

# IRBL迭代二：需求规格说明文档

更新日志：

时间	更新理由	责任人
2021.4.1	创建文档	韩禧
2021.4.5	完善总体描述	韩禧
2021.4.7	完善详细需求描述	韩禧

## 1. 引言

### 1.1 目的

本文档描述了 **IRBL迭代二：软件项目计划书**的功能需求和非功能性需求。开发小组的软件系统实现和验证工作都以此文档为依据。

除特殊说明外，本文档包含的需求都是高优先级的需求。

### 1.2 范围

**IRBL-基于信息检索的缺陷定位系统** 是软件工程与计算III开发小组制作的软件缺陷定位系统，开发目的是为了帮助用户实现bug的辅助定位。功能包括用户友好的terminal，文件的读取与写入功能，文件的预处理功能，word count功能，tf-idf功能，相似bug计算功能，打印结果功能。

### 1.3 参考文献

- 1. IEEE标准
- 2. IRBL 软件项目计划书
- 3. 《软件工程与计算（卷三）》

## 2. 总体描述

### 2.1 项目前景

随着软件工程的发展，对系统缺陷的定位逐渐复杂，也愈发重要。缺陷定位是最常见的开发实践之一，缺陷的存在会影响软件系统的正常使用甚至带来重大危害。本项目将使用基于信息缺陷的定位手

段，将缺陷定位转化为以bug报告为查询，代码文件为搜索对象的搜索问题，为每个缺陷报告生成一份按照程序实体与缺陷相关度降序排序的列表，提升软件开发的效率。该项目在研究与工程上都具有一定的价值，继续精细化可以得到更好的测试数据，如果推动落地则可以带来不错的商业价值。

## 2.2 项目功能

- （一）terminal：有用户友善、稳定性高的命令行系统
- （二）预处理：对bugReport文件以及代码文件进行预处理
- （三）文件的读取与写入
- （四）word count计算及权重计算
- （五）TF-IDF计算
- （六）相似bug计算
- （七）相似度计算
- （七）结果打印

## 2.3 用户特征

用户	特征
企业开发人员	具有熟练的编程技能与大量开发经验，但需要应对复杂系统以及不符合设计目的的变更。工作中会遇到难以定位、难以解决的bug，为此花费大量时间。
编程学习者	在编程学习中会时不时遇到难以解决的bug，编程能力弱，对稍大一些的系统掌控力不足。

## 2.4 约束

- **CON1:** 系统将运行在服务器中，可以通过网页浏览器访问。
- **CON2:** 系统使用terminal界面，会在第三阶段开发简单的web界面。
- **CON3:** 项目要使用螺旋模型进行开发。
- **CON4:** 在开发中，开发人员要提交软件项目计划书、软件概要设计文档、软件需求规格说明文档、软件详细设计描述文档以及测试报告。
- **CON5:** 本项目将使用Java和Python作为后端开发语言，辅以Spring框架。
- **CON6:** 本项目将部署在 CentOS 7.6 服务器中，并使用 Jenkins 作为持续集成工具。

## 2.5 假设与依赖

- 1. 服务器不会被恶意攻击，且解析的代码不会给系统注入恶意信息。
- 2. 超出本系统范围产生的异常（例如：数据库服务器异常、所用工具本身缺陷等第三方错误）。
- 3. 我们使用的算法没有统计学意义上的数据偏好性（即无特定数据情况下的过拟合现象）。
- 4. 所用设备稳定，不会在使用过程中发生蓝屏、系统自动重启、设备失灵等故障。

