# **[Spotify敏捷模式详解三部曲第三篇：工程文化](http://www.scrumcn.com/agile/scrum/21933.html" \o "Permanent Link: Spotify敏捷模式详解三部曲第三篇：工程文化)**

2019年2月22日/分类： [敏捷文章](http://www.scrumcn.com/agile/scrum/category/scrum-agile-practices) /编辑： [Amy Guo](http://www.scrumcn.com/agile/scrum/author/agileeric" \o "由Amy Guo发布)

### ****摘要****

在本系列文章的第一篇和第二篇，我们分别介绍了Spotify的敏捷研发团队和研发过程。

在本篇，我们将介绍Spotify的敏捷工程文化。

### ****引言****

Spotify通过文化和价值观来进行管理，在本篇中，我们从如下八个方面来介绍Spotify的敏捷工程文化：

一、 如何管理小队的自主性

二、 如何管理标准化

三、 如果做到以人为本

四、 如何管理部署上线

五、 如何管理创新

六、 如何管理失败

七、 如何处理浪费

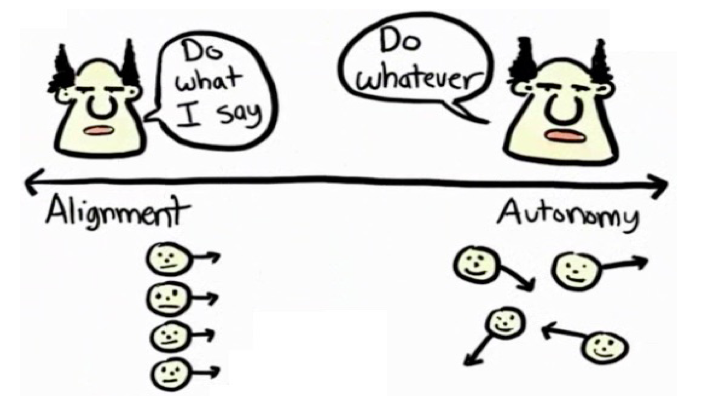
八、 如何管理文化

### ****一、 如何管理小队的自主性****

Spotify的小队，类似于一个高度自治的、迷你的“创业公司”。一方面，小队要保持自主性，同时也要兼顾公司在产品上的整体一致性。

#### ****如何看待自主性和一致性****

通常认为：一致性和自主性就像是天平的两端，自主性高则一致性少。



Spotify认为：这是两个不同的维度：

1. 低一致性、低自主性：

管理者下指令，团队执行。

2. 高一致性、低自主性：

管理者告诉团队要做什么，以及怎么做。

3. 高一致性、高自主性：

管理者聚焦要解决什么问题，由团队自己去找出解决问题的方法。

4. 低一致性、高自主性：

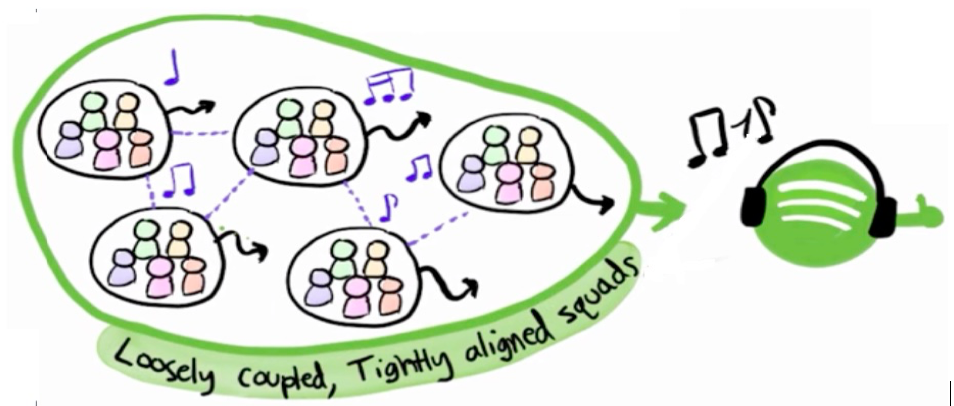
团队各行其事，管理者很无助，产品是个怪胎。

理想情况：具备一致性的自主，只有具备一致性才令团队自主具备可能性，而一致性越高，管理层越能下放自主权。



#### ****Spotify的管理原则：****

* 独立自主，但不能局部优化
* 小队可以自主，但不能追求局部优化。就像是一个大型乐队，每个乐队小队都在独立自主的演奏，但又必须彼此倾听其他小队的演奏，共同聚焦整首曲子的演出，这样才能演奏出好的音乐。
* 目标——建立相互松耦合，但整体高度一致的小队。



