

6.Scrum

6.1.瀑布模型的真相

Winston W.Royce首次提出了著名的传统瀑布方法，但Royce 之所以提供这个模型，就是要拿它当靶子说明不能这样做软件开发！然而，不知怎么的，偏偏是描述瀑布模型的部分受到听众的追捧，并由此变得广为人知。

看到这里感觉自己原来的知识储备被刷新了，以前一直以为传统瀑布方法的提出是在传统软件开发环境下产生并作为一种最佳实践产生的，随着软件需求变更频率加快、规模复杂，瀑布模型不再适用于当前软件开发。然而，事实并非如此，传统瀑布方法在最开始提出就是作为一种反例，并且Winston W.Royce接着又描述了一种他宣称绝对更高级的迭代式流程，该流程和现在的敏捷方法很像。不过，大部分却去追捧瀑布方法，直到他们认识到错误！可以看出，在我们学习一个知识的时候不可以只是一知半解就浮躁的去应用，而是应该深入了解，再将其应用到实践当中，否则会对实践造成巨大影响。

6.2.敏捷方法实践和价值观

不管使用scrum、精益、极限编程，还是要柔和多种敏捷方法论创建混合方法，都要进行以下实践

1. 边做边测试：不要等到最后关头，现在就修复缺陷，等待它有机会在系统里繁衍存活好几个月后，修复成本会变得极高。
2. 及早频繁的交付产品：只有通过展示可工作的软件的方式，你才能发现客户真正想要的是什么。
3. 文档边做边写：只写必须的文档，只写有关的、有效的文档
4. 构建跨职能团队：不要让任何人或者部门成为流程或者信息的瓶颈

可以看出敏捷方法的核心思想在于定期增量地快速交付可工作的软件，我们从这些方法实践中也能感受到敏捷方法的价值观，不过这里要注意，我们在解读敏捷方法价值观时不仅要看前半句，也要结合价值观中的后半句。

1. 个体和互动高于流程和工具：工具和流程是很有用的东西，但是我们不能为了流程而流程、为了工具而工具，要正确的使用和掌控它们，让它们真正为人服务，而非人为它们服务。
2. 工作的软件高于详尽的文档：我们的关注点应该是产品本身而非流程文档。当拥有了可工作软件、自动化的测试、维持在最新状态的产品列表，其实你就有了一切。
3. 客户合作高于合同谈判：合同的缺陷所在在于包商的出价通常就是份赌注，赌他们能够在给定时间内完成工作，而利润则取决于他们能否多快好省地满足客户最低要求。然而，我们应该做的是齐心协力在规定时间内和预算范围内努力构建最有价值的系统。降低客户风险，不是靠前期担保方式转嫁风险给承包商，而是靠流程中客户的持续参与，以及敏捷团队定期交付可以工作软件增量的能力，双方应该是合作的战友，两者之间没必要尔虞我诈，进行一场没有硝烟的战争。
4. 响应变化高于遵循计划：软件开发中，不会变化的只有变化本身，因而认为变化时好事的流程就是最好的流程。敏捷方法通过增量的方式，持续地检验和适应来应对变化。

6.3.Scrum

6.3.1.Scrum角色

- 产品负责人
 - 持有产品愿景
 - 代表业务

- 代表客户
- 拥有产品列表
- 规划故事优先级
- 设立故事的接收标准
- 有空回答团队成员们的问题

产品负责人的责任在于，帮公司得到最高投资回报。即产品负责人控制着团队列表上那些条目的优先级顺序。必然也就意味着，产品负责人需要和干系人密切地合作，判别需要何时构建何物，从而交付出最多业务价值。产品负责人和团队其他人之间有一层天生的紧张关系，产品负责人总想要更多，而团队则必须维护可持续的速率。

- ScrumMaster
 - Scrum 专家
 - 谏言者
 - 教练
 - 障碍推土机
 - 引导者

ScrumMaster 担当教练角色，引领团队达到更高级的凝聚力、自组织和表现。团队的交付物是产品，而 ScrumMaster 的交付物就是自组织的团队。ScrumMaster 就是团队的引导者和 scrum 专家。ScrumMaster 不是团队的老板，此职位和团队是同等的，他们的区分只是职责不同，而不是等级不同 ScrumMaster 肯定会给团队带入一类领导力，但这仅限于通过影响力所获得的领导力，而非由职权所获。ScrumMaster 是团队的 scrum 专家，帮助团队从 scrum 上获取有可能得到的最大价值 ScrumMaster 能发挥用处的地方很多，包括引导 scrum 会议、帮助团队理解和使用 scrum 的工件、指引产品负责人和其他团队成员更好地理解自己在 scrum 团队中所担当的角色。ScrumMaster 还有另一个关键的作用，为团队移除障碍。任何拖累团队的都是障碍，它们形式多样、规模各异。

