软件需求工程作业二

1. **访谈参与人员：**

访谈者（Smart IDE的开发者）： 梁欣悦

被访谈者（Smart IDE的需求者）： 蔡悦

1. **整理后的访谈信息：**
2. **一个Smart IDE应该有什么样的基础功能呢？**

首先是要能够写代码，然后可以编译、运行、Debug。

写代码的时候要有和其他IDE一样的可以自动补全、自动语法分析、代码搜索等功能。然后点击变量名或者函数名可以跳转到声明处。

编译的话，可以参考我们做操作系统相关实验的时候，在Linux平台下GCC运行时，可以有多个优化级别的选择。所以我们这个Smart IDE也可以选择编译的优化等级。

运行的时候要能够实时输出一些日志信息，也要能够Debug，比如要像Visual Studio那样可以断点调试的。

除此之外，要支持多种语言，比如C、C++、JAVA、Python等。

1. **除了一般的基础功能之外，还有什么功能使得它更smart呢？**

刚才说的那些都是很多IDE都有的基础功能。因为我们需要我们的IDE更加Smart，因此要加一些高级的功能。

比如可以辅助我们debug做分析的，代码的静态与动态分析，对，静态分析就是分析代码本身，比如有些内存泄漏可以通过静态分析检测出来。动态分析就是那种可以实时显示运行过程和用到的资源的。比如Android Studio里面的Monitor，监视器这种。动态分析还有一种就是测Coverage的方法，像Eclipse的Eclemma插件就很好，可以使用不同颜色来显示代码测试时覆盖的情况。

然后是一些辅助写代码的实用功能吧，比如自动在使用处生成注释、自动修改和添加包头、提供错误修改方案、代码标记这些。

因为有些专业的程序员喜欢用命令行，所以在软件内部也可以使用命令行来进行一些操作。

云端可以建仓库，支持版本控制。

1. **对于性能方面，有什么要求呢？**

就是正常要求，要求IDE运行和编译代码时效率高、开销低。有一定的兼容性，不可能说win7用得了，win10就用不了。然后不同平台也要有对应的可兼容版本。

然后是无已知的可导致崩溃的bug，也要有如果突然崩溃了，保存当前进度这种功能。写了半天没有保存，就很难受了。

比较重要的是安全，要保护用户隐私，我个人也比较看重这一块。

最后一点是要可扩展，就很市场上很多IDE都是可以从商城里下载扩展插件的，Smart IDE也要有这种功能。然后是可更新，可打补丁这种很常规的维护操作。

1. **对于界面有什么需求呢？**

界面第一需求是要好用，因为要长期注释，所以要简洁美观，看着舒心，支持个性化。然后控件排布合理，可以拖曳。

避免工具栏有太多项吧，支持快捷键。

1. **是否有多人协同开发机制呢？**

支持的，云端仓库支持多人增量式开发，大致就模仿github那样子吧。然后工程文件可以比较方便地合并这样。

1. **除此之外，还有什么需求呢？**

我看到近期有人的研究程序员开发时“心流”状态的记录，如果IDE可以追随前沿科技，记录程序员编写代码时“心流”状态和工作效率就好了。

1. **整理后的需求：**
2. **功能需求：**

*基础需求：*

1. 代码编辑

* 自动补全：变量名、类型名在编辑时，有自动补全的可选项，按tab键可键入。自动对齐：在括号自动按照结构对齐，每一行的空格也根据结构自动对齐。
* 拼写检查：检查变量名、类型名是否有误。
* 语法检查：检查语法否合法，给出错误提示。
* 代码搜索和替换：对一段代码进行搜索和替换，可以进行局部或者全局替换。
* 查看变量或函数的定义：在代码编辑界面，可以通过点击，查看变量或者函数的定义。
* 图形界面开发：可以方便地绘制图形界面和编写功能代码。提供较多工具包。

1. 编译

* 可以自定义编译优化等级，同时编译速度较快。

1. 运行

* 实时并且详细地显示资源占用情况（图形界面），具体包含CPU占用、内存占用、电量消耗等信息。
* 实时显示运行日志，可分类查看，查询和导出。

1. 调试

* 支持断点调试、单步调试、条件调试、单元测试等调试方法。
* 针对调试，可以自定义监视器内容。

1. 支持多种编程语言

* 支持C、C++、JAVA、Python等多种编程语言。

1. 帮助功能

* 有官方网站，提供下载、提问、功能介绍、新闻、联系客服等功能。
* 有用户反馈入口，以及配套的客服服务。
* 有分类详细、管理到位的论坛，用户可以直接导入项目代码进行提问。
* 有新手教程、详细的帮助文档。

