RE_Lab2 实验报告

一、 IDE 项目

Visual Studio

二、 实验数据获取

1．获取路径

从 StackOverflow，Visual Studio 开发者社区，CSDN 等途径获取。对于 StackOverflow，通过获取其中关于 VisualStudio 的问答，从中抽取有意义的能够反映出对 Visual Studio 有功能需求的部分进行一定的分析，作为获取到的数据；对于 VS 开发者社区，其中有很多的来自开发者的关于 VS 的功能需求，改进建议，优化需求等等，我们选取其中有意义的具有代表性的部分问答作为数据来源，得到相应的数据；而 CSDN 中则包含了许多来自国内开发者的对 VS 的吐槽，或是对 VS 的部分缺点的改进方法，我们选取这些数据作为我们获取的数据，来反映最终得到的需求。

2．获取内容

组内四位成员均进行了实验数据的获取，在项目中 Project1 目录下有每人学号所对应的获取到的需求数据。

三、 需求排序

进行等级排序，即将排序问题转化为分类问题，分为 5 级：Highest(5), High(4), Medium(3), Low(2), Lowest(1)

在分级的时候，加入分层的思路：

功能需求：对应 Highest-Medium

系统需求：对应 High-Medium

用户体验：对应 High-Lowest

组内四人分别对每一种需求的重要程度打分，最终取均分，四舍五入后变为 5 档

1. 功能需求

关键功能需求（代码的编辑，编译，调试等）

1) 注释：增加隐藏或显示代码中注释的功能 3
2) 配置文件：优化 VS 配置文件的引用和默认值设置等功能 5
3) 自动补全：改善 VS 的自动补全功能，包括对 VS 的自动补全快捷键的更改 4
4) 子断点删除：VS 的针对不同实例的子断点在某些时候很好用，但是需要增加更便捷的删除子断点的功能 3
5) 调试：需要优化 VS 的调试功能，改善调试器在部分情况下无法响应的问题 3
6) 宏：需要修复对部分宏的显示问题 5

扩展功能需求：

1) 安装包：需要修复部分安装包无法显示问题 5
2) 卸载：增加对无用的扩展程序的卸载功能 5
3) 更新：需要增加对扩展程序的更新检测以及后台更新功能 4

附加功能需求：

1) 文件名：需要修复文件名大小写显示问题 3
2) 用户登录：需要有更加完善，不易出错的用户登录系统 3
3) 用户登录：实现在不同计算机上同步同一用户的用户设置的功能 3
4) 命令行：对命令行支持效果较好，方便对命令行依赖程度较高的程序员，能通过命令行对程序进行编译运行，调试，更新软件，导入必要的库等功能 4
5) 自动保存： 添加基本的保存功能，并允许用户设置自动保存时间等其他选项 5
6) 文件对比：允许进行文件的纵向、横向对比（即不同历史文件对比，两份文件的差异化对比）3
7) 文件处理保存单个文件修改步骤，允许回退到某一个时间点，允许删除文件时，仅仅将其从本项目中删除，但允许文件恢复，只有当删除本项目时才删除所有文件（不可逆）4
8) 引入 Git：鉴于 Git 现在的流行，所以 vs 引入了“发布到 Git 服务”的特性，让开发者可以在 vs 团队服务、GitHub 或一个私有版本库上发布一个新的项目 5
9) 扩展：允许自定义扩展，根据自身喜好进行自定义。此外，vs 还提供了许多丰富，有价值的扩展。3
10) Office 开发人员工具： 包含了附带最新功能和更新的 Office 开发人员工具 4

其他：

1) 多人协作：使用版本控制、具备敏捷性且高效协作（如团队资源管理器中新增 Git 功能并增强了连接体验）4
2) 帮助：增加帮助选项在脱机状态下的可用功能 3
3) 登录:提高用户登录的稳定性 3
4) 同步:实现在不同版本下用户的个性化设置的同步性 4
5) 引入 Git:完善与 Git 存储库的对接，且实现该对接不受更新的影响 3