- 团队成员
 - 负责交付用户故事
 - 做所有的开发工作
 - 自组织地交付用户故事
 - 支配估算流程
 - 支配“如何干活”的决策
 - 避免“与我无关”

Scrum 团队是高度协作的，也是自组织的。团队成员可以全权决定如何完成工作。团队可以自行决定要使用的工具和技术，以及团队成员如何瓜分任务。做具体实现工作的团队成员们，同样也要负责估计实现特性需要的工作量。产品负责人可以决定故事的顺序，但完成特性或任务需要多少时间是开发人员说了算。

6.3.2.Scrum过程

- Sprint规划会议：标志着sprint的开始
 - 团队选举交付物作为当前sprint的承诺：产品负责人按照优先级顺序，逐个的介绍他希望团队在当前sprint完成的那些故事，团队成员们要和产品负责人探讨所有故事，审查其验收标准，确保大家对预期结果有一致的理解。团队成员协商解决依赖性问题，一般还会讨论实现故事要做哪些事情。接着团队成员就要决定他们是否承诺这个故事。对每一个故事重复上述流程，直到团队再也无法承诺更多的工作。
 - 交付用户故事所需：团队把选定的故事分解成任务，要记住故事才是交付物，它们是干系人、用户和客户想要的东西要完成这些任务才能交付故事。获取用户追加输入、新增数据库列式任务。

整个过程产品负责人要一直在产，以便回答问题。

让我想到了在公司实习时，需求移交时的场景，在第一部分基本类似，然而第一部分结束后整个会议就结束了，这会导开发人员对需求理解不够清楚，在设计实现时又会去找产品经理确认需求，有时还会导致返工，降低开发效率。

- Scrum日会：主要目的是同步进度
 - 每天都进行进行一场站立、直接了当的分享会议。每个人轮流快速分享自己昨天已经完成的内容，今天期望完成的内容，导致我慢下来的障碍。

实习时每天都会开站立会议，我们主要通过站立会议确定进度是否能正常进行，在kanban上（众目睽睽之下）更新进度。要注意的是不要在站会中讨论技术细节。

- Sprint评审：也被称作sprint演示，团队有机会在sprint评审会议上展示他们的工作成果。
 - 产品负责人随后可能会根据演示会议上收集到的干系人反馈，相应地调整产品列表和发布安排。收集反馈，但并不会在 Sprint 评审会议上给出承诺。这样更好，它能给产品负责人多一点时间，可以在确定计划或做出承诺之前，把最新收集的反馈也都考虑进去。

实习时我们并没有Sprint评审的环节，一般产品经理验收完后就会直接发布上线，相比于拥有Sprint评审，可能成就感会少一些。

- 回顾：在迭代最后矩形，是专门留给团队的时间，专注于讨论他们当前sprint的心得体会，并用于继续改进。回顾会议帮助我们把握时机，在消极模式尚未根深蒂固时，趁着记忆还很清晰，多捕获一些认识和见解。
 - 准备阶段：要在回顾会议刚开始的时候，先建立起大家对目标的共同理解，这样人们才能在第一时间知道为什么开会。
 - 收集数据：包括燃耗图和燃尽图、故事卡片、缺陷清单、构建受损相关数据、客户意见，以及团队使用的所有大型可视图表
 - 洞察问题：为什么会发生这些事情？记住那个最高指导原则，务必注意要避免指手划脚和抱怨的行为。
 - 确定方案：团队已经查看了过去发生的一切，对事件发生的方式和原因也都有了一定的理解。现在正是选择不同方式进行尝试的时候，也可以创建团队改进型任务并放入随后的 sprint 列表。
 - 结束：团队完成了一些不错的成果。他们投入时间和精力持续地改进。花点时间认可它们并予以称赞。给团队一个机会可以感谢彼此。你要让团队在回顾会议结束离场时充满干劲，比刚入场的时候更团结。

实习时每个月我们会开一次回顾会议，然而我们的会议是相当无效的。我认为主要原因是虽然我们能确定问题，也可以确定方案，但是没有对应的执行与监督，问题太多团队成员往往在下次迭代中忘记或失去改正的动力，所以书中有一个建议很好：“最好的颁发是只选定一个地方做改进，克制住心中想要一并解决所有问题的倾向”，如果每个sprint都能搞定一个新问题那就很好了！