#### ****打造独立自主、松耦合、整体高度一致的小队****

****1. 为什么小队的独立自主如此重要？****

* 这是一种激励方式，受激励的人们能够开发出更好的产品。
* 独立自主能够让小队更快地做决策和行动，而不需要经过层层审批。
* 独立自主，能够尽量避免交接和等待。

****2. 独立自主：****

* 小型（人数少于8）、跨职能、自组织的团队。
* 小队自行决定开发什么产品，如何开发，以及如何协作。
* 坐在一起，共同对研发的端到端事物负责，例如：设计、交付、部署、维护、运营等。

****3. 整体高度一致****：

* 小队有长期的使命，例如：使Spotify成为探索音乐的最佳应用程序，或内部事物，例如执行A/B测试的基础架构。
* 小队要与产品的整体策略保持一致，与公司的整体优先级和其他小队保持一致。Spotify的整体使命的重要性和优先级，高于小队的任务。
* 小队们合作开发产品的整体策略，以及季度重新探索短期目标。

### ****二、 如何管理标准化****

一方面，小队的要保持技术灵活性，另一方面要兼顾公司的整体规范性。

#### ****自由胜过标准化****

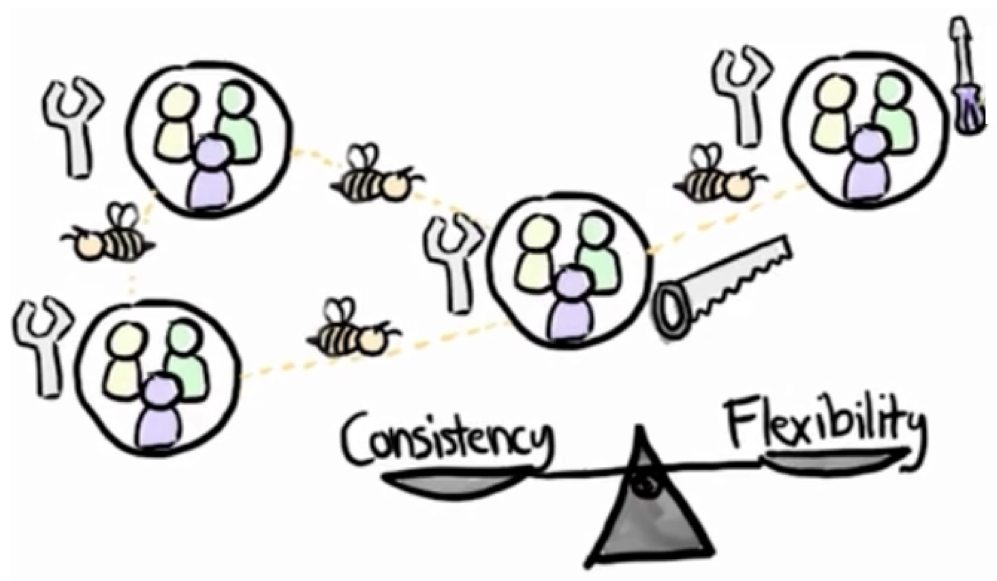
如果有人问：应该使用哪种编程工具，或是如何做计划？答案是：这要看是哪个小队。  
有的会用Scrum，有的会用Kanban，有的会估算故事点并统计速率，有的则不会，实际情况因小队情况而已。

随着时间的积累，发展出了一些设计指引、代码规范等来减少工程上的摩擦，但唯有非常时期才会使用。在权威与自由的天平上，倾向于自由而非权威。



#### ****通过异花授粉而非标准化，来平衡灵活性和一致性****

* 异花授粉（Cross-pollination），而非标准化（Standardization）。
* 当越来越多的小队都使用某种实践方法时，例如使用Git进行版本控制，其他小队也会跟进和开始使用，当小队间都使用这种工具协作时，就成为事实上的标准。
* 通过采用这种非正式的方式，得以在一致性和灵活性间保持平衡。



