Spotify Guilds – Cultivating Knowledge Sharing in Large-scale Agile Organizations

Darja Šmite, Blekinge Institute of Technology

Nils Brede Moe, SINTEF

Georgiana Levinta, Spotify

Marcin Floryan, Spotify

The new generation of software companies such as the American Google and Thoughtworks, the Australian Atlassian or the Swedish Spotify, have revolutionized the way new companies are designed today. These companies are driven by autonomous agile teams and self-motivated engineers, who perform way above average, innovate and create value, further increasing company competitiveness. One of the key enablers of high performance is alternative organizational models with bottom-up governance in contrast to top-down management with hierarchies. However, when non-hierarchical agile environments scale, many challenges emerge, including coordination of multi-team work, agreement on development strategies, and shared code ownership. Which organizational structures take care of such decisions in the absence of hierarchies? In this article, we describe how Spotify cultivates guilds that help the company cultivate knowledge sharing, make collective decisions whenever alignment is required and manage agile development at scale. Further, we provide research-based advice on success criteria for guilds.

New generation organizational structures: the Spotify case

Spotify is a software company providing music streaming service, launched in 2008. Within ten years, Spotify managed to continue growing and become one of the most innovative companies and an icon for the new generation agile organizations. Spotify ways of working and organizational structures are designed to promote innovation, collaboration and teamwork and enable bottom-up governance and autonomy[1] (see Figure 1):

- Teams at Spotify are called ***squads***, which should "feel like mini-startups", be self-organized and cross-functional, and ideally consist of 5-7 people.
- ***Chapter*** is a group of engineers who have the same manager (Chapter Lead) and is focused on personal growth and skills development. Engineers in chapters share knowledge, learn from each other, and discuss common challenges.

- All squads are organized into **tribes** containing 30-200 people each. Tribes have a clear mission, follow a set of foundation principles, and are led by senior experienced leaders responsible for the people, process, technology and culture. All skills necessary to build features are present within a tribe to engineer working software end-to-end.
- A **guild** is a community of interest, a group of people with similar skills that share knowledge, tools or code across Spotify. Guilds are designed beyond the formal structures and unite members with shared interests, whether leisure-related (cycling, photography or coffee drinking) or engineering-related (web development, backend, C++ engineering, or agile coaching).

With the company growth to the size of six research and development offices in three countries, the role of guilds in promoting collaboration among engineers across the company becomes even more important. At the same time, geographic and temporal distribution, and voluntary membership, means that implementing successfully functioning guilds is a challenge.

