



# Introduction à DevOps

Agile Tour Québec 2015



# Mathieu Benoit

---

- Consultant en architecture de solutions informatiques
  - MCSD Applications Web
  - PSM-I
- Directeur-adjoint du centre d'excellence Microsoft et Sitecore
- 8 ans d'expérience
- Privé et Public
- .NET mais aussi Java/Android
- Innovation et apprentissage continu

- 30 ans d'expérience à Québec et ailleurs
- Publicis Group - Razorfish



- Commanditaire de l'Agile Tour Québec
- Centres d'excellences
  - Agile, Microsoft, etc.

# À propos de vous

---

## ➤ Secteur

- Public ? Privé ?

## ➤ Profils

- Gestionnaires ?
- Développeurs ?
- Techno/Infra/Sécurité
- Affaires/Fonctionnels/Product Owner

## ➤ Fréquence de déploiements

- En Acceptation/QA ? en Production ?



# Sommaire

1 ➤ Introduction

---

5 ➤ Conclusion

2 ➤ Déploiement continu

---

6 ➤ Références

3 ➤ Livraison continue

---

7 ➤ Questions

4 ➤ Opération continue

---





# Introduction



# Dev et Ops

---

Je veux du  
changement !



**Développement**

**Mur des confusions**

Je veux de la  
stabilité !



**Opérations**

# DevOps

---

- Ce n'est pas une recette miracle
- Cela ne s'applique pas à chaque entreprise, chaque projet ou chaque personne
- C'est livrer souvent de la valeur ajoutée au client final
- C'est une évolution des pratiques existantes et connues
- C'est un apprentissage continu
- Ce n'est pas une méthodologie
- Ce n'est pas que « Dev et Ops »
- Objectifs de cette rencontre :
  - Introduction à DevOps
  - Aperçu des pratiques associées à DevOps
  - Pistes de réflexion sur des concepts DevOps



# Constats et considérations

---

- Avec un changement d'une seule ligne de code, dans combien de temps cela se retrouvera-t-il en Production ?
- « Ça fonctionne sur mon poste en local »
- « Ça a pourtant passé les tests d'Acceptation »
- « Nous n'avons pas fait graduer la demande XXX par les DBA »
- « La configuration en Production n'est pas la même qu'en Acceptation/QA »
- « Créer un environnement qui reflète la Production est trop compliqué et prend trop de temps »
- « Le document des étapes de graduation en Production n'était pas à jour »
- « On va devoir travailler le soir ou la fin de semaine car une interruption de service en pleine journée n'est pas possible »
- « La réelle valeur d'une fonctionnalité est lorsque l'on obtient des retours de la production »
- « Les entreprises font face à une plus grande vélocité au changement des besoins métiers qui apparaissent chaque mois, semaine ou jour (non plus par an) »

