

Escalando o Agile na Spotify: exemplo de sucesso de Lean Startup, Scrum e Kanban

A implantação da cultura ágil em empresas envolve muita comunicação, integração entre todas as áreas, treinamento, sensibilidade perante às mudanças e evolução gradativa. Após a implantação, e quando as equipes já estão entregando software de forma ágil, surge outro desafio: como escalar as práticas ágeis para as demais equipes. Veja como a Spotify venceu esses desafios.

A motivação da escrita deste artigo surgiu após o estudo de casos de diversas empresas, que estão passando pelo mesmo dilema em direção a escalar o Agile em ambientes complexos e com várias equipes, sistemas e profissionais. Esse artigo tem o intuito de apoiar quem está no mesmo caminho.

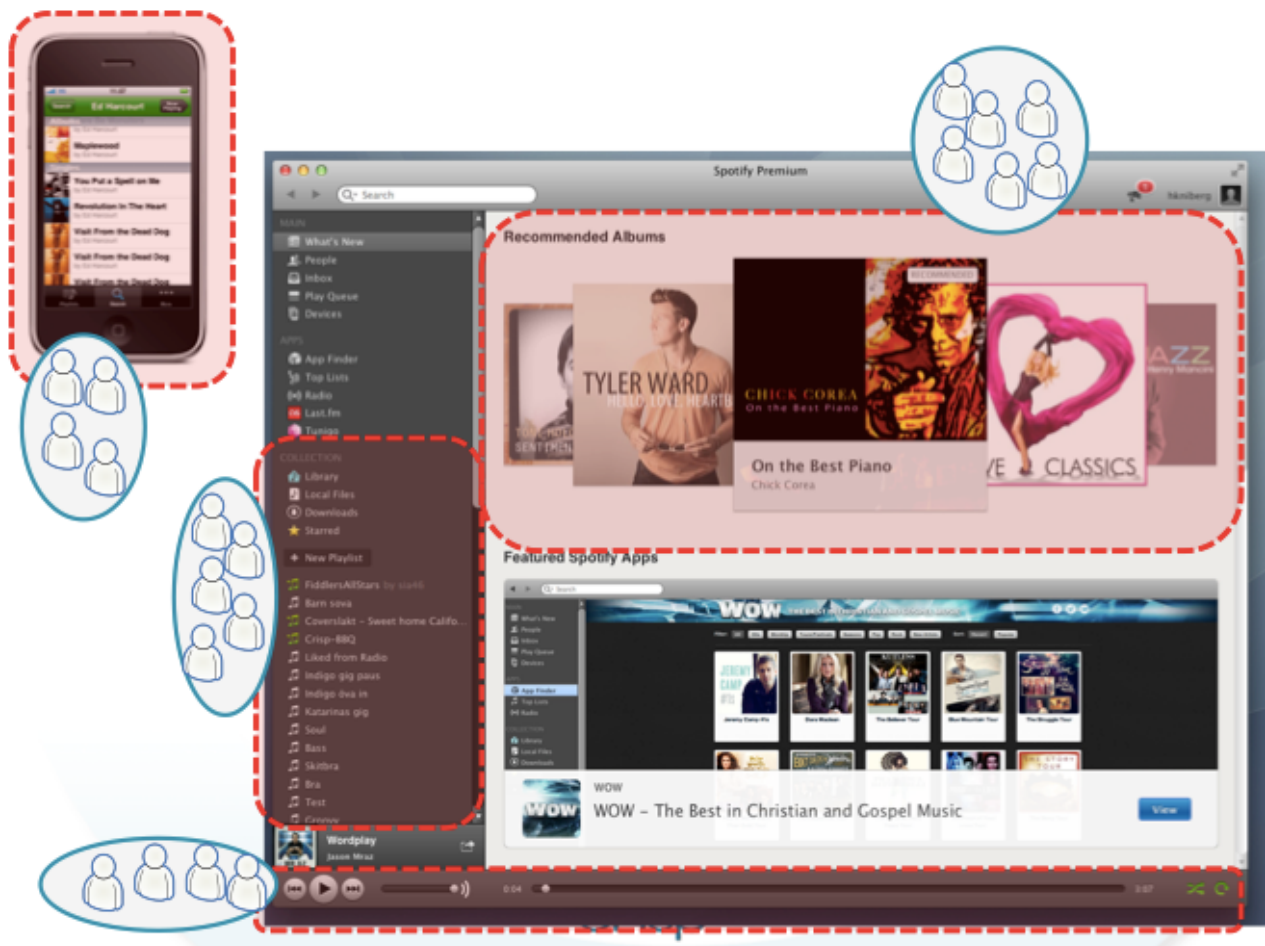
O estudo de caso mais expressivo que encontrei, em relação a escalar as práticas ágeis, foi o da empresa Spotify Ltd, fundada na Suécia em 2006, idealizadora do software Spotify, lançado em outubro de 2008. Atualmente, a sede da empresa está localizada em Londres, e há grupos de desenvolvimento de produto em Estocolmo, Nova York e São Francisco, distribuídas em mais de 30 equipes. O Spotify é um serviço de transmissão de música comercial com fornecimento de conteúdo protegido por direitos digitais, a partir de uma série de grandes gravadoras e independentes, incluindo a Sony, EMI, Warner Music Group e Universal. Em dezembro de 2012, a empresa atingiu 20 milhões de usuários, sendo que 5 milhões possuem assinatura mensal.