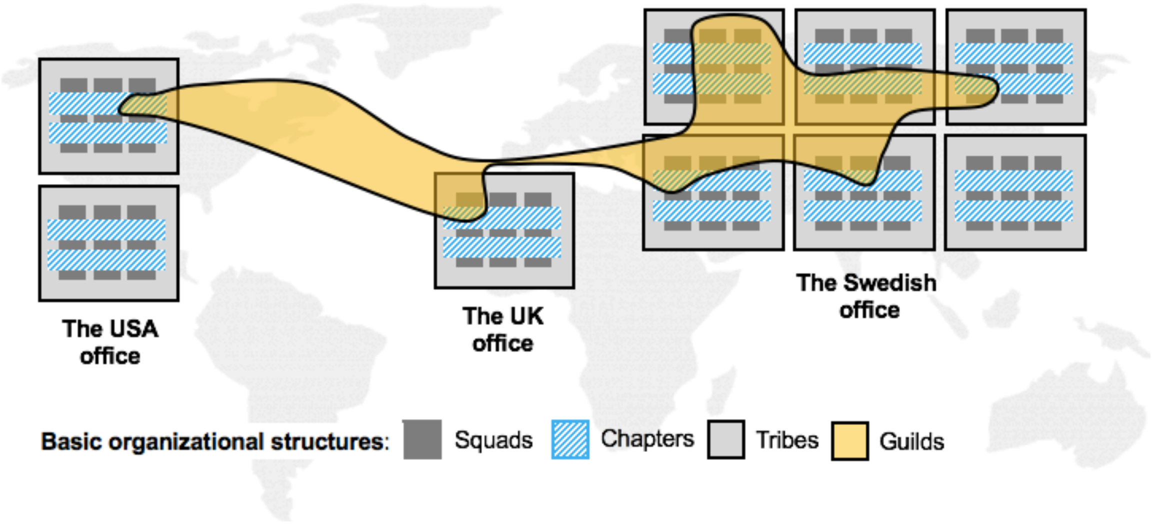


Figure 1. Spotify's Basic Organizational Structures

On communities of practice

Guilds or Communities of practice (CoPs) are not a new phenomenon. Communities existed in the cave times, when people gathered around a fire to discuss strategies for cornering prey[2]. Communities are cultivated for their potential to influence the knowledge culture[3] and bring value on the individual level (e.g. forum for expanding skills and expertise, strong sense of professional identity), team/project level (e.g. arena for problem solving, quick answers to questions) and company level (e.g. coordination and standardization across units, knowledge-based alliances, increased retention of talent)[2]. In large-scale agile organizations, CoPs are recognized for alleviating the inter-team coordination[4,6].

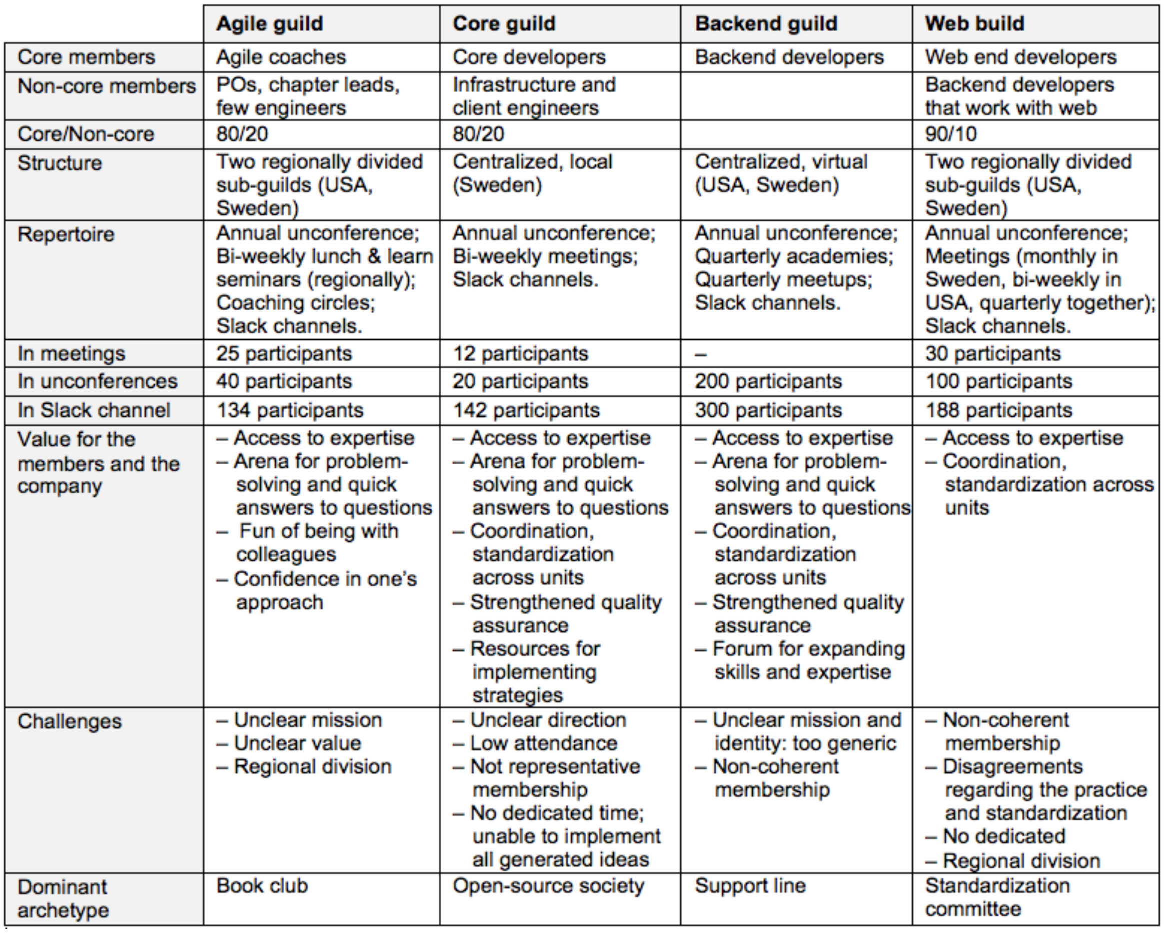
Communities vary in the way they are designed (size, mission, membership, activities). Communities in knowledge-intensive companies are organic emergent groups of practitioners concerned with improvement of a joint practice. It's not uncommon that such communities in early stages of formation focus on solving current problems of community members, and, as the community grows, its repertoire and deliverables become more deliberate and systematic[2]. Implementing well-functioning communities is, however, not an easy task[2-4]. Experience from Oracle Corporation, UK National Health Service, Hewlett Packard, Wipro Technologies, Alcatel and Daimler Crysler suggests that cultivation of knowledge culture requires organizational attention, support and sponsorship for CoPs[3]. Experience from four communities of practice at Ericsson[4] shows that success factors include a good topic, passionate leader, proper agenda, decision-making authority, open community, supporting tools, suitable rhythm, and cross-site participation when needed. Our work is dedicated to extending the current research and advice on implementing CoPs.