6) 对接 Web: 优化 VS 对于 Web 应用程序的响应速度 3

7) 对接 Android:优化 Android 开发，解决 Android 开发不稳定问题 4

8) 多人协作:实现 VS 的实时共享功能，修复实时共享登录问题 5

9) 代码审查:实现代码审查功能的可用性 3

10) 扩展程序：

    a) 实现扩展程序的删除功能 3

    b) 优化对于扩展程序的下载功能，实现扩展程序的后台下载 3

    c) 编译器版本：实现与各个版本编译器的兼容性 5

11) dpi 模式：:需要支持混合模式 dpi3

12) 代码编辑、调试 5

13) 快捷键：Vs 2019 的快捷键需要符合用户使用习惯，最好和其他主流 IDE 类似 3

14) 断点：Vs 可以通过断点折叠而出现子断点，子断点是不同实例对应的断点 3

15) 编译器版本：支持用户自主导入不同版本编译器 4

16) 自动补全：自动补全尽量人性化，尽量做到需要补全时给出待补全内容，但是不会错误理解用户意图（如：用户只是修改某一个部分，但是触发了自动补全，导致用户需要手动取消）3

17) 内存分析：Vs 新增加的内存分析功能可以很好的帮助解决和修复应用中的内存问题 4

18) 测试阶段：使用全面的测试工具编写高质量代码，如：实施单元测试或使用测试资源管理器将自动化与测试用例工作项关联 3

2. 系统需求

1) 增加适用于 Linux 系统的 VS5

2) 系统：优化 VS 对于各个平台的适配性 5

3) 权限：需要系统管理员权限才能进行安装，且需要安装 .NET framework 4

4) 软硬件： Visual Studio 允许在 windows 上进行安装，硬件需要 1.8GHZ 或更快的处理器，2GB 以上的运行内存，硬盘空间一般需要 20GB~50GB 4

3. 用户体验

1) 软件对比：VS 的跨平台移动应用开发功能相较于其他开发工具仍有许多需要改善的功能 4
2) 触控板手势：需要优化在部分菜单内触控板手势滚动敏感度不正确的问题，需要提供对任何情况下敏感度保持统一的触控板手势滚动。4
3) 起始页问题：需要增加起始页的用户自定义功能，增加起始页的禁用功能 3
4) 字体颜色设置：
    a) 需要增加对 c++的 arrtibutes，SQL 的独特的高亮颜色；2
    b) 需要增加代码颜色拾色器 2
    c) 实现不同主题下字体颜色的自定义功能，字体颜色不随主题变化而变化 1
5) 主题设置：
    a) 需要实现在任何情况下的主题选择功能，VS 本身主题不应该受 Windows 本身亮度的影响 2
    b) 需要实现选项对话框与 VS 本身主题一致 2
    c) 需要增加黑色主题下边框亮度的自定义功能 2
    d) 界面设置：需要删除部分代码下无意义的绿色波浪线 3
6) 快捷键设置：
    a) 增加查找和替换对注释是否生效的开关 2
    b) 需要在参考窗口添加刷新按钮 2
    c) 添加用来关闭快速查找窗口的快捷键 1
7) 便捷功能：
    a) 完善开始窗口，允许对开始窗口的自定义 2
    b) 更改解决方案窗口在默认情况下的显示状态 3
8) 主题:
    a) 各个选项窗口应当与 VS 主题相匹配 1
    b) 实现 VS 主题的自定义功能以及对话框与 VS 主题的同步性 1

