

## 实验二 需求排序

### 一、小组成员及得分分配

小组成员：陈轶凡、陈香秀、李昶、徐昌越

得分分配：陈轶凡 25% 陈香秀 25% 李昶 25% 徐昌越 25%

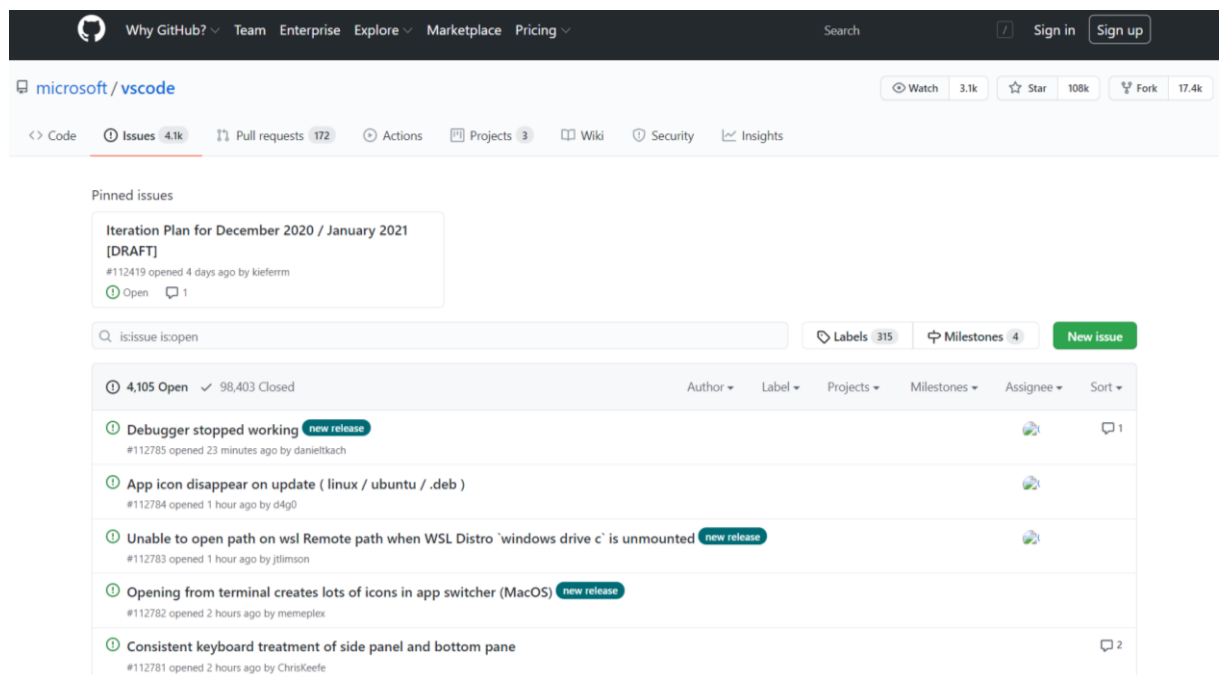
### 二、实验目的

在软件开发维护生命周期中，常常有多个需求被同时提交。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，需要为软件需求进行优先级排序，以合理分配有限资源。

本次实验模拟多个需求同时被提交的情况下，对需求进行排序的过程，并由此认识到在软件开发过程中需求排序的原则以及重要性。

### 三、实验数据及方法

- 1) 本次实验中，我们小组所选取的开源项目是 Visual Studio Code，这是一个开源的文本编辑器，并且在 Github 上有相当多的 issues。实验数据的来源即为 Visual Studio Code 在 Github 上的 issues 模块。详见 <https://github.com/microsoft/vscode/issues>。



- 2) 在 Github 上对应项目的 issues 版块爬取所有关于目标软件 Visual Studio Code 的 200 条问答，将其保存至文本中形成问答文本。