# Dev et Ops



## Culture

- Centrer sur les personnes
- Encourager le changement et l'expérimentation



## Automatisation

- Intégration et Livraison continus
- L'infrastructure comme du code



## Lean

- Produire de la valeur ajoutée finale
- Livraison de petite taille



## Métriques

- Tout mesurer
- Constater les améliorations



## Partage

- Partage d'informations
- Collaboration

# Mouvement DevOps

---

Waterfall

Agile

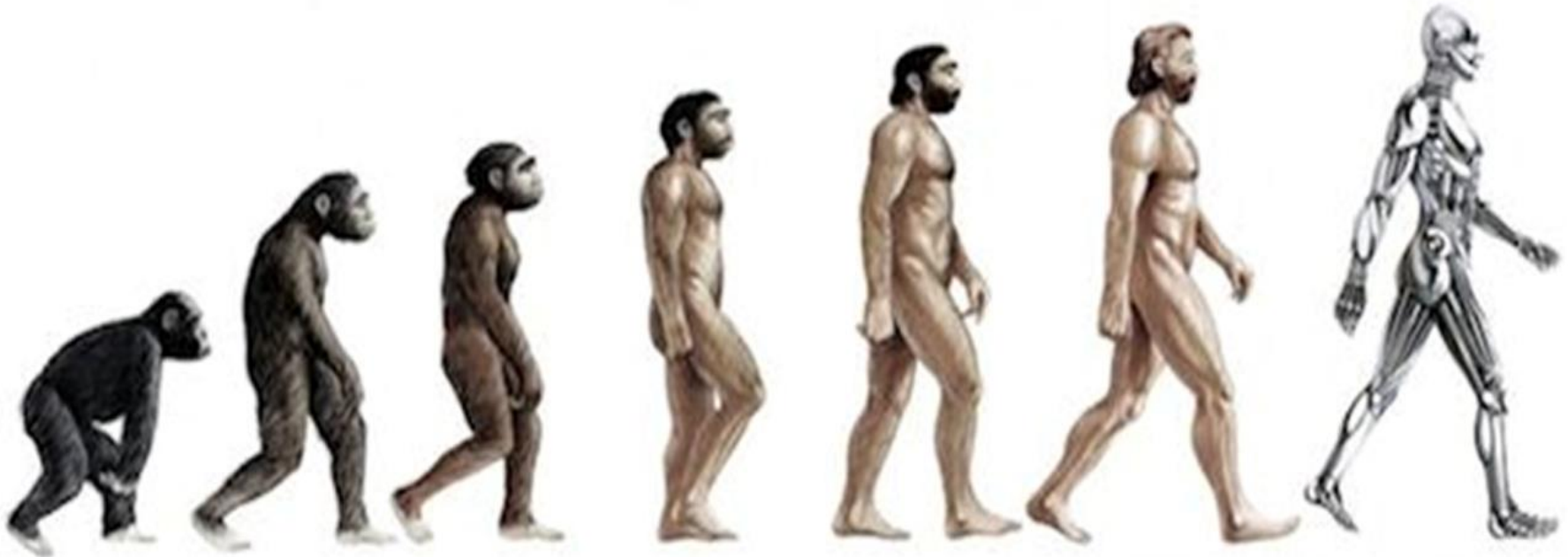
Lean

Intégration  
continue

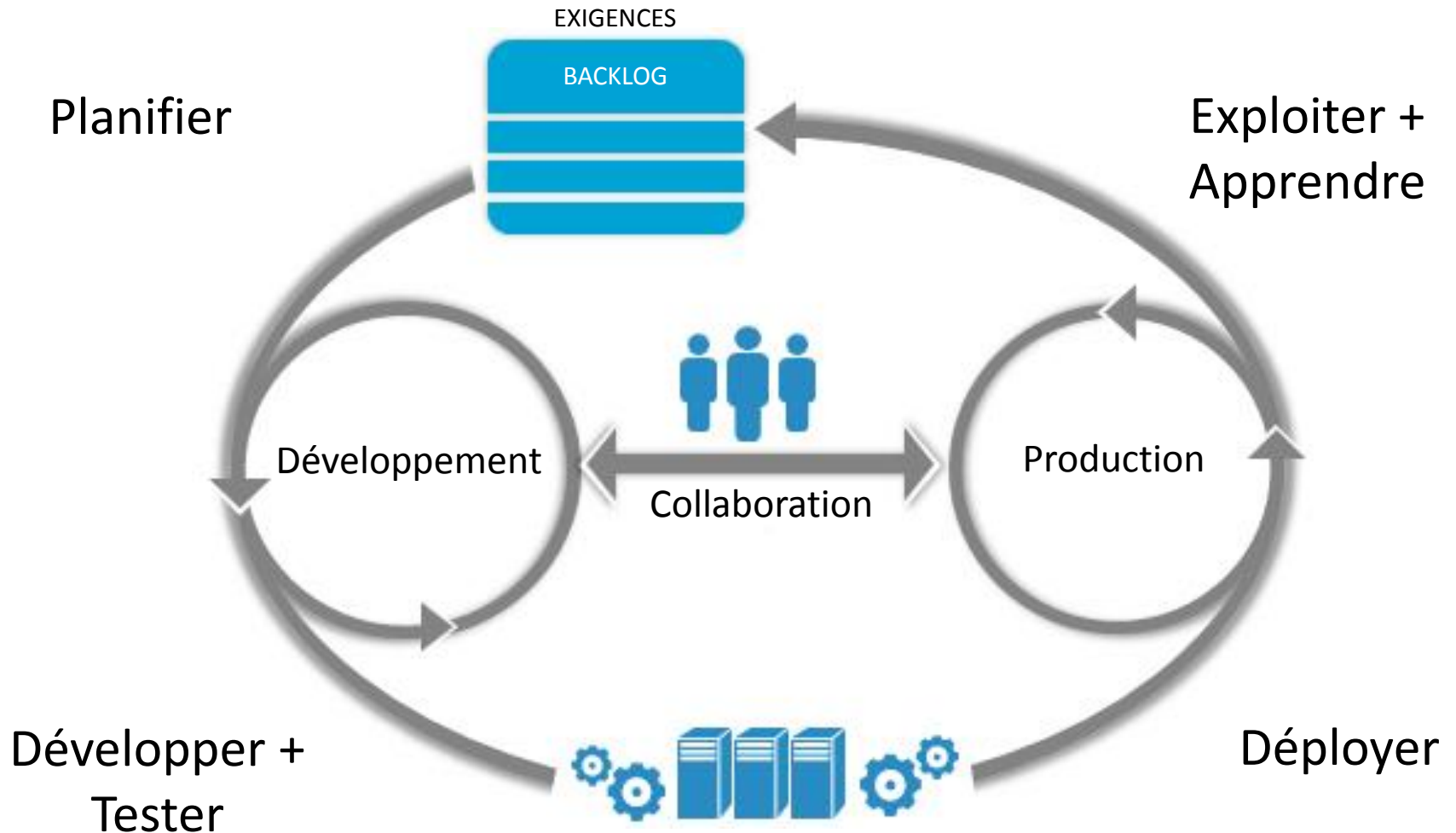
Déploiement  
continu

Livraison  
continue

Opération  
continue



# Gestion du cycle de vie - ALM



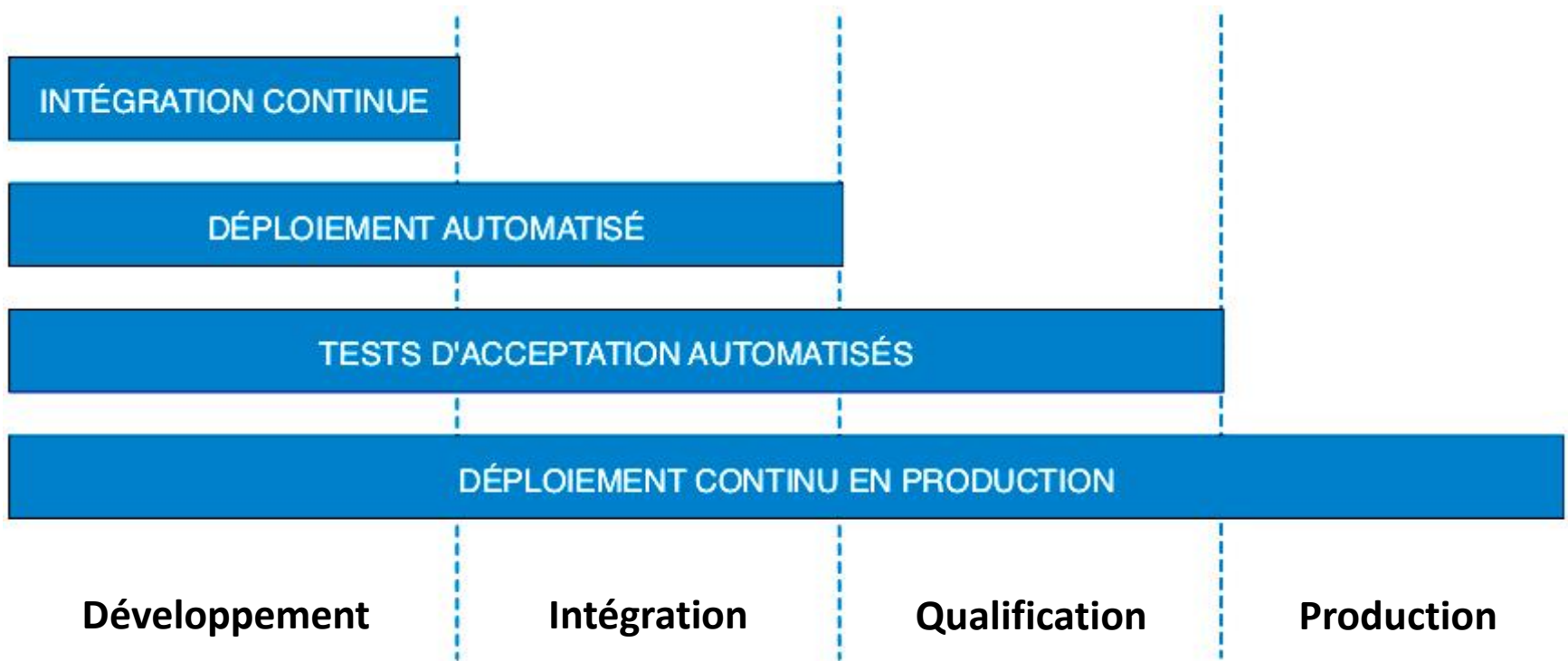


# Déploiement continu



# Déploiement continu

---



# Déploiement continu

---

## ➤ Prérequis : Intégration continue

- Serveur d'intégration continue et d'assemblage
