实验2 需求排序

1.小组成员

- 李舟俊潇 —— 181860051
- 宋超群 —— 171860642
- 童逸舟 —— 181860090
- 郑浩然 —— 181860147

2.小组成员任务及成绩分配

李舟俊潇 — 25%：

需求分类和筛选

宋超群 — 25%：

使用爬虫获取需求等级数据

童逸舟 — 25%：

需求分类和筛选

郑浩然 — 25%：

报告撰写

3.实验目的

`Eclipse`是一个知名的IDE，在这个项目中有丰富的缺陷报告数据。而在软件开发维护生命周期中，由于资源受限，软件团队往往无法同时满足所有需求的实现。因此需要对软件需求进行优先级排序。我们将`Eclipse`的缺陷报告当做一种“软件需求”，对其根据其既定的标签进行优先级排序，从而研究软件开发团队如何高效的处理这些需求。

4.实验方法

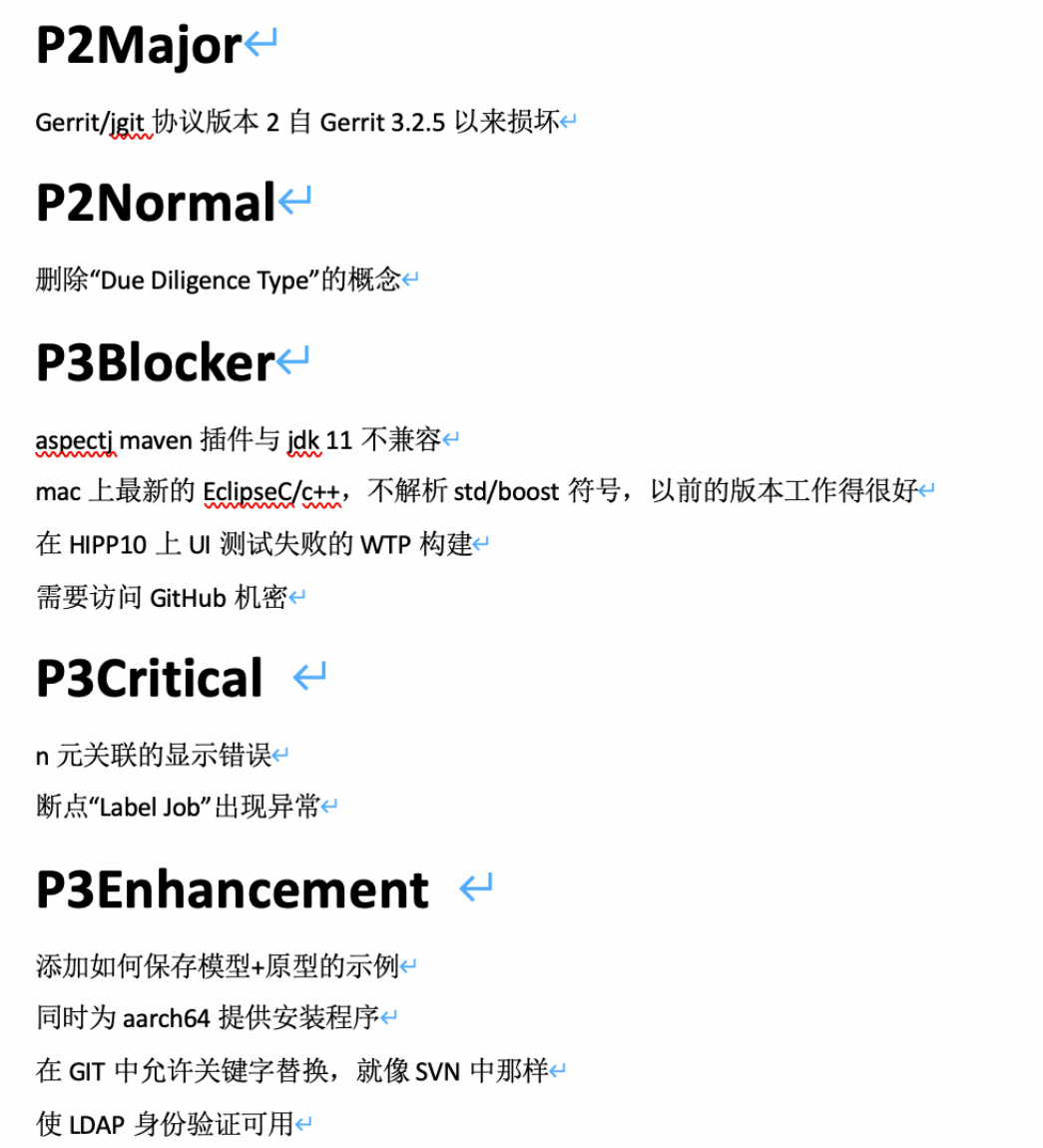
首先通过自己设计的爬虫程序对`Eclipse`的缺陷报告数据的内容和等级进行了爬取。之后由我们人工进行翻译后筛选数据，保留有效的需求条目，并对需求按照等级标签进行分类排序。最后分析并撰写报告

