

- 软件需求工程实验一
 - 小组成员信息
 - 实验目的
 - 实验方法
 - 确定项目
 - 明确信息源
 - 获取数据
 - 分析需求
 - 需求排序
 - 实验结果及效果分析
 - 结论

软件需求工程实验一

小组成员信息

队伍成员姓名	队伍成员学号	得分分配
王麦迪	171860611	20%
吴家玥	171850048	20%
王赠龙	171860631	20%
汪铭潇	171860648	20%
蒋雨宸	171860630	20%

实验目的

该实验旨在对一个高知名度、使用广泛的开源项目进行需求排序，确定可能的需求信息来源以获取有效的需求信息，以及使用特定技术获取潜在需求并进行需求排序。

该实验旨在指导小组成员实践并熟悉需求的获取、需求的理解以及需求的排序，并启发小组成员站在更宏观的角度理解各类需求的优先级。

该实验旨在促进小组成员进行合作继而锻炼任务理解、任务划分、交流平衡、任务合并等能力，并完整地经历大任务分解为小任务、小解决方案合并为大解决方案的过程。

实验方法

确定项目

考虑到熟悉程度和知名程度，我们决定在**GitHub**平台寻找项目。

经过对**GitHub**上的开源项目的综合分析：

- 知名度：**Star**数量，**Watch**数量，**Fork**数量，**Contributor**数量。
- 项目完善程度：是否有**Wiki**。
- 规范程度：**Issue**和**Pull Request**是否有特定模板。

我们最终选出了与实验一相同的项目--前端框架****Vue****作为实验对象。

明确信息源

Issue所能承载的需求信息非常有限，其描述属于非常具体且微观，并且通常不能给出解决方案。经过充分讨论，我们觉得**Issue**不能充分反映该开源项目的需求。

在认真审视**Pull Request**后，我们发现对于**Vue**而言，其**Issues**更多地是为**Pull Request**提供一个前置讨论的场所，而有价值的**Issue**（比如说某个严重的bug，某个需求强烈的功能）最终都会演化成**Pull Request**，进而合并到代码库。**Issue**是一个相对畅所欲言的场所，而**Pull Request**是更有价值更有可能影响到软件演化的需求。

因此，我们决定选择**Pull Request**作为需求的信息源。

获取数据

数据是通过拉取**Vue**的**Pull Request**信息获取的，详情请看**原始需求**文件夹。

分析需求

从高抽象级看，我们获取的**Pull Request**所代表的需求可以划分为两大类：

1. 代码维护：可读性改善；增加测试；缺陷修复
2. 功能迭进：对从前某个**Issue/Pull Request**下的讨论挖掘出的需求进行实现；对开发过程中遇到的痛点需求进行实现。

站在更具体的一层抽象级，我们认识到**Vue**本身提供的**PR**模板给出的划分有极大的合理性，几乎完美覆盖上述观测到的需求。除此之外，我们还发现即使是**Other**类别中的需求也通常能划分到某一个已经存在的类别当中。**Vue**所给出的6个类别：

- 1. Bug修复
- 2. 增加功能
- 3. 代码风格更新
- 4. 代码重构
- 5. 构建相关的修改
- 6. 其他

经过我们的激烈讨论，我们认为这六类需求的优先级排序如下：

- 1. Bug修复
- 2. 构建相关的修改
- 3. 代码重构
- 4. 增加功能
- 5. 代码风格更新
- 6. 其他

需求排序

需求排序	PR编号
Bug修复	#9527; #9479; #9430; #9327; #10454; #10413; #10377; #10272; #10253; #10250; #10299; #10208; #10167; #10117; #10099; #10085; #10073; #10015; #9946; #9938; #9919; #9802; #9787; #9776; #9773; #11178; #11159; #11143; #11138; #11128; #11111; #11107; #11065; #11047; #11031; #10996; #10993; #11740; #11726; #11707; #11706; #11699; #11613; #11578; #11576; #11599; #11583; #11563; #11537; #11523;
构建相关的修改	#9635; #11030; #11099; #11653; #11644; #11640; #11531; #11522;
代码重构	#9527; #9469; #10317; #10066; #9962; #11031
增加功能	#9603; #9601; #9591; #9567; #9564; #9502; #9484; #9482; #9479; #9470; #9430; #9369; #9358; #10361; #10286; #10245; #10202; #10199; #10119; #10037; #10026; #9814; #9765; #9674; #11191; #11152; #11094; #11021; #11047; #11712; #11702; #11488;
代码风格更新	#9635; #9531; #9482;
其他	#9403; #9241; #10394; #10387; #10341; #10279; #10024; #9937; #9872; #9732; #9636; #11189; #11112; #11166; #11139; #11083; #11052; #11024; #11725; #11565; #11518; #11509; #11488;

详情见需求排序文件夹。

实验结果及效果分析

实验结果如下：

1. PR类别最多且优先级最高的是**Bug修复**。
2. PR类别优先级次高的是**构建相关的修改**。
3. 其他类别里面大多是文档完善、增加测试，因此优先级最低。

这样的PR分布与我们的理解是一致的：大多数PR是对功能**可用性**的增强，包括对功能的**鲁棒性**以及**完整性的完善**；代码风格的调整的PR是最少的，因为整体代码库的风格具有持续性和一致性，不会出现大变动；因为这是一个大型开源库，为了方便合作，**可读性**对该库非常重要，因此有很多PR用于完善文档。

实验的效果令人满意：

1. 我们完成了对大型开源库的需求分析，使用了特定技术获取潜在需求并进行需求排序。
2. 我们实践并熟悉需求的获取、需求的理解以及需求的排序，对大型开源项目的需求演变有了第一手的观察经验。
3. 我们锻炼了任务理解、任务划分、交流平衡、任务合并等能力，并完整地经历大任务分解为小任务、小解决方案合并为大解决方案的过程。

结论

著名开源前端框架**Vue**的需求可以分为6个类别：**Bug修复**；增加功能；代码风格更新；代码重构；构建相关的修改；其他。优先级的排序为：**Bug修复**；构建相关的修改；代码重构；增加功能；代码风格更新；其他大多数PR是对功能**可用性**的增强，包括对功能的**鲁棒性**以及**完整性的完善**；代码风格的调整的PR是最少的，因为整体代码库的风格具有持续性和一致性，不会出现大变动；因为这是一个大型开源库，为了方便合作，**可读性**对该库非常重要，因此有很多PR用于完善文档。