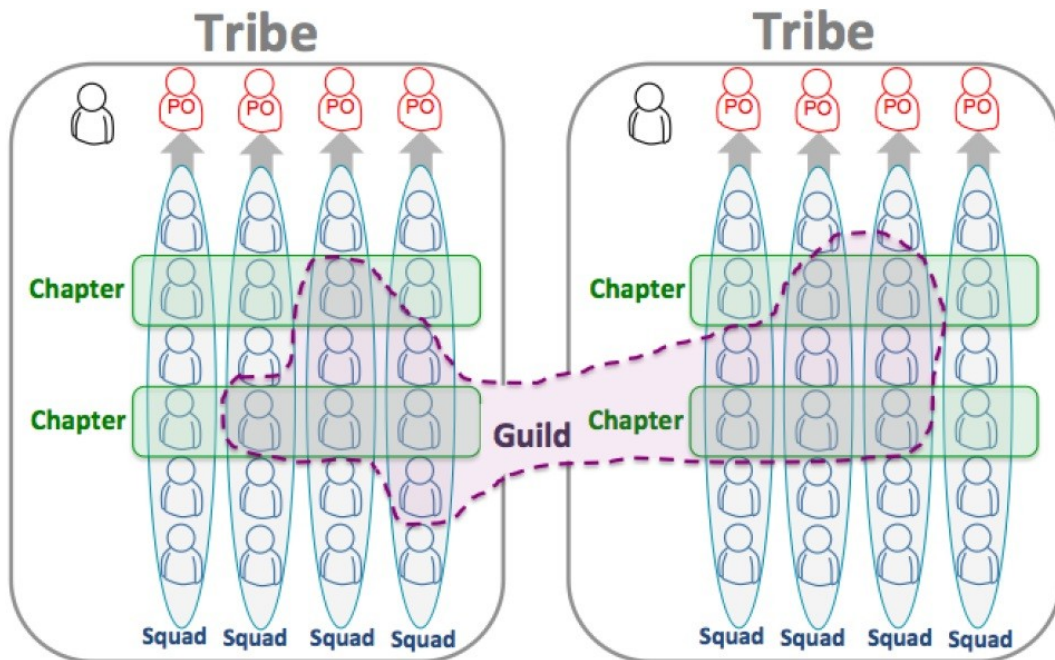


Agilité à grande échelle @ Spotify

avec les Tribus¹, Brigades², Chapitres³ & Guildes⁴

Henrik Kniberg & Anders Ivarsson
Octobre 2012



Gérer plusieurs équipes dans une organisation qui développe des produits est toujours un défi !

L'un des exemples les plus impressionnants que nous avons vu jusqu'ici est Spotify, qui a réussi à préserver un état d'esprit agile même après être monté jusqu'à 30 équipes réparties dans 3 villes.

Spotify est une entreprise fascinante qui est en train de transformer l'industrie de la musique. L'entreprise existe seulement depuis 6 ans et a déjà 15 millions d'utilisateurs actifs dont plus de 4 millions avec l'offre payante. Le produit lui-même peut être associé à « un lecteur de musique magique avec lequel vous pouvez instantanément trouver et écouter toutes les chansons du monde ».

Alistair Cockburn (l'un des pères fondateurs du développement agile de logiciels) a visité Spotify et a déclaré « Super ! Depuis 1992, je cherchais quelqu'un qui puisse mettre en œuvre ce format de matrice :) c'est donc avec grand plaisir que je vois ça ».

Donc, comment ça se gère ?

Nous avons présenté, lors de conférences et de diverses conversations, la façon dont nous travaillons chez Spotify et la manière dont l'entreprise gère l'agilité avec des centaines de

1 NdT : tribes

2 NdT : squads – j'avoue ne pas savoir si 'brigade' est la plus pertinente traduction dans ce contexte.

3 NdT : chapters

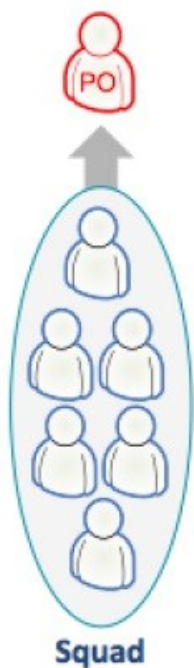
4 NdT : guilds

développeurs. Comme beaucoup de gens sont fascinés par cela, nous avons décidé d'écrire un article sur le sujet.

Avertissement : Spotify évolue rapidement (comme toute bonne entreprise agile). Cet article est donc seulement un cliché de notre manière actuelle de travailler, un voyage en cours, pas un voyage terminé. Le temps que vous lisiez ceci, des choses auront déjà changé.