*Smart需求：*

1. 代码静态分析

* 提供代码静态分析历史记录和文本对比功能
* 对代码（单元或全局）进行静态分析，分类显示错误，错误代码处可追踪。
* 错误分类包括但不限于：

Correctness、Internationalization、Performance、Security、Usability

1. 代码动态分析

* 自动生成测试用例，或自动化生成点击序列。可自定义生成算法。
* 有测试覆盖率、资源占用等动态分析功能。
* 可以自动绘制程序运行流程，如进程、路径、硬件调用等，并可导出。
* 提供运行结果历史记录和对比功能。

1. 错误提示和修改方案

* 编写时，如果检测出错，给出错误类型，并提供几个待选的解决方案，点击即可自动执行。
* 可以直接导入代码和错误类型在论坛上进行提问或搜索。

1. 版本控制

* 可以方便进行本地和云端存储，有连接github的插件。
* 可多人协同增量式开发，能够分配任务、对分配任务进行管理、针对任务快速合并项目。

1. 注释生成

* 搬运变量和函数声明处的注释到使用处，便于用户理解该函数或变量的含义。

1. 自动添加和修改包头或头文件

* 使用函数或变量时，自动识别其依赖于的包头或头文件，自动添加。
* 代码导入时，对应的包头或头文件自动修改。

1. 代码标记

* 对代码标记以区分类别，比如TODO类等，并可以通过标签快速定位。

1. 支持多种操作方式

* IDE内部设置命令行，可以通过内部命令行进行编译等操作。
* 可以用读取指令文件或者批量操作文件的方式对项目进行操作。
* 可以在操作系统终端使用。

1. 记录工作情况

* 可对用户编写代码的编写速度、代码质量做量化和评估。记录程序员编写习惯、工作效率和工作质量，生成自然语言报告。

1. 形式化方法开发

* 可以用逻辑化语言形式化的定义软件开发需求，IDE可以根据形式化方法定义内容，自动生成部分代码。

1. 保存自定义软件框架

* 可以把当前项目作为模板，自定义软件框架。新建项目可以基于模板生成。

1. 跨平台、跨语言测试

* IOS端的测试代码可以直接转换为Android测试代码，实现跨平台测试。
* 不同编程语言之间的测试代码、测试用例也可以进行转换。

1. 针对不同语言的特色功能

* 针对不同语言，可提供不同功能。譬如针对JAVA语言，可提供手动GC功能。针对Python，提供代码对齐检查功能等。

1. 代码共享

* 提供针对其他用户公开的代码的搜索和快速导入功能。便于用户进行代码复用。用户也可对公开代码进行评论和评分。
* 用户也可公开自己的代码与他人分享，支持代码快速导出到公开代码库的功能。
* 支持将项目快速上传至应用市场。

1. 多种插件导入方式

* 内置插件市场，可在插件市场购买和一键安装。
* 也可打开本地文件安装插件，以实现功能的扩展。

1. 内容提供

* 联网状态下，首页提供IDE相关新闻或随机的使用小技巧。

1. **非功能需求：**

*性能需求：*

1. 高效性

* IDE运行和编译代码时效率高、开销低。界面跳转时间小于1S。

1. 兼容性

* 对于同一操作系统，兼容多个版本。也有针对不同操作系统（Windows、Mac、Linux）和硬件条件（32位、64位）的不同版本。
* 稳定性，无已知可导致崩溃的bug，设置针对崩溃的备份应对机制，譬如自动存储当前进度、输出错误日志等。

1. 安全性

* 不泄漏用户隐私，保障代码安全。云端仓库权限管理严格。

1. 可扩展性和可维护性

* 可更新版本、可安装扩展插件。且定期开发新版本，增加新扩展插件。
* 支持用户上传新插件，经过审核后发布于插件市场中。

*界面需求：*

1. 易用性

* 设计风格简洁美观，操作方便。
* 工具栏和编辑界面排布合理，可以自行拖曳和增减。

1. 支持个性化自定义

* 有多种界面主题可选，也可以自行编辑界面主体。
* 支持自定义具体样式，譬如代码高亮颜色、代码前是否有标号、光标样式等。
* 自定义设置可导出和导入。

1. 快捷键

* 支持快捷键和纯键盘操作，快捷键也可以自定义。