

Smart IDE 需求访谈

——精化 Smart IDE 的可能需求

访谈时间：2019 年 10 月 11 日

访谈地点：仙 II -104

访谈对象：梁欣悦

访谈者&记录者：蔡悦

【访谈内容】

1.

蔡：您好，我们从最基础的需求开始吧，首先你对一个 IDE 的基本需求有什么看法？

梁：一个 IDE 的基本需求主要涵盖四个方面，编写代码、编译、运行、Debug。

2.

蔡：那么我们接下来就可以从这四个基本需求入手进行扩展，分析更具体详细的需求了。

梁：嗯。比如，“编写代码”可以包括自动补全、自动对齐、拼写检查、语法检查、代码搜索和替换、查看变量和函数的定义……“编译”功能可以选择优化等级，提供加速机制……“运行”需要生成运行日志、显示开销、使用不同的方式命令运行……“Debug”时可以进行定点调试，提示错误类型，提供修改实例等等。

3.

蔡：嗯。肯定还有一些其他的常见功能，可能是一些软件们为了使用感普遍提供的……

梁：对。诸如支持多种语言、提供良好的用户反馈机制、向用户提供“帮助”，提供官方的论坛，这些面向用户需求和着力提供愉快体验感的功能。

还有很重要的，作为一个图形界面的 IDE，友好的用户界面绝对是关键的。通常简介美观的界面是广受欢迎的，但是根据经验，用户绝对不会讨厌自定义的，当然我们也会提供自带的不同风格界面。

当然也有针对软件自身的，比如所有软件都不能避免的版本更新，还有为了扩展用户群体需要满足一定的兼容性，以及安全性、稳定性、软件性能等必须解决的问题。

4.

蔡：现在我们对一个 IDE 的需求已经比较清楚了，不过作为 Smart IDE，应该还有一些功能是非常“Smart”，而且有创新性的。

梁：是的。比如对于大型项目开发非常必要的版本控制、协同开发、代码共享等等，当然还有不管是项目组还是个人都会提供便利的云端存储。以及我们可以提供简单的代码测试功能，虽然可能不如专业的测试工具，但是胜在方便易用，即使不是专业的测试人也可以简单上手，这可以降低 debug 的劳动量。

5.

蔡：不错，现在用户们的需求已经很丰富了，不过软件开发方一定还有一些别的需求。

梁：的确。商业软件的开发最终目的是为了盈利，那么除了保证软件的质量，提升用户体验，也应该有面向管理者的“反馈机制”，使管理者可以及时方便地了解软件的销售情况、用户群体，以及改良的可能。

6.

蔡：好的，我们已经比较全面地讨论了 Smart IDE 的需求，我们的访谈也可以告一段落，感谢您的支持和配合。

梁：不客气，也谢谢你。

【需求整理】

➔ 业务需求

- ① 具有完善的用户反馈机制，可以及时方便地了解用户需求和满意度，以便对软件进行改进。
- ② 可提供在线客服，包括智能客服和人工客服，增强用户良好的体验感。
- ③ 可提供软件购买情况分析，以便了解用户群体，可以设计有针对性的销售计划
- ④ 随机向用户提供软件使用情况问卷。

