NBA Analysis 测试文档

组48\_高效率小组

目录

[NBA Analysis 测试文档 1](#_Toc414723076)

[组48\_高效率小组 1](#_Toc414723077)

[目录 2](#_Toc414723078)

[1 前言 3](#_Toc414723079)

[1.1 发布日期 3](#_Toc414723080)

[1.2 团队 3](#_Toc414723081)

[1.3 成员 3](#_Toc414723082)

[1.4 变更历史 3](#_Toc414723083)

[2 总体介绍 4](#_Toc414723084)

[2.1 编制目的 4](#_Toc414723085)

[2.2 对象 4](#_Toc414723086)

[2.3 产品概述 4](#_Toc414723087)

[2.4 参考资料 4](#_Toc414723088)

[2.5 词汇表 4](#_Toc414723089)

[3 测试计划 6](#_Toc414723090)

[3.1 单元测试计划 6](#_Toc414723091)

[3.1.1 单元测试范围 6](#_Toc414723092)

[3.1.2 单元测试策略 6](#_Toc414723093)

[3.2 集成测试计划 7](#_Toc414723094)

[3.2.1 集成测试计划范围和资源 7](#_Toc414723095)

[3.2.2 集成测试结束交付的产物 8](#_Toc414723096)

[3.2.3 执行集成测试的入口和出口条件 8](#_Toc414723097)

[3.2.4 如何判断集成测试用通过 8](#_Toc414723098)

[3.2.5 如何判断集成测试用例失败 8](#_Toc414723099)

[4 具体测试 9](#_Toc414723100)

[4.1 测试类型 9](#_Toc414723101)

[4.2 测试环境与配置 9](#_Toc414723102)

[4.3 测试问题总结 9](#_Toc414723103)

[4.4 单元测试用例 10](#_Toc414723104)

[4.5 集成测试用例 12](#_Toc414723105)

[4.6 功能测试 12](#_Toc414723106)

[4.7 性能测试 14](#_Toc414723107)

[4.8 缺陷报告 15](#_Toc414723108)

[4.9 测试结果分析 15](#_Toc414723109)

[5 综合评价 15](#_Toc414723110)

# 前言

## 发布日期

2015年4月12日

## 团队

组48\_高效率小组

## 成员

陈建伟、梅杰、王俊超、刘瀚文

## 变更历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 原因 | 版本号 |
| 梅杰 | 2015/4/12 | 建立文档模板 | V0.1 |
| 刘瀚文 | 2015/4/13 | 完成文档 | V1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 总体介绍

## 编制目的

本文档编写了NBA数据分析系统的单元测试和集成测试的过程和结果，为团队内部对软件成品质量评估提供重要依据

## 对象

本文档读者是本小组全部成员。

## 产品概述

参见《CSEIII项目概述及迭代二需求说明》。

## 参考资料

1.《软件工程与计算（卷三）：团队与软件开发实践》

## 词汇表

|  |  |
| --- | --- |
| 词汇名称 | 词汇含义 |
| NBA | 美国及加拿大职业篮球联盟 |
|  |  |

# 测试计划

## 单元测试计划

### 单元测试范围

单元测试必须能够展示出NBA数据查询系统的类的每个方法都能正常运行，单元测试的概念是将一个单元与该系统的其余部分隔离开来进行独立测试，因此可能需要创建测试驱动代码。要求测试驱动代码用JUIt编写，当被测单元被提交装备完成项目构建时，相关的测试驱动代码必须已经被提交到团队的代码中。

