# **[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" \o "Permanent Link: Spotify敏捷模式详解三部曲第一篇：研发团队)**

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)

### ****引言****

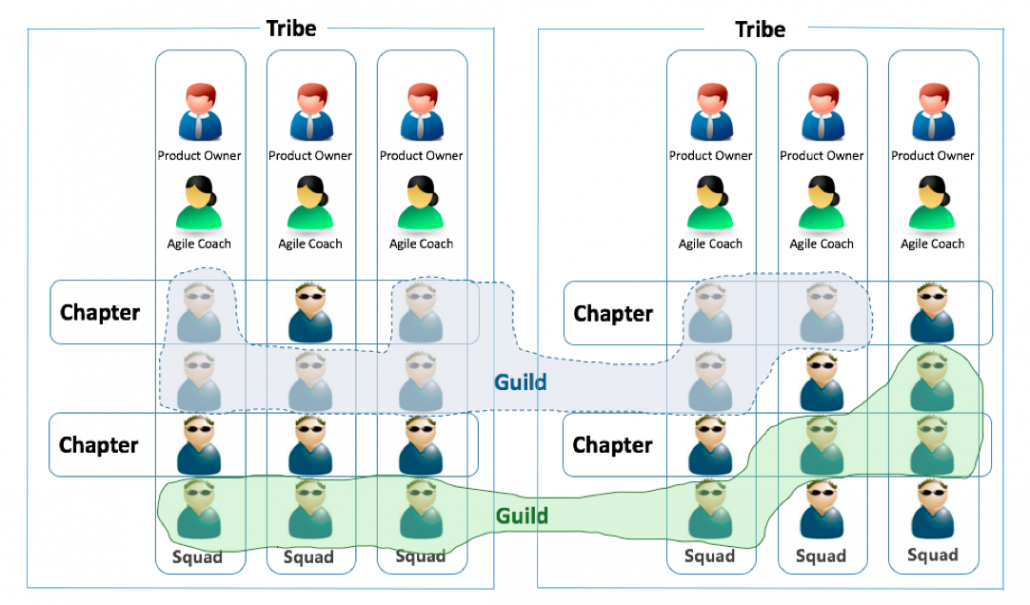
2018年4月，来自北欧瑞典的音乐流媒体公司、百亿美元独角兽Spotify创造了历史，它成为了当代上市公司当中，第一家通过“直接上市”的方式在美国纽交所成功挂牌的公司。这家公司改变的可能不仅是人们听音乐的习惯，而且还有企业进入资本市场的方式。截至2018年初，Spotify拥有1.59亿的全球活跃用户数量、7000万的付费用户数和46%的付费用户增长率。Spotify是当之无愧的音乐流媒体领域的霸主，排名第二的苹果音乐，无论是付费用户还是活跃用户，都只有Spotify的一半。

是什么成就了Spotify？他们是如何打造出这么一款深受用户喜爱的产品的？幸运的是有Henrik Kniberg的存在。Henrik Kniberg是全球知名的敏捷和精益教练，也是一位多产的作家，他参与了Spotify公司的这一过程，并写下了多篇文章，向外界揭开了Spotify公司的神秘面纱。

笔者通过自己的收集和整理，梳理出了Spotify敏捷模式的整个过程，并通过一系列的文章呈现给大家。在本期的文章中，笔者将首先介绍Spotify的研发团队。

### ****组织架构****

Spotify采用的是一种非常独特的组织架构，如下图所示：



整个研发组织有多个称为“Tribe部落”的单元组成，每个部落中包括多个“Squad小队”，从横向的维度把拥有类似技能的人放在一起形成“Chapter分会”和“Guild协会”。

