系统在100个用户同时进行单一文件纠错时的反应。

验证系统在500个用户同时进行单一文件纠错时的反应。

验证系统在10个注册用户同时进行项目纠错时的反应。

验证系统在50个注册用户同时进行项目纠错时的反应。

验证系统在100个注册用户同时进行项目纠错时的反应。

验证系统在500个注册用户同时进行项目纠错时的反应。

## 强度测试

验证系统在进行每天24小时，每周7天的高强度工作下的反应。

## 容量测试

验证系统所允许的最大的用户同时操作数。

## 安全与访问控制测试

验证只有注册用户登录后才能够使用项目纠错。

## 故障转移与恢复测试

无

## 配置测试

验证系统能够使用IE7访问使用本系统。

验证系统能够使用IE8访问使用本系统。

验证系统能够使用FF3访问使用本系统。

## 安装测试

无

# 测试策略

## 测试类型

### 数据与数据库完整性测试

|  |  |
| --- | --- |
| 测试目标 | 确认系统能够对数据库进行正确的访问并且能够正确地完成增，删，改，查操作。 |
| 方法 | 连接数据库，对数据库用合法的和不合法的数据分别进行各类操作。  查看从数据库中所查找得到的数据是否是正确的，所需要的数据。 |
| 完成标准 | 数据库的各种访问和操作都被完善的设计并且不存在数据损坏。 |
| 需要考虑的特殊事项 | 各种操作需要手动的完成。 |

### 功能测试

|  |  |
| --- | --- |
| 测试目标 | 确认OnlineDebug系统的各个功能点被正确实现 |
| 方法 | 对于每一个用例，给出正确或是不正确的输入数据，来验证以下两点：   1. 对于相应的正确的输入，给出正确的结果。 2. 对于不合法的输入会给出相应的合适的错误或是警告信息。 |
| 完成标准 | 1. 每个计划的测试都被执行。 2. 所有被找出的缺陷将被逐一纠正。 |
| 需要考虑的特殊事项 | 无 |

### 用户界面测试

|  |  |
| --- | --- |
| 测试目标 | 确认以下内容：   1. 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。 2. 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。 |
| 需要考虑的特殊事项 | 征询其他非开发人员的意见，是UI尽量能够达到可接受标准 |

### 性能评价

|  |  |
| --- | --- |
| 测试目标 | 核实OnlineDebug系统在以下情况下的性能行为：   1. 正常的预期工作量 2. 预期的最繁重工作量 |
| 方法 | 使用为功能测试制定的测试过程。  增加对一个文件或是一个项目进行纠错时文件或是项目的大小。  增加同一时间同时进行纠错的用户数量大小。  观察系统的响应时间。 |
| 完成标准 | 1. 在对单个用户请求的预期处理时间内完成单个文件或是单一项目的纠错。 2. 在处理多个请求时，在用户所能接受的时间内完成对单个请求的处理。 |
| 需要考虑的特殊事项 | 1. 由于不同机器的运算速度不同，应对不同运算速度的制定不同的标准进行测试，并与同类产品进行比较。 2. 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：  * 直接将“事务强行分配到”服务器上，这通常以“结构化查询语言”(SQL) 调用的形式来实现。 * 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。 |

### 负载测试

|  |  |
| --- | --- |
| 测试目标 | 确认OnlineDebug对大小的负载的处理能力。的并发请求以及对不同大小的单一文件或是项目文件进行纠错的反应。 |
| 方法 | 该项测试可分为三步：   1. 对单一用户的单一文件或是项目的纠错中，不断增大文件大小或是数量。 2. 固定项目或是单一文件大小，不断增加并发请求数，即多个用户同时进行纠错。 3. 在前两个测试的基础上可以修改用户数量或是项目大小。 |
| 完成标准 | 对各种数量的用户同时访问，各种大小的文件或是项目的纠错过程中，OnlineDebug系统都能够在可接受的时间范围内完成工作。 |
| 需要考虑的特殊事项 | 由于不同机器的运算速度不同，应对不同运算速度的制定不同的标准进行测试。 |