Brigades

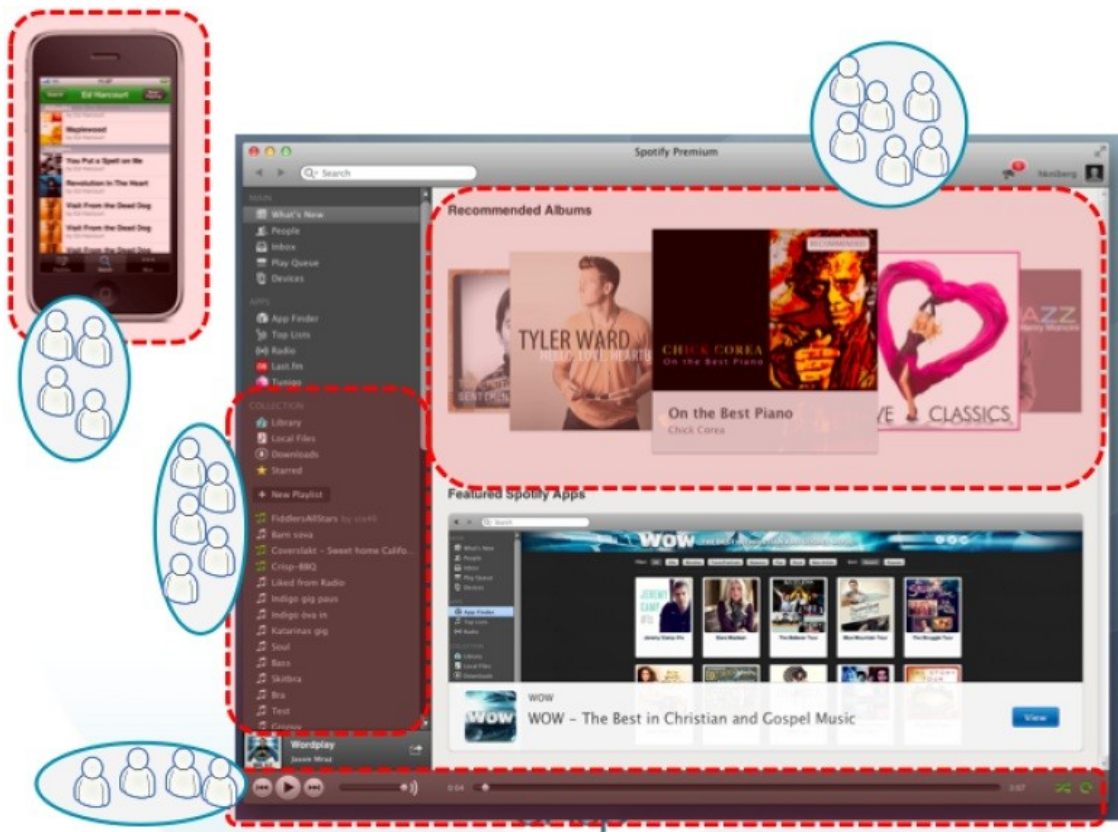
L'unité de base qui développe chez Spotify est la *Brigade*.



Une Brigade ressemble à une équipe Scrum, et est conçue pour se comporter comme une mini-startup. Ses membres sont colocalisés, et ils disposent de toutes les compétences et outils pour concevoir, développer, tester et livrer en production. Ils forment une équipe auto-organisée et décident eux-mêmes de leur manière de travailler, certaines équipes utilisent les sprints Scrum, d'autres Kanban, d'autres encore un mixte des deux approches.

Chaque brigade a une mission long terme telle que construire et améliorer le client Android, recréer une ambiance radio sur Spotify, dimensionner les systèmes backend⁵ ou fournir des solutions de paiement. L'image ci-dessous illustre la manière dont différentes brigades prennent la responsabilité sur différentes parties de l'interface utilisateur.

5 NdT: j'ai renoncé à traduire certains termes anglais, notamment systèmes backend en systèmes dorsaux.



Les brigades sont encouragées à appliquer les principes du Lean Startup tels que le MVP (produit minimum viable) et l'apprentissage validé. Le MVP implique de livrer tôt et régulièrement, l'apprentissage validé implique de passer par des métriques et des tests A/B pour découvrir ce qui marche réellement ou non. C'est ce qui est résumé au travers du slogan « Pensez-le, construisez-le, déployez-le, adaptez-le »⁶.

Étant donné que chaque brigade se voit confier une mission et une partie du produit pendant un temps assez long, ses membres peuvent réellement acquérir une expertise dans le domaine concerné, par exemple sur la mise en œuvre d'une super ambiance radio.

La plupart des brigades ont un environnement de travail exceptionnel qui comprend une zone de bureaux, un coin salon et une salle de réunion. Quasiment tous les murs sont équipés de tableaux blancs. Nous n'avons jamais vu de meilleurs espaces collaboratifs !

⁶ NdT : « Think it, build it, ship it, tweak it »



oui, il s'agit bien d'un requin suspendu... tout est parfaitement normal...

Pour promouvoir l'apprentissage et l'innovation, chaque brigade est encouragée à consacrer en gros 10% de son temps à des « journées libres »⁷. Pendant ces journées libres, les gens font ce qu'ils

⁷ NdT : « hack days »

veulent, typiquement essayer de nouvelles idées et les partager avec leurs collègues. Certaines équipes ont 1 journée libre toutes les deux semaines, d'autres les économisent pour faire une grosse « semaine libre ». Les journées libres ne sont pas seulement ludiques, elles sont aussi un moyen de se maintenir à niveau sur les nouveaux outils et techniques, et mènent parfois à d'importantes innovations produits !

Une brigade n'a pas de chef à proprement parlé, mais elle a un product owner. Le product owner est responsable de la priorisation du travail à réaliser par l'équipe, mais il n'est pas impliqué sur la manière dont l'équipe réalise ce travail. Les product owners des différentes brigades collaborent ensemble pour maintenir une feuille de route au plus haut niveau qui montre la trajectoire globale de Spotify, et chaque product owner est responsable de maintenir le backlog produit correspondant pour sa brigade.

