软件需求工程作业四

171860022 梁欣悦

1. **原来的需求**
2. **功能需求：**
3. 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：检查语法是否合法，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。

1. **Smart需求：**
2. 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。

1. 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。

1. **需求审查清单**
2. 需求是否完整？

不够完整，比如基础功能中的“编译”一栏，没有指定使用什么编译器、以及如何编译器通讯、以及信息储存等，不完整。

又如Smart功能中的“错误提示和修改方案”一栏，只说了“给出几个错误的修改方案”，但是没有说修改方案是如何生成的，以及优先级如何排序。

此外，在非功能性需求中，缺乏一些必要属性，如正确性等。

1. 需求是否一致？

较为一致。

1. 需求是否可理解？

可理解，但是需要对方使用过某些IDE，才能更清楚地了解。

同时还需要具备一定的背景知识，比如了解“云端仓库”等概念。

1. 需求是否明确？

不够明确，比如，基础功能中的“给出错误提示”没说清楚是哪种提示，没有说明是提示错误还是提示错误所在位置。

又如“非功能性需求”中的“高效性”提到“IDE运行和编译代码时效率高、开销低”，“高”与“低”没有相应标准，因此需求不够明确。

1. 需求是否可实现？

基础功能和非功能性需求许多IDE都具备，因此基本可实现。

Smart功能中有一部分功能仍处于科研阶段，如“跨平台、跨语言测试”一栏提到的自动修改测试代码以适应不同环境与语言，这样的功能目前还只是存在于论文上，因此实现起来不太可行。

1. 需求是否可跟踪？

一部分不可跟踪，比如“不泄漏用户隐私，保障代码安全”。一个系统很难说是否是严格保障了安全且没有漏洞。因此该功能的不能够彻底地跟踪审查。

而基本的编写、编译、Debug功能，界面UI，辅助开发的一些可视化功能是可以跟踪的。

1. 需求是否易于修改？

易于修改。在非功能性需求中，有“可扩展性”，即该IDE是由基础平台和插件共同构成的。因此修改的需求可通过增加或者修改插件进行修改。

1. 需求规格说明文档是否完整？

不完整，缺少很多模块，比如缺少“引言”、“综合描述”这些需求规格说明文档应具有的基本模块。

1. **修改建议（基础功能版）:**

**SmartIDE需求规格说明文档**

1. **引言**

1.1 目的

本文档为SmartIDE而编制，该应用旨在建立一个智能的集成化开发平台。在完成了IDE市场的前期调研，同时与多位IDE开发者和审核人员进行了全面的分析和探讨的基础之上，提出了这份SmartIDE需求的说明书。

本文档的编写为下阶段的设计、开发提供依据，为项目组成员对功能等需求的详尽理解，以及在开发过程中的协同工作提供强有力的保证。同时本文档也作为项目评审验收的依据之一。

1.2 预期的读者和阅读建议

本文档面向SmartIDE项目组全体成员，项目评审验收组的审核人员以及使用该应用的广大企业和个人开发者。阅读时，使用文档目录可以快速找到相应内容。

1.3 产品的范围

本文档为SmartIDE做了全面细致的用户需求分析，明确所要开发软件的功能、性能和界面。

1.4 参考文献

集成开发环境：https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83/298524?fromtitle=IDE&fromid=8232086

**2. 综合描述**

2.1 产品的前景

随着计算机技术在人们生活中的强力渗透，软件市场的需求也变得越来越大。在此背景下，一个智能的集成开发环境将为程序员编写软件带来极大的便捷，也将节省测试和维护的成本。前景十分可观。

2.2 产品的功能

本产品集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。

2.3 用户类和特征

主要面向企业、团队和个人开发者，其特征是需要智能化的集成开发环境来减少人力成本并增强软件的可靠性。

2.4 运行环境

兼容不同操作系统：Windows、Mac、Linux，及不同的硬件条件：32位、64位计算机。

2.5 设计和实现上的限制

由于运行环境的不同，部分功能可能受限。

2.6 假设和依赖

假设该运行环境稳定，没有除软件自身外的因素使软件崩溃。

**3．外部接口需求**

3.1 用户界面

3.1.1用户界面应包括框架和组件两部分。框架为IDE的界面提供支撑功能，可以在界面框架上自定义地添加组件。

3.1.2组件包括但不限于：工具栏、代码编辑框、文件导航窗口、调试窗口等。

3.1.3支持个性化自定义

* 提供多种界面主题供用户选择。用户也可以自行编辑界面主题。
* 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。
* 工具栏和编辑界面排布合理，用户可以自行拖曳和增减。
* 自定义设置可导出和导入。

3.1.4快捷键操作

* 支持快捷键和纯键盘操作，快捷键也可以自定义。

3.2 硬件接口

3.2.1支持32位和64位计算机。

3.3 软件接口

3.3.1支持与MySQL数据库连接。

3.3.2支持Windows、Mac、Linux操作系统。

3.3.3支持Python、C、C++、Java、JS等编程语言库。

**4．系统特性**

4.1 说明和优先级

优先级从高到低顺序为：

1. 正确性
2. 稳定性
3. 安全性
4. 高效性
5. 兼容性
6. 可扩展性和可维护性
   1. 激励/响应序列
      1. 用户可以通过IDE的UI界面进行鼠标操作和键盘操作。IDE以用户操作为激励，先响应核心功能（如编译、调试等），再响应界面变化和更新。
      2. 在打开的IDE内部设置命令行，可以通过内部命令行进行编译和UI界面响应。
   2. 功能需求

4.3.1代码编辑

* 自动补全：变量名、类型名在编辑时，有可选可供自动补齐。
* 自动对齐：代码自动规格化，空格对齐。
* 拼写检查：检查变量名、类型名是否有误。
* 语法检查：检查语法是否合法，给出语法错误提示和错误位置。
* 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的搜索和替换。
* 查看符号定义：在代码编辑界面，可以通过点击，查看该符号的定义。

4.3.2编译

* 根据语言自动与编译器连接，传递信息。
* 可以自定义编译优化等级。

4.3.3运行

* 在图形界面实时并且显示资源占用情况，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。

4.3.4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和相关功能。
* 可以自定义调试器的监视器内容。

4.3.5支持多种编程语言

* 支持C、C++、JAVA、Python等多种编程语言。

4.3.6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。

1. **其它非功能需求**

5.1性能需求

5.1.1正确性

* IDE运行时保证文件、编译和调试结果的正确性。

5.1.2高效性

* IDE运行和编译代码时效率不低于市场平均水平。UI界面跳转时间小于1S。

5.1.3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。

5.1.4 稳定性

* 无已知可导致崩溃的bug。
  1. 安全设施需求

5.2.1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. 安全性需求

5.3.1为云端仓库设置分级权限，根据权限保护用户隐私。

5.3.2 保护本地隐私数据的安全性，加密保存。具有防范攻击的机制。

5.4 软件质量属性

5.4.1具有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。
* 有提供给维护人员维护的接口。

5.5 业务规则

5.5.1本IDE提供免费的社区版本，可以供个人开发者使用。

5.5.2购买专业版的企业、团队和个人可使用本IDE的专业版本。

5.6 用户文档

· 5.6.1有具体详细的新手教程和用户帮助文档。

5.6.2有在线帮助和教程。