### 单元测试策略

#### 对方法(method)的测试

检查对所有调用对象的使用。

验证对所有数据结构的处理。

验证对所有文件的处理。

验证控制流的不变性。

检查所有循环的正常终止。

检查所有循环的异常终止。

验证所有错误条件的处理。

检查定时和同步。

#### 对类(classs)的测试

1）结合方法的执行，推荐对类的测试

选择最常见的执行序列。

需要包括可能导致缺陷的序列。

手工计算预期的属性值。

2）集中对每个属性进行单元测试

初始化，然后执行会影响属性的方法序列

3）验证每个类的不变性。

验证默认初始值的不变性

执行方法序列

验证不变性仍然正确。

## 集成测试计划

### 集成测试计划范围和资源

#### 集成测试的主要任务

在链接各个模块时候测试数据是否会丢失

在组合各个模块的子功能后，测试是否达到预期的父功能。

测试一个模块的功能是否会对另一个模块功能产生不利影响

测试单个模块的误差积累是否会放大到不可接受的程度。

#### 集成测试的资源

测试成员为4人，分工如下：

陈建伟，梅杰：界面测试

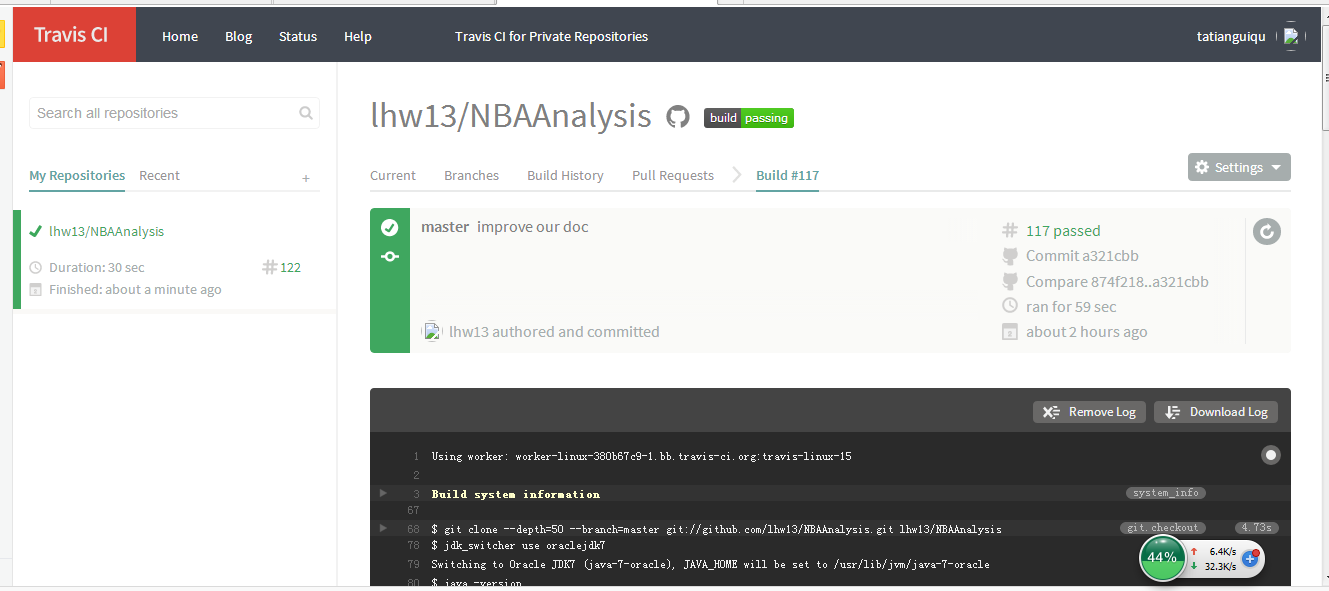
刘瀚文：逻辑测试

王俊超：数据测试

测试时间为4天

#### 集成测试环境

Travis-CI在线持续集成



### 集成测试结束交付的产物

（1）集成测试用例文档的执行情况表

（2）bug报告表

（3）集成测试报告

### 执行集成测试的入口和出口条件

#### 入口条件

单元测试必须完成

开发出完整的系统

搭建了类似于实际使用/配置环境的测试环境

测试人员安排到位

#### 出口条件

所有集成测试用例都被执行，测试结果100%被通过

集成中发现的bug已经被定位或者关闭

### 如何判断集成测试用通过

1）接口集成时，接口提供的功能或数据正确

2）功能点集成时，验证与《项目设计文档》中的描述一致

### 如何判断集成测试用例失败

1）接口集成时，接口提供的功能或数据不正确

2）功能点集成时，验证与《项目设计文档》中的对需求描述不一致

3）集成测试用例未通过，提交测试负责人，由测试负责人刘瀚文判断是否需要转交他人进行测试。

# 具体测试

## 测试类型

|  |  |  |  |
| --- | --- | --- | --- |
| 测试类型 | 测试内容 | 测试目的 | 所用的测试工具和方法 |
| 功能测试 | 1. 球员信息查看 2. 球队信息查看 3. 球员列表查看 4. 球队列表查看 5. 热点球员查看 6. 热点球队查看 7. 进步最快球员查看 8. 动态数据测试 | 核实所有功能均已正常实现，即可按用户的需求使用系统   1. 业务流程检验：各个业务流程能够满足用户需求，用户使用不会产生疑问 2. 数据准确：各数据输入输出时系统计算准确 | 采用黑盒测试，使用边界值测试、进行手工测试及Junit单元测试 |
| 用户界面测试 | 1. 导航、链接、页面结构包括菜单、背景、颜色、字体、按钮、Title、提示信息的一致性等 2. 友好性、易用性、合理性、一致性、正确性 | 核实程序界面风格的可用性和美观性 | 手工测试 |

## 测试环境与配置

|  |  |
| --- | --- |
| 资源名称/类型 | 配 置 |
| 测试PC | 主频2.66GHz，硬盘40G，内存4.0GB |
| 应用服务器 | Github版本控制、Travis-ci持续集成服务器 |