详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/source%26result/main.py](https://github.com/youneta/software_requirement_lab2/blob/main/source%26result/main.py)。

```

1  from lxml import etree
2  import requests
3
4
5  # 获取一个项目的issues列表
6  def get_issues_list(repo_name):
7      issues_list = []
8      url = 'https://github.com' + repo_name + '/issues'
9      # print(url)
10     response = requests.get(url)
11     # 获取源码
12     page_source = response.text
13     tree = etree.HTML(page_source)
14     # # 获取issues数量
15     # number = tree.xpath('//*[@id="js-repo-pjax-container"]/div[1]/nav/ul/li[2]/a/span[2]')
16     # if len(number) == 0:
17     #     number = '0'
18     # else:
19     #     number = number[0].text
20     # if number.isdigit():
21     #     number = int(number)
22     # else:
23     #     number = 200
24     number = 200
25     print(number)
26     # 计算分页数量，每页25个issues
27     page = 0
28     if number % 25 == 0:
29         page = int(number / 25)
30     else:
31         page = int(number / 25) + 1
32     for i in range(1, page + 1):
33         url = 'https://github.com' + repo_name + '/issues?page=' + str(
34             i) + '&q=is%3Aissue+is%3Aopen+sort%3Acomments-desc'
35         response = requests.get(url)
36         # 获取源码

```

### 3) 将爬取的问答保存为问答文档。详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/source%26result/result.txt](https://github.com/youneta/software_requirement_lab2/blob/main/source%26result/result.txt)

```

1  1. https://github.com/microsoft/vscode/issues/10121
2  title: ['Allow for floating windows · Issue #10121 · microsoft/vscode · GitHub']
3  Hi,
4  I suggest floating windows option for:
5
6  Terminal
7  Debug console
8  Problems
9  Output
10
11  Eventually:
12
13  tabs
14  Explorer / search / debug / git / extensions
15
16  This way we could take advantage of large screen space and / or multi monitors.
17  Having to constantly switch between the various windows is not optimum working flow.
18
19  2. https://github.com/microsoft/vscode/issues/519
20  title: ['Allow to change the font size and font of the workbench · Issue #519 · microsoft/vscode · GitHub']
21  At the moment, we can only change the font size / font of the editor. If we want to change the font size, we need to use a roundabout method of "zooming in / out". It would be
22
23  3. https://github.com/microsoft/vscode/issues/10759
24  title: ['Release in the Microsoft Store · Issue #10759 · microsoft/vscode · GitHub']
25  While VS Code is easily obtainable as it is, it still feels like VS Code could benefit from the standardized update procedure that comes with the Windows Store. Perhaps even s
26
27  4. https://github.com/microsoft/vscode/issues/13680
28  title: ['Git: Support git with private key password · Issue #13680 · microsoft/vscode · GitHub']
29  VSCode Version: 1.6.0
30  Commit e52fb0b
31  Date 2016-10-10T18:37:40.189Z
32  Shell 1.3.7
33  Renderer 52.0.2743.82
34  Node 6.5.0
35  OS Version: Windows 7 Pro

```

标题提供了更简练的表达方式。详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/source%26result/result\\_title.txt](https://github.com/youneta/software_requirement_lab2/blob/main/source%26result/result_title.txt)