Une brigade a également accès à un coach agile, qui aide ses membres à évoluer et améliorer leur façon de travailler. Le coach facilite les rétrospectives, facilite les réunions de planification de sprints, fait du coaching individuel, etc.

Idéalement, chaque brigade est pleinement autonome avec un accès direct aux parties prenantes, sans dépendances bloquantes avec les autres brigades. Une mini-startup à la base. Avec 30 équipes, il s'agit d'un véritable challenge ! Nous avons parcouru un long chemin, mais il y a toujours plein d'améliorations à apporter.

Pour faciliter cela, nous menons une revue trimestrielle avec chaque brigade. Cela nous aide à dédier nos efforts à l'amélioration et à découvrir quel type de support est nécessaire de la part de l'organisation. Voici la synthèse visuelle d'une de nos revues, illustrant 5 brigades au sein d'une tribu :

Domaine	Brigade 1	Brigade 2	Brigade 3	Brigade 4	Brigade 5
Product owner					
Coach Agile					
Influencer le travail à faire					
Facilité à livrer					
Processus adapté à l'équipe					
Une mission					
Support de l'orga.					

Les cercles illustrent l'état actuel, les flèches illustrent la tendance. Par exemple, nous pouvons voir que trois brigades remontent un problème sur la livraison et cela ne semble pas s'améliorer. Ce domaine réclame de s'y consacrer d'urgence ! Nous constatons également que 4 brigades ne bénéficient pas d'un grand support du coach agile, mais c'est déjà en cours d'amélioration.

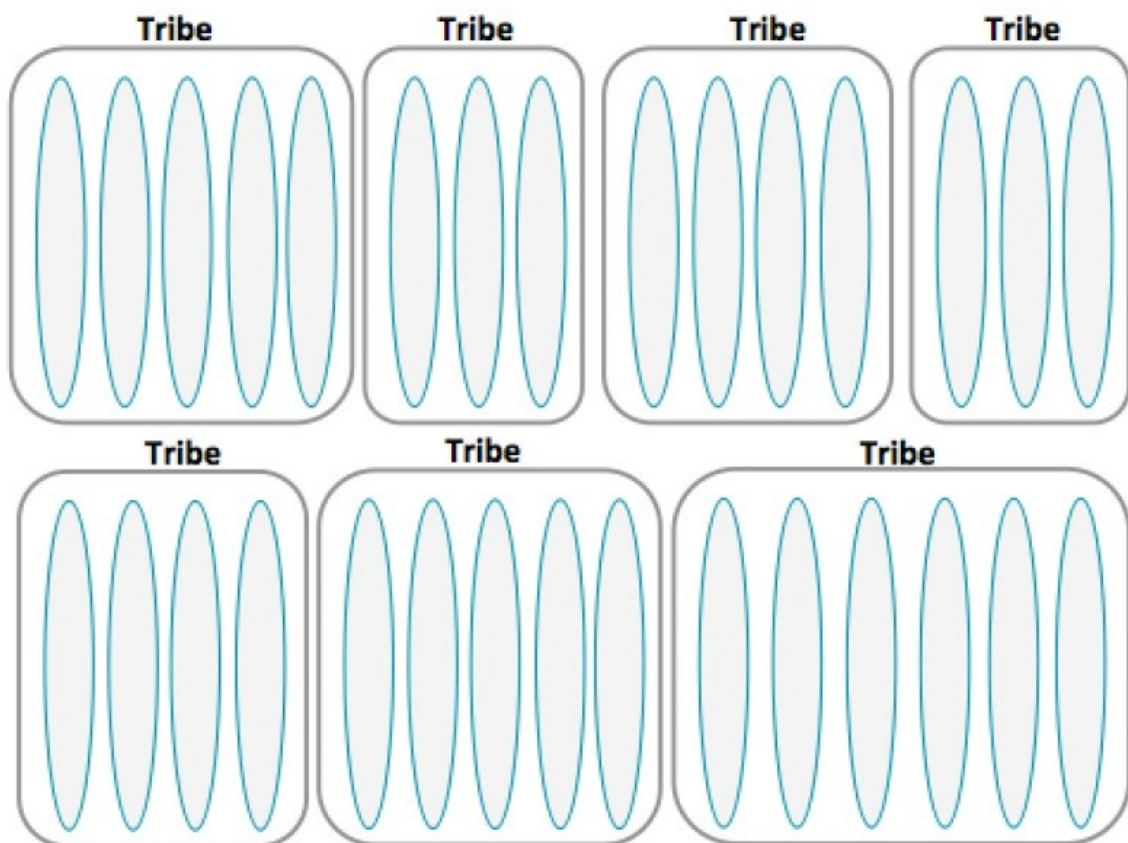
- **Product owner** – une brigade a un product owner dédié qui priorise le travail et qui prend

aussi bien en considération la valeur métier que les aspects techniques.

- **Coach agile** – une brigade dispose d'un coach agile qui aide ses membres à identifier les obstacles et qui les coache pour améliorer continuellement leur processus.
- **Influence sur le travail** – chaque membre d'une brigade peut influencer son travail, c'est-à-dire être un acteur dans la planification et le choix des tâches sur lesquelles il doit travailler. Chaque membre d'une brigade consacre 10% de son temps en journées libres.
- **Facilité à livrer** – une brigade peut (et le fait!) s'en sortir avec un minimum de tracas et de coordination.
- **Processus adapté à l'équipe** – une brigade est propriétaire de son processus et l'améliore constamment.
- **Mission** – une brigade a une mission que chacun connaît et pour lequel il est concerné, et les stories qui sont dans le backlog concernent cette mission.
- **Support de l'organisation** – une brigade sait vers qui se tourner pour obtenir de l'aide sur la résolution d'un problème, aussi bien technique qu'humain.

