

LOG680

Introduction à l'approche DevOps

Par où commencer?

Comment concevoir notre organisation et notre architecture en
tenant compte de la loi de Conway

The DevOps Handbook

Part II, Chap. 7



Francis Bordeleau, 2021

Objectifs d'apprentissage

- Expliquer ce qu'est la loi de Conway.
- En vous basant sur la loi de Conway, expliquer en quoi la manière dont nous organisons nos équipes a un effet direct sur les logiciels que nous produisons, ainsi que sur nos résultats d'architecture et de production. Donner un exemple concret.
- Définir les trois archétypes organisationnels.
- Dans un contexte DevOps, expliquer comment on peut créer des organisations efficaces en utilisant une orientation fonctionnelle.
- Discuter des avantages d'utiliser des membres d'équipe généralistes, plutôt que spécialistes.
- Expliquer quels sont les principaux effets de l'utilisation de petites équipes (2PT).

Sujets

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

- **Introduction**
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

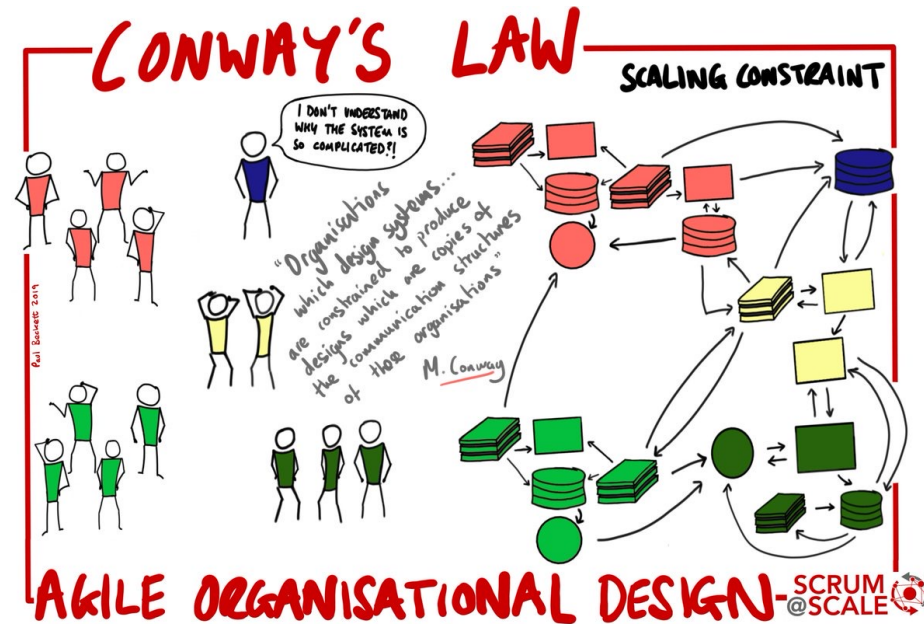
Comment organiser notre équipe?



Conway's Law

“Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations”

Conway, Melvin E. (1968)



Loi de Conway

- **La manière dont nous organisons nos équipes a un effet direct sur les logiciels que nous produisons, ainsi que sur nos résultats d'architecture et de production.**
 - **Pour obtenir un flux de travail rapide du développement aux opérations,** avec une qualité élevée et d'excellents résultats clients, nous devons **organiser nos équipes et notre travail de manière à ce que la loi de Conway Law joue à notre avantage.**
 - **Si elle est mal utilisée, la loi de Conway empêchera les équipes de travailler en toute sécurité et de manière indépendante**
 - Les équipes seront étroitement couplés les uns aux autres, s'attendant les uns les autres pour que le travail soit effectué, même de petits changements entraînant des conséquences catastrophiques et potentiellement globales.