Agradeço ao Henrik Kniberg e Anders Ivarsson, por contribuírem na divulgação desse exemplo de sucesso em um artigo, além das conferências e workshops que estão ministrando ao redor do mundo, com a finalidade de ajudar outras empresas enfrentando o mesmo problema. Henrik Kniberg gentilmente autorizou o uso de imagens do seu blog neste artigo.

Organização das equipes

Uma equipe de desenvolvimento na Spotify é chamada Squad ("Esquadrão"). Uma equipe é similar a uma equipe Scrum, funcionando como uma mini-startup. A equipe é auto-organizada e possui autonomia suficiente para decidir o seu próprio processo interno, além de ter contato direto com os *stakeholders* (partes interessadas). Algumas usam Scrum, outras Kanban (a equipe de operações é um exemplo) e algumas utilizam uma mistura de ambos.

Cada equipe é responsável por diferentes partes da experiência do usuário: software para Android, software para iPad, sistemas de retaguarda, de pagamentos, de rádio, entre outros. Na figura a seguir, observe que uma equipe é responsável pelo aplicativo para iPhone, outra pela interação com o usuário (nesse caso, são sugeridos álbuns de acordo com a preferência); outra equipe cuida da barra lateral esquerda e mais uma do tocador de música. Ou seja, as equipes são divididas basicamente por área de funcionalidade.



Fonte: <http://blog.crisp.se/author/henrikkniberg>

As equipes utilizam também os princípios de Lean Startup, tais como MVP (produto mínimo viável), e validação do aprendizado (utilizar métricas e testes A/B para descobrir o que funciona e o que não). O slogan da empresa é: "Pense, construa, entregue e ajuste".

O espaço e o ambiente da Spotify proporcionam muita colaboração, abertura, informalidade e diversão (veja a foto abaixo). Praticamente todas as paredes são quadros.



Fonte: <http://blog.crisp.se/author/henrikkniberg>

A empresa promove aprendizado e inovação por meio do apoio que oferece para cada equipe investir 10% do seu tempo nos chamados "hack days". Nesses dias, os profissionais são estimulados a aprender e desenvolver novas ideias, compartilhando-as com os seus colegas de trabalho. É uma forma de incentivar que a equipe fique atualizada com as últimas tecnologias e ferramentas. A frequência é quinzenal e as reuniões duram um dia inteiro.

Quanto à formação da equipe, ela é constituída pelos profissionais de desenvolvimento, um Product Owner e um Agile Coach, sendo este último compartilhado por outras equipes relacionadas à mesma parte de um produto. O Product Owner é responsável pelo backlog da sua equipe e pela priorização das histórias de usuários, de forma que todos possam colaborar, no sentido de manter um plano geral de alto nível que mostra para onde a empresa está indo. O Agile Coach ajuda a manter o fluxo de trabalho, facilitando as retrospectivas, as reuniões de planejamento do Sprint, efetuando o coaching de cada um, além de outras atribuições.

Melhoria contínua

Em relação à melhoria contínua e o desafio de gerir mais de 30 equipes, a empresa realiza uma pesquisa quadrimestral, que ajuda a própria estrutura organizacional a prestar a assistência necessária. Veja o resultado da pesquisa, sumarizado de forma visual:

Area	Squad 1	Squad 2	Squad 3	Squad 4	Squad 5
Product owner					
Agile coach					
Influencing work					
Easy to release					
Process that fits team					
A mission					
Org. support					

Fonte: <http://blog.crisp.se/author/henrikkniberg>

Os círculos mostram o estado atual e as setas indicam a tendência. Por exemplo, a Equipe 4 não possui apoio suficiente do Agile Coach (indicado pelo círculo vermelho), mas essa situação já está melhorando (seta com o sinal verde para cima).