Tribus

Une *tribu* est une collection de brigades qui travaillent sur des domaines liés, tels que le lecteur de musique, ou l'infrastructure du backend.



La tribu peut être vue comme un « incubateur » pour les squads mini-startups, elle a un bon degré de liberté et d'autonomie. Chaque tribu dispose d'un leader qui est responsable de fournir le meilleur

habitat possible pour les brigades au sein de la tribu. Les brigades d'une tribu sont physiquement localisées dans le même bureau, normalement très proches l'une de l'autre, et les coins salons qui sont autour favorisent la collaboration entre les brigades.

Les tribus sont dimensionnées en se basant sur le concept du « Nombre de Dunbar »⁸⁹, qui énonce qu'un groupe ne peut maintenir un lien social au-delà d'environ 100 personnes (cette limite est en fait plus grande pour des groupes dont la survie est menacée, ce qui n'est pas réellement le cas chez Spotify, croyez-le ou non...). Lorsque les groupes deviennent trop gros, nous voyons apparaître des règles restrictives, de la bureaucratie, des choix politiques, des couches supplémentaires de management, et toute autre forme de gaspillage.

Les tribus sont donc dimensionnées pour inclure moins de 100 personnes environ.

Les tribus se rassemblent régulièrement de façon informelle pour montrer au reste de la tribu (ou à quiconque vient) ce sur quoi elles travaillent, ce qu'elles ont livré et ce que les autres peuvent apprendre de ce qu'elles sont en train de faire. Cela inclut des démos en live d'un logiciel opérationnel, de nouveaux outils et techniques, de projets réalisés pendant les journées libres, etc.



Dépendances entre les brigades

Lorsqu'il y a beaucoup de brigades, il y a toujours des dépendances. Les dépendances ne sont pas nécessairement mauvaises, les brigades doivent parfois travailler ensemble pour construire quelque chose de vraiment super. Néanmoins, notre objectif est d'avoir des brigades qui soient les plus autonomes possibles, en particulier en minimisant les dépendances qui bloquent ou ralentissent une brigade.

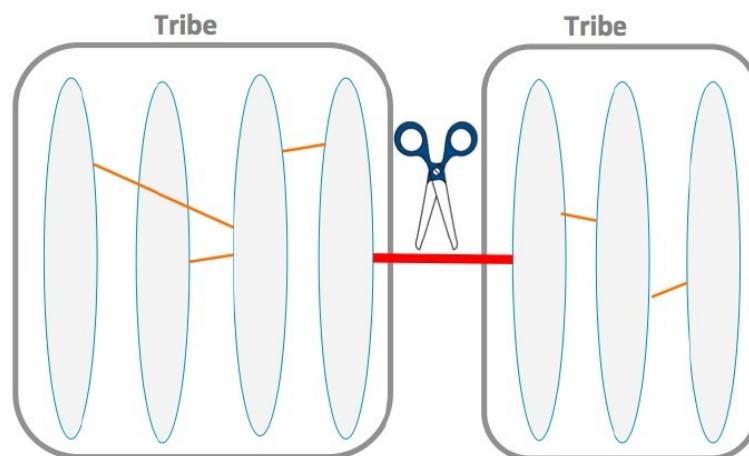
Pour faciliter cela, nous demandons régulièrement à nos brigades de quelles autres brigades elles dépendent, et jusqu'à quel point ces dépendances les bloquent ou les ralentissent. Voici un exemple :

8 http://en.wikipedia.org/wiki/Dunbar's_number

9 <https://agilarium.wikispaces.com/Avant+qu%27il+existe+un+Management>

	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					

Nous discutons ensuite des différents moyens d'éliminer les dépendances problématiques, en particulier les dépendances inter-brigades et bloquantes. Cela mène souvent à des repriorisations, réorganisations, changements d'architecture ou solutions techniques.

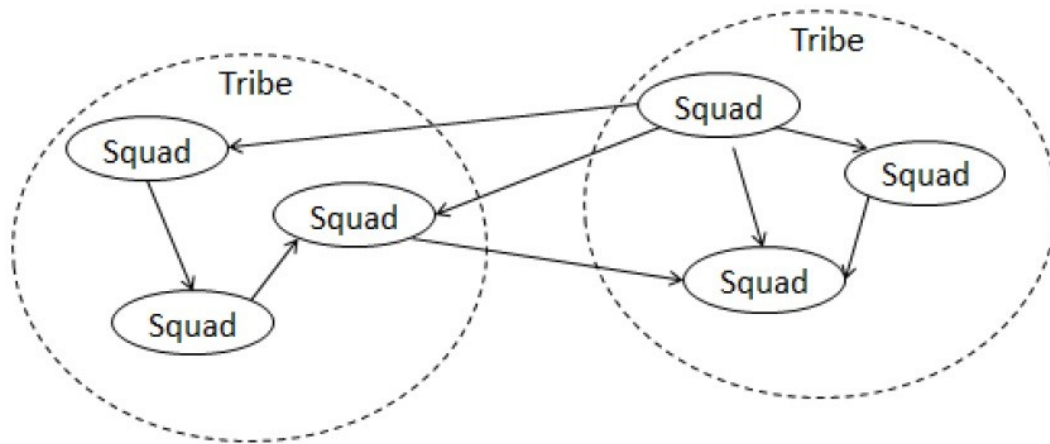


La revue nous aide également à repérer des patterns qui montreraient comment les brigades dépendent l'une de l'autre, par exemple de plus en plus de brigades semblent être ralenties par les opérations¹⁰. Nous utilisons un simple graphe pour suivre l'augmentation et la diminution des différents types de dépendances au fil du temps.