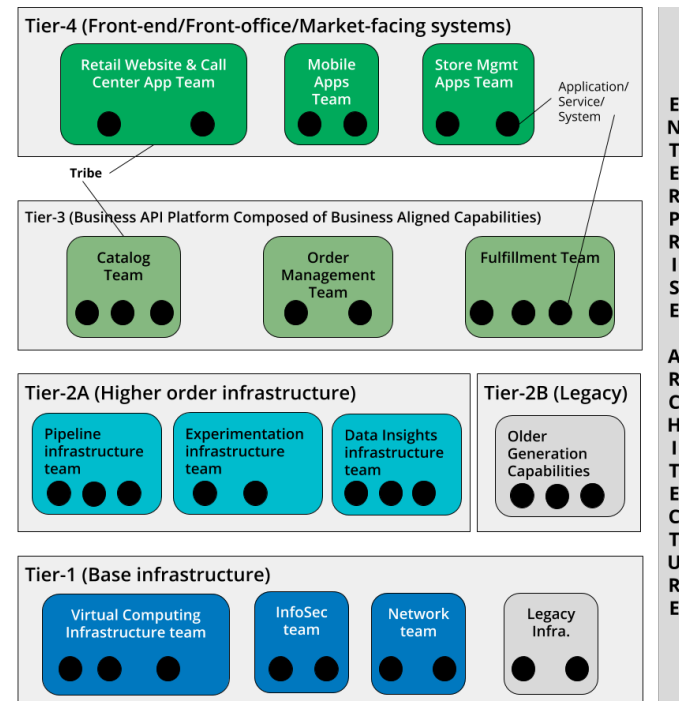
- Introduction
- **Archétypes organisationnels**
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Archétypes organisationnels

- 3 types principaux de structures organisationnelles
 - Les **organisations à vocation fonctionnelle**
 - Les **organisations orientées-marché ("market-oriented")**
 - Les **organisations matricielles**
- La structure organisationnelle détermine la manière dont nous concevons nos flux de valeur DevOps en tenant compte de la loi de Conway

Organisation à vocation fonctionnelle

- But: **optimiser l'expertise, la division du travail ou les coûts.**
- Ces organisations **centralisent l'expertise**
 - Contribue à la croissance de carrière et au développement des compétences
 - Disposent souvent de structures organisationnelles hiérarchiques élevées
- **Méthode d'organisation la plus utilisée pour les opérations**
 - Les administrateurs de serveur, les administrateurs de réseau, les administrateurs de base de données, etc., sont tous organisés en groupes distincts



Organisation orientée-marché

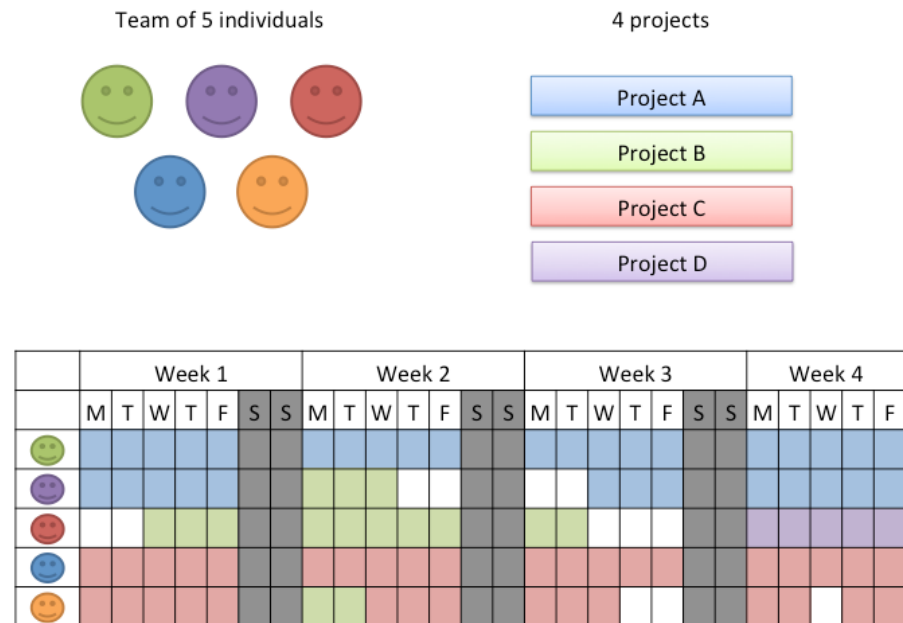
- But: **optimiser la capacité à répondre rapidement aux besoins des clients.**
- Organisations **horizontales** (ou planes) composées d'équipes **interdisciplinaires et inter-fonctionnelles**
 - Par exemple, marketing, ingénierie, etc.
- **Peut entraîner des redondances** dans l'ensemble de l'organisation
- Modèle utilisé par de nombreuses organisations qui adoptent DevOps
- E.g. Amazon, Netflix – Chaque équipe de service est simultanément responsable de la fourniture des fonctionnalités et du support technique.



