基于Scrapy的WebUI开发

——测试结果分析报告

SY1906514 汪凌风 SY1906508 明　昊

SY1906513 汪丽萍 SY1906431 郑泽西

SY1806220 闫奕涛 SY1906509 邵志钧

2020/05/16

# 版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 修改说明 | 参与者 | 审核人 |
| 1.0.0 | 2020.05.16 | 完成第一版 | 全体成员 | 汪凌风 |
| 1.0.1 | 2020.05.24 | 增加管理脚本的白盒测试表格 | 明昊 | 闫奕涛 |

注：全体成员即汪凌风，邵志钧，明昊，汪丽萍，闫奕涛，郑泽西

目　录

[基于Scrapy的WebUI开发 1](#_Toc40799064)

[——测试结果分析报告 1](#_Toc40799065)

[版本记录 2](#_Toc40799066)

[一、测试概述 5](#_Toc40799067)

[1.1 文档说明 5](#_Toc40799068)

[1.2 编写目的 5](#_Toc40799069)

[1.3 测试范围 5](#_Toc40799070)

[1.4 参考资料 5](#_Toc40799071)

[二、测试计划 6](#_Toc40799072)

[2.1 测试方法 6](#_Toc40799073)

[2.2 测试工具 6](#_Toc40799074)

[2.3 测试分工 6](#_Toc40799075)

[2.4 测试环境 7](#_Toc40799076)

[三、功能需求测试结果 8](#_Toc40799077)

[3.1白盒测试过程 8](#_Toc40799078)

[3.1.1 管理脚本模块 8](#_Toc40799079)

[3.1.2 自动化生成脚本模块 10](#_Toc40799080)

[3.2.1 脚本管理测试 11](#_Toc40799081)

[3.2.2 脚本生成测试 12](#_Toc40799082)

[3.2.3 词云生成测试 13](#_Toc40799083)

[3.2.4 选择优化测试 14](#_Toc40799084)

[3.2.5 去重优化测试 16](#_Toc40799085)

[3.2.6 集成测试 17](#_Toc40799086)

[四、非功能需求测试结果 19](#_Toc40799087)

[4.1 可用性测试 19](#_Toc40799088)

[4.2 稳定性测试 19](#_Toc40799089)

[4.3 交互界面测试 20](#_Toc40799090)

[4.4 高定制化测试 20](#_Toc40799091)

[五、测试报告结论 22](#_Toc40799092)

[5.1 综合测试结果 22](#_Toc40799093)

[5.2 现存缺陷 22](#_Toc40799094)

# 一、测试概述

## 1.1 文档说明

　　本文档针对本组在Scrapy框架的基础上进行扩展和改进实现的WebUI界面的测试结果分析报告。文档内容主要包括软件主体功能的白盒测试和黑盒测试、部分非功能测试的测试结果，以及对每个部分测试结果的分析。

## 1.2 编写目的

本文档的编写目的在于检查模块程序内部逻辑结构与运行情况和系统功能的正确性与容错性，进而总结测试阶段自我评审的测试情况以及分析测试结果，描述系统是否符合需求文档，是否已经达到预期的功能目标和非功能目标。

## 1.3 测试范围

测试主要依据《20B\_基于Scrapy的WebUI开发\_项目需求规格说明书V2.5.0》中的需求用例和《20B\_基于Scrapy的WebUI开发\_测试需求规格说明书1.1.0》中的测试用例进行测试。

## 1.4 参考资料

1. 邹欣.构建之法[M].北京.人民邮电出版社,2015:264-296
2. 单元测试[EB/OL].<https://www.liaoxuefeng.com/wiki/897692888725344/953546675792640>
3. 软件工程测试方法[EB/OL]. <https://blog.csdn.net/fightfaith/article/details/47858101>

# 二、测试计划

## 2.1 测试方法

　　对于功能测试，我们选择同时应用黑盒测试与白盒测试的方式进行检查，白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试；黑盒测试又称为功能测试，主要检测软件的每一个功能是否能够正常使用。在测试过程中，将程序看成不能打开的黑盒子，不考虑程序内部结构和特性的基础上通过程序接口进行测试，检查程序功能是否按照设计需求以及说明书的规定能够正常打开使用。我们通过结合两者的方式来检查模块程序内部逻辑结构与运行情况和系统功能的正确性与容错性。

## 2.2 测试工具

对于黑盒测试，测试过程的代码撰写和用例测试主要通过Python的unittest包来实现，模块的一些测试结果通过Pycharm的coverage功能进行分析，对于白盒测试，因为本项目将部署在云端服务器上，测试时通过URL即可访问基于Scrapy的WebUI页面，所以选择在页面上人工按照测试需求和用例完成全部测试工作。