1. ['Allow for floating windows · Issue #10121 · microsoft/vscode · GitHub']
2. ['Allow to change the font size and font of the workbench · Issue #519 · microsoft/vscode · GitHub']
3. ['Release in the Microsoft Store · Issue #10759 · microsoft/vscode · GitHub']
4. ['Git: Support git with private key password · Issue #13680 · microsoft/vscode · GitHub']
5. ['Editor: scroll jumps randomly (related to Chrome, Electron, xinput) · Issue #28795 · microsoft/vscode · GitHub']
6. ['Allow customization of mouse shortcuts · Issue #3130 · microsoft/vscode · GitHub']
7. ['Macro recording · Issue #4490 · microsoft/vscode · GitHub']
8. ['Tabs for integrated terminal · Issue #10546 · microsoft/vscode · GitHub']
9. ['Visual Studio Code for ipad · Issue #70764 · microsoft/vscode · GitHub']
10. ['Add an optional configurable toolbar below the menu · Issue #41309 · microsoft/vscode · GitHub']
11. ['Allow to open multiple workspaces in the same window · Issue #43188 · microsoft/vscode · GitHub']
12. ['Improved column selection (Alt+select) · Issue #5402 · microsoft/vscode · GitHub']
13. ['Alternative locations for the outline view · Issue #49923 · microsoft/vscode · GitHub']
14. ['File Nesting · Issue #6328 · microsoft/vscode · GitHub']
15. ['Windows update failed: Access is denied · Issue #47841 · microsoft/vscode · GitHub']
16. ['Support to print the editor contents · Issue #5953 · microsoft/vscode · GitHub']
17. ['Revisit VS Code folder structure for app data, settings, extensions · Issue #3884 · microsoft/vscode · GitHub']
18. ['Allow tabs to wrap to multi-line · Issue #70413 · microsoft/vscode · GitHub']
19. ['Support to open a project folder in multiple VS Code windows · Issue #2686 · microsoft/vscode · GitHub']
20. ['Support variables when resolving values in settings · Issue #2809 · microsoft/vscode · GitHub']

- 4) 将英文版问答文本以及标题文本翻译简化为中文版本，以便查看和处理。

详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/source%26result/result%20%E7%BF%BB%E8%AF%91%E4%B8%8A%E5%8D%8A%E9%83%A8%E5%88%86.docx](https://github.com/youneta/software_requirement_lab2/blob/main/source%26result/result%20%E7%BF%BB%E8%AF%91%E4%B8%8A%E5%8D%8A%E9%83%A8%E5%88%86.docx)

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/source%26result/result%20%E7%BF%BB%E8%AF%91%E4%B8%8B%E5%8D%8A%E9%83%A8%E5%88%86.docx](https://github.com/youneta/software_requirement_lab2/blob/main/source%26result/result%20%E7%BF%BB%E8%AF%91%E4%B8%8B%E5%8D%8A%E9%83%A8%E5%88%86.docx)

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/source%26result/%E7%B7%BB%E8%AF%91title.txt](https://github.com/youneta/software_requirement_lab2/blob/main/source%26result/%E7%B7%BB%E8%AF%91title.txt)

- 5) 手工从问答文档以及标题文本中选取 150 条需求进行概括，从而筛选出其中无用的 issue，并将其作为排序的原材料。详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/%E9%97%AE%E9%A2%98%E6%A6%82%E6%8B%AC.docx](https://github.com/youneta/software_requirement_lab2/blob/main/%E9%97%AE%E9%A2%98%E6%A6%82%E6%8B%AC.docx)

1. 改变窗口布局，现有窗口布局不合理，需要不断切换窗口。
2. 允许改变工作台的大小和字体。
3. 将 VS Code 放入 Microsoft Store 中，更易获取。
4. 允许使用具有私钥密码的 git。
6. 允许使用自定义鼠标+修饰符的快捷键。
7. 允许记录一连串输入并重现。
8. 令终端更为实用。
9. 允许 iPad 上使用 Visual Studio 代码。
10. 在菜单下方添加一个可选的可配置工具栏。
11. 允许在同一窗口中打开多个工作空间。
12. 改进列选择 (Alt+select)。
13. 允许轮廓视图位于任何位置。
14. 将文件嵌套变成 VS2015 的工作方式。
15. Windows 更新失败，访问被拒绝。
16. 支持打印编辑器内容。
17. 改变配置的设置目录。
18. 允许标签页被包装成多行。
19. 支持在多个窗口中打开一个项目文件夹。
20. 在设置解析值时支持变量。
21. 在比较视图中增添左右复制、替换功能。
22. 隐藏文件时允许排除部分不需要隐藏的文件。
23. 改善全局文件搜索，使得不需要全部路径和名称。
24. 下载更新补丁并关机后会造成 VSC 丢失。

6) 根据

a) 该 issue 在 Github 上的评价

b) 该 issue 在 Eclipse 的对应问题的分级结果

从上一步概括出的 150 条需求中，找到 40 条进行分级，生成“标准”排序。

详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/%E5%88%86%E7%BA%A7.txt](https://github.com/youneta/software_requirement_lab2/blob/main/%E5%88%86%E7%BA%A7.txt)