| 应用软件 | Eclipse 4.4 |
| 单元测试工具 | Maven+SureFire2.18+Junit4.11 |

## 测试问题总结

在整个系统测试执行期间，项目组开发人员高效地及时解决测试人员提出的各种缺陷，在一定程度上较好的保证了测试执行的效率以及测试最终期限。

## 单元测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例标识符 | 测试用例名称 | 状态 | 预期结果 | 测试结果 | 备注 |
| BL1 | testPlayerAnalyse | 通过 | 得到球员列表 | 得到球员列表 |  |
| BL2 | testPlayerAnalyseByName | 通过 | 得到单个球员对象 | 得到单个球员对象 |  |
| BL3 | testPlayerByTeamAnalyse | 通过 | 得到球队中所有球员列表 | 得到球队中所有球员列表 |  |
| BL4 | testTeamAnalyse | 通过 | 得到球队列表 | 得到球队列表 |  |
| BL5 | testTeamAnalyseByName | 通过 | 得到单个球队对象 | 得到单个球队对象 |  |
| BL6 | testDirtyData | 忽略 | - | - | 此例用于变成初期检测数据可靠性 |
| BL7 | testPlayerResult | 通过 | 球员计算结果正确（属性过多，不再赘述，参见程序） | 球员计算结果正确（属性过多，不再赘述，参见程序） |  |
| BL8 | testTeamResult | 通过 | 球队计算结果正确（属性过多，不再赘述，参见程序） | 球队计算结果正确（属性过多，不再赘述，参见程序） |  |
| UI7 | testPlayeruiSelection | 通过 | - | - | 测试球员选择界面能否正常显示，内容是否完全、正常 |
| UI8 | testPlayeruiInfo | 通过 | - | - | 测试球员个人信息界面能否正常显示，内容是否对应、合理 |
| UI9 | testuiPlayeruiRank | 通过 | - | - | 测试球员技术统计排名界面能否正常显示，能否进行升降序排列，能否筛选 |
| UI10 | testTeamuiRank | 通过 | - | - | 测试球队技术统计排名界面能否正常显示，能否进行升降序排列 |
| UI11 | testTeamuiSelection | 通过 | - | - | 测试球队选择界面能否正常显示，能否跳转 |
| UI12 | testTeamuiInfo | 通过 | - | - | 测试球队详细信息正常显示，内容是否对应、合理 |
| DS1 | MatchesDataTest | 通过 | 1 | 1 | 测试比赛数据读取是否正确 |
| DS2 | PlayersDataTest | 通过 | 89 | 89 | 测试球员数据读取是否正确 |
| DS3 | TeamsDataTest | 通过 | ATL | ATL | 测试球队数据读取是否正确 |
| Consel1 | Team1 | 通过 | 对比文件 | | 测试命令1正确性 |
| Consel2 | Team2 | 通过 | 测试命令2正确性 |
| Consel3 | Team3 | 通过 | 测试命令3正确性 |
| Consel4 | Player | 通过 | 测试命令4正确性 |