- Règles et Tests d'intégration/unitaires

## ➤ Déploiement continu

- Supprimer les tâches manuelles et fastidieuses
- Fiabiliser le processus
- Augmenter la fréquence
- Intégrer les Tests d'acceptations automatisés
- Réduire le « Time-To-Market »

## ➤ Kata de déploiement

- Prendre une application et réaliser un déploiement, cela ne doit ni être compliqué ni être long



# Automatisation

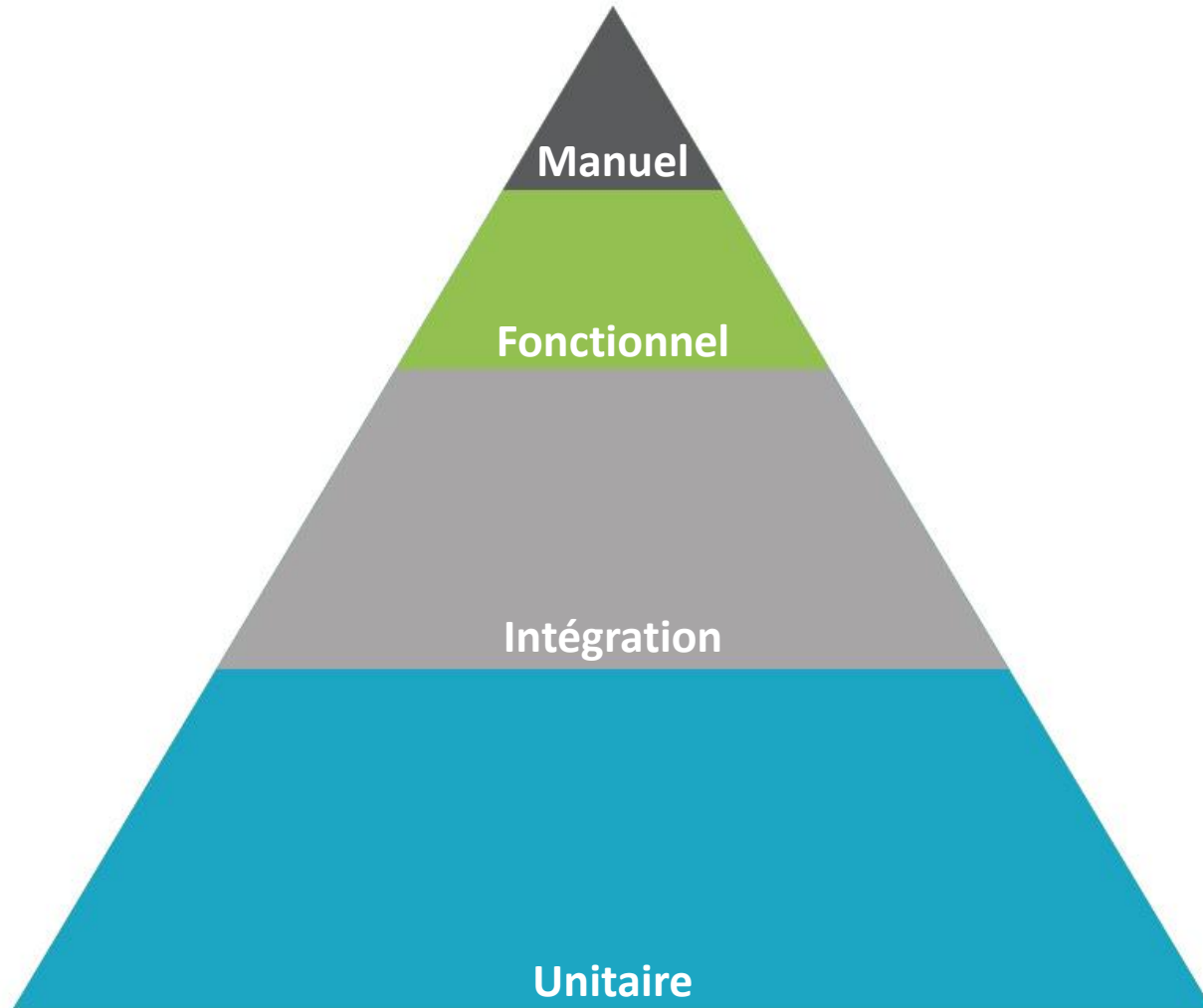
---

- Infrastructure et configuration comme du code
- Pas d'action manuelle avec des assistants visuels
- Scripts
  - Documentations toujours à jour
  - Encouragent le partage et la coopération
- Simplification
  - Utilisation de scripts prédéfinis
  - entraînement des équipes avec des activités ludiques
  - ajout dans la définition de « fini »
- L'automatisation rend les processus répétables, testables et rapides