### ****Squad（小队）****

****组织定位：****

• 类似于一个高度自治的、迷你的“创业公司”。

• 肩负一个长期的使命，长期从事某一类任务或者开发产品的某一个部分。例如：

1）功能特性小队：专注于某块功能特性，例如搜索功能。

2）客户端APP小队：专注于使特定的客户端平台更容易发布，例如：iOS、Android.（注意：他们不进行业务特性开发）。

3）基础设施小队，专注于让其他小队更有效率，提供工具和例行事物，例如：持续交付、A/B测试，监控和运维。

****团队成员：****

• 不存在官方任命的团队领导。

• 有一位产品负责人（Product Owner）。

1） 各小队的产品负责人紧密合作，共同维护一个宏观的产品路线图（Roadmap），指引整个 Spotify 公司的产品发展方向。

2） 每个产品负责人也分别维护一个自己所在小队的产品待办列表（Product Backlog）。

• 可能会有一位敏捷教练（Agile Coach），帮助团队改进工作方式。

• 具备开发产品需要的所有知识和技能，例如设计、开发、测试、发布等。

• 通常少于8个人。

****工作方式：****

• 坐在一起工作。

• 自组织管理自己的工作。

• 定义和改进自己的工作流程。

****办公环境****



### ****Tribe（部落）****

****组织定位：****

• 在相关领域工作的多个小队的集合，例如音乐播放器、后台基础架构等等。

• 可以看成是迷你创业小队的“孵化器”。

****团队成员：****

• 每个部落有一名酋长，他负责为部落内的各小队提供最好的栖息地（Habitat）。

• 部落规模大小基于“邓巴数（Dunbar Number）理论”而定（小于100人）。

****工作方式：****

• 一个部落中的所有小队在同一个办公地点工作，通常各小队的办公区是彼此相邻的，办公区附近的休息区能促进了小队间的交流与合作。

• 会定期在部落内开展非正式的聚会，在聚会上相互展示自己正在做什么、已交付了什么、其他人能从自己正在进行的事项中得到什么经验教训。展示内容包括能工作的软件、新工具与新技术、非常酷的黑客日/黑客周项目……

“邓巴数理论”：即150定律，由英国牛津大学的人类学家Robin Dunbar在1990s年代提出。

该定律认为：人类智力允许人类拥有稳定社交网络的人数是148人（四舍五入为150）。Spotify认为在超过 100 人的组织中，大部分人很难维持稳固的社会关系。当一个组织变得过大后，我们就会开始看到限制性的规定、官僚主义、政治斗争、冗余的管理层级，以及其他各种浪费。

****组织对小队的支持****

每个季度，组织会对每个小队做一次调查，以确定小队需要在哪些方面改善以及需要在组织层面提供哪些支持。



各个调查项的评判参考标准：

1. 产品负责人（Product Owner）——小队内是否有专职的产品负责人对任务的优先级进行排序？排序时，产品负责人是否能够综合考虑商业价值和技术因素？

2. 敏捷教练（Agile Coach）——小队是否有敏捷教练帮助团队识别障碍、指导团队进行持续地过程改进。

3. 支配自己的工作（Influencing Work）——小队内的每个成员都可以支配自己的工作？可以积极参与制定工作计划？可以选择自己做什么任务？每个成员是否都可以把自己 10%的工作时间投入到黑客时间？

4. 容易发布（Easy to Release）——小队是否可以（并且确实做到）轻松发布产品，而不需要很多的争论和同步？

5. 合适的流程（Process that fits the team）——小队拥有自己的工作流程并且持续对其进行改进？

6. 使命（Mission）——小队是否有一个小队所有人都知晓并关注的使命？产品待办项中的故事是否都与此使命息息相关？

7. 组织级支持（Organizational Support）——小队是否知晓该从何处寻求解决问题所需的支持，无论是技术问题，还是“软”问题（“soft” issues）？

### ****管理小队间依赖****

****核心思想：****

• 依赖就意味着堵塞和等待。

• 要尽量减少小队间的依赖项。

****实践方法：****

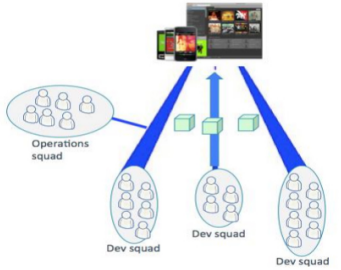
1. 经常对小队进行依赖调查：你们的工作依赖于哪些小队？这些依赖是否阻塞或拖慢了你们的工作？严重到了什么程度？