#### ****解耦系统****

****1. 现状&问题：****

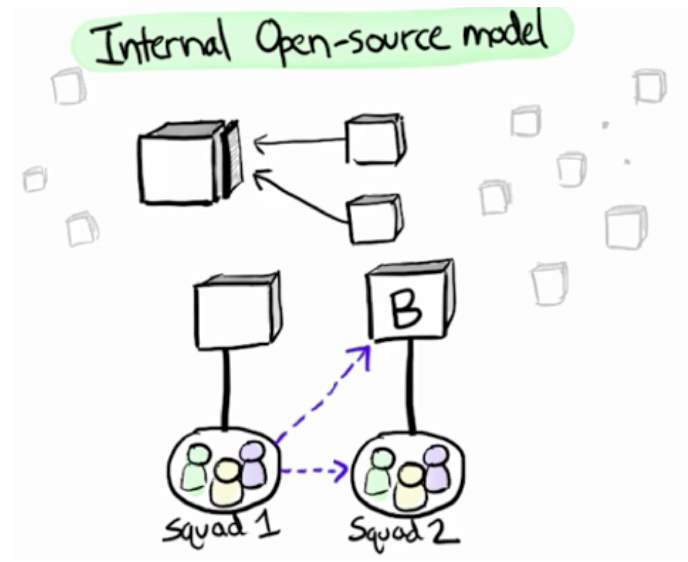
* 产品中包含100多个独立开发和部署的系统。
* 在技术上，每个系统由某一个小队负责；大部分小队会同时负责好几个系统。
* 系统间存在交互，而各小队又分别专注于自己的特性。例如播放清单的管理、搜索或监控。
* 如何让小队在专注自己的特性和系统的同时，还能与公司整体以及其他小队保持一致？

****2. 实践方法：****

以清晰的接口和规范，力求实现需求之间以及系统之间的解耦。

#### **内部开源机制**

内部开源模式：谁都可以改代码+同舟共济的代码评审文化。



例如，上图中有两个小队：Squad1和Squad2

* 小队1需要在B系统完成某项事情时，而小队2对B系统的程序编写比较在行。
* 通常小队1会找小队2协助处理。
* 如果小队2没有时间或者有更重要的任务，小队1也不会等待。组织鼓励小队1自行去编写系统B的代码，然后再请小队2评审修改。消除等待的同时，这样有助于提升质量和传播知识。

### ****三、 如何做到以人为本****

唯有以人为本，工作才能得以顺利进行。

#### **以人为本的文化**

1. 坚实的“互相尊重”文化：经常听到像是“我的队友真棒”的评论。

2. 人们常常把事情归功于其他人的杰出表现，而非自私的为自己争功。

3. Spotify确实有很多天才，但却很少有人狂妄自大。

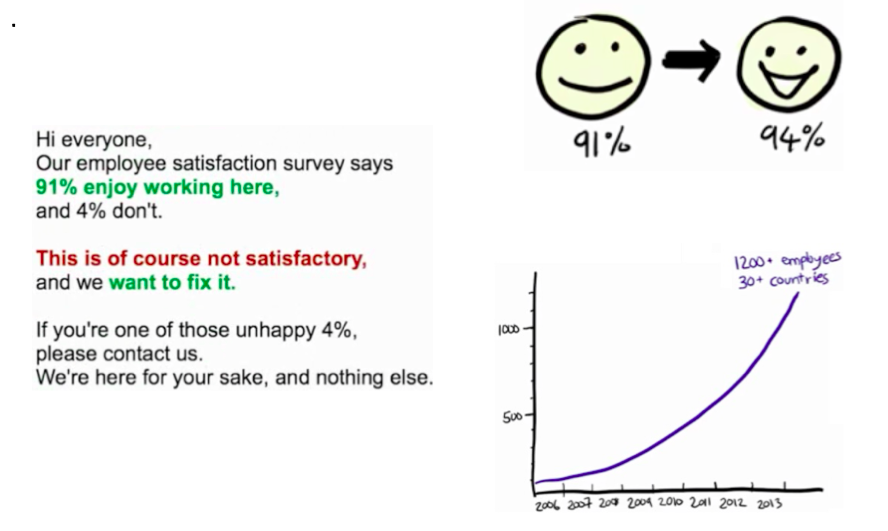
4. “不干涉+同舟共济”的文化：

* 你和你的队友会被期望自己去找答案，没人会告诉你该做什么。
* 但当你需要协助的时候，你会很快获得许多援助。
* 大家一致认同的真理是：我们在同一条船上，而且必须帮助其他人成功。