<https://marketbusinessnews.com/financial-glossary/market-orientation-definition-meaning/>

Organisation matricielle

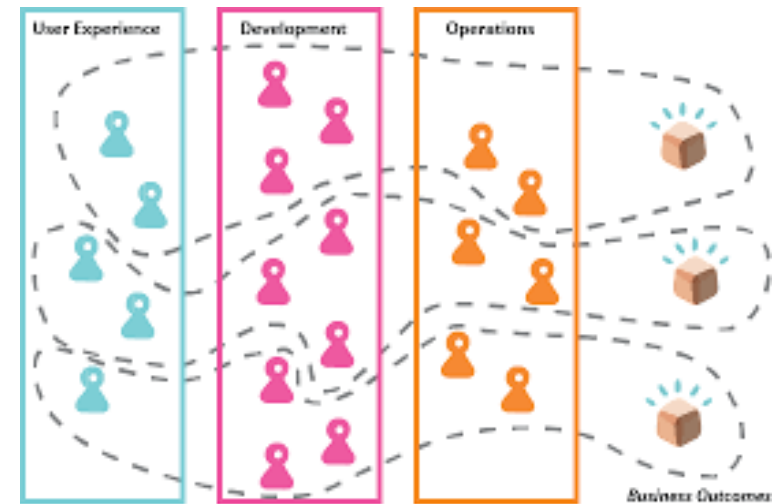
- But: **combiner une orientation fonctionnelle et une orientation marché.**
- Les organisations matricielles **aboutissent souvent à des structures organisationnelles compliquées**
 - Exemple, contributeurs individuels relevant de deux gestionnaires (ou plus), et ne réalisant parfois aucun des objectifs d'orientation fonctionnelle ou de marché.



- Introduction
- Archétypes organisationnels
- **Orientation excessivement fonctionnelle**
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Orientation excessivement fonctionnelle

- **But: optimiser les coûts**
- Organisations IT traditionnelles
 - Organisation des équipes en fonction de leurs spécialités
 - E.g. administrateurs de base de données dans un groupe, administrateurs réseau dans un autre, administrateurs de serveur dans un troisième, etc.
- Problèmes causés par une orientation excessivement fonctionnelle
 - **Longues files d'attente et longs délais d'exécution**
 - **Manque de visibilité sur les objectifs globaux**
 - Transferts médiocres, nombreuses reprises de travail, problèmes de qualité, goulots d'étranglement et retards.
 - Empêche la réalisation d'objectifs organisationnels importants, qui dépassent souvent de loin le désir de réduire les coûts.
- Problème **exacerbé lorsque chaque domaine fonctionnel d'opérations doit servir plusieurs flux de valeur**
 - E.g. plusieurs équipes de développement nécessitant les mêmes services/ressources



- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- **Équipe orientée-marché**
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Équipe orientée-marché