9) 快捷键:

a) 添加撤销更改的设置的快捷键 3

b) 添加更多的快捷键功能，加入用户自定义快捷键 3

10) 语句对齐:实现对不同语言不同语句的自动化设置格 4

11) 头文件搜索:实现对所需头文件的自动搜索及补全 3

12) 模板搜索:实现对最近的模板的搜索功能 3

13) 起始页:优化起始页的页面布局 2

14) 设置记忆:实现对属性窗口的配置的记忆与保留 4

15) 任务栏图标: 优化 VS 在任务栏的图标显示 3

16) 菜单:扩大菜单栏的点击区域 3

17) 修复提示符:删除代码下的绿色波浪线提示符 2

18) 字体高亮: 优化针对部分语言的突出显示 3

19) 按钮:添加针对查找窗口的刷新按钮 3

20) 代码保存: 添加对代码的自动保存功能 4

21) 启动与运行：优化 VS 启动过程与运行过程 3

22) 语言：优化 VS 对多语言的支持 4

23) 下载：Vs 下载速度慢，需要手动修改特定的源，否则可能出现下载卡顿 3

24) 安装卸载：可能会出现无法卸载干净，以及因为没有卸载干净导致无法安装新版本或者版本回退的情况。例如：“安装向导”在您的机器上检测到 Visual Studio 6.0 产品潜在的版本冲突软件启动速度较慢，可能是由于安装时下载了不必要的组件 3

25) 不单一：由于没有将支持不同语言的功能单独拆分成新的 IDE，导致 vs 能支持多种语言，使得 vs 有些杂乱 2

26) 冗余项：下载时出现很多暂时用不到的其他语言组件，导致软件庞大 3

27) 崩溃：在 win7 系统休眠而 vs2017 仍然处于打开状态，则重新进入 win7 系统后，vs2017 立即崩溃，只能将电脑重启 2

28) 界面设计：界面设计扁平化，界面简洁，将同一类型的功能隐藏了一个下拉选项内部，但同时，不能让该选项包含过多内容。支持不同背景颜色修改，最好能支持自定义背景图片（透明化，尺寸等）2

29) 软件对比：由于每次 vs 推出新特性时，工程人员都不免将其与 Eclipse 进行对比，由于 vs 存在占用磁盘空间和机器内存的缺点，所以微软最近加入了 Eclipse 基金，开始将二者结合起来，未来，开发人员可以在 vs 内部使用 Eclipse，反之，亦然 3

四、 效果分析

1) 对于需求的分析、分级排序有一定的主观色彩，建议可以结合大数据分析，取得各个需求的提出者人数，结合各种其他情况一起分析，得到更加客观、有效的结果
2) 设定需求优先级的目标之一：为能够给客户带来最大的业务利益或易用性利益的需求最高的优先级。从这点来看，功能需求、系统需求及用户体验中又分成不同的优先级。
   A. 在功能需求中：文件配置、自动补全、宏及安装卸载更新等功能的优先级比其他功能的高。
   B. 系统需求的整体优先级都较高，因为系统支持不足无法使用户体验达到期望值。
   C. 在用户体验中，不同软件之间的对比、触控板手势及语句对齐的优先级较高。对于编程人员来说，使用的 IDE 种类较多，无法避免对 IDE 进行比较，一旦某个软件的某种优势突出就会导致用户形成依赖，自然会筛选不具备此功能的 IDE。语句对齐可以增加代码的美观性，也减轻了用户手动对齐的负担，让用户书写代码更加流畅。
3) 相较于全排序，等级排序的优点在于对于每一个需求，都能找到一个与其对应的优先级层次，包括 Highest，High，Medium，Low，Lowest。相应的我们可以通过其所在层次，直观的得出该需求的重要程度。但其缺点在于对于同层次的多个需求，不能直观的体现出其相互之间的优先级关系，或者说等级排序并没有对处于同一层次的几个需求再进行排序。

全排序可以直观的表现出上述的对于每两个需求之间的比较关系，但是相应

的，对于每一个单个的需求，我们只能通过他在全排列中的位置来估计他的优先级程度，但这一位置所表现出来的会受到全排列中最高优先级与最低优先级所在层次的影响。即我们不能直观的得到他的需求层次。

举例来说，假设一个全排列得到了一组需求，其优先级最高与最低的需求都对应着 Lowest 层次，则我们对于其中一个需求所在层次的判定就要结合其两端需求的层次来判断而不能直观得到。但是等级排序可以做到每个需求所在层次的直观表示。