## 集成测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例标识符 | 测试用例名称 | 状态 | 预期结果 | 测试结果 | 备注 |
| UI1 | testPlayerAnalyse | 通过 | - | - |  |
| UI2 | testPlayerAnalyseByName | 通过 | - | - |  |
| UI3 | testPlayerByTeamAnalyse | 通过 | - | - |  |
| UI4 | testTeamAnalyse | 通过 | - | - |  |
| UI5 | testTeamAnalyseByName | 通过 | - | - |  |
| UI6 | testTeamuiInfo | 通过 | - | - |  |
| UI7 | testPlayeruiSelection | 通过 | - | - |  |
| UI8 | testPlayeruiInfo | 通过 | - | - |  |
| UI9 | testPlayeruiRank | 通过 | - | - |  |
| UI10 | testTeamuiRank | 通过 | - | - |  |
| UI11 | testTeamuiSelection | 通过 | - | - |  |
| BL1 | testPlayerAnalyse | 通过 | 得到球员列表 | 得到球员列表 |  |
| BL2 | testPlayerAnalyseByName | 通过 | 得到单个球员对象 | 得到单个球员对象 |  |
| BL3 | testPlayerByTeamAnalyse | 通过 | 得到球队中所有球员列表 | 得到球队中所有球员列表 |  |
| BL4 | testTeamAnalyse | 通过 | 得到球队列表 | 得到球队列表 |  |
| BL5 | testTeamAnalyseByName | 通过 | 得到单个球队对象 | 得到单个球队对象 |  |

## 功能测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试目的 | 测试步骤 | 预期结果 | 实际结果 | 状态 | 执行成本 |
| 测试系统能否正常打开 | 1. 点击运行系统 2. 等待系统响应 | 系统正常打开 | 系统正常打开 | Pass | 0.25人日 |
| 测试所有球队的选择跳转和球队信息显示是否正常 | 1. 分别点击所有球队图片 2. 系统响应点击并跳转到该球队详细信息面板 | 跳转匹配无误，球队信息显示正常 | 跳转匹配无误，球队信息显示正常 | Pass | 0.25人日 |
| 测试球队排名的跳转和显示是否正常 | 1. 点击球队排名 2. 系统响应点击并跳转到球队排名面板 | 跳转及球队排名信息显示正常 | 跳转及球队排名信息显示正常 | Pass | 0.25人日 |
| 测试球员选择和球员个人信息跳转和显示是否正常 | 1. 点击球员选择 2. 系统响应点击并跳转到所有球员选择面板 3. 随机点击50个球员名字 4. 系统响应点击并跳转到对应球员详细信息面板 | 球员选择列表显示正常，球员个人信息跳转和显示正常 | 球员选择列表显示正常，球员个人信息跳转和显示正常 | Pass | 0.5人日 |
| 测试球员筛选排名跳转和显示是正常 | 1. 点击球员排名筛选 2. 系统响应点击并跳转到球员筛选面板 | 跳转及球员显示正常 | 跳转及球员显示正常 | Pass | 0.25人日 |
| 测试所有表格升降序操作是否正常 | 1. 分别点击所有表格的所有表头进行升降序操作 2. 系统响应点击并显示升降序操作后的表格信息 | 所有表格升降序操作正常 | 所有表格升降序操作正常 | Pass | 0.5人日 |
| 测试球员筛选功能是否正常 | 1. 分别点击单个筛选条件和组合筛选条件复选框 2. 系统响应点击并显示筛选后的表格 | 筛选功能正常 | 筛选功能正常 | Pass | 0.25人日 |

