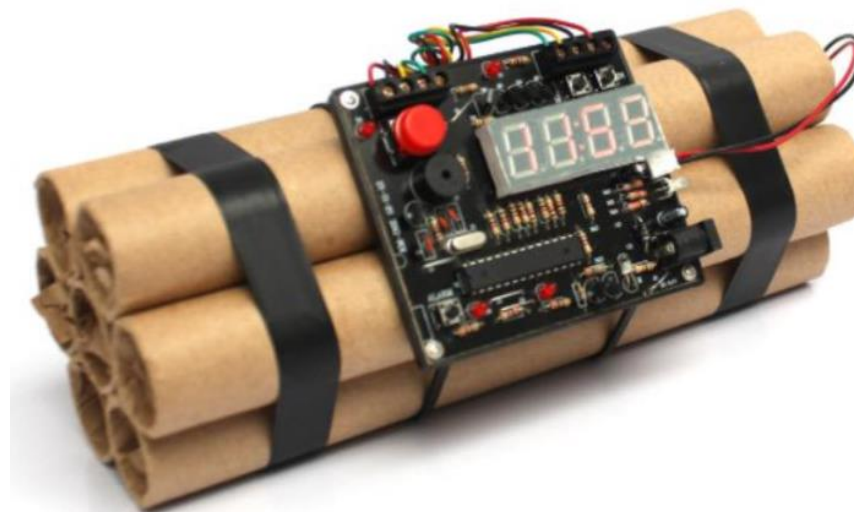


Comment saboter la transition devops de son entreprise

Pièges et bonnes recettes d'une transition devops

@alaindelafosse



Bienvenue !

Quelques points



Objectif

Aborder d'une manière décalée le sujet des transitions devops et leur impact sur la sécurité, l'organisation, ainsi que les autres points à surveiller.



Prérequis ?

Connaître un peu devops et le contexte d'une transition



Avertissement 1

On parle pas beaucoup de technique, désolé (Le snowcamp est là pour ça ;-)



Avertissement 2

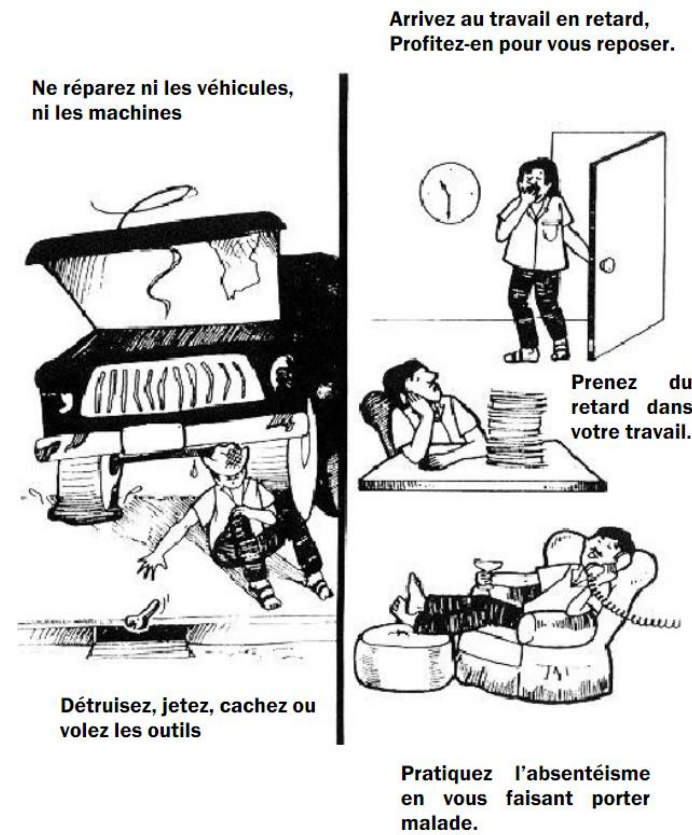
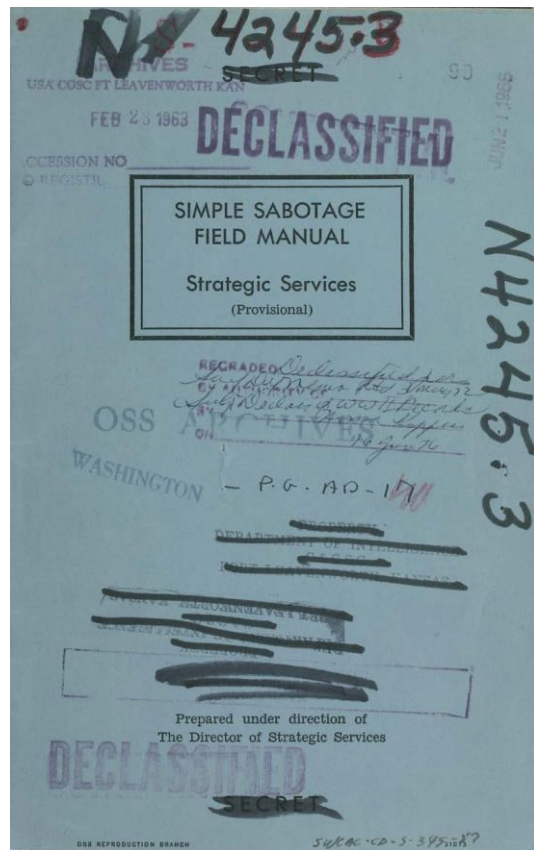
Les personnages et les situations de cette présentation étant purement fictifs, toute ressemblance avec des personnes ou des situations existantes ou ayant existé ne saurait être que fortuite.