- **But: optimiser pour la vitesse**
 - Pour obtenir les résultats de DevOps, il faut réduire les effets de l'orientation fonctionnelle et permettre l'orientation vers le marché
- Permettre à des petites équipes de travailler de manière indépendante et en toute sécurité, générant rapidement de la valeur pour les clients.
 - Équipes multifonctionnelles et indépendantes
 - Équipes responsables non seulement du développement des fonctionnalités, mais aussi des tests, de la sécurisation, du déploiement et de l'opération des services en production, de la conception de l'idée jusqu'au retrait des fonctionnalités.
 - Permet de concevoir et exécuter des expériences utilisateur, créer et fournir de nouvelles fonctionnalités, déployer et exécuter les services en production et résoudre les problèmes de manière indépendante.
- Modèle adopté par Amazon et Netflix
 - Présenté par Amazon comme l'une des principales raisons de leur capacité à agir rapidement, même en cas de croissance.



Équipe orientée-marché

- Pour mettre en place une orientation-marché, il est **déconseillé de procéder à une vaste réorganisation descendante ("top-down")**.
- Au lieu de cela, nous pouvons
 - **Intégrer les ingénieurs fonctionnels et les compétences (e.g. Ops, QA, Infosec) dans chaque équipe de service**, ou
 - **Fournir leurs capacités via des plates-formes automatisées en libre-service** offrant des environnements de type production, de tests automatisés ou de déploiement.
- **Permet à chaque équipe de service de fournir de manière indépendante de la valeur au client** sans avoir à ouvrir de tickets avec d'autres groupes, tels que les opérations, le contrôle qualité ou Infosec.

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- **Orientation fonctionnelle en DevOps**
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Orientation fonctionnelle en DevOps

- Possible d'obtenir de hautes performances avec une orientation fonctionnelle et un groupe d'Ops centralisé
 - À condition que les équipes de service obtiennent ce dont elles ont besoin d'Ops de manière fiable et rapide (idéalement à la demande) et vice-versa.
 - La plupart des organisations DevOps les plus reconnues ont une orientation fonctionnelle des opérations, e.g. Etsy, Google et GitHub.
- **Culture de grande confiance**
 - Permet à tous les départements de travailler efficacement
 - **Tout le travail est hiérarchisé de manière transparente**
 - Suffisamment de ressources disponibles pour permettre l'achèvement rapide des tâches prioritaires.
- **Rendu possible par les plateformes de libre-service automatisées**
 - Intègrent la qualité dans les produits que tout le monde construit.

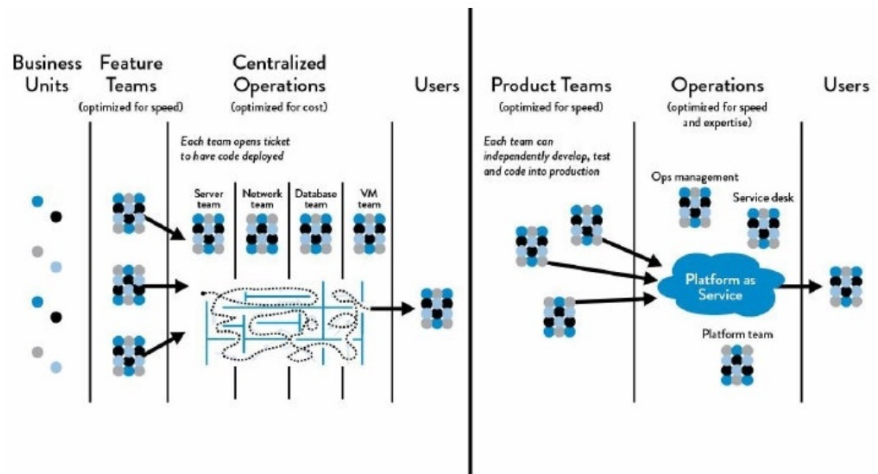


Figure 12: Functional vs. market orientation

Left: Functional orientation: all work flows through centralized IT Operations; Right: Market orientation: all product teams can deploy their loosely-coupled components self-service into production. (Source: Humble, Molesky, and O'Reilly, Lean Enterprise, Kindle edition, 4523 & 4592.)