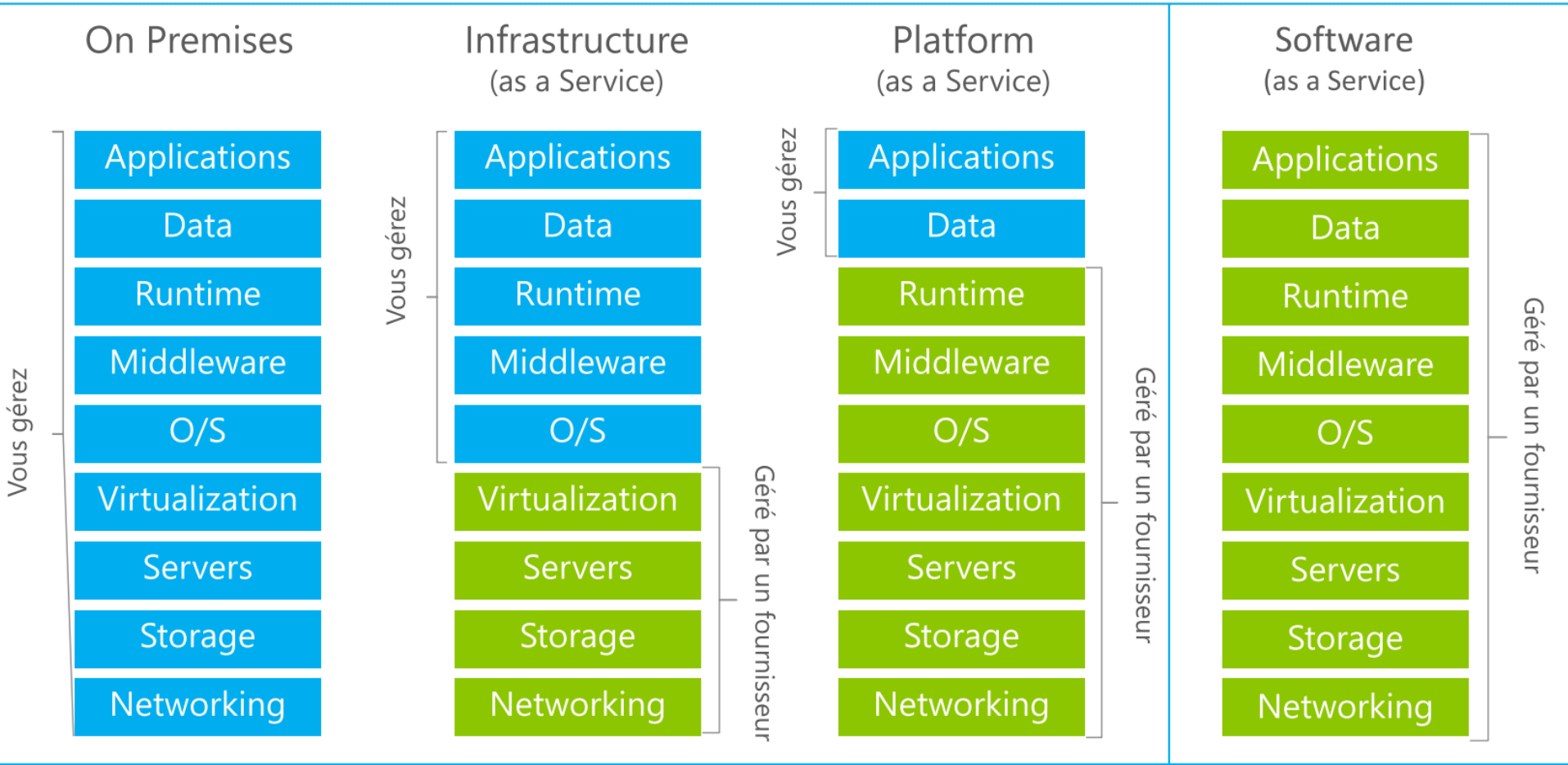


# Pyramide des Tests

---



# Dématérialisation





# Livraison continue



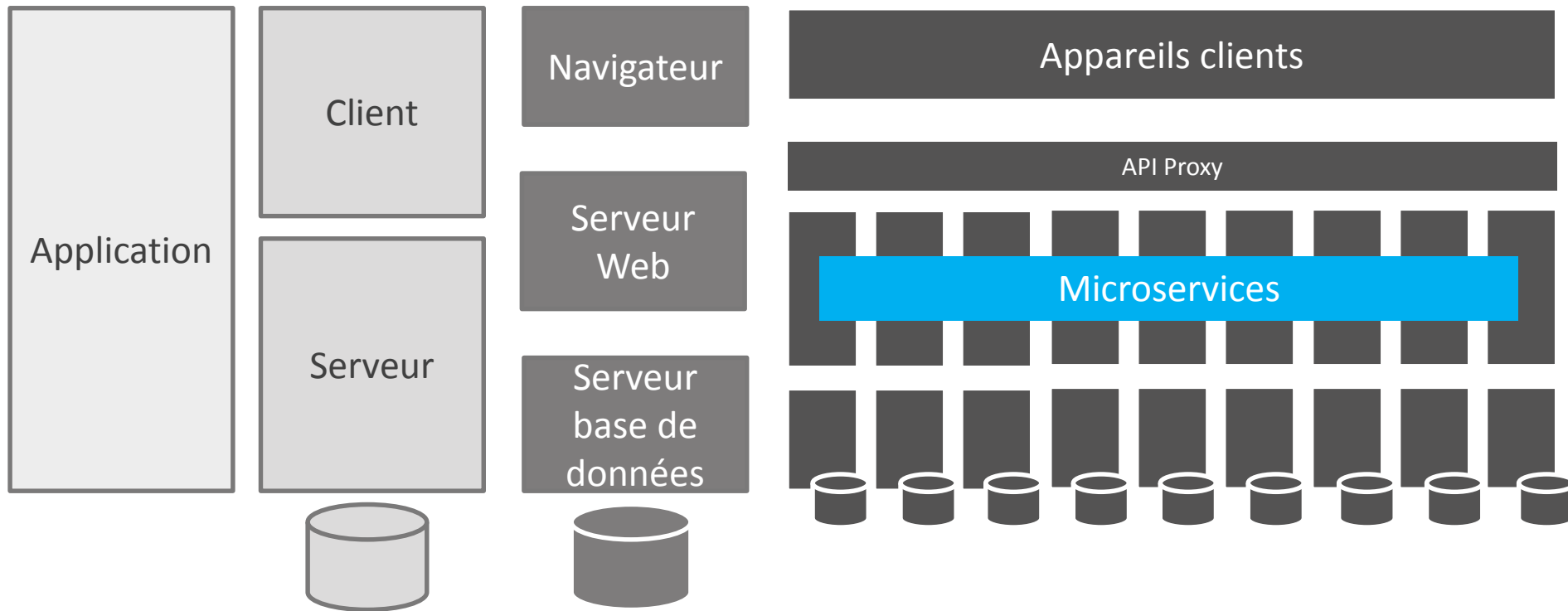
# Livraison continue en Production

---

- « La réelle valeur d'une fonctionnalité est lorsque l'on obtient des retours de la production »
- « Chaque client est un client impatient »
- Quelques approches :
  - « Staging » - Pré-Production
  - Activation de fonctionnalités
  - Tests A/B
  - Limitations des adresses IP
- Fréquence
  - Par an (avant)
  - Par mois, semaine ou jour (aujourd'hui)
- Google, Amazon, Netflix, Facebook, etc.
  - > 100 déploiements en Production par jour

# De Monolithique à Microservice

---



# Microservices

---

- Autonome
  - Non couplé aux autres services
- Isolé
  - Mises à jours indépendantes
- Résilient
  - Tolérant aux défaillances et hautement disponible
- Élastique
  - Échelonnable