Cada área pesquisada é explicada a seguir:

- Product Owner: prioriza o trabalho, levando valor de negócio e aspectos técnicos em consideração.

- Agile Coach: ajuda a identificar os impedimentos e treina a equipe para melhorar continuamente o processo.
- Influência no trabalho: cada membro da equipe possui auto-suficiência, tem "voz ativa" no planejamento, escolhe quais tarefas irá trabalhar e pode investir 10% do seu tempo nos hack days.
- Facilidade para implantação: a equipe pode implantar uma release com o mínimo esforço.
- Processo da equipe: a equipe é a dona do processo e o melhora continuamente.
- Missão: a equipe tem uma missão clara e as histórias do backlog são relacionadas a ela.
- Apoio organizacional: a equipe conhece o "caminho das pedras" para se obter o suporte necessário em caso de algum problema corporativo e questões técnicas.

Grupos com foco em produto

Um grupo de equipes de várias esquadras que trabalham em um mesmo produto ou áreas relacionadas é chamado de "Tribo". Exemplos de tribos são: reprodutor de música e infraestrutura do backend.

Cada tribo possui total autonomia, sendo que um líder fica responsável por manter o melhor ambiente de trabalho para as equipes que estão inclusas nele. As equipes de cada grupo ficam localizadas em um mesmo escritório, normalmente uma ao lado da outra, com uma área de lazer próxima, afim de promover melhor colaboração.

Os integrantes de cada grupo se reúnem frequentemente e de maneira informal, e exploram o que cada equipe lançou recentemente, apresentando demos, novas ferramentas e tecnologias, além de projetos inovadores que surgiram fruto dos hack days.

Dependências de outras equipes

As equipes se reúnem diariamente com a finalidade de identificar e solucionar as dependências que existem entre as funcionalidades, mantendo um quadro com anotações de quais não foram tratadas e assim poderem acompanhar. Essa reunião é similar com a prática do Scrum chamada "Scrum of Scrums", mas a dinâmica é diferente.

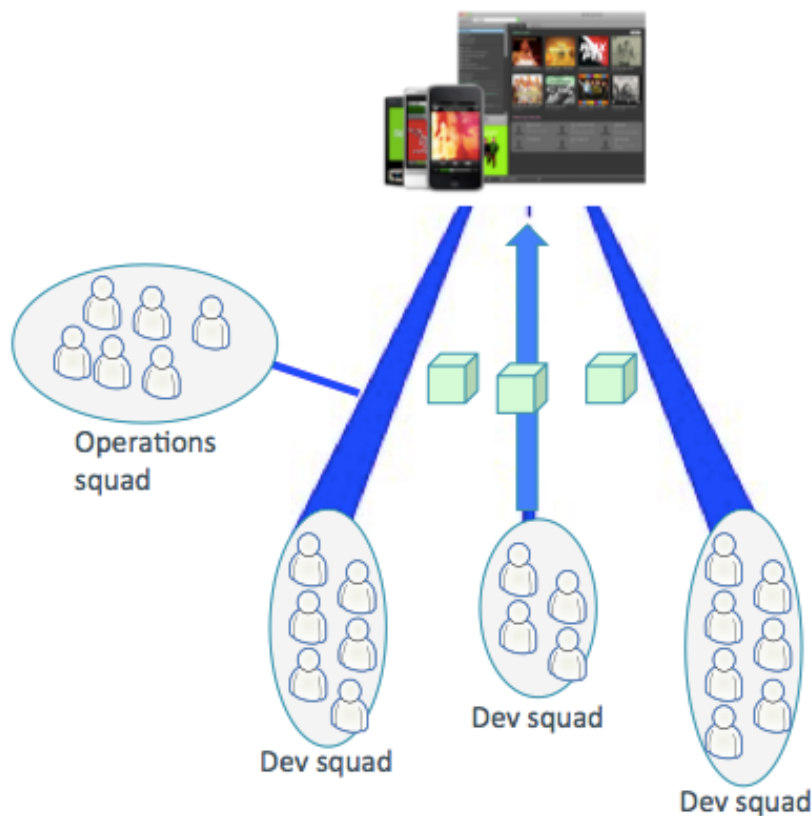
Um exemplo de como mantêm as anotações das dependências que surgem é apresentado a seguir:

