

实验二：需求排序

小组成员：171860566 王盼盼

171860565 刘振国

成绩比例 1:1

实验思路

选定一个开源 IDE 项目，确定可能的信息来源，获取有效信息，对所获取的需求进行优先级排序。

- (1) 确定 IDE 项目；
- (2) 明确信息源；
- (3) 获取需求；
- (4) 提出一种方法，对需求进行优先级排序。
- (5) 对所提方法的效果进行分析。

数据获取

选择的 IDE 仍然是 `vscode`

根据实验指导中思路一的提示，我们明确 `vscode` 托管在在 `GitHub` 上的地址作为信息来源，实验的思路是对该项目中的 `Pull Request`（地址：<https://github.com/microsoft/vscode/pulls>）作为一种需求，获取相关信息。

我们发现这些 `PR` 的标题可以概括相应的需求信息，为了获得众多的标题信息，我们使用 `Python` 语言写代码（文件：`vs_pr_get.py`）来抓取这些信息并存入文件（`pulls3.txt`）。

之后利用实验一词频分析的思路，这里不做过多赘述

需求排序

优先级排序基于下面的算法：

根据价值、成本和风险来设定优先级

• 使用优先级设定模型，可以遵循如下步骤：

1. 在电子表格中列出要设定优先级的所有特性、用例或需求。
2. 让客户代表来估计每一个特性提供给客户或业务的相关利益，并用1~9划分等级，1代表对任何人都没用的特性，9代表具有最大价值的特性。
3. 估计出如果没有把某一特性包括到产品中，将会给客户或业务上带来的损失。
4. 该电子数据表中“总价值”一栏是“相对收益”和“相对损失”的总和。
5. 根据特性的复杂度、所需要的用户界面的实际情况、重用前代码的潜在能力、所需的测试量和文档等等，开发人员可以估算出成本。
6. 类似地，让开发人员估计出与每个特性相关的技术风险或其他风险的相对程度，并利用1~9来划分等级。
7. 把所有的估算都填入电子数据表之后，就可以利用如下公式计算出每一特性的优先级。

$$\text{优先级} = \frac{\text{价值}\%}{(\text{成本}\% * \text{成本价值}) + (\text{风险}\% * \text{风险价值})}$$

8. 按计算出的优先级的降序排列表中的特性。处于列表最顶端的特性是价值、成本和风险之间的最佳平衡。

一名组员扮演客户代表，一名组员扮演开发人员代表，分别为每一个需求特性打分，利

用上面的算法计算出每一个需求特性的基础优先级，然后乘以该需求所在大类的权重，得到最终的优先级。为了验证我们方法的有效性，我们设置了三种大类权重分配方式，计算出三种情况下的优先级，进行对比。排序结果在 lab2_sort.xlsx 文件中。

方法效果分析

起初我们的想法是直接按照打分情况计算出基础优先级即可，然而结果却很不理想，因为有些需求很重要但它的成本相对较高，这样会使得它最终的优先级反而低于那些并不是很重要但是成本却很低的需求，这种情况显然很不合理。此外，对于优先级排序这种主观性很强的任务，如何评判某种方法的好坏标准不一，单独对一项结果进行排序分析难以做到，但是我们却可以对比不同结果的好坏，我们可以很容易找到合理与不合理的地方。

于是我们提出了先按照最初的想法计算出基础优先级，再用基础优先级乘以该需求所在大类的**权重**，得到最终的优先级。然后，在此基础上，我们横向对比三种权重的不同结果，从而得到某些结论。

基于对 IDE 的认知，我们认为在需求方面：**编译>调试>资源管理>编辑>用户界面**

	权重1	权重2	权重3
编译	5	16	9
调试	4	8	7
资源管理	3	4	5
编辑	2	2	3
用户界面	1	1	1

分别对三种优先级进行排序，提取出每种情况下的 Top10，结果如下：

优先级 1:

需求	优先级1
文件路径（在窗口标题中显示文件所在的文件夹路径）	10.80
单词拼写检查功能，突出显示错误	10.71
显示行号、列号	9.00
语法检查，提示出错位置和出错类型	9.00
如果项目中有未保存的文件，提交时显示警告	9.00
支持多种语言的语法	8.75
允许在工作区编辑中重命名/创建/删除文件	8.40
查找指定内容	8.00
语义检查，提示出错位置和出错类型	7.50
暂存新文件	7.50

优先级 2:

需求	优先级2
单词拼写检查功能，突出显示错误	34. 29
语法检查，提示出错位置和出错类型	28. 80
支持多种语言的语法	28. 00
语义检查，提示出错位置和出错类型	24. 00
文件路径（在窗口标题中显示文件所在的文件夹路径）	14. 40
如果项目中有未保存的文件，提交时显示警告	12. 00
允许在工作区编辑中重命名/创建/删除文件	11. 20
暂存新文件	10. 00
断点调试	9. 60
显示行号、列号	9. 00

优先级 3:

需求	优先级3
单词拼写检查功能，突出显示错误	19. 29
文件路径（在窗口标题中显示文件所在的文件夹路径）	18. 00
语法检查，提示出错位置和出错类型	16. 20
支持多种语言的语法	15. 75
如果项目中有未保存的文件，提交时显示警告	15. 00
允许在工作区编辑中重命名/创建/删除文件	14. 00
显示行号、列号	13. 50
语义检查，提示出错位置和出错类型	13. 50
暂存新文件	12. 50
查找指定内容	12. 00

从上面的结果中可以看出，最高优先级是 IDE 所必备的功能，如：编译、汇编、文件、编辑等功能，这符合程序员对于一个 IDE 的客观要求。有些功能虽然使用起来非常的便捷，如查找、替换指定内容等，但没有它们并不会使 IDE 无法正常工作，因此它们的优先级处于中上的位置。综合来看，需求优先级的计算，必备需求的优先级权重较高，尽管它的成本和风险可能也较高，但这种成本和风险是开发人员必须克服的。

这些结果也会有具体排名上的不同，对于不同的权重分配问题，会引起不同的排名结果，但它们在一定方面都具有合理性，这取决于不同项目对实际需求的重视程度。

仓库文件说明

data

word_frequency.txt	词频统计结果
pulls3.txt	PR 爬取结果
lab2_sort.xlsx	基于不同大类权重的成本风险优先级排序结果

code

Word_frequency.cpp	词频统计程序
vs_pr_get.py	vscode 的 PR 标题爬取程序