➔ 功能需求

- ① IDE 的基本功能。

a) 编写代码。在此基础上可扩展：

- 1>添加注释
- 2>自动补全；
- 3>自动对齐；
- 4>拼写检查；
- 5>语法检查；
- 6>搜索代码；
- 7>替换指定代码；
- 8>光标移动至变量名或函数名时可查看变量或函数的定义；
- 9>可以选择在变量或函数使用处显示定义处的注释；
- 10>点击函数声明处可跳转至函数实现处；
- 11>支持多种语言（虽然编写代码时普遍采用英文，但是由于程序需要输入输出以及注释，所以支持多种语言是有必要的）；
- 12>可将某文件中代码直接插入正在编写的代码段中，或将文件直接加入项目中；
- 13>代码和项目可以云存储，采用账号登录即可访问云端代码和项目（同时也会在本机保存，这是必然的）；
- 14>可以选择对代码进行版本控制，可以保存、查看、编辑、搜索、删除历史版本
- 15>自动添加、修改、删除包头/头文件（例如，使用到某头文件中某个函数时，自动添加该头文件）
- 16>可以自定义常用函数，并在下次使用时直接选择并插入代码
- 17>分析代码，生成代码流程图
- 18>可进行代码标记（更换代码颜色/底色等）
- 19>支持多个程序员协同开发
- 20>支持代码共享、合并

b) 编译。在此基础上可扩展：

- 1>可选择优化等级

- 2>可选择查看/导出汇编文件
- 3>编译加速机制

c) 运行程序。在此基础上可扩展：

- 1>生成并保存运行日志
- 2>生成实时时间内存开销图
- 3>可在软件内部通过命令行运行

d) Debug。在此基础上可扩展：

- 1>断点调试（包括单步、多步调试、进入函数内部等）
- 2>错误提示（包括含链接的错误码）和提供代码修改方案实例

②程序测试。可对代码进行简单的测试，包括：

- 1>代码静态分析（检查内存泄漏等明显漏洞）
- 2>代码动态分析（测试覆盖率）
- 3>自动生成测试样例（生成输入，和输出，但无预期输出）

③ 界面需求

- 1>界面简介美观大方
- 2>高度支持自定义（但提供成套的多种风格界面）
- 3>工具栏和编辑界面可分离/合并（拖曳）
- 4>支持快捷键，包括软件和系统自带快捷键和用户自定义快捷键
- 5>支持更换界面背景（风格），包括软件自带和用户自定义
- 6>可调整显示分辨率

④用户需求

- 1>提供“帮助”，帮助第一次使用时的“使用引导”和软件使用详细说明，以使用户更好地使用软件。
- 2>提供软件论坛，以使用户群体互相交流
- 3>提供版本更新支持
- 4>提供在线客服（包括智能客服/人工客服）
- 5>提供完善便捷的反馈机制

➔ **非功能需求**

- ①（可扩展性）版本更新。修复已知的 bug，增加新功能。
- ②（兼容性）兼容同一操作系统不同版本，提供适应不同操作系统的版本。
- ③（安全性）能够抵御一定攻击，保护用户隐私和项目不被泄漏。
- ④（稳定性）无已知 bug；减少崩溃可能，设置崩溃闪退等风险应对机制
- ⑤（性能）IDE 编译和运行效率高、开销低（可选择不同的软件运行模式）。

➔ **Smart 需求**

①评估工作效率

| ID 和名称 | 评估工作效率 |
|--------|-------------|
| 首要角色 | 程序员（代码编写人员） |

| | |
|------|--|
| 次要角色 | 项目组长（项目管理人员） |
| 描述 | 程序员在使用 Smart IDE 进行代码编写时，IDE 会对其工作效率（即编写代码的过程）进行分析、评估和生成记录。此功能提供了及时效率、当日效率、本周/月（一段时间内）效率分析。程序员可根据此功能查看本人工作效率，管理人员可根据此功能管理项目进度，人员调配。 |
| 触发条件 | 在设置选项中选择开启“评估工作效率”功能 |
| 前置条件 | 1.对程序员，拥有 IDE 使用权/账号 2.对管理员，拥有管理员身份认证，且拥有对程序员们的管理权限 3.Smart IDE 软件运行正常 4.网络连接正常 |
| 后置条件 | 1.在本地生成并保存结果分析文件 2.通过网络将程序员工作效率分析结果整合发送至管理员 |
| 正常流程 | 1.程序员/管理员设置本功能开启（程序员只可开启自己的功能，而管理可选择对所有程序员开启此功能） 2.程序员正常编写代码，IDE 显示及时效率 3.程序员完成工作，关闭 IDE，生成效率分析结果，并保持至本地 4.工作效率分析文件上传给对应管理员用户 |
| 异常 | 1.设置异常，开启功能失败 2.程序员长时间开启但未使用 IDE 3.无存储空间 4.无网络 5.软件非正常退出 |

