



vivo研发效能量化提升与平台建设之路

张文博





目录

- 1 研发效能如何量化
- 2 vivo研发效能平台建设之路
- 3 敏捷与OKR实践
- 4 研发效能体系展望





01

研发效能如何量化



研发效能是什么

持续高效的交付有价值的需求

用户价值

- 需求的市场价值；
- 优先级；

交付效率

- 需求吞吐率；
- 需求状态分布；
- 迭代完成周期；

交付质量

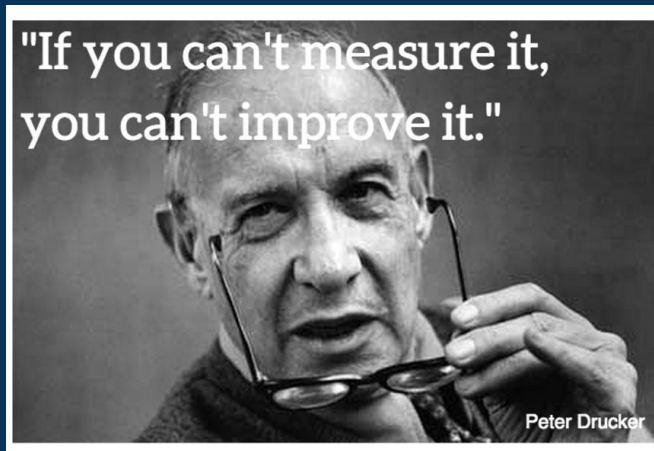
- 缺陷趋势；
- 缺陷平均修复时长；
- 线上缺陷遗留数量；

项目跟踪与风险

- 工作进度；
- 人力排期；
- 工时偏差；
- 延期风险；



软件研发效能的不同声音



VS



软件开发效能度量的误区

过度依赖工时、代码行数
等易于获得的数据

忽视团队成员的发展需求

缺乏统一的标准，主观
性强

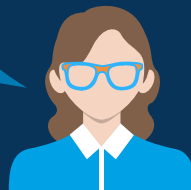


手段
不等于
目标

缺少过程工具，指标数
据人工录入

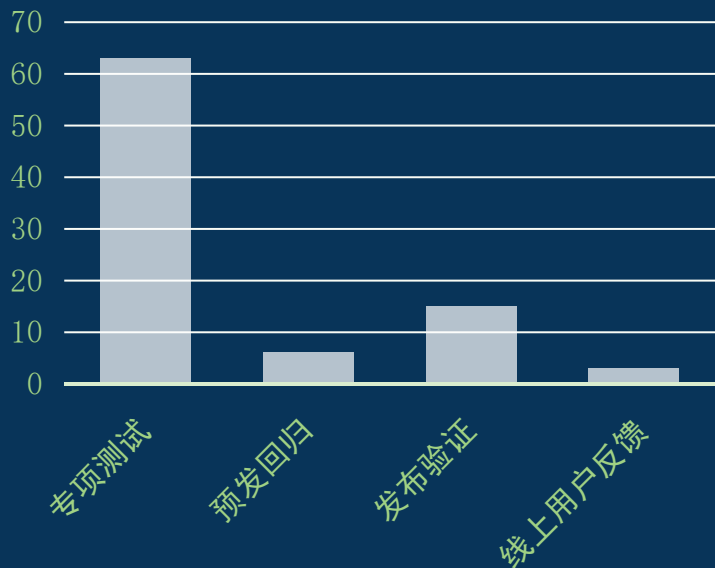
度量指标的滞后性

单一评价维度



案例：研发质量问题专项分析

各阶段缺陷数



如何把发布验证的bug提前到预发回归阶段？





02

vivo 研发效能平台建设之路



研发效能实践方案



数据采集

- 迭代过程数据
- 成员任务数据
- 项目质量数据
- 需求周期数据



指标分析

- 指标有效性
- 策略合理性



模型指导

- 效率模型
- 质量模型
- 成本模型
- 业务模型



风险决策

- 延期风险
- 趋势/预测
- 回馈业务



一体化的项目研发流程



系统架构

项目规范化、流程自动化、过程可视化



度量因子标准化

工时估算的问题：

1. 每个人能力不同，缺乏统一标准；
2. 工作项重新分配给其他人时存在项目风险；
3. 不能突出复杂度；
4. 不能代表总工作量；



引入故事点

这是一个测试任务

状态: 完成 处理人: 文峰 预计时间: 2021-09-14 ~ 2021-09-17

工时

预计工时 5故事点 已用工时 5故事点 添加工时花费

剩余工时 0故事点 预估偏差 0故事点

进度 100%

- 张文博 2021-09-17 花费工时 1 故事点
代码分支合并
- 张文博 2021-09-16 花费工时 1 故事点
自测以及编写单元测试
- 张文博 2021-09-15 花费工时 2 故事点
接口对接，数据适配
- 张文博 2021-09-14 花费工时 1 故事点
数据库以及增删改查逻辑开发