5.实验结果及效果分析

由于网络的原因，我们的爬虫程序总是异常中断，并且爬取速度非常缓慢，每次爬取都要花费大约24小时。最多的一次，我们也只爬取到了247条bug目录，进过去重和筛选之后（其中有非常多的重复需求和无效需求），我们最终保留了 $1+1+4+2+10+12+4+95=129$ 条需求。

对于这些通过分类和筛选的需求，我们按照标签将其分为了8类，按照优先级分别是：`P2Major`，`P2Normal`，`P3Blocker`，`P3Critical`，`P3Enhancement`，`P3Major`，`P3Minor`，`P3Normal`，他们的数量分别为 1，1，4，2，10，12，4，95。

这是我们需求分析的一部分截图，具体见相同目录下的`lab2需求排序.docx`具体分析：



在Bug的等级划分中，P1被定义为**系统无法执行、崩溃或严重资源不足、应用模块无法启动或异常退出、无法测试、造成系统不稳定**。经过我们的爬取，`Eclipse`中并不存在此等级的bug，可见其本身的系统质量并没有什么大问题。

P2被定义为**影响系统功能或操作，主要功能存在严重缺陷，但不会影响到系统稳定性**。在我们的数据中，`P2Major`类只有一条：`Gerrit/jgit协议版本2自Gerrit 3.2.5以来损坏`；`P2Normal`也有一条：`删除“Due Diligence Type”的概念`，相比于P3类bug的数量非常少，但都较为严重。

P3被定义为**界面、性能缺陷，次要功能存在部分错误**，在这一部分的bug数量相当多，有127条需求，可见大部分的bug都是一些次要功能或是界面方面的缺陷。`P3Blocker`是最严重的P3类bug，多是一些系统崩溃或无法执行的错误，比如`aspectj maven插件与jdk 11不兼容`、`mac上最新的EclipseC/c++，不解析std/boost符号`等次要功能的缺陷问题；`P3Critical`大多为一些不影响系统稳定的严重问题，在我们爬取的数据中有类似`断点“Label Job”出现异常`的问题。`P3Enhancement`是一类自定义等级的bug，通过分析发现其大多数是改进建议类的问题，涉及范围广泛，包含界面，编译器，关键字等等，类似于`在GIT中允许关键字替换，就像SVN中那样`。`P3Major`大多数是一些界面或是兼容性问题，类似于`APT类加载器在JDK中找不到类`，`Mac BigSur上的搜索字段大小不正确`，最近Mac更新出了`BigSur`，因此也出现了不少相关的新问题。最后就是`P3Minor`，`P3Normal`类的bug，这类bug数量非常多，多是一些易用性的建议，或是一些无伤大雅的bug，类似于`Java编辑器错误行号显示区和刷新行为`，`更改布局位置并重置为菜单项`。种类繁多，也不太好总结。

经过分析，发现bug多集中在编译器和界面方面，结合上一次实验对`vscode`的分析，这也是大多数编译器的通病。毕竟每个人的机器配置不同，审美也不太一样，因此调试起来非常麻烦。但总的来说，`Eclipse`是一款非常优秀的IDE，其bug管理机制非常高效，bug问题也大多可以接受。

结论

通过本次实验，我们了解了`Eclipse`如何管理自己的bug需求，同时也更加深刻的认识到了如何处理众多的需求。对于类似于`Eclipse`这样的大型软件，随时都会遇到用户提出的新的需求，这些需求当然是不能全都处理的，否则需求就会越积越多，需求的实现效率低下。因此对于需求进行等级排序是一项必须的工作。合理的对需求进行排序也是软件管理中非常重要的一部分，优先处理高优先级的需求，才能保证软件的正常运作。