②逻辑语言自动转化高级语言代码

| | |
|--------|---|
| ID 和名称 | 逻辑语言自动转化高级语言 |
| 首要角色 | 程序员（代码编写人员） |
| 描述 | 程序员应用本功能，通过编写标准化的逻辑语言（如代数语言/几何语言/逻辑模型等），由 IDE 智能生产对应代码段或函数体 |
| 触发条件 | 选择逻辑语言生成代码 |
| 前置条件 | 1.程序员拥有 IDE 使用权/账号 2.Smart IDE 软件运行正常 |
| 后置条件 | 生成的代码段/函数体插入指定位置 |
| 正常流程 | 1.程序员选择使用逻辑语言生成代码 2.按照标准，使用逻辑语言编写输入过程；也可选择 IDE 库中提供的标准模板 3.完成编写，确认无误，选择转化的高级语言（C, C++, Java, Python 等） 4.IDE 检查逻辑语言是否语法正确，若正确则自动转化逻辑语言为高级程序设计语言，否则返回提示，转化失败 5.程序员选定插入位置 6.IDE 将代码插入指定位置，同时将原本的逻辑语言作为注释插入生成代码后，进行标注 |
| 异常 | 1.设置异常，开启功能失败 2.IDE 运行异常 |

| | |
|--|--|
| | 3.IDE 转化逻辑语言至高级语言异常 4.逻辑语言非法 5.软件非正常退出 |
|--|--|

③图形界面便捷绘制和管理

| | |
|---------------|---|
| ID 和名称 | 图形界面便捷绘制和管理 |
| 首要角色 | 程序员（代码编写人员） |
| 描述 | 便捷简明的图形界面绘制和编写关联程序，提供大量常用标准模板，易于更改设置和基础要素 |
| 触发条件 | 选择编写图形界面 |
| 前置条件 | 1.程序员拥有 Smart IDE 使用权/账号 2.Smart IDE 运行正常 3.安装匹配的图形界面插件 |
| 后置条件 | 1.IDE 调用库中的标准图形界面模板 |
| 正常流程 | 1.程序员选择编写图形界面程序 2.使用提供的图形界面素材或标准模板 3.IDE 自动生成关联代码 4.程序员调整图形界面参数，修改填充代码段 5.完成图形界面程序编写，编译运行，调试 bug 6.IDE 响应程序员后续操作 |
| 异常 | 1.设置异常，图形界面编绘模式开启失败 2.IDE 运行异常 3.调用库中图形素材或标准模板失败 4.编译/运行/调试图形界面程序失败 5.软件异常退出 |

④自动化测试

| | |
|---------------|--|
| ID 和名称 | 自动化测试 |
| 首要角色 | 程序员（代码编写/测试人员） |
| 描述 | 根据程序入口和出口，分析程序内部逻辑，自动生成期望测试样例，运行程序，生成测试结果，分析测试结果。 |
| 触发条件 | 选择对程序进行自动测试 |
| 前置条件 | 1.程序员拥有 Smart IDE 使用权/账号 2.Smart IDE 运行正常 3.安装匹配的自动测试插件 |
| 后置条件 | 1.IDE 保存测试日志 |
| 正常流程 | 1.程序员选择开启自动测试 2.选定待测试程序或局部函数 3.选定测试模式/方法（例如语句覆盖，分支覆盖等等） 4.IDE 运行插件，生成测试样例集，对选定部分进行自动测试 5.生成测试结果，分析测试结果，给出程序可能存在错误及缺陷 |
| 异常 | 1.设置异常，自动化测试功能开启失败 2.IDE 运行异常 |