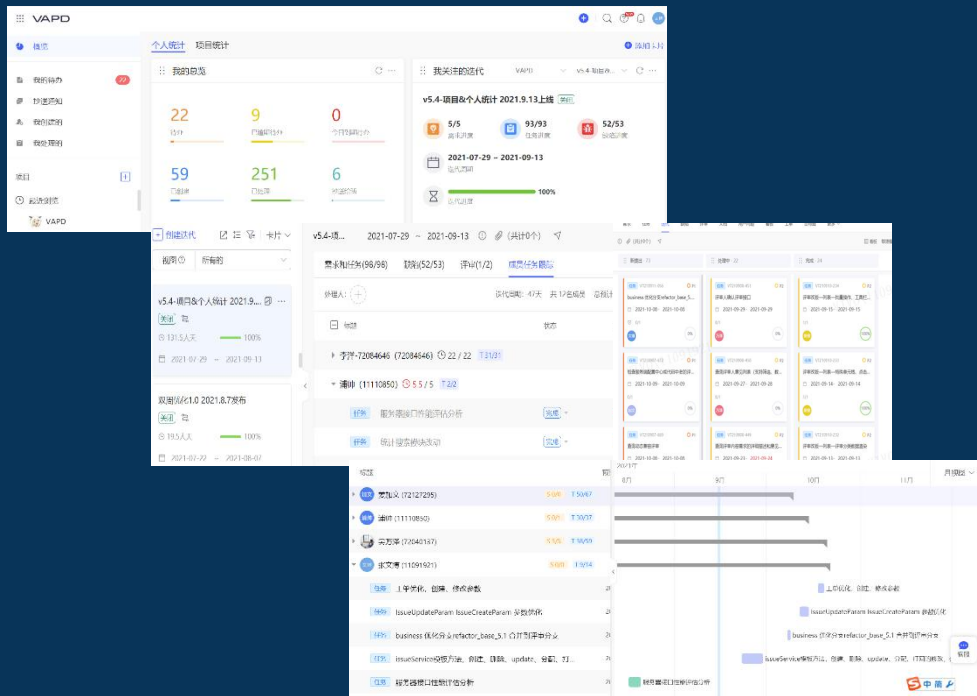
故事点的建议：

1. 项目共同制定0.5人日的基准故事点；
2. 故事点取值1, 2, 3, 5；
3. 把任务按照故事点进行更新总结；



项目过程可视化

- 这个迭代包含哪些需求任务？
- 某个任务当前处于什么阶段了？
- 哪些人有空余时间支撑一个紧急任务？



指标数据系统化

项目分析

项目总览、项目进度、延期风险

成员分析

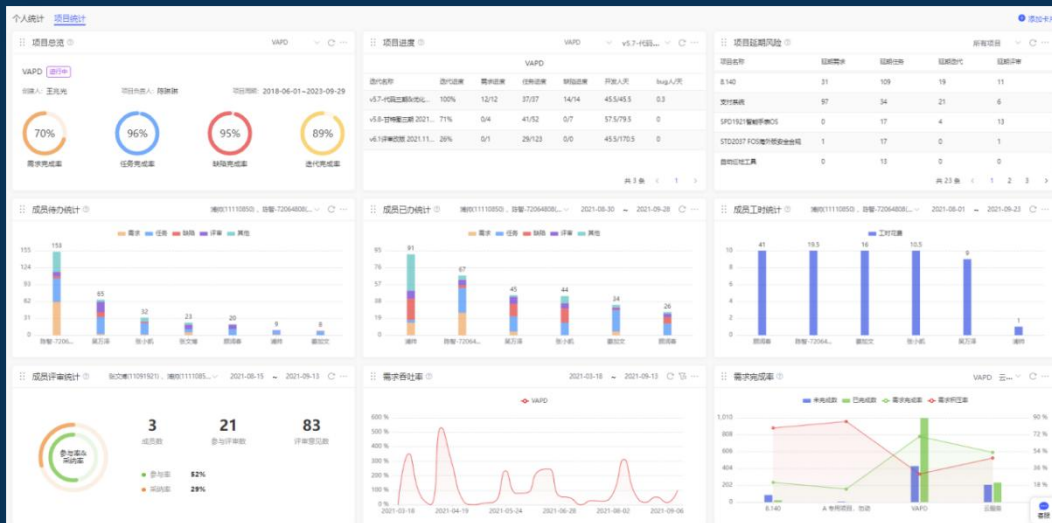
待办、已办、工时、评审，及人力排期

需求迭代

吞吐率、完成率、需求状态分布，缺陷趋势

个人统计分析

工作台待办与关注，工时、已办、缺陷等



1. 统计指标类型20+, 接口平均耗时100ms
2. 沉淀项目数据量1000W+
3. 扩展性强, 支持其他统计类型





03

敏捷与OKR实践



识别“假敏捷”

敏捷软件开发宣言

我们一直在实践中探寻更好的软件开发方法，
身体力行的同时也帮助他人。由此我们建立了如下价值观：

个体和互动 高于 流程和工具

工作的软件 高于 详尽的文档

客户合作 高于 合同谈判

响应变化 高于 遵循计划

也就是说，尽管右项有其价值，
我们更重视左项的价值。

项目中敏捷误区：

- 拥抱变化=需求频繁变更
- 依赖晨会进行沟通同步
- 对需求进行不合理拆分
- 敏捷就是更紧密的迭代



OKR实践



措施：1. 做好自测记录，写好单元测试。
2. 多站在其他角色的角度去思考，去使用产品。比如：开发站在测试角度，测试站在实际用户度。

O：版本缺陷率低于0.1