Scrum a une pratique qui s'appelle le « scrum de scrums », une réunion de synchronisation où une personne de chaque équipe rencontre les autres pour discuter des dépendances. Nous ne menons généralement pas de scrum de scrums chez Spotify, principalement parce que la plupart des brigades sont assez indépendantes et n'ont pas besoin d'avoir de telles réunions de coordination.

Au lieu de cela, les scrum de scrums se font « sur demande ». Par exemple, nous avons récemment eu un gros projet qui a nécessité le travail coordonné de plusieurs brigades sur quelques mois.

¹⁰ NdT : exploitation, helpdesk, ...

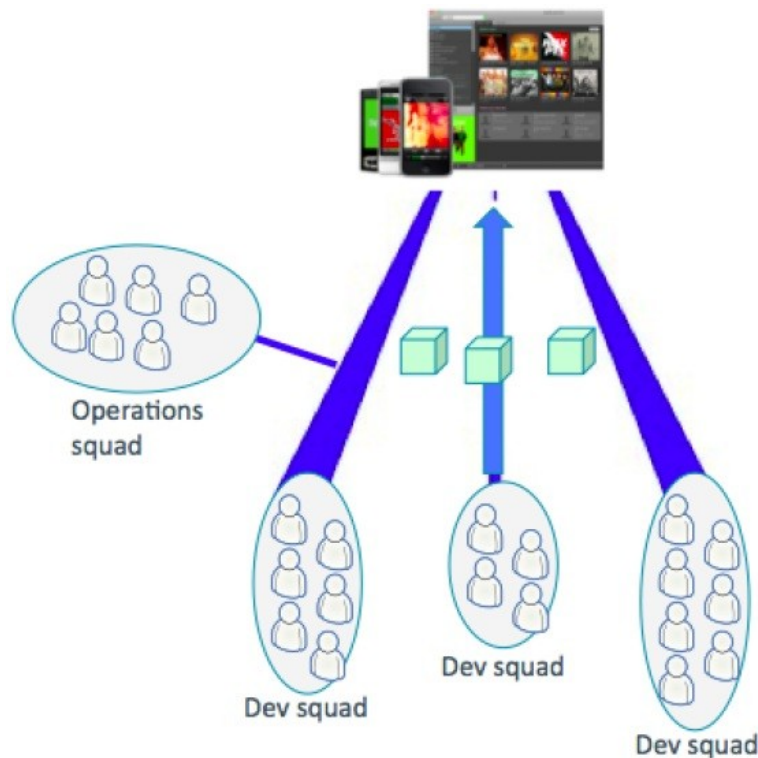


Pour réaliser ce travail, les équipes ont une réunion quotidienne de synchronisation pendant laquelle elles identifient et résolvent les dépendances entre les brigades, et maintiennent un tableau de post-its pour garder la trace des dépendances non résolues.



Une source commune de problèmes de dépendances dans de nombreuses entreprises se situe entre le développement et les opérations. La plupart des entreprises dans lesquelles nous avons travaillé, ont une sorte de passage de relais entre le développement et les opérations, avec des frictions et des retards.

Chez Spotify, il y a une équipe opérations séparée, mais leur travail n'est pas de faire les livraisons pour les brigades, leur travail est de donner aux brigades le soutien nécessaire pour qu'elles livrent leur code elles-mêmes ; support sous la forme d'infrastructure, de scripts et de routines. Elles « construisent la route vers production » en un sens.



Il s'agit d'un mode de collaboration informel mais efficace, basé sur la communication en face à face plutôt que sur une documentation détaillée du processus.

Chapitres et guildes

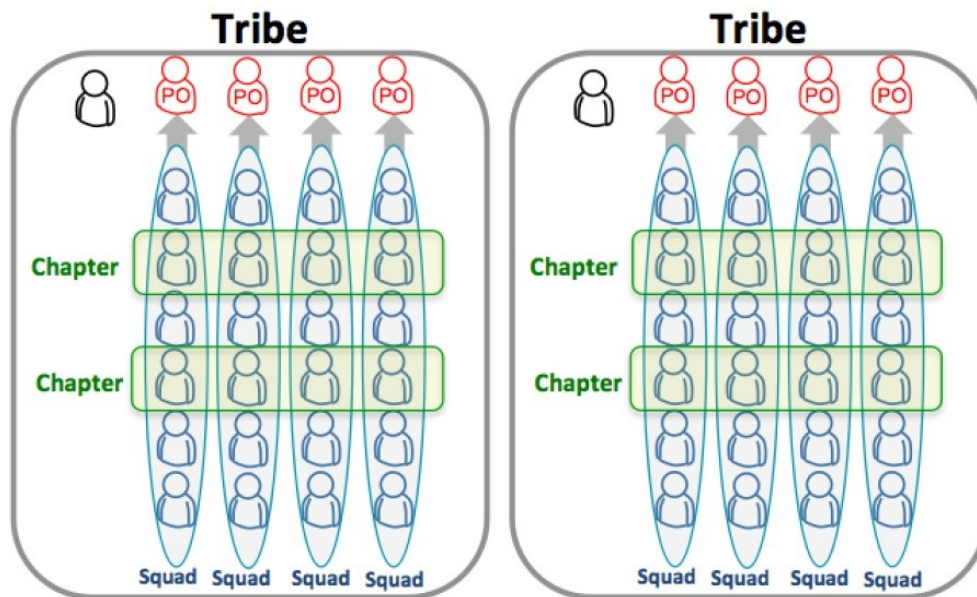
Il y a un inconvénient à chaque chose, et le potentiel inconvénient à la pleine autonomie est une perte d'économies d'échelle. Le testeur de la brigade A est peut-être en train de se battre avec un problème que le testeur de la brigade B a résolu la semaine dernière. Si tous les testeurs étaient ensemble, à travers les brigades et les tribus, ils pourraient partager leurs connaissances et créer des outils pour le bénéfice de toutes les brigades¹¹.