Orientation fonctionnelle en DevOps

- Dans le mouvement de la fabrication "Lean" des années 1980, de nombreux chercheurs étaient déconcertés par **l'orientation fonctionnelle de Toyota**
 - Allait à l'encontre des meilleures pratiques consistant à avoir des équipes inter-fonctionnelles et orientées marché.
 - «le **deuxième paradoxe de Toyota**»
- Mike Rother (Toyota Kata):

«Aussi tentant que cela puisse paraître, vous ne pouvez pas réorganiser votre chemin vers l'amélioration continue et l'adaptabilité.

Ce qui est décisif n'est pas la forme de l'organisation, mais la façon dont les gens agissent et réagissent.

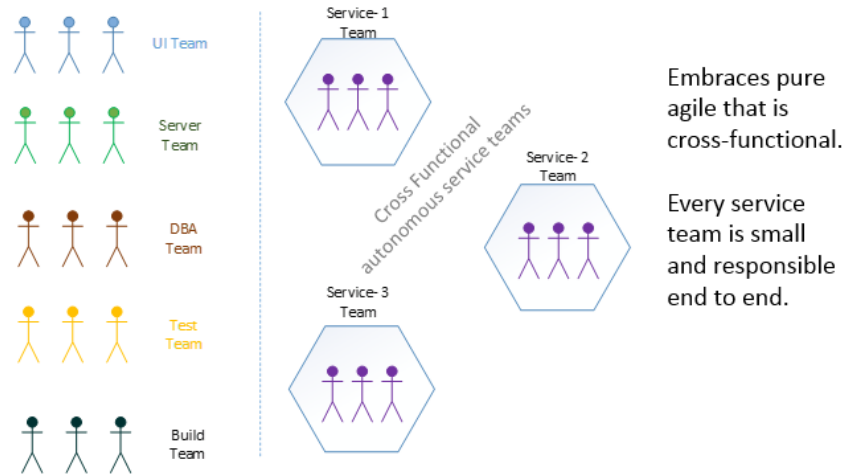
Le succès de Toyota ne réside pas dans ses structures organisationnelles, mais dans le développement des capacités et des habitudes de ses employés.

En fait, cela surprend beaucoup de gens de constater que Toyota est en grande partie organisée dans un style traditionnel de département fonctionnel. »

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- **Ingénieurs multifonctionnels**
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Ingénieurs multifonctionnels

- But: intégrer le test, les opérations, et la sécurité dans le travail de tous les membres de l'équipes au quotidien.
- Dans les **organisations très performantes, tous les membres de l'équipe partagent un objectif commun.**



- Exemple Facebook
 - En 2009, Facebook connaissait une croissance énorme ... et étaient confrontés à des problèmes importants liés au déploiement de codes
 - Une des choses les plus importantes qu'ils aient faites pour changer les résultats des déploiements a été de faire en sorte que **tous les ingénieurs, directeurs techniques et architectes de Facebook effectuent une rotation sur appel pour les services qu'ils ont créés.**
 - **Tous les membres de l'équipe ont bénéficié d'un retour d'information viscéral sur les décisions d'architecture et de codage prises en amont,** ce qui a eu un impact positif énorme sur les résultats en aval.

