

Smart IDE 需求分类分析

(1)业务需求

从公司老板或者项目经理的角度来说,开发一个尽可能满足程序员编辑代码开发项目过程中的各种编辑习惯和需求的 IDE 是业务需求,这个 IDE 需要突出'Smart'的特点,即开发的功能应该尽量像'人'一样贴心地了解 and 满足用户的各种编辑需求

(2)用户需求

对于使用 Smart IDE 的用户群体,也会有不同的需求,在第二次访谈中划分细化了可能的几个用户群体,对于不同身份的用户,他们对于 IDE 的需求有如下的区别,包括功能性需求,以及侧重点不同的非功能性需求。

• 学生群体

- (1) IDE 轻便, 易安装和使用
- (2) 能满足基本的编译和调试功能
- (3) 代码检索和补全功能
- (4) 对于调用的库函数, 可以给出用法和实例提示
- (5) 简单易懂的编译报错信息
- (6) 良好的代码缩进

• 企业/工程师

- (1) IDE 功能要齐全, 支持大型项目的开发, 编译和运行
- (2) 可靠性, 有详细的运行异常和报错日志信息进行 bug 排除
- (3) 安全性, 在 Smart IDE 开发的企业级项目代码应该能确保不会因为 IDE 而泄露出去, 或者是类似于文档编辑权限的代码编辑权限, 阻止非授权用户使用, 以及恶意攻击
- (4) 兼容性, 要支持不同平台不同操作系统下都能兼容 Smart IDE 的使用
- (5) 高效性: 保持高速处理的能力
- (6) 支持自定义插件, 对 IDE 进行功能扩展

• 维护人员

(1) 交互性, 考虑到维护 Smart IDE 的维护工程师也算作用户的一部分, 要提供 Smart, 良好的反馈机制, 比如普通用户对于 IDE 的体验和意见能通过 Smart IDE 迅速反馈到工程师, Smart IDE 要具有类似论坛的交互窗口, 便于及时收集意见

(2) 可维护性, 在接到修改需求之后, 应该能在 1~2 天内迅速完成设计并上线, 需要 Smart IDE 提供良好的故障追踪功能

• 通用的 Smart 需求

1.功能需求:

- (1) 能够完成源代码编辑
- (2) 能支持大部分常用语言的代码编译和运行
- (3) 支持大量的插件

2.非功能需求:

(1) 易用性:代码智能补全, 为了强调'Smart'特点, 最好是能根据用户平时的代码编辑习惯显示常用的几个补全, 减少用户选择某项来补全的时间

(2) 易用性: 源码检索, 代码跳转功能, 包括本地和联网检索

(3) 易用性: 在编辑代码中经常要进行删除, 撤回, 复制, 粘贴, 编译运行等繁琐操作, 要提供这些功能的快捷键操作, 特别地, 可以由用户自定义地设置他习惯的快捷键位, 因为目前很多 IDE 产品的快捷键都不相同, 用户在使用不同的 IDE 时需要适应不同的快捷键操作, 非常不合理

(4) 可拓展性: 支持不同功能的插件, 支持多语言的编译和运行, 可根据用户自身需求, 安装不同的插件, 定制专属功能的 Smart IDE

(5) 功能性: 代码结构可视化功能, 例如可显示清晰的项目文件目录结构, 或者是能便捷生成代码结构的结构图或者树状视图等等

(6) 高效性: 支持高效的文件或代码导入和导出功能

(7) 高效性: 高性能的文本搜索功能和插件搜索功能

(8) 可移植性: 兼容 Windows, Mac, Linux 系统, 能在不同的平台使用, 应该满足或兼容大部分插件或者扩展, Smart IDE 在设计时应当考虑到多平台多语言的情况下如何兼容和运行插件或扩展

(9) 鲁棒性: 不能频繁出现内存泄漏, CPU 占用率过高, 窗口崩溃等异常情况