Avertissement 3

Je n'encourage évidemment pas le sabotage et toute utilisation de cette présentation pour le sabotage d'une transition est néfaste et je nierai toute implication ou celle de mes services..

Manuel de sabotage de la CIA



But du document

Utilisé pour former des agents chargé de perturber l'activité de pays dont on veut changer le régime.
Des points opérationnels et des points plus méthodologiques

Eléments applicables

Dès que possible, soumettez toutes les questions à des comités, pour « une étude plus attentive ».

Tentez de former les comités les plus larges possibles – jamais moins de cinq personnes.

Soulevez des problèmes futiles aussi fréquemment que possible.

Comprenez les ordres de travers. Posez d'innombrables questions ou lancez-vous dans de longs échanges écrits à propos de ces ordres.

Ergotez sur ces ordres quand vous le pouvez.

Pinaillez sur les éléments de langage, les minutes, les décisions.

Soyez irréaliste et poussez ceux qui vous écoutent à être « réalistes » et à éviter toute précipitation qui pourrait entraîner des difficultés par la suite.

Dans le but de dégrader le moral général et la productivité avec, soyez agréable avec les employés incompetents.

Accordez des promotions non méritées. Discriminez les employés efficaces, plaignez-vous injustement de leur travail.

Remplissez les circulaires de manière illisible pour qu'on soit obligé de les refaire ; faites des erreurs ou omettez les informations requises.

Répandez de fausses rumeurs qui sonnent comme de vrais bruits de couloirs.

Pourquoi le sabotage ?

Pour conserver son rôle actuel Par peur du changement

- Installé dans une posture de Héros que l'on souhaite conserver
- The opposite of [#KISS](#) (Keep It Simple Stupid) is [#KICKME](#) (Keep It Complicated Keep Me Employed) [@rhundhausen](#)
- Investissement dans une carrière, un poste potentiellement remis en cause
- Peur de perdre du pouvoir et de l'influence (budget, reports)
- Peur des contraintes, désintérêt
- Shadenfreunde