#### ****以激励为公司文化的重点****

Spotify每年都会进行员工满意度调查。



### ****四、 如何管理部署上线****

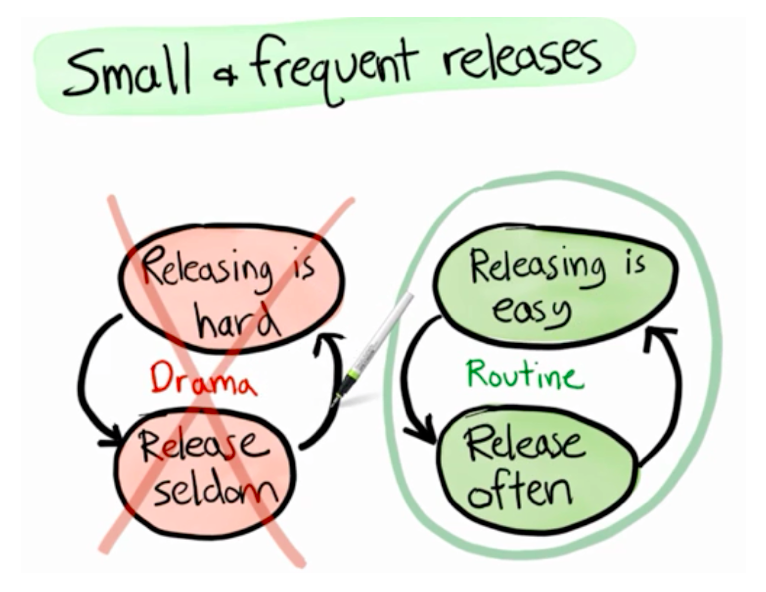
对自主性最重要的一点是部署上线的难度。

#### **追求小而频繁的发布：**

1. 程序发布应该是例行的（Routine），而非戏剧性的（Drama）。

2. 拒绝恶性循环：发布困难-》发布得少-》发布规模大-》发布困难。

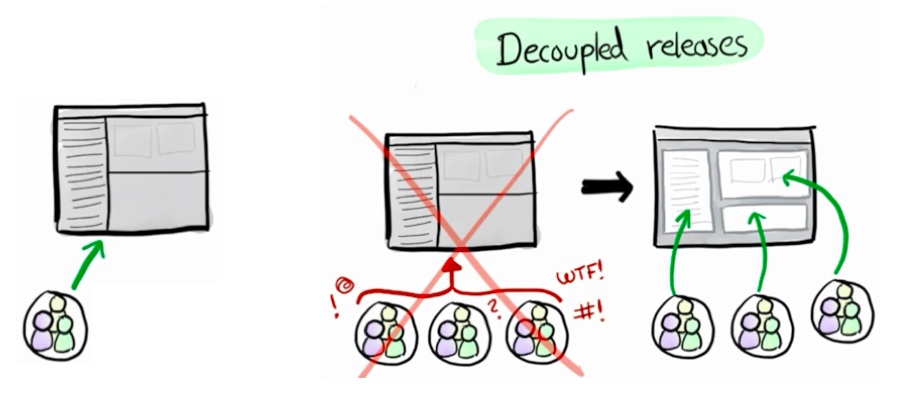
3. 追求良性循环：发布容易-》频繁发布-》发布规模小-》发布容易。



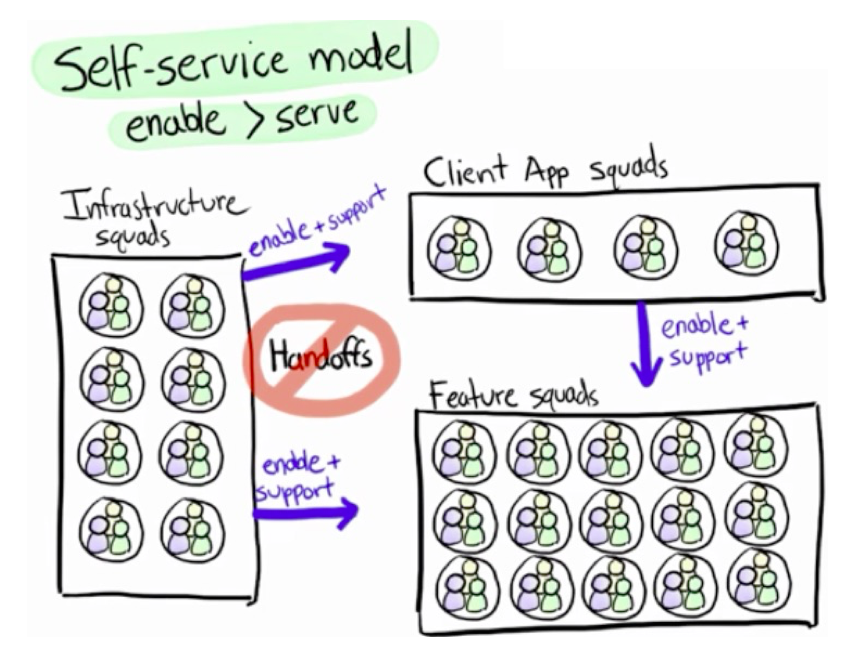
#### ****投资令发布更加容易****

不是建立流程、规范来管理发布，而是通过投资测试自动化和持续交付的基础设施，例如：改变架构，让发布解耦。

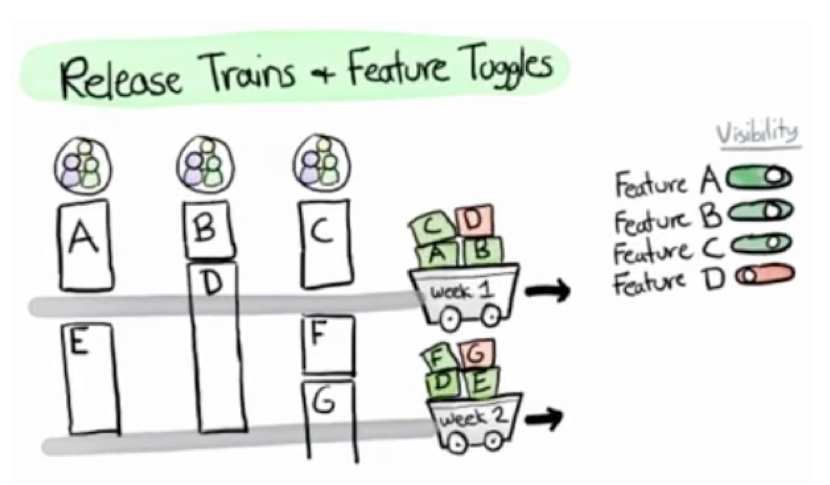