## 2.3 测试分工

|  |  |  |
| --- | --- | --- |
| **成员** | **主要职责** | **测试用例** |
| 明昊 | 完成管理脚本功能需求测试，并对WebUI进行总体设计和完善 | 管理脚本 |
| 汪丽萍 | 完成生成脚本功能需求测试，并撰写对应测试文档 | 生成脚本 |
| 郑泽西 | 完成对应功能需求测试，并撰写对应测试文档 | 暂无 |
| 闫奕涛 | 完成生成词云功能需求测试，负责测试需求规格说明书的整体框架，并撰写对应测试文档 | 生成词云 |
| 邵志钧 | 完成选择优化功能需求测试，并撰写对应测试文档 | 选择优化 |
| 汪凌风 | 完成去重优化功能需求测试，并撰写对应测试文档 | 去重优化 |

## 2.4 测试环境

Chrome 7.0版本以上，Windows自带edge浏览器等。

# 三、功能需求测试结果

按照第二章测试方法和工具的介绍，对于功能需求的测试，我们选择同时应用黑盒测试与白盒测试的方式进行检查，通过Python的unittest包和Pycharm的coverage功能来实现黑盒测试过程的用例测试，通过人工访问基于Scrapy的WebUI页面来完成部分功能的白盒测试。

## 3.1白盒测试过程

对于白盒测试过程，我们通过Python的unittest包和Pycharm的coverage功能来实现部分模块的测试，因为脚本生成和选择优化功能集成在一起，且API何在一起，所以这两个模块的单元测试一起进行,测试过程按照《20B\_基于Scrapy的WebUI开发\_测试需求规格说明书1.1.0》中的测试用例进行测试。

### 3.1.1 管理脚本模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **名称** | 管理脚本模块单元测试 | | | | | |
| **模块功能** | 管理脚本的保存和运行 | | | | | |
| **模块优先级** | 高 | | | | | |
| **测试目的** | 测试API函数合法的输入参数及正确的调用顺序 | | | | | |
| **测试描述** | 在每一个模块测试中，调用所有的API接口函数。对API函数参数输入合法参数值，并且以正确的顺序调用，打印输出函数返回值。 | | | | | |
| **前提和约束** | Redis与Scrapyd正常运行 | | | | | |
| **测试用例** | **序号** | **输入数据** | **预期输出** | **实际输出** | **是否通过** | **备注** |
| 1 | File: ../DistributedSpider-master.zip,  Project:”,  Spider: Dis\_Douban | need project name | need project name | 通过 | 上传项目 |
| 2 | File: ../DistributedSpider-master.zip,  Project: DistributedSpiders,  Spider: “ | need spider name | need spider name | 通过 | 上传项目 |
| 3 | File: ../test.txt,  Project: DistributedSpiders,  Spider: Dis\_Douban | no need file | no need file | 通过 | 上传项目 |
| 4 | File: ../DistributedSpider-master.zip,  Project: DistributedSpiders,  Spider: Dis\_Douban | success | success | 通过 | 上传项目 |
| 5 | Name: DistributedSpiders, | success | success | 通过 | 运行项目 |
| 6 | Name: DistributedSpiders, | such project is running or pending | such project is running or pending | 通过 | 运行项目 |
| 7 | Name：dis | no such project exists | no such project exists | 通过 | 运行项目 |
| 8 | Name: DistributedSpiders | Fail,  such project is running or pending | Fail,  such project is running or pending | 通过 | 删除项目 |
| 9 | Name: DistributedSpiders | Success | success | 通过 | 停止项目 |
|  | 10 | Name：etstdemo | such project not exist | such project not exist | 通过 | 停止项目 |
|  | 11 | Name: DistributedSpiders | Success | success | 通过 | 删除项目 |
|  | 12 | 无 | False | False | 通过 | 查看所有项目 |
| **覆盖率** | C:\Users\minghao\AppData\Local\Temp\WeChat Files\0283bad1dfe745d939f01af2110dffd.png | | | | | |
| **备注** | 针对脚本管理整个文件的单元测试，测试用例必须顺序运行，且后继测试用例需要使用前趋测试用例创建的环境，所以合并在一个里面，覆盖率72%的原因：1. 由于查看输出和下载输出相关功能与服务器环境有关，在本地环境下测试困难，改为黑盒测试 2.API代码内有的分支只是为了鲁棒性设置，但是触发非常困难，所有没有选择覆盖这部分的分支 | | | | | |