2. 基于调查结果，讨论如何消除有问题的依赖，特别是引起了工作阻塞的以及跨部落的依赖关系。为了消除这些有问题的依赖，经常会调整任务优先级、团队组成、架构或技术方案。

3. 必要时，召开SoS（Scrum of Scrums）协调会议。

4. 开发小队自行发布成果，运营小队只负责“铺路”。





### ****Chapter（分会）****

****组织定位：****

• 负责传播知识和开发工具。

• 类似于传统的职能部门。

****团队成员：****

• 同一个部落、相同技能领域、拥有相似技能的一些人。（技能领域，例如：例如敏捷教练，或Web开发。）

• 分会有一个领导，就是分会成员的直线经理，是一个服务式的领导，负责教导和指导分会成员的工作，执行员工发展、定薪等。

• 分会的领导，同时也是某个小队的成员，参与小队日常工作。

****工作方式：****

• 每个分会定期聚会，讨论专业知识及遇到的挑战。

### ****Guild（协会）****

****组织定位：****

• 分享知识、工具、代码和实践。

• 轻量级的“兴趣爱好社区”。

****团队成员：****

• 囊括整个公司的成员，聚集和分享特定领域的知识，例如领导力，Web开发或持续交付。

• 每个协会都有一个“协会协调人”，负责组织和协调协会的活动。

• 任何人都可以随时加入或离开协会。

****工作方式：****

• 每个协会定期组织一些研讨会。

### ****如何解决系统的整体一致性？****

****1. 现状&问题：****

1) Spotify 的技术是高度面向服务的。

2) 一共有 100 多个独立的系统，每个系统都可以单独地维护和部署。

3) 通常小队需要更新多个系统来完成新特性的开发。

4) 如果没人关注系统整体一致性的话，那么系统架构就会变得一团糟。

****2. 解决方案：****

• 系统负责人（System Owner）：每个系统都有一个或一对系统负责人（鼓励结对）。对某些关键的运营系统，系统负责人由开发和运营结对组成（Dev-Ops Pair）—— 一个人拥有开发视角，另一个人拥有运营视角。

• 系统负责人负责协调和指导开发人员，以避免开发人员之间的冲突。

• 系统负责人关注于：质量、文档、技术债、稳定性、可扩展性和发布流程。

• 系统负责人并非专职，通常也是小队成员或分会领导，还有其他的日常工作。

• 系统负责人会不时地执行一次“系统负责人日”，以整理一下系统。（系统负责人要在“系统负责人日”上投入的时间，不同的系统差异较大，但是通常不少于10%。）

• 一个首席架构师，负责协调跨越多个系统的、较高层面的架构问题。

• 评审新系统的开发工作，以避免一些常见错误，并确保新系统和现有的架构设计是一致的。架构师的反馈只是建议和输入——系统设计的最终决策取决于小队。

### ****与传统矩阵式组织的区别****

传统矩阵式组织形式

1) 项目立项时，“职员”由“职能经理”从职能部门“分配”到“项目”。

2) 在项目中，“职员”由项目的“项目经理”分配任务。

3) “职员”要例行向“职能经理”进行工作的“汇报”。

4) 项目结项后，“职员”从“项目”释放，回到“职能部门”，等待分配到下一个“项目”。

****• Spotify的组织形式：****

1) “职员”在“小队”中“持续”工作，与小队中其他人员共同“打造”一款优秀的“产品”。

2) “分会”和“协会”为“职员”提供“服务”，分享知识、工作、代码和实践，解决如何小队成员“如何做得更好”的问题。

****• Spotify的组织设计思想：****

1) 采用社区（Community）来替代传统的层级式组织（Hierarchical Structure）。

2) “教授和企业家”模型

3) 产品负责人（PO）就是“企业家”，关注于交付优秀的产品，

4) 分会领导是“教授”，关注于技术卓越。

本文是Spotify敏捷模式详解三部曲第一篇：研发团队。

### ****相关阅读：****

[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)

[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" \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）