- Orienté messages asynchrones
- Configurable
  - Accès à une console d'administration avec une infrastructure flexible
- Réactif
  - Répond dans un temp raisonnable
- Exemple
  - Musiciens dans un orchestre
  - Danseur dans un ballet
- Bénéfices
  - Évidents : déploiements de codes plus petits et mises à jour fréquentes
  - Non-évidents : encourage l'innovation et améliore la flexibilité

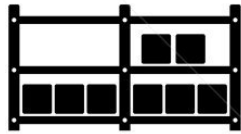
# Containers



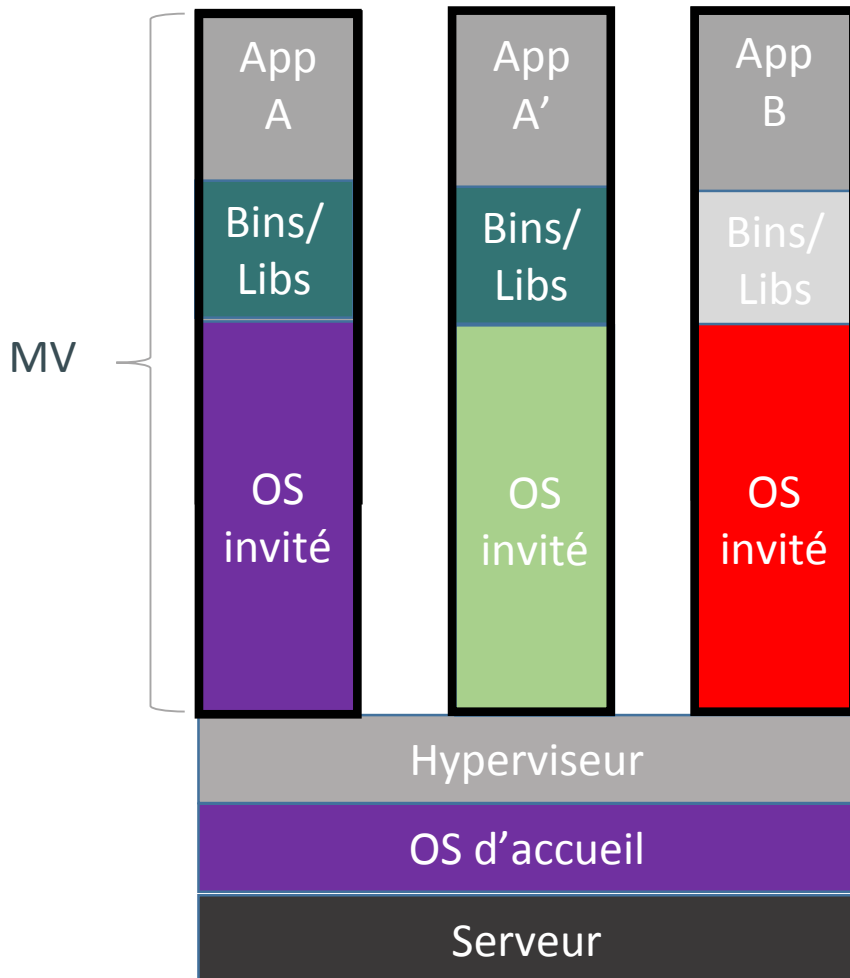
Un container standard est chargé avec des objets et reste scellé jusqu'à sa livraison ...



... entre temps il peut être chargé, déchargé, empilé et transporté efficacement ainsi que transféré d'un mode de transport à un autre.



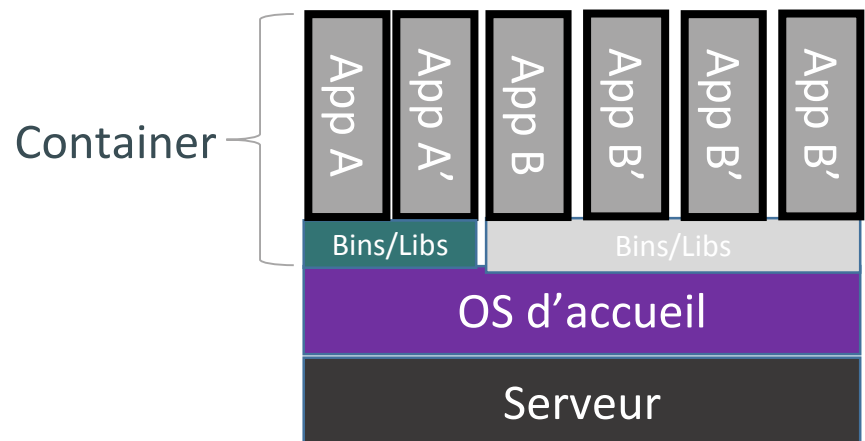
# Machines Virtuelles vs Containers



Containers isolés avec ressources partagées

Résultats significatifs :