How we conducted the study: During the winter of 2018, we conducted nine semi-structured interviews with core members from four Spotify guilds (sponsors and coordinators), focusing on the main characteristics, success factors and challenges. We then studied one specific guild in detail through four semi-structured interviews with active and inactive guild members. All interviews were 45-60 min long and were recorded. We discussed our findings in iterations, aiming at deriving good practices for designing, running and supporting guilds. The results contain guild archetypes, which emerged from cross-guild comparison and success criteria for guilds, which emerged from the qualitative analysis of the interviews, during which we validated and extended the published success factors for CoPs[3,4].

Spotify guilds

Spotify guilds are open to anyone, and have representatives from different squads, tribes and chapters, providing a potential for making better decisions, helping others and sharing valuable knowledge across the organization. There are both organically emerging guilds and designed guilds. The latter are called sponsored guilds and have a stakeholder (sponsor) and a budget per member. Spotify guild activities are run by one or several *guild coordinators*, who are the main contact persons, "bootstrap" the guild to enable self-organization, ideally trying to get rid of the need for the coordinator role. All guilds have a Google group, Slack channels, and most have regular meeting fora, including yearly co-located unconferences (open space and discussions). An engineer can join multiple guilds, depending on the interest. Guild members can choose how active or inactive they want to be. Some of the guilds have many members, some have few, some have a lot of activity and some activate only once a year. Some die and new emerge. Some are seen as successful while others struggle (see a summary of four sponsored Spotify guilds in Table 1). Yet, the importance of guilds is undisputed and therefore we sought improvement ideas by visiting four different guilds and interviewing their core members.

Table 1. Guild profiles



	Agile guild	**Core guild**	**Backend guild**	**Web build**
Core members	Agile coaches	Core developers	Backend developers	Web end developers
Non-core members	POs, chapter leads, few engineers	Infrastructure and client engineers		Backend developers that work with web
Core/Non-core	80/20	80/20		90/10
Structure	Two regionally divided sub-guilds (USA, Sweden)	Centralized, local (Sweden)	Centralized, virtual (USA, Sweden)	Two regionally divided sub-guilds (USA, Sweden)

Repertoire	Annual unconference;
	Bi-weekly lunch & learn seminars (regionally);
	Coaching circles;
	Slack channels.
		Annual unconference;
		Bi-weekly meetings;
		Slack channels.
			Annual unconference;
			Quarterly academies;
			Quarterly meetups;
			Slack channels.
				Annual unconference;
				Meetings (monthly in Sweden, bi-weekly in USA, quarterly together);
				Slack channels.

In meetings	25 participants	12 participants	–	30 participants
In unconferences	40 participants	20 participants	200 participants	100 participants
In Slack channel	134 participants	142 participants	300 participants	188 participants
Value for the members and the company	– Access to expertise – Arena for problem-solving and quick answers to questions – Fun of being with colleagues – Confidence in one's approach	– Access to expertise – Arena for problem-solving and quick answers to questions – Coordination, standardization across units – Strengthened quality assurance – Resources for implementing strategies	– Access to expertise – Arena for problem-solving and quick answers to questions – Coordination, standardization across units – Strengthened quality assurance – Forum for expanding skills and expertise	– Access to expertise – Coordination, standardization across units
Challenges	– Unclear mission – Unclear value – Regional division	– Unclear direction – Low attendance – Not representative membership – No dedicated time; unable to implement all generated ideas	– Unclear mission and identity: too generic – Non-coherent membership	– Non-coherent membership – Disagreements regarding the practice and standardization – No dedicated – Regional division
Dominant archetype	Book club	Open-source society	Support line	Standardization committee