Si chaque brigade est totalement autonome et ne communique pas avec les autres brigades, alors quel est l'intérêt d'être dans une entreprise ? Spotify pourrait donc être découpé en 30 entreprises plus petites.

C'est pourquoi nous avons des Chapitres et des Guildes. C'est la colle qui relie tout le monde au sein de l'entreprise, qui nous fournit quelques économies d'échelle sans trop sacrifier d'autonomie.

Le chapitre est votre petite famille de personnes qui disposent de compétences similaires et qui travaillent dans le même domaine d'expertise au sein de la même tribu.

11 NdT : <https://agilarium.wikispaces.com/Comment+faire+Yokoten+%3F>

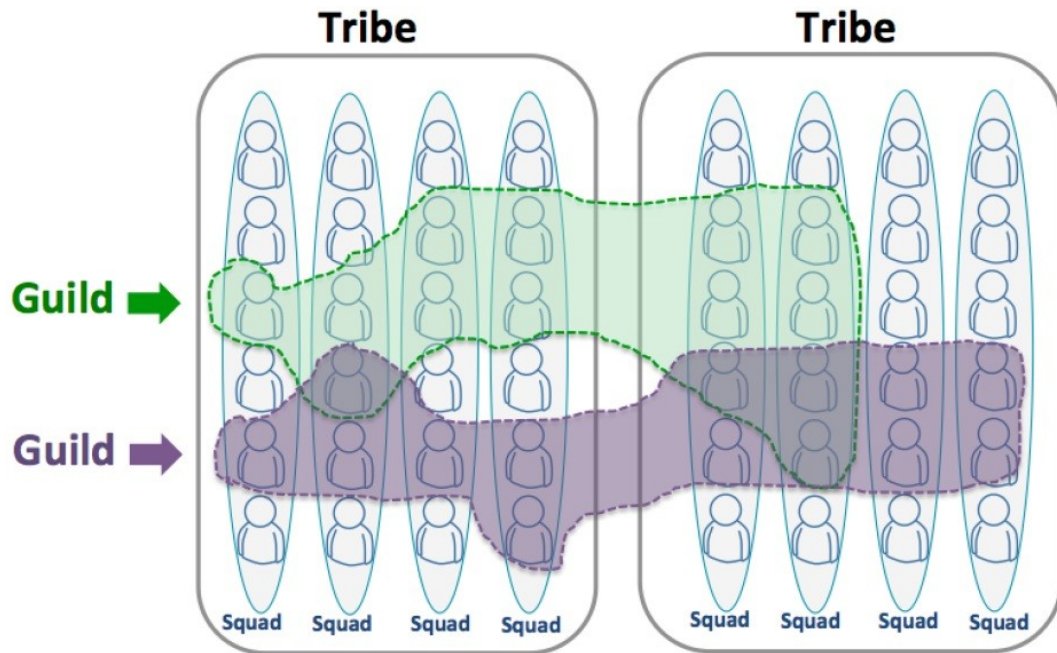


Chaque chapitre rencontre régulièrement les autres pour discuter de leur domaine d'expertise et de leurs défis spécifiques, par exemple le chapitre des tests, le chapitre des développeurs web ou le chapitre du backend.

Le leader du chapitre est le manager des membres du chapitre, avec toutes les responsabilités traditionnelles telles que le développement des personnes, la négociation des salaires, etc. Pourtant, le leader du chapitre fait aussi partie d'une brigade et est impliqué dans le travail de tous les jours, ce qui lui permet de rester au contact de la réalité.

Sauf que la réalité est toujours plus désordonnée que les jolis schémas ci-dessus. Par exemple, les membres du chapitre ne sont pas uniformément distribués entre les brigades ; certaines brigades ont beaucoup de développeurs web, d'autres non. Mais le schéma peut vous donner une idée générale.

Une Guilde est une « communauté d'intérêt » plus organique et très large, un groupe de personnes qui souhaitent partager leurs connaissances, leurs outils, leurs codes et leurs pratiques. Les chapitres sont toujours au sein d'une Tribu, alors qu'une guilde traverse généralement toute l'organisation. Quelques exemples : la guilde de la technologie web, la guilde des testeurs, la guilde des coachs agiles, etc.