## 性能测试

Table 示例数据时间-接口(getPlayerAnalysis)访问次数表

|  |  |
| --- | --- |
| 接口访问次数 | 耗时（5次测试耗时最大值上取整）/s |
| 10 | 0.07 |
| 100 | 0.10 |
| 200 | 0.11 |
| 400 | 0.19 |
| 800 | 0.28 |
| 1600 | 0.50 |
| 3200 | 0.90 |
| 6400 | 1.90 |

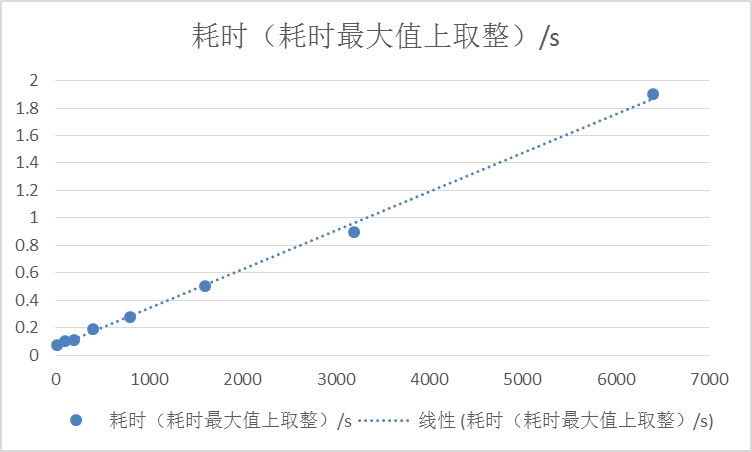


Figure 上表的线性回归分析

上述测试说明在频繁的界面跳转或之后设计为CS模式产生的多用户访问，时间复杂度保持在O(n),,满足《需求规格说明书》中的性能需求。

## 缺陷报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 缺陷编号 | 错误或问题描述 | 严重等级 | 发现者 | 解决者 | 错误或问题状态 |
| UI1 | 球员身高和出生日期显示错误 | 低 | 陈建伟 | 刘瀚文 | 已解决 |
| UI2 | 命中率为0 | 中 | 陈建伟 | 刘瀚文 | 已解决 |
| UI3 | 胜率为0 | 中 | 陈建伟 | 刘瀚文 | 已解决 |
| UI4 | 访问球员出现空指针 | 高 | 陈建伟 | 刘瀚文 | 已解决 |
| BL7 | 计算抢断率公式书写错误 | 中 | 刘瀚文 | 刘瀚文 | 已解决 |
| 功能测试 | 球员提升率数据计算错误 | 中 | 刘瀚文 | 刘瀚文 | 已解决 |
| 功能测试 | 球队名称变更在动态数据时产生错误 | 高 | 王俊超 | 王俊超 | 已解决 |
| 功能测试 | 比较器异常 | 高 | 梅杰 | 梅杰、刘瀚文 | 已解决 |
| 功能测试 | 球员位置类型错误 | 低 | 陈建伟 | 陈建伟 | 已解决 |
| 功能测试 | 界面排序错误 | 高 | 梅杰 | 梅杰 | 已解决 |
| 功能测试 | 界面数组越界 | 高 | 梅杰 | 陈建伟 | 已解决 |