——使用Chromium嵌入式架构，每个客户端就是一个伪装的网页浏览器。每个区块就像是网站的一个框架，小队可以直接发布他们的产品。



#### ****客户APP小队、基础设施小队和“自助服务”模式****

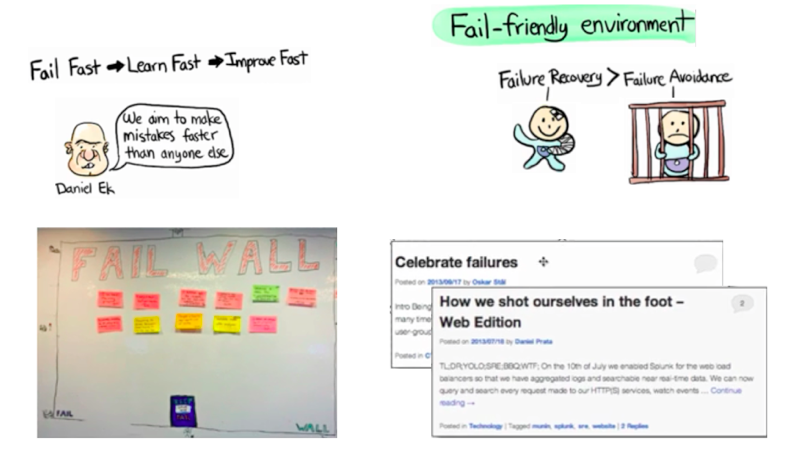


#### ****利用发布火车和特性开关解决小队间的同步问题****



### ****五、 如何管理失败****

#### ****包容失败的文化****

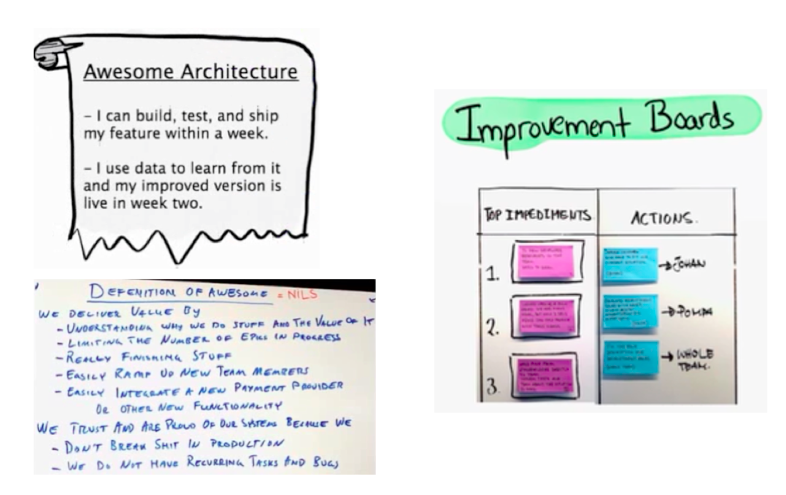


#### ****自下而上驱动，自上而下支持的持续改进文化。****

* 没有任何学习成果的失败就真的只是失败。
* 当出问题的时候，通常要进行事后检视。不关注“这是谁的错”，只关注“怎么发生的？学到了什么”，以及“我们如何改变”。
* 事后检视是事故管理流程的一部分，任何事故单，问题解决了还不算完，仅当有所学习和收获时才能关闭。不仅要改进产品，还要改进过程。
* 所有小队，每隔几周就要召开一次回顾会议，讨论什么地方做得好，什么地方该改进