Une guilde inclut souvent tous les chapitres travaillant sur un domaine et donc leurs membres, par exemple la guilde des tests inclut tous les testeurs de tous les chapitres de tests, mais toute personne intéressée peut rejoindre la guilde.

Chaque guilde a un « coordinateur de la guilde » qui, eh bien, fait juste ça :o)

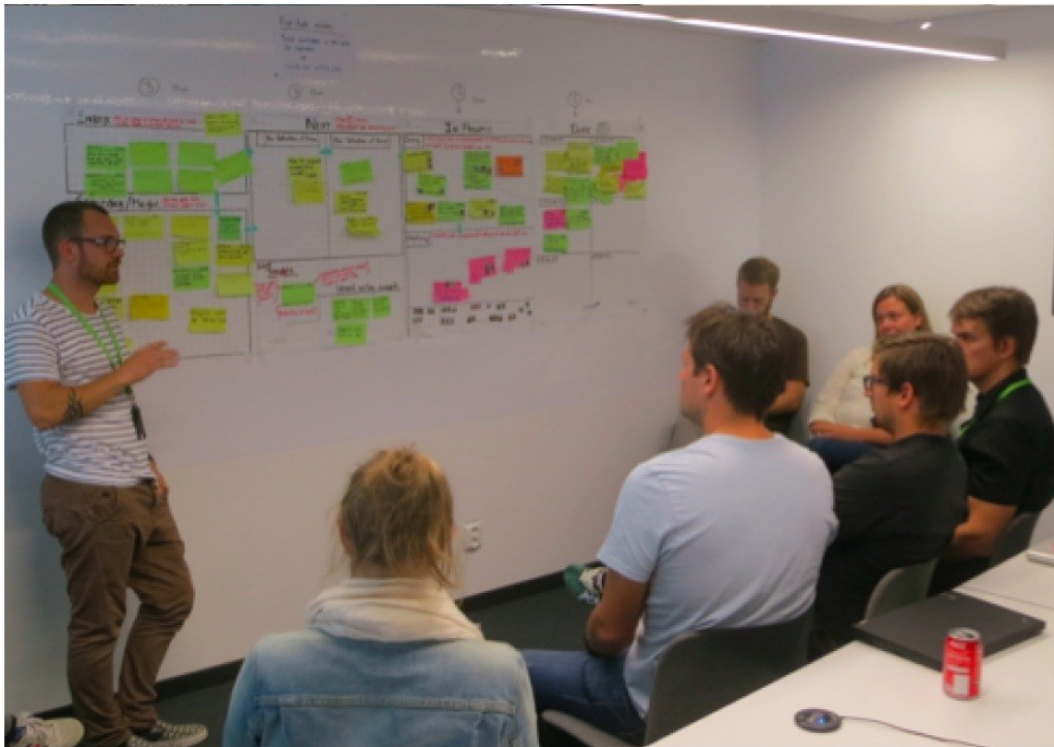
Pour donner un exemple du travail d'une guilde, nous avons récemment eu une « Unconference¹² de la Guilde Web », un événement sous forme de forum ouvert où tous les développeurs web de Spotify se sont rassemblés à Stockholm pour discuter des défis et solutions dans leur domaine.



Un autre exemple est la guilde des coachs agiles. Les coachs sont répartis dans toute l'organisation,

¹² NdT : <http://en.wikipedia.org/wiki/Unconference>

mais partage constamment leurs connaissances et se rencontrent régulièrement pour collaborer sur les grands domaines d'amélioration de l'organisation, ce que nous suivons sur un tableau d'amélioration.



Attendez une minute, on ne serait pas en train de parler d'une organisation matricielle ?

Oui. Eh bien, une sorte de... C'est un type différent de matrice que celle que nous avons l'habitude de rencontrer.

Dans beaucoup de matrices organisationnelles, les gens de compétences similaires sont « parqués » ensemble dans des départements fonctionnels et « affectés » à des projets, et font leur reporting à un manager fonctionnel.

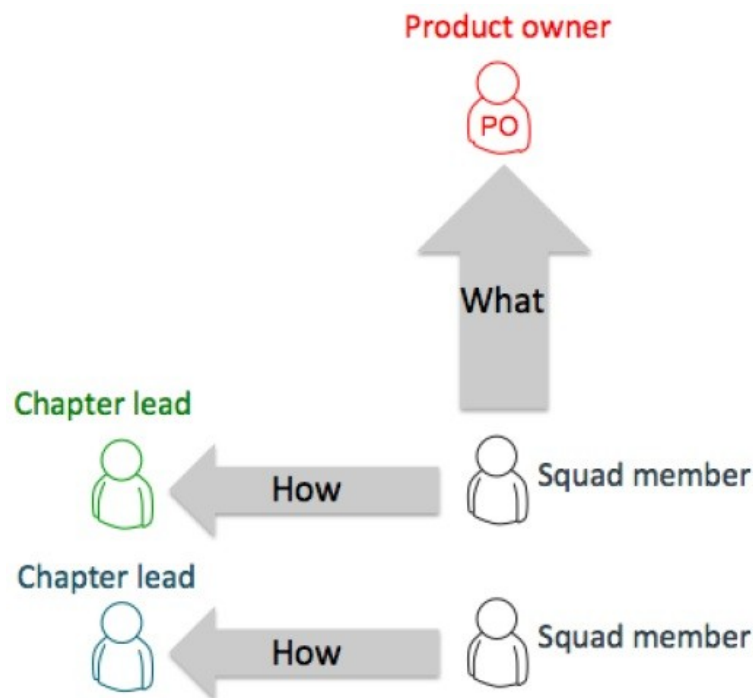
Spotify fait très rarement ce genre de choses. Notre matrice est orientée livraison.