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- **Membres d'équipe généralistes**
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Membres d'équipe généralistes

- Dans les cas extrêmes d'**organisation fonctionnelle d'équipes d'opérations**, nous avons des **départements de spécialistes**
 - E.g. administrateurs réseau, administrateurs de stockage, etc.
 - Mène à la **silosation**
 - Multiples transferts et files d'attente entre les différents domaines de l'infrastructure
 - Entraîne des délais plus longs
- Puisque nous nous appuyons sur un **nombre croissant de technologies**, nous devons disposer d'**ingénieurs spécialisés et maîtrisant les domaines technologiques** dont nous avons besoin.
- **Nous ne voulons pas créer des spécialistes «figés dans le temps»**, seulement capables de comprendre et de contribuer à des domaines spécifiques de la chaîne de valeur.
- **Contre-mesure: permettre et encourager chaque membre de l'équipe à être un généraliste.**
 - E.g. Ingénieur "full-stack"



Membres d'équipe généralistes

Table 2: *Specialists vs. Generalists vs. "E-shaped" Staff (experience, expertise, exploration, and execution)*

"I-shaped" (Specialists)	"T-shaped" (Generalists)	"E-shaped"
Deep expertise in one area	Deep expertise in one area	Deep expertise in a few areas
Very few skills or experience in other areas	Broad skills across many areas	Experience across many areas Proven execution skills Always innovating
Creates bottlenecks quickly	Can step up to remove bottlenecks	Almost limitless potential
Insensitive to downstream waste and impact	Sensitive to downstream waste and impact	
Prevents planning flexibility or absorption of variability	Helps make planning flexible and absorbs variability	

(Source: Scott Prugh, "Continuous Delivery," ScaledAgileFramework.com, February 14, 2013, <http://scaledagileframework.com/continuous-delivery/>.)

Membres d'équipe généralistes

- **Approche en contradiction avec les pratiques d'embauche traditionnelles**
 - Souvent considéré comme plus onéreux
 - "je peux embaucher deux administrateurs de serveur pour chaque ingénieur d'exploitation polyvalent. "
 - Cependant, les avantages commerciaux de permettre un flux plus rapide sont énormes
- **Il faut encourager l'apprentissage, aider les gens à surmonter l'anxiété liée à l'apprentissage, faire en sorte que les gens possèdent les compétences requises et une feuille de route définie pour leur carrière, etc**
 - Aider nos ingénieurs à développer l'esprit de croissance.
 - **Une organisation apprenante a besoin de personnes prêtes à apprendre.**
- Jason Cox, directeur de l'ingénierie des systèmes chez Disney
 - «Dans les équipes d'opérations, nous avons dû modifier nos pratiques d'embauche. **Nous recherchions des personnes qui avaient «curiosité, courage et franchise»** et qui étaient non seulement capables d'être des généralistes, mais aussi des renégats... Nous voulons promouvoir des bouleversements positifs afin que notre entreprise ne reste pas bloquée et puisse aller de l'avant.

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- **Mettre l'accent sur les services et les produits**
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Mettre l'accent sur les services et les produits

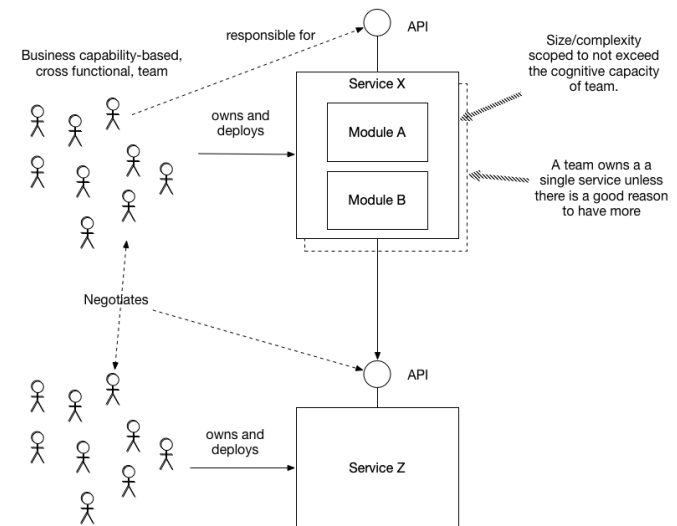
- **But: financer le développement de services/produits, plutôt que des projets**
 - Créer des équipes stables dotées d'un financement permanent pour mettre en œuvre leur propre stratégie et leur feuille de route.
 - Ces équipes disposent des ingénieurs dédiés nécessaires pour respecter les engagements concrets pris envers les clients internes et externes, tels que les fonctionnalités, les récits et les tâches.
- Modèle de financement par projet
 - Les équipes de développement et de test sont affectées à un «projet», puis réaffectées à un autre projet dès que le projet est terminé et que le financement est épuisé.
 - **Impossibilité pour les développeurs de voir les conséquences à long terme des décisions qu'ils prennent**
- **Objectif:** valoriser la réalisation des **résultats organisationnels et clients**, e.g.
 - **Revenus**
 - **Valeur client sur la durée de vie du produit**
 - **Taux d'adoption client**, idéalement avec un minimum de version
 - Plutôt que, par exemple, si le projet a été achevé dans les limites de budget et de temps.