## 测试结果分析

### 覆盖分析

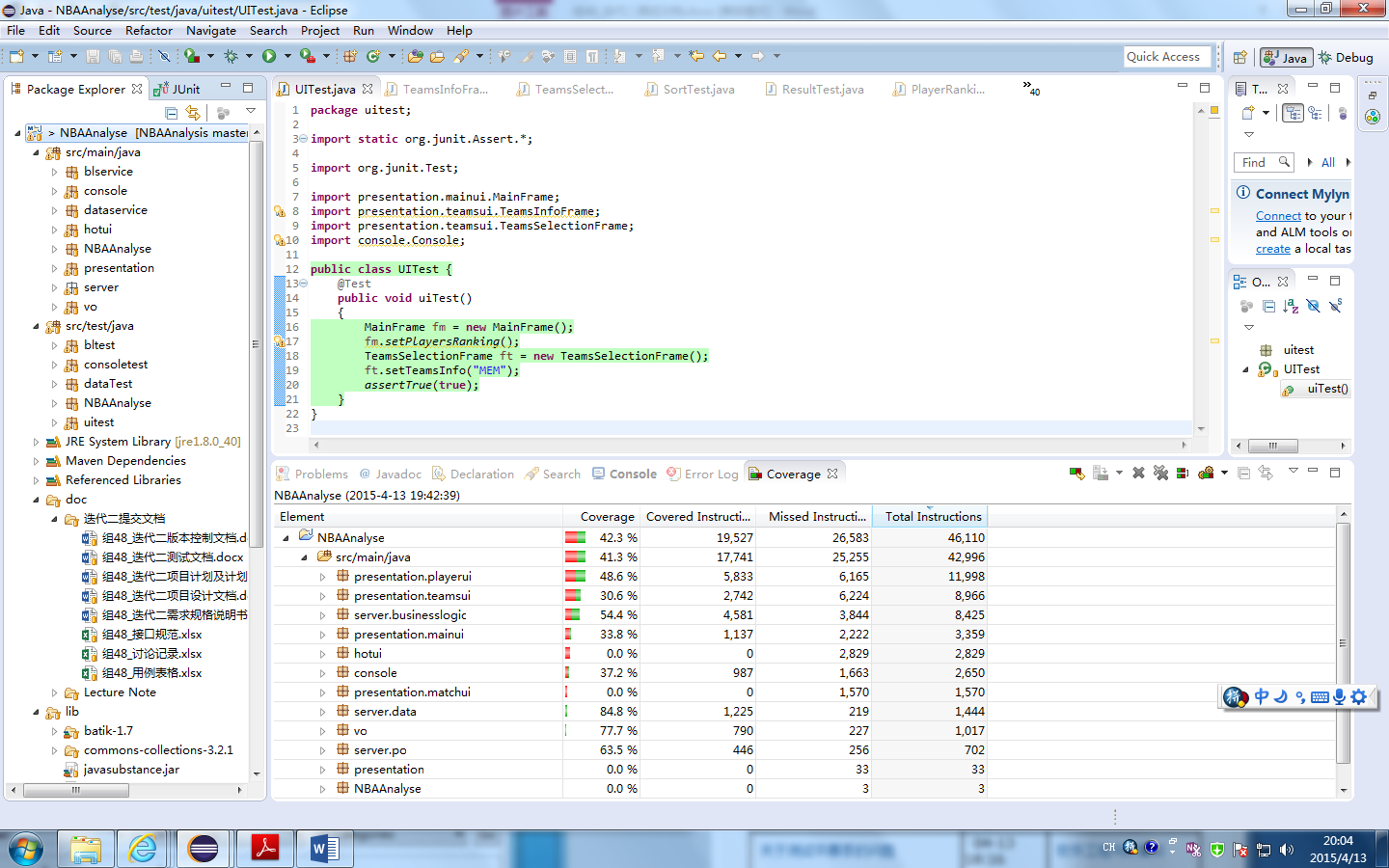


Figure Emma插件的分支覆盖度分析

由上方截图得，缺陷集中的数据层与逻辑层分支覆盖率在有大量的比较器和界面美化代码时达到50%以上，达到预期测试目的。

### 缺陷分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 严重级别  需求 | A-严重影响系统运行的错误 | B-功能缺陷，影响系统运行 | C-不影响运行但必须修改 | D-合理的建议 | 总数 |
|  | 5 | 1 | 5 | 0 | 11 |

# 综合评价

组48的NAB数据分析系统已达到交付标准。该系统能够实现用户需求说明书上的功能。