C'est-à-dire que les personnes sont regroupées dans des brigades stables et colocalisées, où les personnes de compétences différentes collaborent et s'auto-organisent pour livrer un super produit. C'est la dimension verticale de la matrice, et c'est la principale puisque ce sont de cette manière que les personnes sont regroupées et c'est là où elles passent la plupart de leur temps.

La dimension horizontale est là pour partager la connaissance, les outils et le code. Le travail du leader du chapitre est de faciliter et soutenir cela.

En utilisant des termes de matrice, pensez à la dimension verticale comme le « quoi » et à la

dimension horizontale comme le « comment ». La structure de la matrice garantit que chaque membre de la brigade peut être aussi bien guidé pour savoir « quoi construire prochainement » que « comment bien le construire ».



Cela correspond bien au modèle du « professeur et de l'entrepreneur » recommandé par Mary et Tom Poppendieck. Le PO est « l'entrepreneur » ou « champion du produit », qui se concentre sur la livraison d'un super produit, tandis que le leader du chapitre est le « professeur » ou le « leader des compétences », se concentrant sur l'excellence technique.

Il y a une tension très saine entre ces rôles, puisque l'entrepreneur essaye d'accélérer en prenant des raccourcis, tandis que le professeur essaye de ralentir et de construire les choses proprement. Les deux aspects sont nécessaires, c'est pourquoi l'on parle de tension « saine ».

Et l'architecture dans tout ça ?

La technologie Spotify est très orientée service. Nous avons plus de 100 systèmes différents, et chacun peut être maintenu et déployé séparément. Cela inclut des services backend tels que des gestionnaires de listes de lecture, la recherche de chansons ou le paiement, et des clients tels que le lecteur iPad, ainsi que des composants spécifiques comme la radio ou la section « quoi de neuf ? » sur le lecteur de musique.



Techniquement, tout le monde peut accéder à n'importe quel système. Puisque les brigades sont effectivement des équipes features¹³, elles ont normalement besoin de mettre à jour plusieurs systèmes pour mettre une nouvelle feature en production.

Le risque de ce modèle est que l'architecture d'un système soit compromise si personne ne s'intéresse à la cohérence globale du système.

Pour pallier ce risque, nous avons un rôle qui s'appelle le « System Owner ». Tous les systèmes ont un system owner, ou un binôme de system owners (nous encourageons le binômage). Pour les systèmes opérationnels critiques, le System Owner est un binôme Dev-Ops, c'est-à-dire une personne ayant la perspective développeur et une autre personne ayant la perspective opérations.

Le system owner est la personne qu'il faut « aller voir » pour les problèmes techniques ou d'architecture concernant le système. C'est un coordinateur qui guide les personnes qui codent dans le système et qui leur garantit qu'elles ne vont pas se marcher dessus. Il se concentre sur des choses comme la qualité, la documentation, la dette technique, la stabilité, la scalabilité et le processus de livraison.

Le System Owner n'est pas un goulet d'étranglement ou un architecte dans une tour d'ivoire. Il ne doit personnellement pas prendre toutes les décisions, ou écrire tout le code, ou faire toutes les livraisons. C'est typiquement un membre d'une brigade ou un leader de chapitre qui a d'autres responsabilités quotidiennes en plus de celles de system owner. Pour autant, il prendra de temps en temps une journée « system owner » pour faire le ménage sur le système. Normalement, nous essayons d'avoir une charge de system owner représentant 10% d'un équivalent temps plein, mais cela varie beaucoup entre les systèmes bien entendu.

Nous avons également un rôle d'architecte en chef, une personne qui coordonne les problèmes d'architecture à un haut niveau tous systèmes confondus. Il mène des revues sur les développements de nouveaux systèmes pour se prémunir des erreurs classiques, et que tout le monde soit bien aligné sur notre vision de l'architecture. Le feedback consiste toujours en de simples suggestions, la décision de la conception finale du système relevant toujours de la brigade qui le construit.

13 NdT : <https://agilarium.wikispaces.com/Equipe+Feature>

Comment est-ce que ça marche ?

Spotify a connu une croissance très rapide, en 3 ans nous sommes passés de 30 à 250 personnes dans le domaine technique, nous avons donc eu notre lot de souffrances en terme de croissance ! Ce modèle à grande échelle avec les Brigades, les Tribus, les Chapitres et les Guildes, a été introduit petit à petit au cours de l'année dernière, les personnes sont donc encore en train de s'y habituer. Mais jusqu'à maintenant, en se basant sur les revues et rétrospectives, le modèle à grande échelle semble parfaitement fonctionner ! Et il nous donne le moyen de nous « transformer ». En dépit de cette croissance rapide, la satisfaction des employés a continuellement augmentée ; en Avril 2012, elle était de 4,4 sur 5.

Pourtant, comme dans toute organisation en pleine croissance, les solutions d'aujourd'hui donnent naissance aux problèmes de demain. Donc, restez à l'écoute, l'histoire n'est pas terminée :o)

Henrik & Anders

henrik.kniberg@spotify.com www.crisp.se/henrik.kniberg
anders.ivarsson@spotify.com