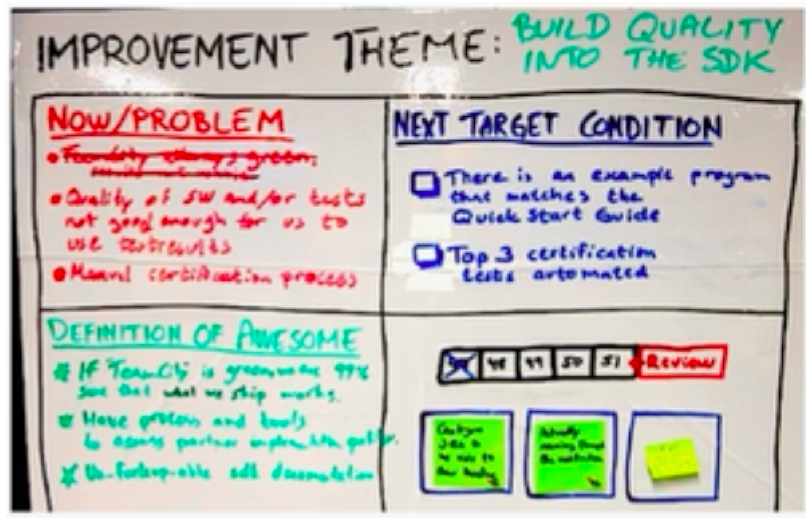
#### ****“改进清单”和“牛逼的定义”****



#### ****实例：某小队的改进看板****

****1. 改进看板的内容：****

* 左上：现状，小队面临质量问题。
* 左下：牛逼的定义，理想中不会有质量问题。
* 右上：真正的目标。如果我们靠近牛逼一步是什么样子？
* 右下：三个让我们达到目标的行动项，完成后会增加新的行动项。



#### **控制失败造成的损失**

****1. 失败必须是非致命的，否则我们无法存活到下次失败。****

* 通过架构解耦（Decoupled Architecture），控制爆炸范围(Limited Blast Radius)。
* 当一个小队犯错时，只会影响到系统的少部分，而非搞垮整个系统。
* 由于无需交接切换，小队通常能够快速修复问题。

****2. 采用A/B测试****

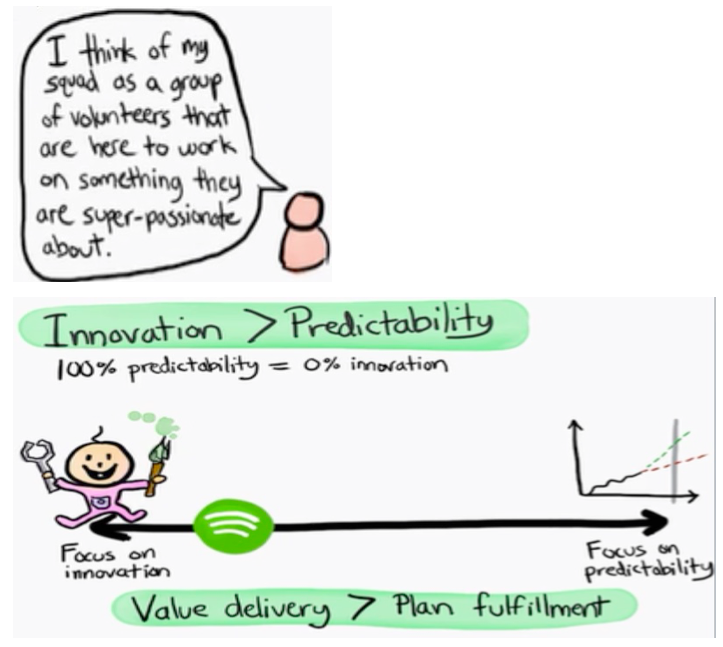
新功能发布时，先发布给少量用户体验并进行密切观察。仅当确定功能已经稳定后，才会逐步发布给全世界的用户。