- Déploiement plus rapide
- Simplification des mises à jour
- Redémarrage plus rapide







# Opération continue



# Retours rapides et constants

---

- Satisfaction du client
- Proactivité
- Retours instantanés
  - Construction rapide, retours visibles à toute l'équipe, enregistrer des statistiques
- Log d'erreurs
- Analytics d'usage
- Retours clients
  - UserVoice
  - Courriel/Centre d'appel
- Monitoring
  - La surveillance et l'exploitation doivent faire partie intégrante du cycle
  - Mettre en place des alertes et des points de vérifications
  - Collecter des métriques et indicateurs

# Métriques

---

## ➤ Indicateurs

- Fréquence de déploiement
- Disponibilité
- Performance
- Utilisation

## ➤ MTTD (Mean Time To Diagnose)

- Le temps moyen que cela prend pour diagnostiquer un problème

## ➤ MTTR (Mean Time To Repair)

- Temps moyen que cela prend pour corriger un problème

## ➤ MTTF (Mean Time To Failure)

- Temps moyen pour qu'une erreur revienne à la normale

## ➤ MTBF (Mean Time Between Failures)

- Temps moyen entre deux erreurs
- Disponibilité =  $\text{MTTF} / \text{MTBF}$



# Conclusion



# Ce que nous avons vu

---

## ➤ Déploiement continu

- Intégration continue
- Automatisation
- Tests automatisés
- Dématérialisation

## ➤ Livraison continue

- Livraison en Production
- Microservice
- Container

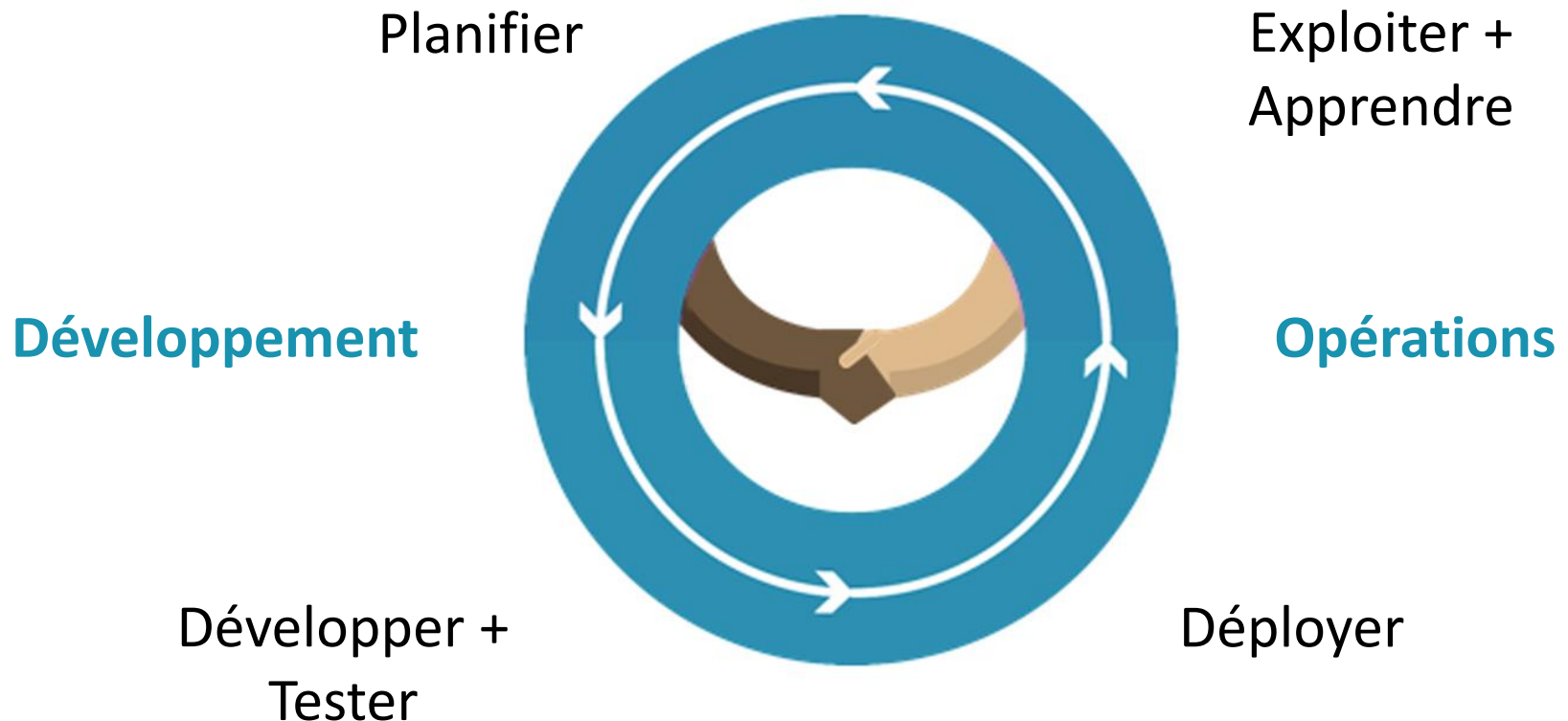
## ➤ Opération continue

- Exploitation des retours rapides
- Indicateurs, statistiques et critère



# DevOps = ALM + Dev et Ops

---



# DevOps

---

- Ce n'est pas une recette miracle
- Cela ne s'applique pas à chaque compagnie, chaque projet ou chaque personne
- C'est livrer souvent de la valeur ajoutée au client final
- C'est une évolution des pratiques existantes et connues
- C'est un apprentissage continu
- Ce n'est pas une méthodologie
- Ce n'est pas que « Dev et Ops »
- Objectifs de cette rencontre :
  - Introduction à DevOps
  - Aperçu des pratiques associées à DevOps
  - Pistes de réflexion sur des concepts DevOps



