

SCRUMMASTER的检查清单

一个全职的引导者？

一个普通的SCRUMMASTER能够同时应付2-3个团队。如果你把你的角色局限在组织会议、确保时间盒和响应组员提出来的障碍，并对此感到满意的话，你可以只用部分时间来曾担SCRUMMASTER的角色。在此情况下，团队的绩效可能还是会超过组织的预期，可能也不会有很糟糕的事情发生。但是如果你展望一个能在转型的组织里做出前人难以想象成就的团队，你需要考虑成为一个很棒的SCRUMMASTER。很棒的SCRUMMASTER一次只支持一个团队。我们推荐一个7人左右的团队能有一个专职的SCRUMMASTER，特别是在团队的初始阶段。

如果你还未发现有很多的工作要做，仔细看一下你的产品负责人、你的团队、你团队的工程实践和他们所在的组织。尽管没有对每人都适用的药方，我做了一个检查清单，包含了一些经常被SCRUMMASTER忽视的事情。

我的产品负责人做得怎样？

- 产品待办列表是根据产品负责人最新的想法优先级排序的吗？
- 来自所有利益相关人的对产品的需求和愿望都被放到产品待办列表了吗？记得待办列表是涌现的。
- 产品待办列表的规模可管理吗？为了维护可管理的条目数目，需要把高优先级的条目保持在细粒度，低优先级的条目粗粒度。过分分析优先级不高的条目会适得其反，因为你的需求会在产品开发和利益相关人/客户的不断交流中变化。
- 有任何需求（特别是处在高优先级的）能够更好地写成独立的、可磋商的、有价值的、可估计的、小的和可测试的用户故事吗？
- 你有让你的产品负责人了解到关于技术负债和如何避免它吗？一种方式是把自动化测试和重构加入到对每一个待办条目的“完成”的定义里。
- 待办列表是作为信息辐射体，对所有利益相关人立即可见吗？
- 如果你们使用自动化工具来管理待办列表，每个人都知道如何容易使用它吗？自动化的管理工具引入了使待办列表变成信息冷藏箱而没有被积极地传播的危险。

- 你能通过给大家打印件来帮助传播待办列表包含的信息吗？
- 你能通过创建可见的大图来帮助传播待办列表包含的信息吗？
- 你有没有帮助过产品负责人来把待办条目规划到合适的发布或优先级组？
- 所有的利益相关人（包括团队）都知道是否发布计划还和团队现有的速率符合吗？你可以尝试在每个SPRINT评审会议上在条目被确认“完成”后向所有人展示产品发布燃尽图。这会帮助我们更早地发现范围和时间的漂移。
- 你们的产品负责人会在SPRINT评审后调整发布计划吗？少数能够准时交付充分测试过的产品的产品负责人会在每个SPRINT调整发布计划。因为会发现有更重要的工作，所以一些原有的工作通常会被推迟到后续发布。

我的团队做得怎样？

- 团队成员大多数时间处在流畅状态吗？这种状态的一些特征：
 - 明确的目标（对期望和规则是了解的，目标是可达到的、和团队的能力相匹配）
 - 全神贯注并全力以赴
 - 处在无意识状态，行动和认知融为一体
 - 直接及时的反馈（活动中的成功和失败是显露的，因而行为可以随需要调整）
 - 能力和挑战的平衡（不要太容易也不要太难）
 - 对局面和任务的个人掌控感
 - 所做的事情是有内在奖励的，因此感到工作不费力
- 团队成员看起来相互喜欢，经常在一起，并且庆祝各自的成功吗？
- 团队成员相互以高标准监督，相互挑战以促进成长吗？
- 有没有一些话题因为大家感觉难受，所以在团队里没有进行讨论的？
- 你尝试过用不同的形式和地点做SPRINT回顾吗？
- 团队在SPRINT中一直关注验收标准吗？也许你应该举行一个SPRINT中的检查来重新看一下这个SPRINT承诺条目的验收标准。现在列出的SPRINT任务够吗？
- SPRINT任务反映团队实际在做的事情吗？
- 团队是由5-9个具备混合技能的人员组成吗？