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- **Limites des équipes conformément à la loi de Conway**
- Architecture à faible couplage
 - Two-pizza-team
- Conclusion

Limites des équipes conformément à la loi de Conway

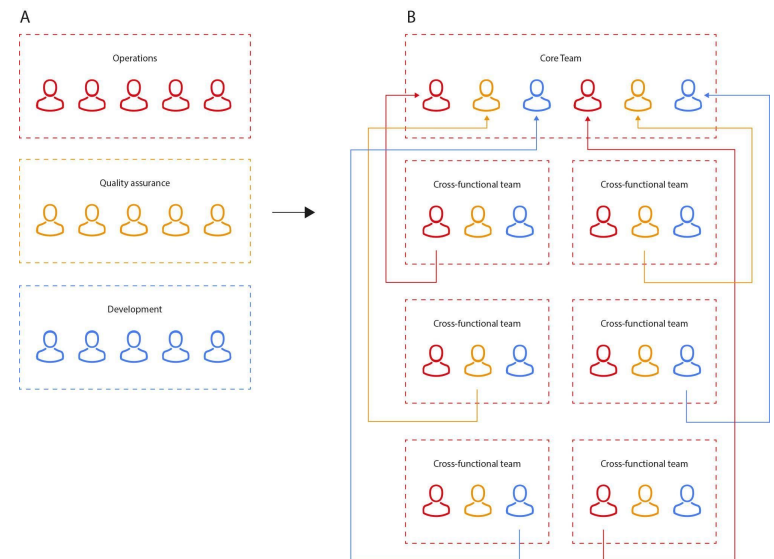
- **Maintenir une communication et une coordination efficaces** entre les personnes et les équipes **constitue l'un des plus grands défis.**

- Trop souvent, lorsque des personnes et des équipes résident sur un étage, un bâtiment ou un fuseau horaire différent, il devient plus difficile de créer et de maintenir une compréhension partagée et une confiance mutuelle, ce qui empêche une collaboration efficace.
- La collaboration est également entravée lorsque les principaux mécanismes de communication sont des fiches de travail et des demandes de modification, ou pire, lorsque des équipes sont séparées par des limites contractuelles, par exemple lorsque le travail est effectué par une équipe sous-traitée.



Limites des équipes conformément à la loi de Conway

- Effet secondaire de la loi de Conway: **la manière dont nous organisons les équipes peut engendrer de piètres résultats.**
- Exemple, division des équipes par fonction
 - Placer les développeurs et les testeurs dans des emplacements différents, externaliser les tests, ou utiliser des équipes responsables d'une couche architecturale (e.g. équipe application, équipe de base de données).
 - **Nécessitent une communication et une coordination importantes entre les équipes**
 - **Entraînent un nombre important de "reworks", des désaccords sur les spécifications, des transferts médiocres et des personnes inactives attendant quelqu'un d'autre.**