### 强度测试

|  |  |
| --- | --- |
| 测试目标 | 确认OnlineDebug系统能够能够在以下强度条件下正常运行，不会出现任何错误：   1. PC上几乎没有或根本没有可用的内存（RAM和 DASD） 2. 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”）。 |
| 方法 | 使用为性能评价或负载测试制定的测试。  使用其他应用程序将机器的内存消耗殆尽，并产生最繁重的事务量或最差的事务组合进行测试。 |
| 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。 |
| 需要考虑的特殊事项 | 用其他应用程序占据内存的方式非常不准确，故本测试的参考意义值得商榷。 |

### 容量测试

|  |  |
| --- | --- |
| 测试目标 | 本项测试是为了确认在以下两点下OnlineDebug系统能够正确运行的。   1. OnlineDebug系统的所能接受的最大并发数以及最大的单一文件大小和最大项目文件个数。 2. 数据库已达到最大规模并且同时执行了多个查询或是其他操作。 |
| 方法 | 使用为性能评价或负载测试制定的测试。   1. 应该使用多台客户机来运行相同的测试或互补的测试，以便在长 时间内产生最繁重的事务量或最差的事务组合（请参见上面的”强度测试”）。 2. 创建最大的数据库大小（实际的、按比例缩放的、或输入了代表性数据的数据库），并使用多台客户机在长时间内同时运行查询和其他操作。 |
| 完成标准 | 所计划的测试已全部执行，而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. |
| 需要考虑的特殊事项 | 对于上述的大容量条件，哪个时段是可以接受的时间？ |

### 安全性和访问控制测试

|  |  |
| --- | --- |
| 测试目标 | 确认不同权限的用户所能访问的域的限制 |
| 方法 | 检查代码中的Session检查. |
| 完成标准 | 在对所有只有注册用户登录后才能进行的操作的代码中需事先进行Session的检查已达到控制用户访问权限的目的. |
| 需要考虑的特殊事项 | 无 |

### 配置测试

|  |  |
| --- | --- |
| 测试目标 | 1. 验证能够通过各主流浏览器访问OnlineDebug系统 2. 能够在所提供的基础配置的计算机中访问OnlineDebug系统. |
| 方法 | 通过各主流浏览器(包括IE7,IE8,FF3,Chrome)访问系统,来验证以下:   1. 能够正确的使用系统所提供的各个功能. 2. 页面元素正确,美观,有效.   使用所预期的最低配置的计算机访问系统.能够正确的使用系统所提供的各个功能. |
| 完成标准 | 1. 使用各主流浏览器访问时界面元素能够一致或是大致一致地被显示. 2. 使用所预期的最低配置的计算机访问系统时能够正确的使用系统所提供的各个功能. |
| 需要考虑的特殊事项 | 无 |

## 工具

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **角色** |
| Baldur开发组 | 项目的开发组 | 负责项目的开发工作，成员包括徐昊，季尚松，胡超 |
| 服务端管理者 | 负责维护服务端的系统 | 对服务其进行维护，包括备份数据库等工作 |
| Java开发者 | 一般的JAVA开发者 | 使用JAVA语言的开发者 |

# 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名 | 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 | 2名 | 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名 | 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名 | 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 | 2名 | 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 | 2名 | 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 | 2名 | 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

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 |  |
| —网络或子网 | TBD |
| —服务器名服务器名 | TBD |
| —数据库名 | MySQL |
| 客户端测试 PC | TBD |
| 包括特殊的配置需求 | 操作系统：Windows Vista ultimate  处理器:Intel core Duo T2350 1.87Ghz  内存: 3070MB |
| 测试开发 PC | TBD |

# 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 | 1 | 27/07/2008 | 28/07/2008 |
| 设计测试 | 1 | 28/07/2008 | 29/07/2008 |
| 实施测试 | 2 | 29/07/2008 | 30/07/2008 |
| 执行测试 | 2 | 30/07/2008 | 31/07/2008 |
| 评估测试 | 1 | 31/07/2008 | 01/08/2008 |

# 可交付工件

在测试过程中将会提交的有

* .测试计划
* .测试用例报告
* 测试评估报告
* 缺陷报告

# 附录A

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