

# 一、需求

## 业务需求：

企业内部使用，不考虑商用。

## 功能需求：

提供软件集成开发环境。

/\* “集成开发环境”这个说法不够明确，过于笼统，建议给出具体定义和具体标准。 \*/

有健全的文本编辑功能。

/\* “健全”词义不明确，需要给定“健全”的标准。 \*/

支持插件。

/\* 对插件的类型、大小、数量、安全性等有无要求？需要明确指出。 \*/

支持市面上主流高级语言和汇编语言。

/\* “市面上主流”并不明确。请指定高级语言的具体类型以及版本，汇编语言也需指定。

\*/

调试时运行状态可视化。

/\* 哪些状态可视化？对可视化的风格有无具体要求？需要明确。 \*/

## smart 功能：

**名称自动联想功能：**当输入一个名称（如变量、函数等）时，会根据已输入的字符联想可能需要的全名供用户选择，省去输入全名的麻烦。

/\* 应给出实例便于理解。 \*/

**函数（方法）自动补全代码功能：**例如使用 C++时在一个类中编写 `get` 某成员变量的函数，本产品能够直接补全其功能代码，即返回这个成员变量的值。

/\* 应给出实例便于理解 \*/

**名称拼写检测：**检测用户的命名是否是正确英文拼写，若有误则以醒目的下划线或背景色标注。应当具有一定智能，如果命名显然非英文风格，则不予标注。另外也应允许用户手动局部地取消标注，或全局地关闭该功能。

/\* 对“正确英文拼写”的检查，需要指定参照的英文标准或英文单词库，或它们的集合。  
判断一个命名是否属于英文风格不易实现，应当降低要求。 \*/

**字符染色功能：**能够自动对字符进行染色。同种类型的字符串颜色相同，不同类型的颜色不同，相对于背景色更醒目。

/\* “字符”和“字符串”的含义并不相同，用在此处容易引起混淆，建议修改表述。未说明是否所有内容都需要分类并染色。建议结合实例说明。 \*/

**同义字符高亮功能：**当用户选定一个字符或字符串时，应当将同义的字符或字符串高亮显示。如选定局部变量 *i*，该部分与之所指相同的变量 *i* 应当全部高亮，而其他部分的意义不同的 *i* 则不变。

**代码段的优化提示功能：**简单地分析代码内容，将可优化的点提示给用户。如 C++ 中大段的“if-else”语句提示改为“switch”，某段代码逻辑上永远不可能执行到提示用户可删除该段。

**绘制模块或类的依赖关系图：**将模块或类之间的关系可视化。例如 C++ 中绘制类之间的继承、组合等等的关系。

/\* 建议对可视化提出具体要求，如颜色、图形、需要呈现哪些信息等。 \*/

**非功能需求：**

支持 Windows 和 macOS。

/\* 需要具体到 Windows、macOS 的哪个或哪些版本。 \*/

软件外观有科技感。

/\* “科技感”的概念较为主观。建议给出实例，或人工绘制样板提供参考。 \*/

能以树形结构展示文件。

/\* 需要明确在树形结构中可以对文件进行哪些操作。 \*/

提供便捷工具栏。

/\* 需要给出非常具体的标准，工具栏的大小、可否拉伸和移动、包含哪些功能、能否自定义功能、具有什么样的外观、能否自定义外观、能否开启多个工具栏等等。建议此功能需求另外单独说明。 \*/

软件可靠性好。

/\* 需要给出“可靠性好”的标准，例如通过某项可靠性测试等。 \*/

### **约束：**

半年推出可用的测试版。其后的一年内逐渐维护成型，维护期间要定期汇报。需要长期维护。

/\* 这里的需求不明确。“可用”的标准是什么，需要明确。“成型”的标准是什么，同样需要明确。“定期”指代多长的周期，需要给出明确的要求。“长期”指代的时间长度也需要明确。  
“汇报”的内容需要规定。“维护”包含哪方面的维护，需要明确。 \*/