### 3.1.2 自动化生成脚本模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **名称** | 自动化生成脚本模块单元测试 | | | | | |
| **模块功能** | 自动生成爬虫脚本并压缩项目 | | | | | |
| **模块优先级** | 无 | | | | | |
| **测试目的** | 测试API函数合法的输入参数及正确的调用顺序 | | | | | |
| **测试描述** | 在每一个模块测试中，调用所有的API接口函数。对API函数参数输入合法参数值，并且以正确的顺序调用，打印输出函数返回值。 | | | | | |
| **前提和约束** | 无 | | | | | |
| **测试用例** | **序号** | **输入数据** | **预期输出** | **实际输出** | **是否通过** | **备注** |
| 1 | test.json文件 | success | success | 通过 | 无 |
| 2 | 无 | “name”:”demo.zip” | “name”:”demo.zip” | 通过 | 本API为get方法，因此无输入 |
| **覆盖率** |  | | | | | |
| **备注** | 因为该部分的自动生成脚本代码主要为被调用函数，其中包含大量字符串，因此对覆盖率有一定影响；另一方面，由于下载脚本功能在服务器环境和本机环境存在较大差异，因此单元测试中没有对该功能进行测试。 | | | | | |
| **测试人员** | 邵志钧 | | | | | |
| **测试时间** | **2020年5月17日** | | | | | |

## 3.1黑盒测试过程

### 3.2.1 脚本管理测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 运行脚本 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户是否能运行脚本 | |
| **前提和约束** | 用户打开WebUI | |
| **测试步骤** | **1** | 用户点击选择脚本上传项目与必要参数 |
| **2** | 用户选中项目点击运行 |
| **预期结果** | 脚本状态变为运行 |
| **测试结果** | 通过 | |
| **测试人员** | 明昊 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 查看输出 | |
| **简要描述** | 本测试验证用户是否能查看日志输出 | |
| **前提和约束** | 用户打开WebUI | |
| **测试方法** | 黑盒测试 | |
| **测试步骤** | **1** | 用户选择脚本 |
| **2** | 用户点击查看输出 |
| **3** | 界面弹出小框显示输出 |
| **预期结果** | 在弹出小框内正常显示日志输出 |
| **测试结果** | 通过 | |
| **测试人员** | 明昊 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 下载输出 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户是否能下载输出 | |
| **前提和约束** | 用户选择脚本查看输出 | |
| **测试步骤** | **1** | 用户点击下载输出按钮 |
| **预期结果** | 浏览器开始下载文件 |
| **测试结果** | 通过 | |
| **测试人员** | 明昊 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 管理记录 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证系统是否能够保存用户的过往爬取信息 | |
| **前提和约束** | 用户在WebUI内运行过项目 | |
| **测试步骤** | **1** | 用户重新打开WebUI |
| **预期结果** | 用户可以看到自己之前上传的项目并且查看输出 |
| **测试结果** | 通过 | |
| **测试人员** | 明昊 | |
| **测试时间** | **2020年5月17日** | |

### 3.2.2 脚本生成测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 自动生成scrapy项目压缩文件测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户是否能通过提交表单相应信息自动生成scrapy项目压缩文件 | |
| **前提和约束** | 用户运行系统 | |
| **测试步骤** | **1** | 选择“自动生成”子功能 |
| **2** | 填写项目名，网址，item和对应xpath等表单内容 |
| **3** | 选择“提交”按钮进行提交 |
| **预期结果** | 表格显示自动生成的scrapy项目压缩文件的文件名 |
| **测试结果** | 通过 | |
| **测试人员** | 汪丽萍 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 下载scrapy项目压缩文件测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户是否能下载自动生成的scrapy项目压缩文件 | |
| **前提和约束** | 用户自动生成的scrapy项目压缩文件名显示在表格上 | |
| **测试步骤** | **1** | 点击需要下载的scrapy项目压缩文件的“下载”按钮 |
| **预期结果** | 浏览器下载相应scrapy项目压缩文件 |
| **测试结果** | 通过 | |
| **测试人员** | 汪丽萍 | |
| **测试时间** | **2020年5月17日** | |

### 3.2.3 词云生成测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 生成词云 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 用户在选择框中选取URL，该用例能够成功生成词云图 | |
| **前提和约束** | 用户进入生成词云页面 | |
| **测试步骤** | **1** | 在选择框中选取URL |
| **2** | 点击“生成词云” |
| **3** | 运行脚本 |
| **预期结果** | 脚本正确运行并成功生成词云 |
| **测试结果** | 通过 | |
| **测试人员** | 闫奕涛 | |
| **测试时间** | **2020年5月17日** | |