- 团队的任务估计和你们的任务板包含最新信息吗？
- 团队用于自管理的工件（任务板、SPRINT燃尽图，等等）对团队可见吗？方便使用吗？
- 这些工件有没有被保护以防止用于微观管理？团队外部的人对每日活动多余的审查会阻碍团队内部的透明性和自我管理。
- 团队成员自己选任务吗？
- 偿还技术负债的条目有没有包括在待办列表里？如果没有的话，团队有没有和产品负责人沟通？
- 团队成员能否抛开各自的岗位定义，对约定工作的所有方面（测试、用户文档等）集体负责？
- 管理层是否以集体的成功来衡量团队？

我们的工程实践做得怎样？

- 你们在开发中的系统有没有一个“按下测试”的按钮，从而每个人（同一团队或不同团队的）都能方便地检测到系统被破坏了？通常可以使用XUNIT框架来实现。
- 你们在自动化的端到端系统测试（或功能测试）与自动化的单元测试之间有适当的平衡吗？
- 团队用开发系统的同种语言来写系统功能测试和单元测试（而不是用只有部分团队知道如何维护的专用脚本语言或捕捉回放工具）吗？
- 你们团队已经发现在系统测试和单元测试之间存在有用的灰色区域吗？
- 当有人引起回归失败时，是否有个持续集成服务器会在一小时甚至几分钟内自动发出警报？（KENT BECK说过，“每日构建只是弱者的水平”）
- 所有的测试都会总集成进持续集成服务器的结果吗？
- 团队成员已经发现了，作为一种和预先做大量设计不同的实践，做持续设计和不断重构的乐趣和成功了吗？重构有严格的定义：改变系统的内部结构而不改变其外部行为。重构应该发生在有重复代码、复杂的条件逻辑、不好的命名标识符、对象之间的过度耦合等情况下，每个小时就有几次。有自动化测试覆盖才能有信心地去重构。测试覆盖中的漏洞和推迟的重构就是糟糕的技术负债。
- 你们针对每个功能性产品待办条目的“完成”的定义（验收标准）包含了完整的自动化测试覆盖和重构吗？我从未看到过或听到过一个没有采用极限编程实践的超高效团队。

- 团队成员大多数时间都结对编程吗？结对编程显著地增加了代码的可维护性和减少了缺陷率。然而它挑战了开发人员之间的界限，有时候看起来花了更长时间（如果我们把数量放在质量之上的话）。以身作则与团队成员发起结对工作日，而不是迫使大家这么做。一些成员会开始喜欢这样工作。

我们的组织做得怎样？

- 有适量的跨团队沟通吗？SCRUM OF SCRUMS只是其中的一种方式。
- 团队在需要跨越架构边界的情况下还能独立交付工作的特性吗？
- 你们的SCRUMMASTERS聚在一起工作在组织的障碍列表上吗？
- 在适当的时候，组织的障碍有被贴在开发总监办公室的墙上吗？这些障碍带来的浪费能被量化为钱、失去的进入市场的时间、失去的质量或者失去的客户机会吗？（但是记得KEN SCHWABER的发现：“一个不存在的SCRUMMASTER是个没用的SCRUMMASTER。”）
- 你们组织是为数不多的能提供和团队集体目标一致的职业道路的组织吗？如果存在职业激励去做编码或架构工作，而牺牲测试、测试自动化或用户文档，回答“不是”！
- 你们组织已经被专业刊物或其它独立来源公认为最佳公司或行业领导了吗？
- 你在帮助创建学习型组织吗？

结论

如果你能勾掉多数条目而且还有时间多，我想收到你的来信。创造性的工作没有一个标准方程，这里列出的点可能帮到你，也可能帮不到。当你意识到你做这些事情能有所作为，你也可能发现自己不敢做。这样的话你应该是走上了成为很棒SCRUMMASTER的正道。