KR1：发布当天点检缺陷数量低于5个；

KR2：重新打开的缺陷数占比小于5%；

KR3：每人每次测试用例评审，提出有效意见数不低于3个；

使用流量回放工具，常用接口100%自动化回归，减少集成测试阶段工作量；

测试用例区分优先级，并且考虑异常场景，比如：历史数据迁移；

提交代码时候，对于新增Service的public方法要写单元测试；

预上线阶段增加内部使用点检，对于发现有效问题最多的同事给予奖励；



04

研发效能体系展望



研发效能行业发展趋势

洞察与成长

- 团队问题诊断与建议；
- 学习成长；



DevMind

工具更“懂”研发，更好更精准的提供服务：

- 构建研发画像：分析成员的研发行为特点，项目的指标，生成标签；
- 赋能团队：精确定义项目风险、团队问题，挖掘潜力；

智能化分析

- 自动分析处理；
- 历史故障形成解决方案；



AIOps

引入算法，让经验成为工具：

- 系统问题预测与自愈；
- 专家经验沉淀到案例库；
- 模型训练；

流程与规范

- 减少人工操作；
- 开发运维一体化；



DevOps

Aone , Teambition , TAPD



平台规划

以提效为中心，
融合研发全领域知识和经验的智能洞察平台。

项目
场景

团队
视角

项目概览

团队资源

迭代度量

风险预警

个人
视角

工作助手

团队知识库

计划与排期

核心
能力



DevOps

移动端



流转编排

自定义配置



分析统计

数据
平台

指标中心

规则中心

事件中心

模型中心

研发数据资产

运算引擎（离线分析、算法）



提问时间

谢谢大家！