| | |
|--|---|
| | 3.生成测试样例集失败 4.自动运行测试程序出错 5.软件异常退出 |
|--|---|

⑤跟踪运行程序

| | |
|---------------|--|
| ID 和名称 | 跟踪运行程序 |
| 首要角色 | 程序员（代码编写/测试人员） |
| 描述 | 对正在运行的程序进行内存、接口调用、进程、外部设备调用等进行跟踪和记录分析 |
| 触发条件 | 程序员选择对程序进行运行跟踪 |
| 前置条件 | 1.程序员拥有 Smart IDE 使用权/账号 2.Smart IDE 运行正常 3.程序可运行 |
| 后置条件 | 1.Smart IDE 保存跟踪日志 |
| 正常流程 | 1.程序员选择跟踪运行程序 2.IDE 响应设置 3.程序开始运行，同时进行跟踪，同步显示跟踪结果 4.记录分析并生成跟踪文件 |
| 异常 | 1.设置异常，跟踪运行程序功能开启失败 2.IDE 运行异常 3.程序运行异常 4.软件异常退出 |

⑥自定义程序框架

| | |
|---------------|--|
| ID 和名称 | 自定义程序框架 |
| 首要角色 | 程序员（代码编写人员） |
| 描述 | 可以使用 IDE 提供的模板框架，或者根据已有程序/项目，选择生成并保持自定义框架，以便下次开发相似项目；在联网状态下可选择分享个人自定义的框架或查看使用他人分享的框架 |
| 触发条件 | 程序员选择使用自定义程序框架 |
| 前置条件 | 1.程序员拥有 Smart IDE 使用权/账号 2.Smart IDE 运行正常 3.安装自定义程序框架插件 4.网络正常 |
| 后置条件 | IDE 生成运行日志 |
| 正常流程 | 1.程序员自定义程序框架 2.选择已有程序项目，选择待生成保存的框架部分 3.IDE 相应操作，生成自定义程序框架 4.新建项目，选择自定义框架或 IDE 库中提供的模板框架 5.开始在框架基础上编写代码 |
| 可选流程 | 1.登录账号，网络情况正常状态下，可选择分享自己的框架 2.登录账号，网络情况正常状态下，可选择查看使用他人分享的框架 |
| 异常 | 1.设置异常，自定义程序框架功能开启失败 2.IDE 运行异常 |

| | |
|--|--------------------|
| | 3.网络异常 4.软件异常退出 |
|--|--------------------|

⑦项目管理和任务分配

| | |
|---------------|---|
| ID 和名称 | 项目管理和任务分配 |
| 首要角色 | 管理员（项目管理人员） |
| 次要角色 | 程序员（代码编写人员） |
| 描述 | 管理员可对较大的、需要多人合作的项目进行管理，包括任务分配，验收完成代码部分，选择是否通过，查看项目进度，版本迭代等； 程序员可以接收任务分配，完成任务，提交程序 |
| 触发条件 | 管理员选择项目管理和任务分配模式 |
| 前置条件 | 1.程序员拥有 Smart IDE 使用权/账号 2.管理员拥有 Smart IDE 使用权/账号，用管理员认证，并与程序员进行关联 2.Smart IDE 运行正常 3.安装项目管理和任务分配插件 4.网络正常 |
| 后置条件 | IDE 生成项目日志 |
| 正常流程 | 1.管理员选择开启项目管理和任务分配模式 2.管理员编写项目简要说明，确定项目需求，决定项目模块 3.管理员向下分配程序编写任务，设置提交期限 4.程序员接受任务，完成任务后提交代码 5.管理员审核程序员提交的代码 6.管理员可查看项目进度，代码迭代情况等 |
| 异常 | 1.设置异常，项目管理和任务分配功能开启失败 2.IDE 运行异常 3.网络异常 4.软件异常退出 |