因此，失败只会影响到系统的少量用户，并且影响时间很短暂。这种有限的损害范围，让小队勇于做更多的尝试，并加速学习的步伐，而不是把时间浪费在风险的预测和控制上。

### ****六、 如何管理创新****

通过降低可预测性，来促进创新

* 100%可预测性=0%创新。
* 如果必须做出交付时间承诺，通常会推迟承诺——直到产品特性已经被证实或者接近完成时，才给出承诺。
* 通过降低预测性要求，使小队能聚焦于价值交付，而不是成为某人的武断计划的奴隶。
* 有位PO说：我把我们的小队看做一群聚集起来从事自己狂热事业的志愿者。



#### **鼓励人们玩耍和尝试，以促进创新**

* 惊艳的新产品，始于人的灵感，唯有允许人们玩耍与尝试，才能产生灵感。
* 鼓励人们花10%的时间，执行“黑客日”或者“黑客周”。在这个时间里，人们可以根据自己的想法，尽情的去试验或者开发任何自己想要的产品。
* 在Spotify，每年会举行两次黑客周。

1. 头号标语：Make cool things real！Build everything you want, with whoever you want, in whatever you want!
2. 好几百人整周都在闭门当黑客，并在周五时大型会展中展示成果。

* 开发出来的产品真的有用吗？

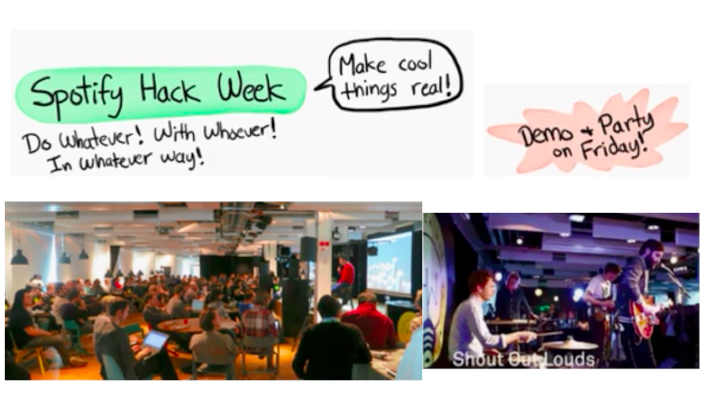
1. 这不重要。
2. 关键是只要你尝试的点子够多，我们就能不断接近目标。
3. 而且往往，做黑客时学到的知识，比黑客行为本身更有意义。
4. 而且，也挺好玩的。

#### ****黑客创意产品——“拨打一首歌”****

只需要拿起电话，拨打一首歌的号码，就能听歌。



#### ****“黑客周”活动****



#### ****鼓励试验的文化（Experiment-friendly Culture）****

1. 例如：

* 该用工具A还是工具B？不知道，让我们两个都试用一下然后做个比较。
* 或我们是否真的需要Sprint计划会议？不知道，我们跳过一次会议，看看后面是否会怀念它。
* 或我们该在歌手页放5首还是10首排行榜歌曲？不知道，让我们两种情况都尝试一下，然后评估效果。

2. 对于决策问题，以其争论半天，不如试着讨论下：

* 有什么假设？
* 我们学到了什么？
* 接下来我们尝试什么？

3. 这让我们的决定多基于客观数据，而非来自某个人的想当然或者屈从于权威。

### ****七、 如何处理浪费****

#### ****排斥浪费的文化****

* 虽然我们乐于尝试，并且试着用不同的方式来做事情，但我们的文化非常排斥浪费。
* 人们会快速停止做任何不能带来增值的事情。如果发现有效，就继续；反之，则抛弃。

a) 通过试验发现有用，需要继续的一些实例：回顾会议、每日站会、google文档、GIT、协会的研讨活动。

b) 通过试验发现没用，需要抛弃的一些实例：例行报告、交接、独立的测试团队或测试阶段，任务估算，无用的会议，企业官话。

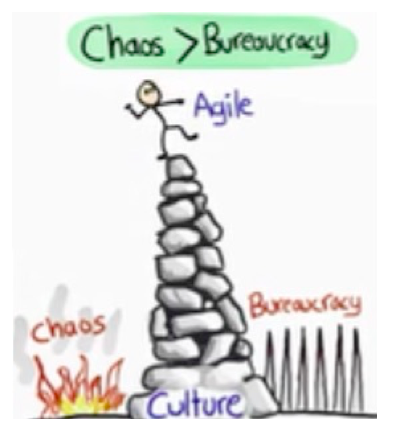
#### ****大型项目****

* 大型项目是一种最常见的浪费。（大型项目，通常指需要多个小队一起合作好几个月来完成的项目。）
* 大型项目，意味着巨大的风险。（Big Project = Big Risk）
* 要尽量减少大项目，尽可能把大项目拆成一系列的小工作。
* 有时候，大项目有其合理性，并且潜在收益远大于风险，这种情况下，需要采取一些措施：