- Notre **architecture logicielle devrait permettre aux petites équipes d'être indépendamment productives et suffisamment dissociées** les unes des autres **pour que le travail puisse être effectué sans communication et coordination excessives ou inutiles.**

- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- **Architecture à faible couplage**
 - **Two-pizza-team**
- Conclusion

Architecture à faible couplage

- **But: créer des architectures à faible couplage pour améliorer la productivité des développeurs**
- **Architectures orientées services (SOA)**
 - Décrites pour la première fois dans les années 90
 - Architectures composées de **services faiblement couplés avec des contextes bien définis**
 - Les services **interagissent avec leurs pairs strictement via des API** et ne partagent donc pas de structures de données, de schémas de base de données ni d'autres représentations internes
 - **Permet de tester et déployer les services de manière indépendante**
 - Permet de mettre-à-jour les services de manière indépendante
- Randy Shoup, ancien directeur technique de Google App Engine

«Les organisations dotées de ce type d'architectures orientées services, telles que Google et Amazon, offrent une flexibilité et une évolutivité incroyables.
Ces organisations comptent des dizaines de milliers de développeurs où les petites équipes peuvent encore être incroyablement productives. »

Garder les équipes petites – 2PT

La **loi de Conway** nous aide à définir les limites de notre équipe en tenant compte des schémas de communication souhaités, mais elle nous encourage également à réduire la taille de notre équipe.

Cette limite de taille a quatre effets importants:

1. Garantit que l'équipe a une compréhension claire et partagée du système sur lequel elle travaille.
2. Limite le taux de croissance du produit ou du service sur lequel on travaille.
3. Décentralise le pouvoir et permet l'autonomie.
4. Diriger une 2PT est un moyen pour les employés d'acquérir une certaine expérience de leadership dans un environnement où l'échec n'a pas de conséquences catastrophiques.

THE POWER OF SMALL TEAMS

B.

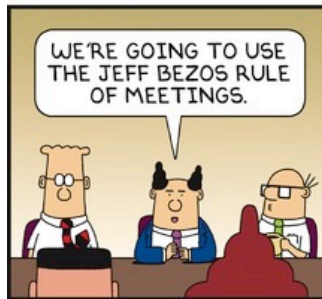
If a team can't be fed with two pizzas - it's too big

Jeff Bezos
Amazon

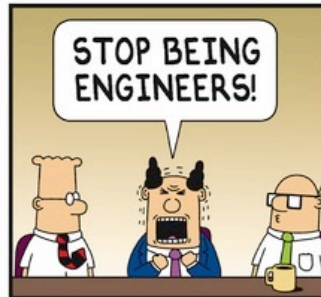
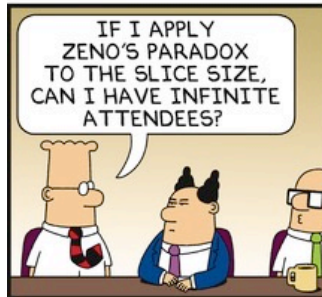


Garder les équipes petites – 2PT

DILBERT



BY SCOTT ADAMS



- Introduction
- Archétypes organisationnels
- Orientation excessivement fonctionnelle
- Équipe orientée-marché
- Orientation fonctionnelle en DevOps
- Ingénieurs multifonctionnels
- Membres d'équipe généralistes
- Mettre l'accent sur les services et les produits
- Limites des équipes conformément à la loi de Conway
- Architecture à faible couplage
 - Two-pizza-team
- **Conclusion**

Conclusion

- Plusieurs études de cas (e.g. Etsy et Target) démontrent comment l'architecture et la conception organisationnelle peuvent améliorer considérablement nos résultats.
- La loi de Conway peut faire en sorte que l'organisation crée de mauvais résultats, prévenant la sécurité et l'agilité.
- Si elle est bien réalisée, l'organisation permet aux développeurs de développer, tester et déployer de la valeur en toute sécurité et en toute indépendance pour le client.