# Références





# Références

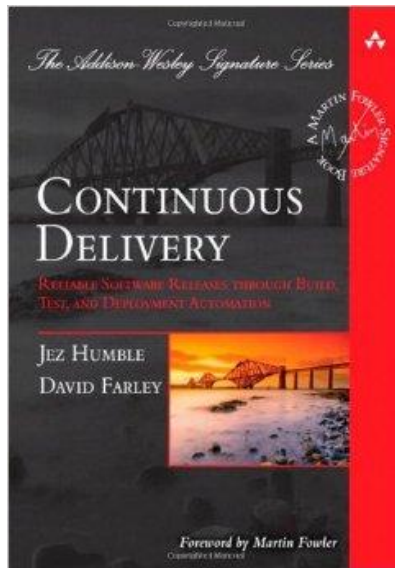
---

- [Evolution of Modern DevTest Labs in DevOps World](#)
- [Assessing and Improving your DevOps Capabilities](#)
- [DevOps an IT Pro Guide](#)
- [Enabling DevOps Practices with Visual Studio Online Build](#)
- [Open Sources for DevOps Practices](#)
- [DevTest Scenarios in the DevOps World](#)
- [Exploring Microservices in Docker and Microsoft Azure](#)

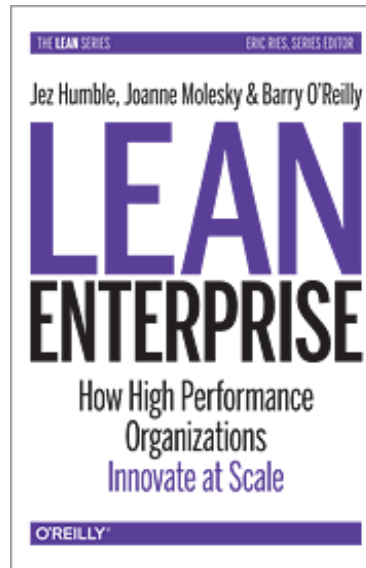


# Lectures (1/2)

---



Continuous Delivery  
Jez Humble  
David Farley



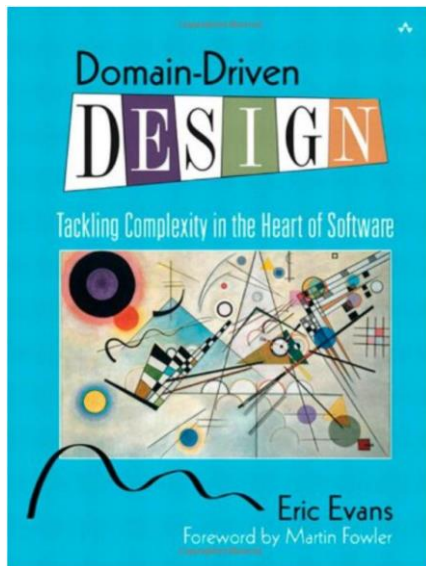
Lean Enterprise  
Jez Humble  
Joanne Molesky  
Barry O'Reilly