### 3.2.4 选择优化测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 随机代理IP测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户选择随机代理IP优化选项后，能否生成正确的脚本并正确运行 | |
| **前提和约束** | 用户进入自动化生成脚本页面 | |
| **测试步骤** | **1** | 选择随机代理IP后设置优化选项“随机代理IP”优化选项并点击生成脚本 |
| **2** | 检查脚本内容 |
| **3** | 运行脚本 |
| **预期结果** | 脚本正确运行并使用代理IP进行爬取 |
| **测试结果** | 通过 | |
| **测试人员** | 邵志钧 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 随机User-Agent测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户选择随机User-Agent优化选项后，能否生成正确的脚本并正确运行 | |
| **前提和约束** | 用户进入自动化生成脚本页面 | |
| **测试步骤** | **1** | 选择“随机User-Agent”优化选项并点击生成脚本 |
| **2** | 检查脚本内容 |
| **3** | 运行脚本 |
| **预期结果** | 脚本正确运行并使用随机User-Agent进行爬取 |
| **测试结果** | 通过 | |
| **测试人员** | 邵志钧 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 爬取间隔测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户设置爬取间隔后，能否生成正确的脚本并正确运行 | |
| **前提和约束** | 用户进入自动化生成脚本页面 | |
| **测试步骤** | **1** | 设置优化选项“页面爬取间隔”参数并点击生成脚本 |
| **2** | 检查脚本内容 |
| **3** | 运行脚本 |
| **预期结果** | 脚本正确运行并使用设定页面爬取间隔进行爬取 |
| **测试结果** | 通过 | |
| **测试人员** | 邵志钧 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | Cookie测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户设置是否携带Cookie后，能否生成正确的脚本并正确运行 | |
| **前提和约束** | 用户进入自动化生成脚本页面 | |
| **测试步骤** | **1** | 设置优化选项“是否携带Cookie”并点击生成脚本 |
| **2** | 检查脚本内容 |
| **3** | 运行脚本 |
| **预期结果** | 脚本正确运行并按要求携带Cookie进行爬取 |
| **测试结果** | 通过 | |
| **测试人员** | 邵志钧 | |
| **测试时间** | **2020年5月17日** | |

### 3.2.5 去重优化测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 选择去重算法优化 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户选择去重优化后能否正常运行以及内存消耗是否降低 | |
| **前提和约束** | 用户运行一个能够正确运行的爬虫项目 | |
| **测试步骤** | **1** | 点击选择去重算法优化 |
| **2** | 点击确定运行优化后的项目 |
| **预期结果** | 爬虫项目与为选择优化时最终结果一致，且在百万级URL数目的项目中，服务器内存占用比原先降低。 |
| **测试结果** | 通过 | |
| **测试人员** | 汪凌风 | |
| **测试时间** | **2020年5月17日** | |

### 3.2.6 集成测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 用户查看说明 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 用户查看本网站的使用说明 | |
| **前提和约束** | 用户能够正确打开本网站 | |
| **测试步骤** | **1** | 打开网页 |
| **2** | 打开其他页面，再点击右上角的Instructions |
| **预期结果** | 初始页面显示网站简介和使用说明，显示使用说明 |
| **测试结果** | 通过 | |
| **测试人员** | 汪凌风 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 选择所有优化选项进行自动生成 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 选择所有优化选项，查看是否能在网站上正确运行 | |
| **前提和约束** | 用户能正确打开本网站 | |
| **测试步骤** | **1** | 点击自动生成脚本，任意合法填写URL等项目，勾选所有优化选项并点击创建脚本。 |
| **2** | 刷新后下载刚创建的脚本 |
| **3** | 在脚本管理界面上传刚才下载的zip文件，正确填写项目名和脚本名 |
| **4** | 选中刚才上传的脚本，点击运行 |
| **预期结果** | 成功创建脚本，成功下载，成功上传脚本，刷新后在列表显示，显示运行成功提示 |
| **测试结果** | 通过 | |
| **测试人员** | 汪凌风 | |
| **测试时间** | **2020年5月17日** | |

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 删除脚本测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 查看是否能正确删除脚本 | |
| **前提和约束** | 用户能正确打开本网站，脚本管理列表中存在脚本 | |
| **测试步骤** | **1** | 选中一个脚本，点击删除 |
| **2** | 上传同名项目 |
| **预期结果** | 删除后刷新后脚本消失，能够正常上传 |
| **测试结果** | 通过 | |
| **测试人员** | 汪凌风 | |
| **测试时间** | **2020年5月17日** | |

