

Eclipse 需求的抽取与分析

小组成员：201220141 王怡涵 201220145 马琳琳

成绩分配比例：5:5

一、实验目的

根据用户问题抽取 Eclipse 的用户需求并进行分类

二、实验方法

1. 确定开源 IDE 项目：

Eclipse

2. 获取需求信息：

数据信息来源于 Eclipse 的缺陷报告数据网 <https://bugs.eclipse.org/bugs/> 利用 Chrome 的 Web Scraper 插件从该网址中获取了 1000 条用户提交的有关该软件功能缺陷的信息。

3. 提取潜在需求：

通过 Excel 的检索功能提取高频词汇，人工分析词汇所对应的问题条目并根据这些问题分析用户的潜在需求。

4. 对需求进行分类：

汇总步骤 2 中的所有问题，人工提炼问题的共性需求，按照功能性需求和非功能性需求两个大类及非功能需求下的操作需求、性能需求和安全性需求三个小类对这些需求进行分类。

三、实验分析及结果

1. 用户使用过程中遇到的常见问题：

（其中部分需求仅截取了错误路径或代码，未阐明具体问题，此处未进行分析）

①根据关键词检索提炼得到的高频问题

- A. 用户操作导致程序崩溃或挂起（146 条）
- B. 使用插件或组件出现错误（57 条）
- C. 版本更新后导致运行错误或不兼容问题（47 条）
- D. 用户指南不准确或不完善（29 条）
- E. 复制、粘贴、拖拽等操作无效（28 条）
- F. 软件打开、项目创建等用户操作速度慢，响应时间长（15 条）
- G. 使用软件编译 C/C++ 文件时出现问题（10 条）
- H. 存在安全漏洞（8 条）

②对上述问题可能原因的分析：

排除用户误操作的情况，程序占用的内存过高、系统更新后对旧插件不兼容、用户指南存在错漏、系统某些功能的程序编写存在考虑不全面的地方（或错误）等可能是造成以上常见问题的原因。

2. 上述问题所对应的潜在需求：

- A. 程序应降低运行时的内存消耗，缩短响应时间，最大程度避免系统崩溃情况。
- B. 程序应对所有可使用版本的插件具有较高的兼容性和适配性。
- C. 减少程序运行中用户常用操作失效的情况
- D. 程序可以提供一份正确、全面且使用便捷的用户指南。
- E. 系统应具备基本完善的防火墙等安全保障体系，防止用户数据的丢失或泄露；同时应增强鲁棒性，防止用户误操作导致的安全或其他问题。

3. 对潜在需求的分类：

①功能性需求

对用户操作模块设计的全面考虑

完善的用户指南

②非功能性需求:

- A. 插件的高兼容性和可扩展性 —— 操作需求
 - B. 较短的系统响应时间
 - C. 较小的内存占用
 - D. 较高的系统可靠性
 - E. 较高的系统安全性 —— 安全性需求
- } 性能需求

四、实验结论

本实验旨在对 Eclipse 的需求进行抽取与分析。Eclipse 是一个成熟的开源 IDE 项目, 实验中通过爬虫从 Eclipse 的缺陷报告中获取了部分用户对于该软件功能缺陷的反馈, 从中分析出用户遇到的共性问题, 提炼总结出其中所包含的潜在用户需求并进行分类。该实验指出了软件中暂时缺少和不甚完善的功能, 为软件的进一步精化提出了较为具体和清晰的方向目标, 有助于软件更好满足用户需求, 提供更高的用户体验。

五、附录

项目仓库:

<https://github.com/ml1201220145/Software-Requirements-Engineering-lab1.git>