	A	B	C	D	E
1	Squad	Depends on	Dependency	Comment	Same tribe?
2	Music Player				
3	Content	Ops	Slowing	Need machines, connections, help set-up things etc. Works really well in general, but at times the workload on operations causes the lead times to grow and slow us down	No
4	Content	NeXT	No problem	Storage. Not big, mostly information/communication needs to happen.	No
5	Content	BFS	No problem	Replacement service	Yes
6	Content	Team 2	No problem	Communication around next story	No
7	Content	Team 1	Future	Content ingestion	No
8	BFS	UX	Slowing	Need UX to discuss, review and provide mock-ups.	No
9	BFS	Content	No problem	Normal dependencies, sprint work.	Yes
10	BFS	Mobile	Slowing	No internal mobile developers within Squad.	No
11	BFS	Analytics	Slowing	A/B test results slowing down roll outs of features	No
12	BFS	Team 3	Blocking	Waiting for data dumps	No
13	BFS	Team 1	Future	Waiting for data dumps	No
14					

Fonte: <http://blog.crisp.se/author/henrikkniberg>

Uma dependência comum em várias empresas está relacionada à equipe de operações, o que geralmente causa bastante atrito com a equipe de desenvolvimento. A adoção do DevOps está sendo largamente difundida em empresas, com a missão de melhorar a comunicação entre as duas equipes e estreitar os laços de colaboração e integração.

Na Spotify, isso não é necessário, pois a equipe de operações não tem como função implantar software para as equipes. Por outro lado, o objetivo da equipe de operações é apoiar as equipes no que precisam para implantar o software por eles mesmos, além de ajudar na infraestrutura necessária, criação de scripts e rotinas. A missão de operações é: "construa a estrada para produção" (veja a figura abaixo). É uma colaboração informal, mas efetiva, baseada na comunicação cara a cara, ao invés de em um processo detalhado:

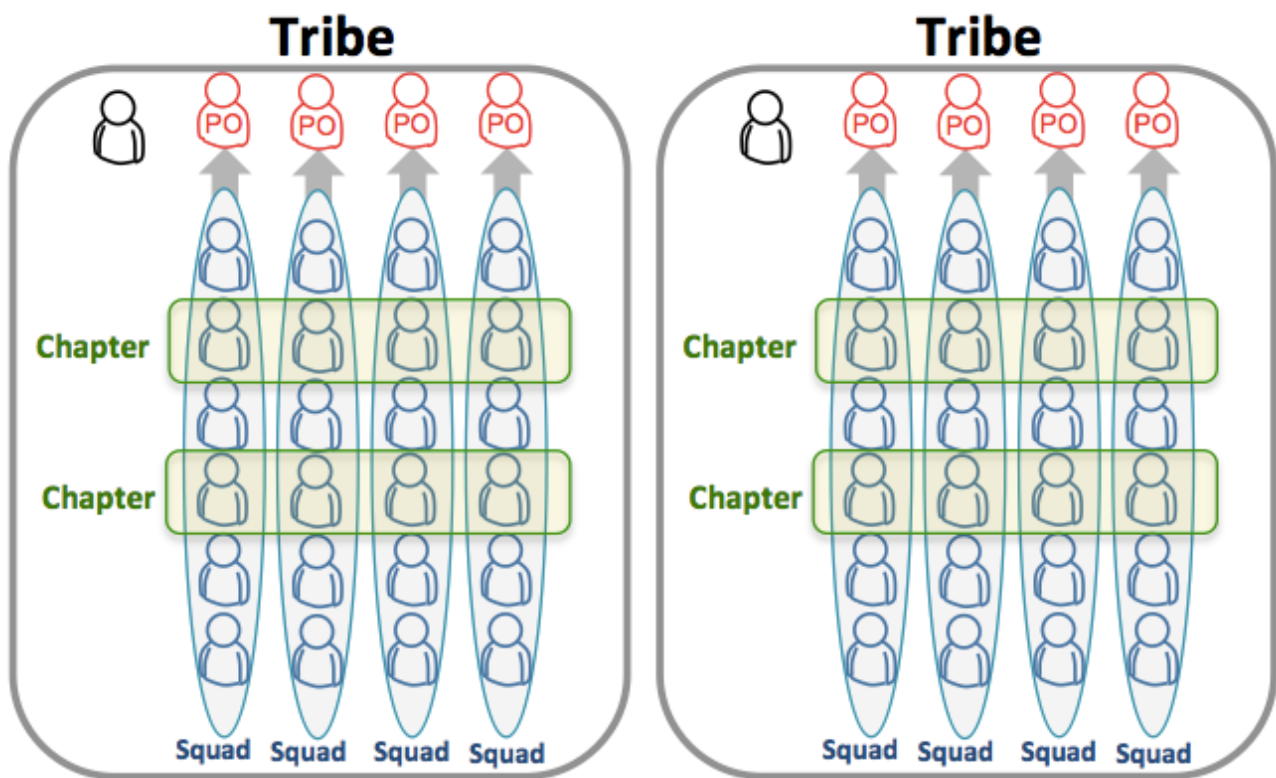


Fonte: <http://blog.crisp.se/author/henrikkniberg>

Comunidades de competências específicas e de interesse

Existe uma desvantagem, em relação a essa total autonomia das equipes e uma delas é a perda de economias de escala. Por exemplo, um analista de teste em uma equipe está enfrentando um problema que um outro analista de testes já solucionou na outra semana. Se todos os analistas de teste se reunissem frequentemente, eles poderiam compartilhar o conhecimento e criar ferramentas para o benefício de todas as equipes. Qual foi a maneira que a Spotify equacionou essa questão? E como podemos compartilhar o conhecimento entre os profissionais com a mesma função?

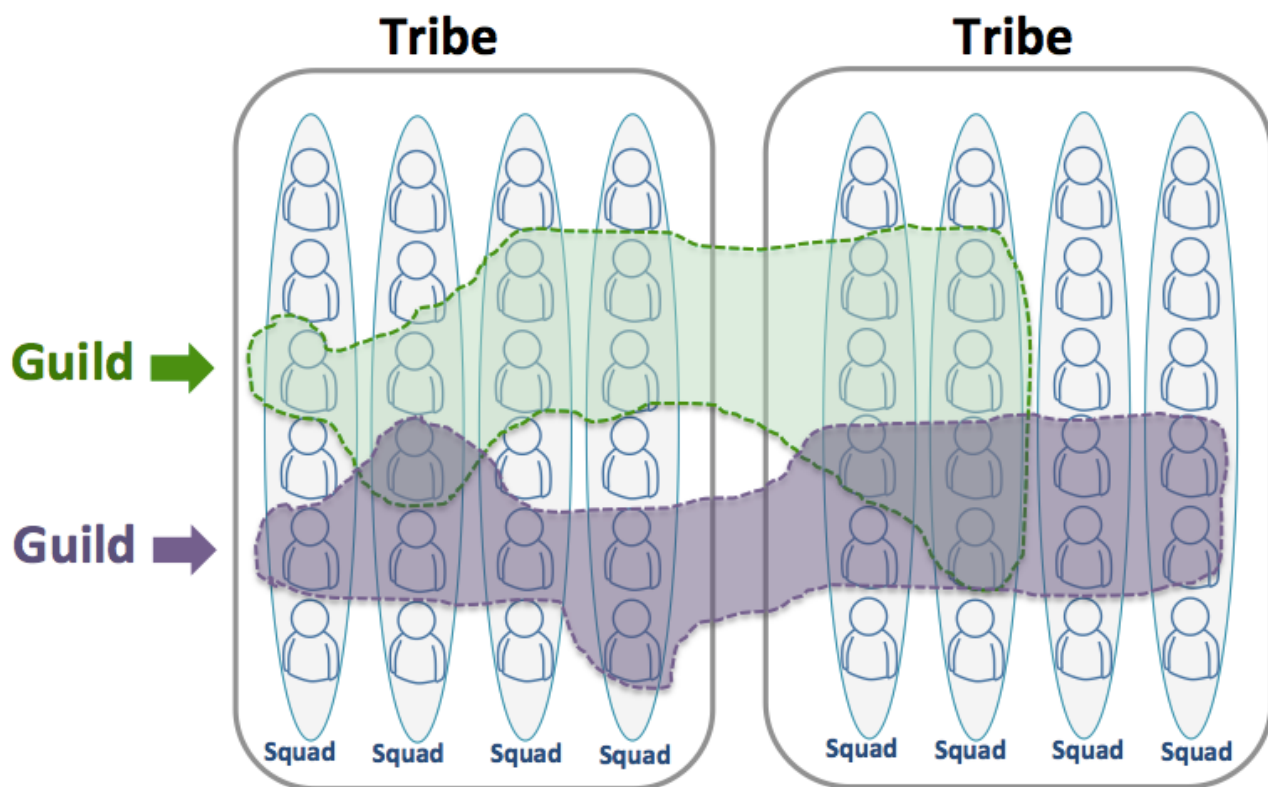
Na Spotify foi criado o conceito de "Chapter" (Divisão), um conjunto de profissionais com as mesmas habilidades e dentro da mesma área de competência, dentro da mesma tribo. Como pode ser observado na figura seguinte, existem quatro divisões, que por ventura podem ser relacionados a diferentes funções, por exemplo: uma divisão é composta por programadores, outro formado por analistas de teste, outro podendo ser os Agile Coaches e, por último, o dos Product Owners.