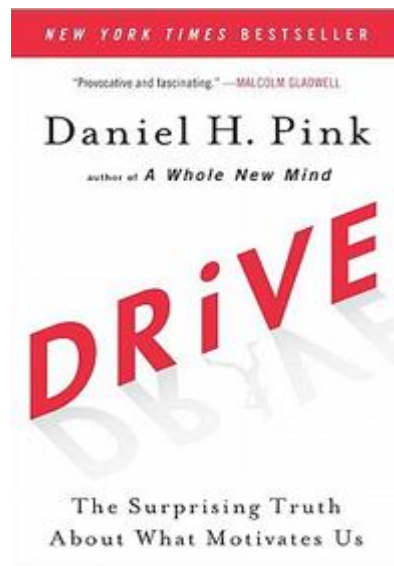
Building Microservices  
Sam Newman

# Lectures (2/2)

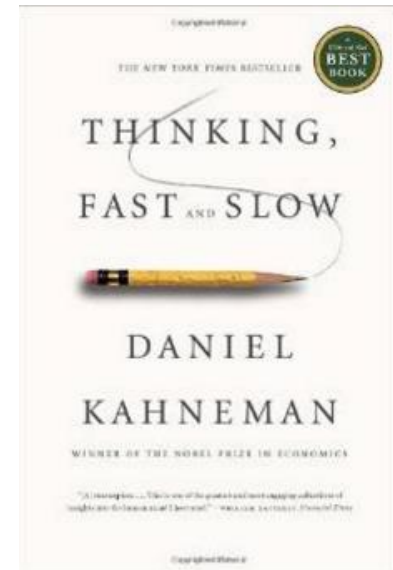
---



Domain Driven Design  
Eric Evans

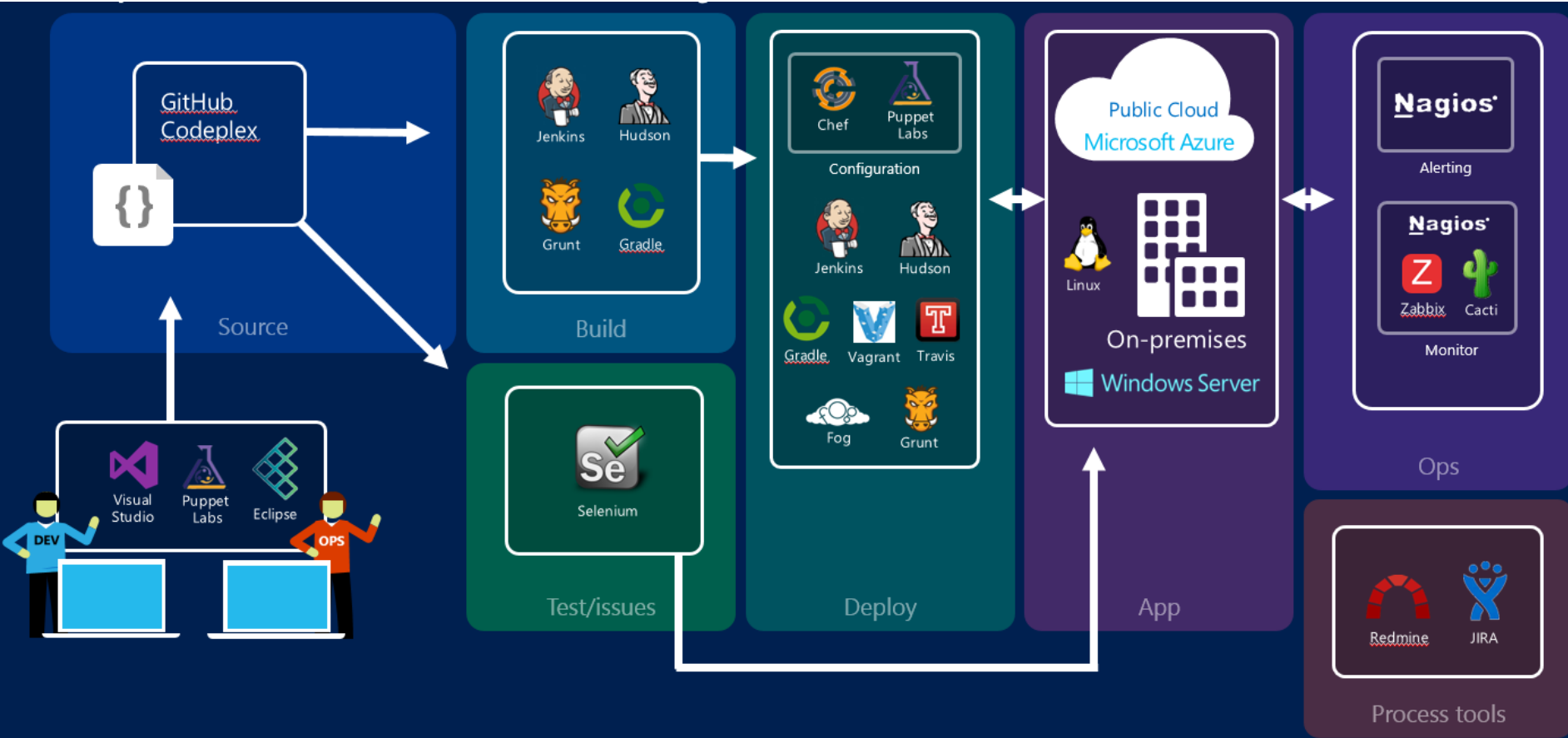


Drive  
Daniel H. Pink

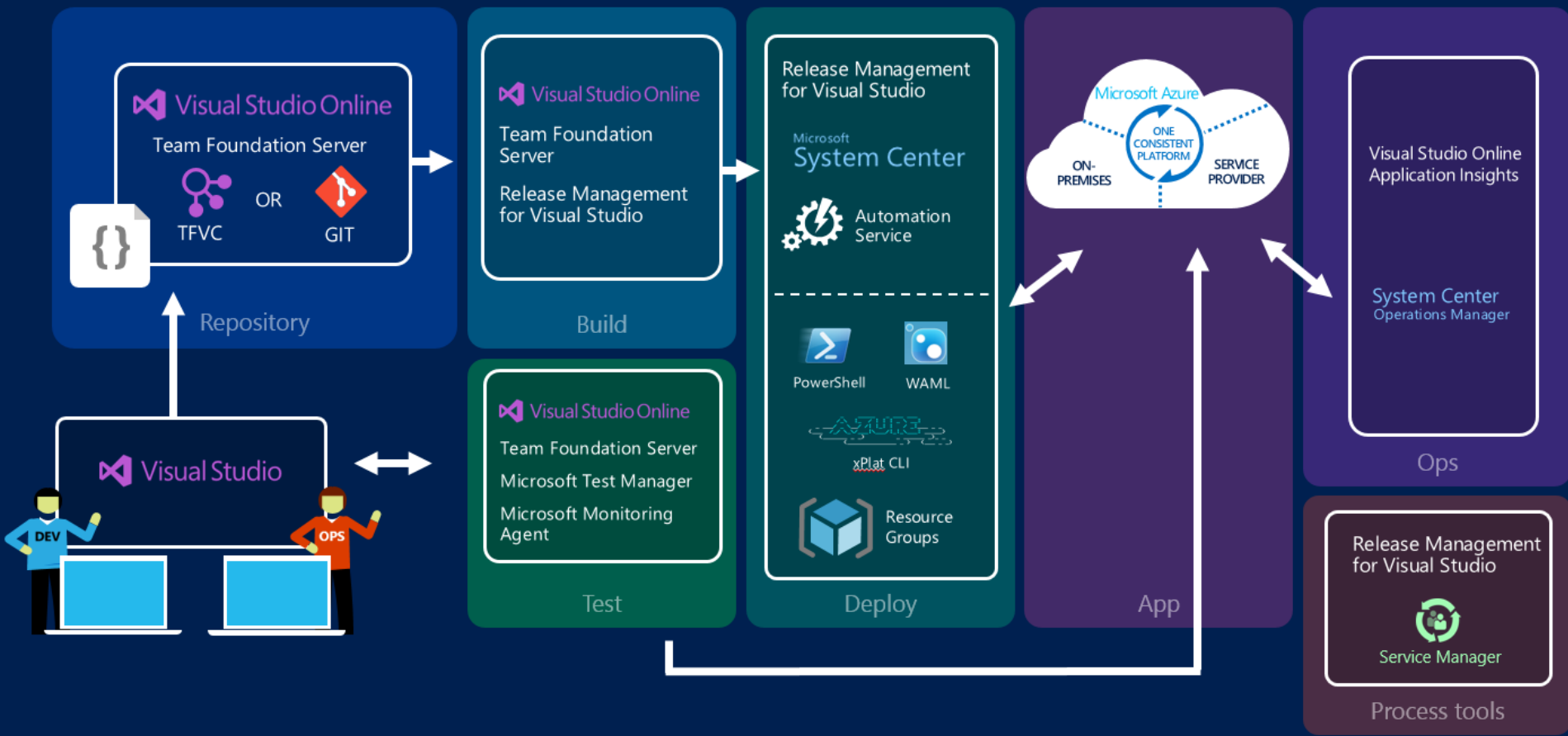


Thinking, Fast and Slow  
Daniel Kahneman

# Écosystème Open-source



# Écosystème Microsoft



# Restez connectés près de chez vous

---

- [DevOps Québec](#)
- [Agile Québec](#)
- [Microsoft Azure Québec](#)
- [Agile Montréal](#)



Questions ?

Merci pour votre attention !  
Bon Agile Tour 2015.