7) 提出一种分级标准，结合个人评价，手工对上一步中选取的 40 条需求进行排序，生成“定级”排序。

分级的标准为：

P1:

严重阻碍测试和开发工作，包括但不限于  
用户需求功能完全没有实现（影响用户完成任务）  
用户需求功能实现错误（影响用户完成任务）  
软件闪退/崩溃 无法运行

P2:

不完全阻碍测试执行的严重问题，包括但不限于  
软件常用功能卡死，响应时间较长，超出用户忍耐限度  
后台数据丢失或受损  
影响到系统的某一功能完全无法运行（不影响其余功能）  
安全性问题，如数据加密，数据泄露  
严重的数据计算错误

P3:

用户常用功能存在缺陷，但不影响应用与系统的稳定运行（如初学者都会感受到的缺陷）

内存泄露  
对低版本的兼容性  
功能实现逻辑覆盖面不全  
出现概率近较低的系统异常，如崩溃、闪退、卡死  
用户简单操作即可感知到的明显 UI 问题

P4:

用户不常用，对软件熟悉度高才会感知到的功能缺陷  
插件的兼容性  
用户使用不频繁，且不影响用户完成任务的需求实现错误或未实现  
出现概率低于 20% 的系统异常  
用户使用不频繁的功能易用性差（复杂，效率低）  
用户使用不频繁的功能实现与需求文档不完全匹配

P5:

软件额外功能的性能与软件界面缺陷，以及用户对产品的意见  
插件的优化问题  
非正常操作导致 UI 显示异常  
提示信息与界面效果展示错误  
对于产品设计方面的意见和建议

对于产品界面优化方面的意见和建议  
对于产品需要优化增强用户体验方面的意见和建议

详见

[https://github.com/youneta/software\\_requirement\\_lab2/blob/main/%E5%88%86%E7%BA%A7.txt](https://github.com/youneta/software_requirement_lab2/blob/main/%E5%88%86%E7%BA%A7.txt)

8) 将“定级”排序和“标准”排序的相似性进行比较。详见下一节的实验结果。

#### 四、 实验结果

个人评价的“定级”需求排序如下：

P1:

- 15 Windows 更新失败，访问被拒绝
- 29 虚拟空间未实现
- 164 Win10 上 VSC 冻结/崩溃

P2:

- 24 下载更新补丁并关机后会造成 VSC 丢失
- 32 Windows 的 VSC 打开速度慢
- 55 无法启动共享进程
- 64 Linux 上使用'setxkbmap'的键盘映射不工作
- 124 代码重启更新在 mac 上不起作用
- 25 使用 XRDP 连接时，VS 代码无法在 Ubuntu 上工作
- 40 文件被另一个进程使用时无法更新
- 39 文件资源管理器与文件更新不同步

P3:

- 1 改变窗口布局，现有窗口布局不合理，需要不断切换窗口
- 2 允许改变工作台的大小和字体
- 6 允许使用自定义鼠标+修饰符的快捷键
- 13 允许轮廓视图位于任何位置
- 14 将文件嵌套变成 VS2015 的工作方式
- 19 支持在多个窗口中打开一个项目文件夹
- 31 文件监视器导致 CPU 使用率过高
- 66 滚动和打字反应慢

P4:

- 10 在菜单下方添加一个可选的可配置工具栏
- 26 在 mac 版提供一个隐藏标题栏和内联窗口控件的设置
- 30 使用 VSC 作为 git 合并工具
- 34 终端机从睡眠状态恢复时有时会显示损坏的纹理
- 69 files.watcherExclude 设置在多根工作目录下不起作用
- 187 不支持非标准编码的操作
- 38 XHR 通过 VSC 安装插件时失败
- 20 在设置解析值时支持变量
- 33 查找当前文件所有项目操作太繁琐
- 51 支持私人扩展库

P5:

- 11 允许在同一窗口中打开多个工作空间
- 12 改进列选择 (Alt+select)
- 27 显示打开和关闭的文件所有文件的错误警告
- 82 滚动滞后无响应
- 131 允许 100%的面板宽度和高
- 189 将资源管理器排序与平台排序对齐
- 22 隐藏文件时允许排除部分不需要隐藏的文件
- 35 支持通用语言注入
- 101 支持特定平台扩展

根据标准生成的“标准”需求排序如下：

P1:

- 39 文件资源管理器与文件更新不同步
- 164 Win10 上 VSC 冻结/崩溃

P2:

- 1 改变窗口布局，现有窗口布局不合理，需要不断切换窗口
- 2 允许改变工作台的大小和字体
- 10 在菜单下方添加一个可选的可配置工具栏
- 14 将文件嵌套变成 VS2015 的工作方式
- 15 Windows 更新失败，访问被拒绝
- 29 虚拟空间未实现
- 32 Windows 的 VSC 打开速度慢

P3:

- 6 允许使用自定义鼠标+修饰符的快捷键
- 13 允许轮廓视图位于任何位置
- 19 支持在多个窗口中打开一个项目文件夹
- 24 下载更新补丁并关机后会造成 VSC 丢失
- 25 使用 XRDP 连接时，VS 代码无法在 Ubuntu 上工作
- 30 使用 VSC 作为 git 合并工具
- 31 文件监视器导致 CPU 使用率过高
- 33 查找当前文件所有项目操作太繁琐
- 40 文件被另一个进程使用时无法更新
- 55 无法启动共享进程
- 64 Linux 上使用'setxkbmap'的键盘映射不工作
- 66 滚动和打字反应慢
- 82 滚动滞后无响应
- 124 代码重启更新在 mac 上不起作用

P4:

- 20 在设置解析值时支持变量
- 22 隐藏文件时允许排除部分不需要隐藏的文件
- 26 在 mac 版提供一个隐藏标题栏和内联窗口控件的设置
- 34 终端机从睡眠状态恢复时有时会显示损坏的纹理
- 38 XHR 通过 VSC 安装插件时失败
- 51 支持私人扩展库
- 101 支持特定平台扩展

189 将资源管理器排序与平台排序对齐

P5:

- 11 允许在同一窗口中打开多个工作空间
- 12 改进列选择 (Alt+select)
- 27 显示打开和关闭的文件所有文件的错误警告
- 35 支持通用语言注入
- 69 files.watcherExclude 设置在多根工作目录下不起作用
- 131 允许 100%的面板宽度和高 z
- 187 不支持非标准编码的操作

其中,

“定级”需求排序与“标准”需求排序相同的概率

$$17 / 38 = 44.737\%$$

“定级”需求排序与“标准”需求排序相等或小于 1 的概率

$$(17 + 19) / 38 = 94.737\%$$

“定级”需求排序与“标准”需求排序相等或大于 1 的概率

$$(17 + 2) / 38 = 50.00\%$$

## 五、 结论

1. 根据提出的分级标准进行的分级排序, 和“标准”的需求排序相比, 呈现出三个特点:
  - 1) 排序相同的概率为 44.737%, 相等或小于 1 的概率为 94.737%, 可见提出的分级标准和开发者的分级和排序意图相当接近, 得出的需求排序具有相当的参考性。
  - 2) 需求排序相等或大于 1 的概率达到 50.00%, 说明提出的分级标准具有一定的激进性, 一些需求的重要性可能被激进地认为处于较高的位置。
  - 3) 需求排序相等或小于 1 的概率比需求排序相等或大于 1 的概率更大, 说明提出的分级标准相比激进性呈现出更加明显的保守性, 贴近开发者的分级和排序意图。
2. 实验过程采取了自动爬取、人工筛选、人工制定分级标准和人工进行需求排序相结合的方法, 简化了需求获取的重复操作, 滤去了 issue 中无效的需求, 从人的主观感觉的角度进行需求排序, 并将其与“标准”需求排序进行比较, 很好地进行了实验结果的分析。小组成员在实验过程中体验了需求排序的过程及其重要性。