1. 用物理看板或者电子看板，通过各种组合方式，可视化项目进度。
2. 进行每日同步会议：所有小队共同参与讨论，解决相互依赖。
3. 每一到两周，进行一次演示：将产品各部分进行集成，召集干系人进行整体评估。
4. 一个小而紧密的领导团队，来持续地掌控整个大局。通常有：一位技术主管，一位产品主管，有时候还有一位设计主管。三者通力合作。

#### ****在混乱与官僚主义之间取得平衡****

* 当我们成长时，面临者陷入混乱的风险。当如果增加太多的结构和流程，又会陷入官僚化的风险。
* 关键问题：什么是最小可行官僚系统？可以让我们以最少的结构和流程来避免陷入完全混乱。
* 排斥浪费的文化和敏捷思维能帮助我们在混乱和官僚主义之间保持平衡。



### ****八、 如何管理文化****

#### ****为何Spotify如此重视文化？****

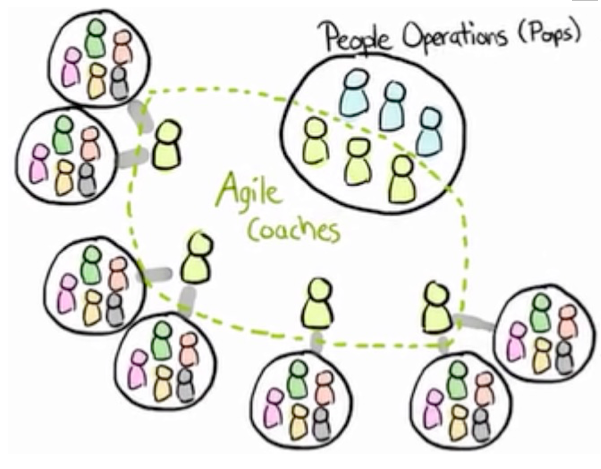
在企业成长的过程中，会面临各种问题，以及采取各种解决方案，包括改造产品架构、流程和组织等。

但是企业面临的情况也是一直在变化的，今天看似聪明的解决方案，可能会在明天造成一个难缠的新问题。而健康的文化能够修复有问题的流程。



#### ****关注文化的角色（Culture-focused Role）****

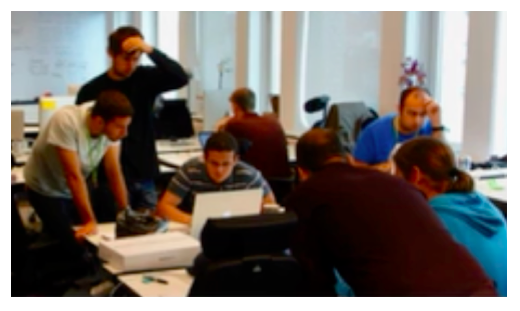
* 人力资源运作团队
* 30+位敏捷教练



#### ****新员工训练营（Boot Camp）****

* 时间：为期一周
* 人员：由新招募的人员组成临时的小队
* 内容：

1. 解决实际问题。
2. 同时学习技术栈和工作过程。
3. 以及学习如何作为一个团队协同工作。
4. 这种紧张而有趣的训练，能让你真正融入到文化中。



#### ****通过讲故事来传播文化****

在博客、午餐以及各种场合分享自己的成功以及失败学到的经验教训。



#### ****每个人都是组织文化的一份子。****

* 组织文化，是组织中每个人的态度和行为的总和。
* 每个人都是组织文化的一份子。
* 每个人都在塑造组织的文化。



### ****相关阅读：****

[Spotify敏捷模式详解三部曲第一篇：研发团队](http://www.scrumcn.com/agile/scrum/21929.html" \t "http://www.scrumcn.com/agile/scrum/_blank)

[Spotify敏捷模式详解三部曲第二篇：研发过程](http://www.scrumcn.com/agile/scrum/21931.html" \t "http://www.scrumcn.com/agile/scrum/_blank)

### ****参考资料：****

本文部分内容及引用的图片主要来自于Henrik Kniberg和Anders Ivarsson的文章《Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds》。

### ****本文作者：****

Jerry Li(李洁)， Eric Liao（廖靖斌）