Archetypes

While analyzing the guilds, it became evident that there is a need to describe the differences between guilds. We derived four archetypes (patterns) in relation to the guild mission and activities. All guilds consist of a mixture of archetypes but have one dominant:

- **Book clubs** focus on learning rather than doing and activities as lunch-and-learn seminars, invited guests that inspire, and discussing ways to conduct a practice in a better way. Book clubs rarely make decisions or standardize the practice, since the emphasis is on the versatile competence development rather than putting limitations.
- **Open source societies** focus on maintaining, improving and setting the future strategy for owned components. Such societies resemble an onion structure with a small group of core contributors in the middle, then active developers, and then readers and passive users of the components (engineers who shall integrate their code, or reuse pieces of the code).
- **Support lines** focus on onboarding new engineers into a practice, providing quick answers to technical questions and facilitating solution discussions. Support lines may have hundreds of members who never meet, but rely on a few core experts whose engagement is paramount for guiding less experienced engineers.
- **Standardization committees** focus on aligning a practice across the company through establishment of concrete artefacts, such as coding standards, toolset recommendations.

Common challenges

We found a number of recurring challenges in the guilds studied:

- **Defining the purpose:** Defining the purpose and expected value was the number one challenge. Some of our interviewees said that their guilds are designed for one purpose (archetype), but in practice fulfill a different one.
- **Finding dedicated time:** In guilds that have tasks (developing guidelines, improving repositories or piloting new ways of working) members often struggle with finding time for the guild work. This is a common challenge especially for standardization and open source type of guilds, threatening turning these guilds into book clubs.
- **Achieving engagement and attendance:** All guilds that arrange regular meetings experienced challenges to achieve representativeness and engagement beyond the 10-15% of the members. We found that obstacles for attendance include practical reasons (time zone challenges and conflicting priorities) and motivational reasons (perceived low value of attending meetings).
- **Cross-site links:** Guilds with cross-site members, especially from far-off locations, reported a challenge of scheduling joint meetings. As a result, it was not uncommon to have decentralized guild activities, inhibiting cross-site knowledge sharing.

Prerequisites for success

Building upon existing research and elicited suggestions for improving guilds at Spotify, we extend the current success criteria (see Figure 2).

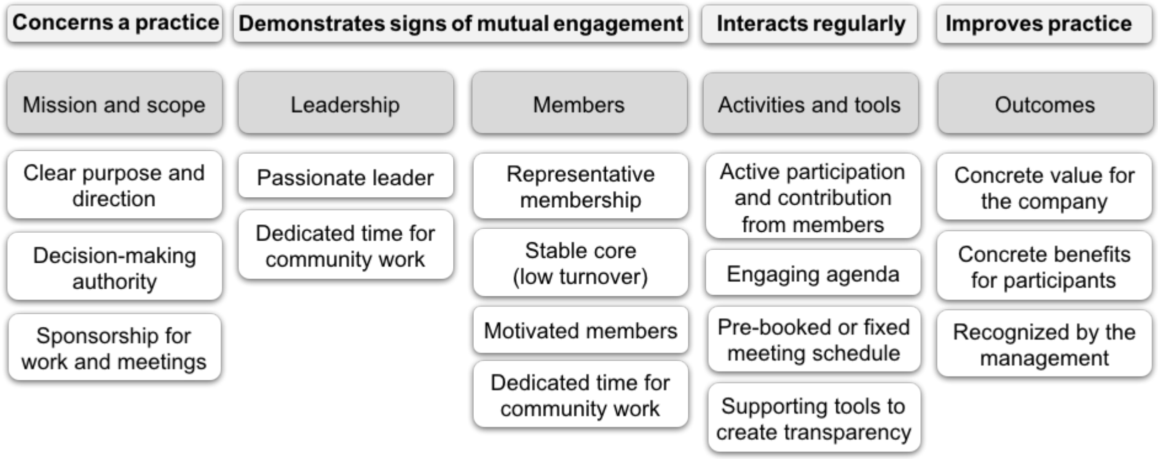

Concerns a practice
Demonstrates signs of mutual engagement
Interacts regularly
Improves practice
Mission and scope
Clear purpose and direction
Decision-making authority
Sponsorship for work and meetings
Leadership
Passionate leader
Dedicated time for community work
Members
Representative membership
Stable core (low turnover)
Motivated members
Dedicated time for community work
Activities and tools
Active participation and contribution from members
Engaging agenda
Pre-booked or fixed meeting schedule
Supporting tools to create transparency
Outcomes
Concrete value for the company
Concrete benefits for participants
Recognized by the management

Figure 2. Success criteria for guilds.