## 3 详细需求描述

### 3.1 对外接口需求

#### 3.1.1 命令行界面

采用命令行进行用户输入，以及获得程序反馈。

### 3.2 功能需求

#### 3.2.1 读取用户指令

##### 3.2.1.1 特征描述

通过命令行读入用户输入的指令。

##### 3.2.1.2 刺激/响应序列

刺激：用户输入指令

响应：系统读入指令

##### 3.2.1.3 相关功能需求

Terminal.GetCommand	系统获取用户输入的指令
Terminal.Response	系统对用户的输入给予反馈

#### 3.2.2 解析用户指令

##### 3.2.2.1 特征描述

通过对用户指令的解析，执行相应的操作。

##### 3.2.2.2 刺激/响应序列

刺激：用户指令进入指令解析模块

响应：系统对指令进行解析，并执行相应的操作

##### 3.2.2.3 相关功能需求

Parse.Process	系统处理用户输入的指令
Parse.Response	系统对用户的输入给予反馈

3.2.3 命令行操作帮助

3.2.3.1 特征描述

在命令行中打印为用户正确输入指令准备的提示信息。

3.2.3.2 刺激/响应序列

刺激：用户输入错误指令

响应：系统打印错误提示，并引导用户查看命令行操作帮助

刺激：用户输入 /h ，查看操作帮助

响应：系统打印

3.2.3.3 相关功能需求

Terminal.Warning	系统反馈用户输入的错误指令
Terminal.Help	系统打印命令行操作帮助

3.2.4 读取文件

3.2.4.1 特征描述

读取特定路径下的文件

3.2.4.2 刺激/响应序列

刺激：用户将需要处理的文件放在特定路径下，并启动读取程序相关的功能

响应：系统读取特定路径下的文件

3.2.4.3 相关功能需求

FileReader.Read	读取特定路径下的文件
-----------------	------------

3.2.5 写入文件

3.2.5.1 特征描述

在特定路径下创建并写入文件，如果已有文件就进行覆盖。

3.2.5.2 刺激/响应序列

刺激：用户启动写入程序相关的功能

响应：系统进行写入文件操作

3.2.5.3 相关功能需求

Writer.Write	在特定路径下写入文件
--------------	------------

3.2.6 文件预处理

3.2.6.1 特征描述

对项目文件进行预处理，去掉特殊词以及限定词。

3.2.6.2 刺激/响应序列

刺激：用户输入预处理指令

响应：系统执行预处理操作

3.2.6.3 相关功能需求

File.Pretreatment	对读取到的文件进行预处理，并记录结果
-------------------	--------------------

3.2.7 TF-IDF

3.2.7.1 特征描述

通过TF-IDF计算获得项目中的主题词

3.2.7.2 刺激/响应序列

刺激：用户输入TF-IDF执行指令

响应：系统执行TF-IDF算法

3.2.7.3 相关功能需求

FileReader.ReadJson	读取记录相似度计算结果的json文件
FileWriter.WriteJson	将计算结果写入JSON文件

Algorithm.Tfidf	使用TF-IDF算法提取主题词
-----------------	-----------------

### 3.2.8 Word Count

#### 3.2.8.1 特征描述

通过对预处理后的整个项目中代码文件的分析，获取word count

#### 3.2.8.2 刺激/响应序列

刺激：用户输入word count执行指令

响应：系统执行相应算法，计算出项目中的word count值

#### 3.2.8.3 相关功能需求

FileReader.ReadJson	读取记录相似度计算结果的json文件
FileWriter.WriteJson	将计算结果写入JSON文件
Algorithm.WordCount	使用word count算法进行词数统计

### 3.2.9 打印结果

#### 3.2.9.1 特征描述

在控制台打印出相似度计算的结果，为用户展现处理的结论作为缺陷定位依据

#### 3.2.9.2 刺激/响应序列

刺激：用户输入打印结果的指令

响应：系统打印相似度计算的结果

#### 3.2.9.3 相关功能需求

FileReader.ReadJson	读取记录相似度计算结果的json文件
Algorithm.PrintResult	在控制台打印出相似度计算的结果

### 3.2.10 相似bug计算

#### 3.2.10.1 特征描述

对已修复的bug的报告进行向量化处理，计算相似度并映射到对应的源码文件，得出源码文件的相似度得分，并将结果存储为JSON数据

3.2.10.2 刺激/响应序列

刺激：用户输入计算相似bug的指令

响应：系统运行相似bug计算模块，并将计算结果保存

3.2.10.3 相关功能需求

FileReader.ReadJson	读取记录相似度计算结果的json文件
FileWriter.WriteJson	将计算结果写入JSON文件
Algorithm.BugCalcular	对bug报告进行相似度计算

3.2.11 RepportTagger

3.2.10.1 特征描述

对报告进行预处理的过程中，对不同内容解析为不同标记，有NL，SC，ST三种类型，结果存储为JSON数据

3.2.10.2 刺激/响应序列

刺激：用户输入运行RepportTagger的指令

响应：系统运行RepportTagger模块，并将处理结果保存

3.2.10.3 相关功能需求

FileReader.ReadJson	读取记录相似度计算结果的json文件
FileWriter.WriteJson	将计算结果写入JSON文件
Algorithm.ReportTagger	对bug报告进行标记预处理

3.3 非功能需求

3.3.1 安全性

- **Safety1:** 系统只应对Java项目文件进行处理。
- **Safety2:** 系统处理的文件应小于1G。

### 3.3.2 可维护性

- **Modifiability1**: 系统的Java部分使用面向对象方式进行开发，Python部分按功能分为不同模块，Java通过RunTime对Python方法进行调用。
- **Modifiability2**: 模块之间职责分离，便于后阶段的迭代更新。

### 3.3.3 易用性

- **Usability1**: 所有类型人员都不需要专门的培训，都可以在10 分钟内完成任一个操作任务。
- **Usability2**: 本项目完成部署后，在所有支持es6的web浏览器中均可正常使用。
- **Usability3**: terminal中可以查看操作提示（输入 /h 指令可以打印出输入提示）
- **Usability4**: terminal中输入不符合要求的指令时，系统会给出错误提示

### 3.3.4 可靠性

- **Reliability1**: 在运行计算时Java线程会进入等待，不接收新的指令。
- **Reliability2**: 程序的各步骤分离，如果运行失败可以明确错误发生点。
- **Reliability3**: 如果遇到网络问题，系统会给出相应提示。

### 3.3.5 约束

- **IC1**: 部署在CentOS服务器中，迭代一、迭代二采用terminal进行交互，迭代三使用简单的web前端页面进行交互。
- **IC2**: 只能解析规范的Java项目。

## 3.4 其他

### 3.4.1 安装需求

**Install1**: 部署系统时，要安装Java 1.8版本以及Python 3.8版本。

**Install2**: 用户使用该系统时，需要安装兼容ECMAScript 5的浏览器。

### 3.5.2 系统需求

运行系统目前只在Windows、Mac OS、CentOS上进行测试，其它版本暂不保证运行良好。