Fonte: <http://blog.crisp.se/author/henrikkniberg>

Em intervalos regulares, cada divisão se reúne para discutir a sua área de expertise e os seus desafios. Há o líder da divisão que é o gerente funcional dos membros, com as mesmas responsabilidades de um gerente tradicional, tais como desenvolver pessoas, promover, demitir, e assim por diante. Porém, esse líder também é parte da equipe e é envolvido nas tarefas diárias, o que o ajuda a encarar a realidade.

Existe um outro grupo chamado "Guild" (associação), uma espécie de comunidade de interesse que deseja trocar conhecimento, ferramentas, códigos e práticas. Enquanto os capítulos são locais a uma tribo, as associações possuem integrantes provenientes de outras tribos (veja a figura abaixo). Alguns exemplos de associações são associação de tecnologia web, associação de testes e associação de coaches ágeis.



Fonte: <http://blog.crisp.se/author/henrikkniberg>

Cada associação também possui um coordenador que ajuda a liderar as iniciativas de cada comunidade de interesse.

À primeira vista, essa estrutura aparenta ser uma estrutura de empresa matricial. Pelo contrário: as equipes são organizadas para lançar produtos, ou seja, cada uma delas é composta por desenvolvedores de backend e frontend, analistas de teste, analistas de usabilidade, Product Owner e Agile Coach. A dimensão horizontal é organizada justamente para compartilhar o conhecimento, ferramentas e código. A função do líder da divisão é facilitar palestras de discussões técnicas, conferências internas, além de abrir o caminho para que a comunicação entre as equipes ocorram com clareza.

Arquitetura de sistemas

A arquitetura de sistemas na Spotify é orientada a serviços, sendo que existem mais de cem sistemas distintos dos quais são mantidos e implantados separadamente. Qualquer um pode manter qualquer sistema, o que geralmente acontece, pois uma funcionalidade sempre requer que sejam atualizados vários sistemas. O risco a ser mitigado nessa situação é quanto à integridade de cada sistema, já que vários profissionais podem atualizá-lo.

Para tanto, foi criado um papel de "System Owner" (Proprietário do Sistema). Cada sistema possui um ou um par de System Owners. Para sistemas críticos, esse par é constituído de um desenvolvedor e alguém de operações. O System Owner foca na qualidade, documentação, dívidas técnicas, estabilidade, escalabilidade e o processo de implantação. Mas, ele não é um gargalo, e sim um membro da equipe que possui outras responsabilidades; tem apenas a responsabilidade maior de ser o proprietário daquele determinado sistema. Normalmente, o System Owner aloca um dia para fazer uma limpeza do código.

Existe também um papel de Arquiteto Chefe, que coordena o trabalho nas questões de arquitetura de alto nível e que cruza múltiplos sistemas. Ele revisa o desenvolvimento de novos sistemas, a fim de evitar descuidos comuns e permitir que estejam alinhados com a visão arquitetural. O feedback é sempre por meio de sugestões e conversas, a decisão final é da equipe.

Conclusões

A organização da Spotify e a forma como escalaram o Agile e Lean estão em constante processo de melhoria. No entanto, os princípios da empresa não mudarão tão cedo: transparência, informalidade, propósito e missão bem definidos; equipes auto-organizadas e com autonomia; poucos papéis, ambiente altamente colaborativo e foco em inovação. Vimos aqui um estudo de caso de uma grande empresa que conseguiu escalar o Agile e Lean com sucesso. Esse sucesso é mostrado pelos próprios números: 5 milhões de clientes pagantes, de um total de 20 milhões de usuários ativos. Os usuários já criaram mais de 1 bilhão de listas de reprodução.