

敏捷的本质

白杨

<http://baiv.cn>

2010-10-29

重新认识到软件研发的工艺性

即：软件开发过程除了有类似于汽车、手机制造的工程性之外，也包含了类似绘画、雕塑创作的艺术（或技艺）成分。为此：

- ★ 正视不同开发人员之间，个体生产力和产出质量上的巨大差异（最典型的例子：我们经常能看到一个程序员需要两周时间才能完成的任务，交给另一个程序员来做的话，可能只需要 2、3 个小时）。

进而认识到程序员不同于流水线上可替换的零件，从而重视并强调开发人员的专业素养和团队协作能力等各项技能指标。

- ★ 强调人和交互比流程和工具更重要。这其中不仅涵盖了开发人员之间的交流，也包括了开发人员和客户间的合作。

拥抱变化

归根结底，敏捷所致力于解决的两个根本问题都是与快速适应变化有关的，其一是：如何以最小的代价快速适应不断变化的市场和需求；另一个问题是：如何在客户仅有一个美好愿景，但缺乏具体需求的前提下，以最小的浪费逐步开发出使客户满意的产品。

由此可以看出，敏捷骨子里的基因是拥抱变化，敏捷强调“可用的软件重于完善的文档”、“响应变化重于遵循计划”等务实的价值观。在实践上，敏捷提供了四种手段保证其能够快速响应变化：

1. **用户故事**：用户故事将不同的需求按照用户的使用案例进行分组，理清需求之间的关联性。从而区分出哪些功能是最重要、最稳定，需要尽早实现的；哪些是次要的，甚至是潜在（即：很有可能在以后发生变化）的。

通过整理用户故事，我们可以优先挑选最重要、最稳定的需求予以实现，避免那些次要的、变化风险较高的需求。并且对于用户自己对需求也不是很清楚的情况（其实这才是大部分用户的常态，大部分软件产品都是从一个

个模糊的方向性创意开始的), 用户故事也可以起到逐步迭代式开发的作用。

2. **迭代交付 (其实也可以叫“持续交付”?)**: 即我们常说的“早交付、常交付”将需求放在不同的迭代进行交付, 每次交付都让用户有机会检验真实产品与其脑中的理想产品间差异, 并及早做出沟通和调整。越早调整, 造成的返工和浪费就越少。

除此之外, 迭代式交付也利于开发人员与客户合作, 使客户可以在真实的使用体验中较为准确地提出自己的下一步需求。毕竟比起抽象的概念和想象来说, 人类总是更善于处理实际的问题。通过对实际产品的操作和体会, 用户更容易判断其下一步演进方向 (或者下一步需要改进的方面)。

3. **持续重构**: 持续地对产品的实现和架构进行优化和重构, 使产品保持在一个健康的状态下; 让代码易于阅读、理解和维护; 使组件之间保持高内聚和低耦合, 尽可能缩小需求变化的影响范围, 降低需求变化所引起的额外工作量。
4. **持续集成**: 依靠自动化测试和构建工具搭建持续集成环境。尽早发现并修正每次提交中包含的缺陷。