# 四、非功能需求测试结果

## 4.1 可用性测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 可用性测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证应用运行时，管理脚本、自动化生成脚本、生成词云、在线编辑调试和选择优化四大主要功能是否能正常运行 | |
| **前提和约束** | 用户打开ScrapyWebUI网址 | |
| **测试步骤** | **1** | 用户进行各项功能的使用 |
| **2** | 测试各项功能是否正常运行，是否存在功能失效的情况 |
| **预期结果** | 各项功能正常运行，不存在功能失效的情况 |
| **测试结果** | 通过 | |
| **测试人员** | 郑泽西 | |
| **测试时间** | **2020年5月17日** | |

## 4.2 稳定性测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 稳定性测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证从用户打开应用开始，2小时以内可以稳定运行无故障，出故障时，能够发出故障信息 | |
| **前提和约束** | 用户打开ScrapyWebUI网址 | |
| **测试步骤** | **1** | 用户进行各项功能的使用 |
| **2** | 测试从用户打开网址开始，是否可以2小时以内稳定运行无故障，若出现故障，是否能够发出故障信息 |
| **预期结果** | 从用户打开网址开始，2小时以内可以稳定运行无故障，出现故障时，能够发出故障信息 |
| **测试结果** | 通过 | |
| **测试人员** | 闫奕涛 | |
| **测试时间** | **2020年5月17日** | |

## 4.3 交互界面测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 交互界面用户友好性测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证应用运行时，其实用过程是否有较为全面的用户使用说明、提示及引导 | |
| **前提和约束** | 用户启动本应用 | |
| **测试步骤** | **1** | 用户启动应用 |
| **2** | 用户进行各项功能的使用 |
| **3** | 检查各项功能的用户引导是否完整 |
| **预期结果** | 用户提示及引导完善且正常显示 |
| **测试结果** | 通过 | |
| **测试人员** | 汪凌风 | |
| **测试时间** | **2020年5月17日** | |

## 4.4 高定制化测试

|  |  |  |
| --- | --- | --- |
| Test Case Specification | | |
| **名称** | 高定制化测试 | |
| **测试方法** | 黑盒测试 | |
| **简要描述** | 本测试验证用户可以使用本系统是否可以实现高自由度的爬虫定制 | |
| **前提和约束** | 用户进入自动化生成脚本页面 | |
| **测试步骤** | **1** | 用户进入自动化生成脚本页面 |
| **2** | 用户对爬虫进行定制 |
| **3** | 用户使用定制爬虫可以爬取大部分不使用动态JS编写的网站 |
| **预期结果** | 爬虫脚本运行正常。 |
| **测试结果** | 通过 | |
| **测试人员** | 汪丽萍 | |
| **测试时间** | **2020年5月17日** | |

# 五、测试报告结论

## 5.1 综合测试结果

在测试过程中，项目小组的成员根据项目规格说明书的要求，制定合理的测试计划，撰写了覆盖所有需求并与之一一对应的合理的测试用例， 并依据测试需求规格说明书进行测试。

通过对《20B\_基于Scrapy的WebUI开发\_测试需求规格说明书1.1.0》中的测试用例进行测试，并对结果进行分析，可以看出对功能测试的白盒测试部分，管理脚本模块和自动化生成脚本模块的程序内部逻辑结构与运行情况正常，符合开发人员的预期，对于黑盒测试部分，每个模块的功能运行正常，能正确实现需求规格说明书里提出的需求。而对非功能需求测试结果的分析可以看出该软件的可用性、稳定性、交互界面和高定制化的非功能需求已经得到了满足。经过总体的测试结果进行综合分析，本组在Scrapy框架的基础上进行扩展和改进实现的WebUI界面的实现覆盖了项目需求规格说明书里的所有需求并达到其中规定的要求，其中在测试过程中发现的部分缺陷在下一小节提出，后续会继续改进。

## 5.2 现存缺陷

脚本管理：项目上传成功之后脚本运行出错没有有效返回信息。

　　脚本生成：1.项目名不能重名，2.由于没有数据库和用户登录功能，返回给用户的项目是所有用户创建的项目文件。

　　生成词云：本计划是由用户自主输入URL即可爬取数据，但在调研过程中发现，现存的所有成熟的词云工具网站都不支持此功能，并且实际实现时面临的不确定性过多，所以改成了由系统提供固定的几个URL选项。