Concerning a practice: Similarly to previous research[2], our study shows that successful guilds have a shared understanding and concern for a practice. Guilds whose members had disagreements about what a practice represents, or did not have a clear direction, experienced what we call "existential problems". We also found that guilds fulfillment of a mission was dependent on their authority and organizational attention, support and sponsorship, as also found in prior research[3].

Demonstrating signs of mutual engagement: We found that challenges with engagement, attendance and representative membership were often related to a lack of dedicated time, low motivation, high turnover and lack of collocation, which guild coordinators tried to overcome by proactively contacting the members. As one coordinator explained: *"We try to get people to the meetings, that we think should be there, that probably want to join, but some people are in US that are part of the time zones, and some are doing other projects currently..."*

Interacting regularly: Regular interactions were found important for guilds, as suggested in previous research[2,4]. We found that Slack channels, Google groups, Trello boards and other tools provide valuable means for interaction and transparency. However, face-to-face sessions were said to be crucial for boosting guild activity. Successful guild meetings were associated with active participation and member contribution, an engaging agenda, and suitable schedule (ideally pre-booked).

Improving practice: Delivering value is key to community life, especially because participation in most communities is voluntary[2]. We found that guilds seek value creation for both members and Spotify. Challenged guilds with existential concerns doubted that they deliver value, or that the value created is recognized by the management. As one of the interviewees explained: *"If we had a better idea of what we are, and what our value proposition is, then we would be in a better place to sell that"*.

Finally, we found that well-functioning of the guilds in the large requires extra effort for enabling regular interaction and mutual engagement across remote sites.

Conclusions

"**A**gile in the large" has been[5] and still is[6] one of the top burning research questions. Based on a study of guilds in Spotify, we found that guilds play an important role in scaling agile as a bottom-up function enabling knowledge sharing, sharing responsibilities for the common code, developing new and aligning current development practice across many teams and sites. These organizational establishments go beyond the formal structures and bring together people from different organizational units and locations enabling natural communication and lightweight coordination, and avoiding unnecessary and hard to find documentation as in many traditional large companies[7]. Yet, we learned that implementing guilds in Spotify is not an easy task. There was no guaranteed how-to recipe for guilds due to their differences in purpose, design, membership, and repertoire. We classified the guilds into four archetypes, which overlap with existing research. Prior research on CoPs[4] found that they are related to knowledge sharing and learning (our book clubs), coordination and technical work (our open source societies), and participation in organizational development (our standardization committees). In addition, we found support lines as a special type of CoPs that provide important technical guidance for inexperienced practitioners, may have hundreds of members but never meet.

Based on our results, we recommend starting the implementation of guilds by identifying the mission, scope and expected value on an individual and organizational levels. The identified archetypes can help in choosing the targeted repertoire. Members of established guilds will benefit from conducting a retrospective using the identified challenges and proposed success criteria as input, and by revisiting the mission and scope for guilds on a regular basis.

References

1. Olsson, H. H. & Bosch, J. (2016). No More Bosses? A Multi-case Study on the Emerging Use of Non-hierarchical Principles in Large-Scale Software Development. In Proceedings of PROFES conference, pp. 86-101.
2. Wenger, E., McDermott, R. A., & Snyder, W. (2002). *Cultivating communities of practice: A guide to managing knowledge*. Harvard Business Press.
3. Oliver, S., & Reddy Kandadi, K. (2006). How to develop knowledge culture in organizations? A multiple case study of large distributed organizations. *Journal of knowledge management,* 10(4), 6-24.
4. Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization–Case Ericsson. *Information and Software Technology*, 56(12), 1556-1577.
5. Dingsøyr, T., & Moe, N. B. (2014). Towards principles of large-scale agile development. In Proceedings of XP conference (pp. 1-8).
6. Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87-108.
7. Dingsoyr, T., & Šmite, D. (2014). Knowledge Management Strategies in Global Software Development

Projects. *Journal of IT Pro*, January/February, 22-29.