**需求评审**

**评审对象**

符冰瑶的软件需求工程第二次作业

**评审结果**

**一.总体评审：**需求不完整：缺少IDE的基本功能需求：代码编辑、编译、调试、图形化界面等功能。

**二.分条评审：**

* 1. **需求：**

变量/函数定位：在代码中双击变量/函数或者在搜索框搜索函数/变量时，立即跳到定义点；按下“→”键，跳到声明点；用阴影标记项目中所有的引用点标记，按下⬇，跳到下一个引用点。

**评审意见：**

1. 不明确。未说明变量/函数未定义和未声明时，用户进行相关操作后系统的处理方法。
2. 语法错误。“用阴影标记项目中所有的引用点标记”应改为“用阴影标记项目中所有的引用点”。
3. 这条需求与“智能搜索”不一致，存在冲突。该需求中使用的搜索框与“智能搜索”中使用的搜索框是同一个吗？若是，那么在用户搜索时如何区别用户的意图并显示相应内容？
   1. **需求：**

基于人工智能的智能预测:

1. 当程序员编写代码时，IDE能够当前的代码上下文和模式，利用人工智能技术，从网上的源代码或者程序员自己的代码学习、预测程序员此时最可能需要的 API，提供一个API参考列表。
2. 利用人工智能学习用户的行为，如：输入习惯、代码风格、偏好、常犯错误等。根据学习的结果，在用户写代码时，可能发生错误的地方自动提示，按照用户习惯调整推进、自动填充空格、换行等，同时保护用户隐私。

**评审意见：**

1. 不易理解、具有二义性。第（1）条需求中的“IDE能够当前的代码上下文和模式”中，当前的代码模式具体指的是什么？
2. 不明确、不可验证。第（2）条需求中的“保护用户隐私”具体应该做到什么？ “保护隐私”应该如何验证？
   1. **需求：**

监测、可视化显示栈：

1. 代码调试时，监测每一个函数的栈从函数调用到函数返回的变化情况。
2. 进行可视化处理，以图的形式显示。

**评审意见：**

1. 不正确。除非使用断点调试，否则程序运行过程中函数调用栈的变化速度相当之快，即使能够实时的显示栈的变化情况，用户也难以看清变化过程。
2. 建议修改为：代码调试时，监测并记录每一个函数的栈从函数调用到函数返回的变化情况和发生变化时的代码行数，进行可视化处理，以图的形式显示。用户可异步地对其进行查看。
   1. **需求：**

代码可视化和逆视化：

1. 静态可视化：
2. IDE能够分解出代码中的各语法成分，分析它们间的逻辑关系和代码的语义、结构。
3. IDE定义各种可视化结构对应的标准视化形式。可视化结构包括类、源文件、复杂数据结构、指针及其指向等。视化形式包括流程图/流程树、数据结构图、类图、反映指针的指向情况的图等。
4. IDE能够根据代码生成相应的标准视化形式。
5. 动态可视化：
6. 在调试/测试/执行过程前，IDE对代码中的所有变量进行可视化，生成该程序的程序状态（代码中所有变量及其值的集合）结构框架。
7. 在调试/测试/执行过程中，IDE记录下每一个（或者指定位置的）程序状态（向框架中填值）。
8. 最终生成一个可用类似PPT放映的形式播放的状态转移动画，程序员每点击一次鼠标，就播放下一个状态。例如：动态播放插入排序的过程。
9. 逆视化

由程序员输入一个符合IDE标准定义的视化形式，IDE根据输入的视化形式，生成相应的可视化结构的代码。

**评审意见：**

1. 对于（3）逆视化：
2. 可行性较低。该需求中描述的逆视化，实际上是指IDE支持模型驱动开发（MDD）。若如该需求中所述，由IDE自己定义各种“视化形式”，则会大大增加用户的学习成本，使得该项功能难以推广，通用性差。IDE应该支持市场上现有的软件建模工具，如UML等，直接使用其成果作为将模型转化为代码的输入。
3. 应该修改为:IDE支持模型驱动开发（MDD），用户输入由UML等软件建模工具导出的模型，IDE将其转换为合适的代码。 然后，IDE以高度自动化的方式调试和测试模型驱动的代码。
4. 对于(2)动态可视化：
5. 不易理解。
6. 部分需求没有必要。在程序运行过程中，系统需要记录下所有的程序状态，若程序规模较大，则工作量巨大、十分影响性能。没有必要在执行和测试过程中，进行这项工作。也没有必要对所有的变量、所有的语句进行可视化，仅对用户需要的部分进行可视化计即可。
7. 修改建议：缩小动态可视化的范围。
   1. 仅当调试时，用户可选择进行动态可视化。
   2. 由用户选择需要进行可视化的变量和代码段。
8. 对于(1)静态可视化：

可行性较低。目前的技术较难实现由机器分析出代码中的逻辑关系。并且如上述所言，IDE应当支持市场上现有的代码可视化/软件建模工具，在此基础上导出的相应的模型/视化图，应当能通用于其他平台。

* 1. **需求：**

实时错误预防：能够立即了解写代码时所做的每个更改的影响，并在用户键入代码时对其进行测试。在错误发生时立即提示/修改错误，并查看现有测试是否覆盖了新更改。

1. 实时对代码中的语句，生成执行该语句的测试输入；实时针对不同的方法，生成测试用例，并进行实时的单元测试。
2. 在代码编辑器中，在每行代码的最左端，显示它是否由单元测试覆盖/覆盖这些代码的测试是否通过。程序员可以点击查看每行代码的测试信息（如：未通过的测试名和用例、未通过的原因、人工智能给出的修改建议）。

**评审意见：**

1. 不正确。“在错误发生时立即提示/修改错误”，在错误发生时提示即可，修改应由用户完成。
2. 具有二义性。“能够立即了解写代码时所做的每个更改的影响，并在用户键入代码时对其进行测试”中的测试对象指的是谁？