

# Virtualme开发者个人画像-细化方案

**目的：**服务的对象：帮助用户了解自己，或支持智能体完成下游任务

**暂无方案：**较为复杂，暂无计算方案

**当前数据不支持：**vscode+插件的方案无法获取相关数据源，需要IDE提供相应的API

## 基础信息

目的	指标	数据源和算法
用户自身了解；codechat	常用编程语言占比	可以读打开过的repo直接统计，可以画图。
用户自身了解；codechat	技术栈：常用哪些框架/库	可以读打开过的repo直接统计，结合常用插件生态，可以画图。
用户自身了解	代码阅读/开发/修复行为占比：参考github profile的陈列。	可以画图统计

## 技术能力

目的	指标	数据源和算法
用户自身了解	代码补全依赖度：完成一段编码时自动补全的比例	当前数据可识别tab补全或手敲
用户自身了解	快捷键使用偏好：重构、tab、撤销重做等	可直接统计，可以画图。
用户自身了解	IDE功能依赖程度	结合快捷键和IDE功能使用频率/覆盖率/偏好
用户自身了解；codechat	AI助手依赖程度	人智交互频率。当前阶段无数据可记录。
用户自身了解；代码生成/重构	代码片段复用频率：保存和调用自己写的代码	复用分为：跨项目代码、项目内封装。比较抽象的指标。 <b>暂无方案</b>
用户自身了解	代码生产力：每个周期内提交的代码量	数据带git记录，可以计算一个周期内的代码提交情况。
用户自身了解；codechat	语言熟练度：对每种语言的掌握程度	需要结合撤销重做/生产力/语言feature使用量/代码质量/构建成功率等多重指标？
用户自身了解；codechat，代码生成	项目复杂度：常用repo是否涉及多语言、多领域技术	可以读打开过的repo，统计涉及的编程语言数量、标志性配置文件/文件夹识别对应框架

目的	指标	数据源和算法
用户自身了解；codechat	缺陷修复效率：从错误发生到解决的平均时间	如何判别发生了错误？以及错误的解决？当前可以判别终端命令的成功与否。结合识别的partten行为模式转移情况。只管IDE能看到的部分。
用户自身了解；codechat	单元测试编写频率	如何识别单元测试的覆盖率？ <b>暂无方案</b>
用户自身了解；	构建编译测试能力：终端命令成功率	可直接统计
用户自身了解；	合并冲突处理频率与效率	如何判别冲突处理，有IDE接口也有手动处理，也有使用第三方软件处理的。 <b>当前数据不支持</b>

## 工作习惯

目的	指标	数据源和算法
用户自身了解；	每天/每周/每月的活跃时刻与长度：参考github的粒度是天，我们可以做到小时分钟	数据带时间戳，可以直接算。可以画热力图。
用户自身了解；	提交粒度：小步提交/大块提交，持续集成/批量提交	数据带git记录，可以直接算
用户自身了解；	本地/远程操作偏好	远程需要识别repo是不是ssh远程仓库，以及terminal是不是ssh终端。 <b>当前数据不支持</b>
用户自身了解；	注释编写频率：体现对代码可读性和可维护性的关注	代码快照+LLM可以识别注释编写记录
用户自身了解；codechat	插件生态	安装的插件都属于哪些类型，可以画图
用户自身了解；代码调试	代码调试的频率/调试断点的密度： <b>有助于支持AI帮助调试</b>	<b>当前数据不支持</b> ，可以画图

## 学习能力

目的	指标	数据源和算法
用户自身了解	新repo探索/理解效率	结合程序员认知负荷+Developer Behavior论文，可以量化在repo探索时的认知指标
用户自身了解	新插件尝试频率	插件安装记录
用户自身了解	新技术栈尝试频率	可以读打开过的repo直接统计

# 代码质量

需要使用静态分析工具，跟我们自身的数据源关系不大。IDE内统计这个指标的意义不太大？

指标
圈复杂度
注释覆盖率
单元测试覆盖率
缺陷密度
代码风格一致性
耦合度与内聚度

## 参考资料和标准

1. [SWEBOK \(Software Engineering Body of Knowledge\)](#)
2. Github Profile提供的个人repo统计指标
3. [SFIA](#)
4. ISO/IEC 24773 软件工程专业能力认证和评估环节
5. PSP、TSP
6. [CMMI能力成熟度模型](#)