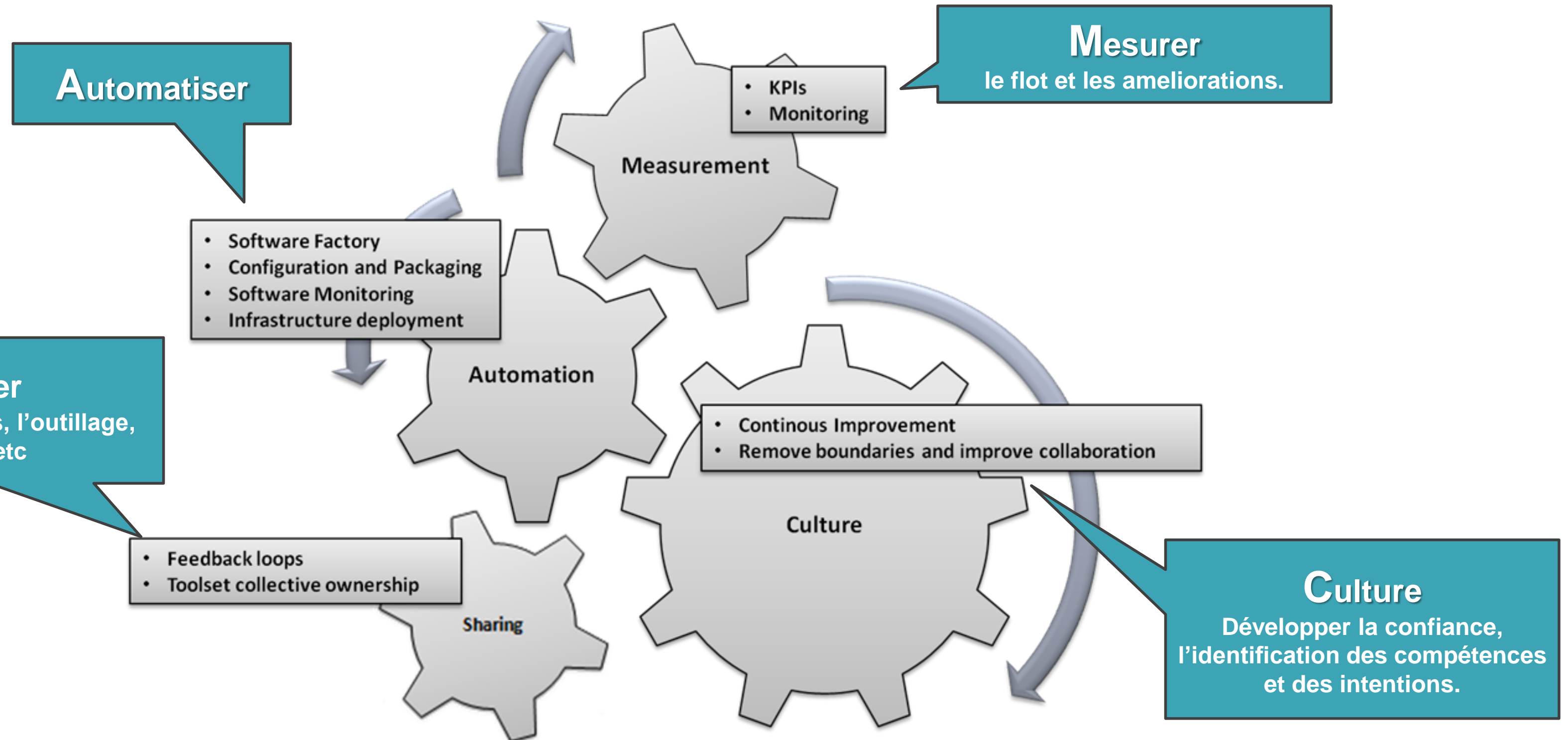


Intentions maladroites Impact non volontaire

- Posture de sauveur: devops ne marchera pas et va mettre en danger l'entreprise
- Souhait de perfection: inclure et traiter tous les cas particuliers



DevOps ? CAMS



Transition DevOps ?

Gestion du changement

Faire évoluer son organisation, ses outils, ses processus et surtout SA CULTURE pour atteindre des objectifs tels que la qualité et la rapidité de déploiement.



Impliquer la sécurité

Sabotage

Excellent angle d'attaque !

Points à mettre en avant auprès d'oreilles attentives:

- Accès aux données sensibles possible à tous (RGPD)
- Tout le monde pourrait casser le service en production
- Mots de passe de production disponibles pour tous dans les scripts (OS, BDD, etc)



Prévention

Anticiper et travailler avec les responsables sécurité dans la mise en place du pipeline et des environnements de production.

Intégrer et mettre en place une ségrégation des rôles et responsabilités par plateforme (au moins au début)

Points positifs à valoriser:

- Réinstallation rapide
- Application des patches rapide
- Auditabilité des interventions
- Contrôle continu et automatisé des droits

L'outillage et les processus

Sabotage

- Mettre en avant que le choix de l'outillage est primordial et doit devancer tout autre sujet
- Sujet important donc tout les processus associés doivent être finement définis, approuvés et suivis avec les métriques associées (rigueur, sécurité)
- Crier au scandale si on ne traite pas tout le périmètre et tout les cas de figure possibles...

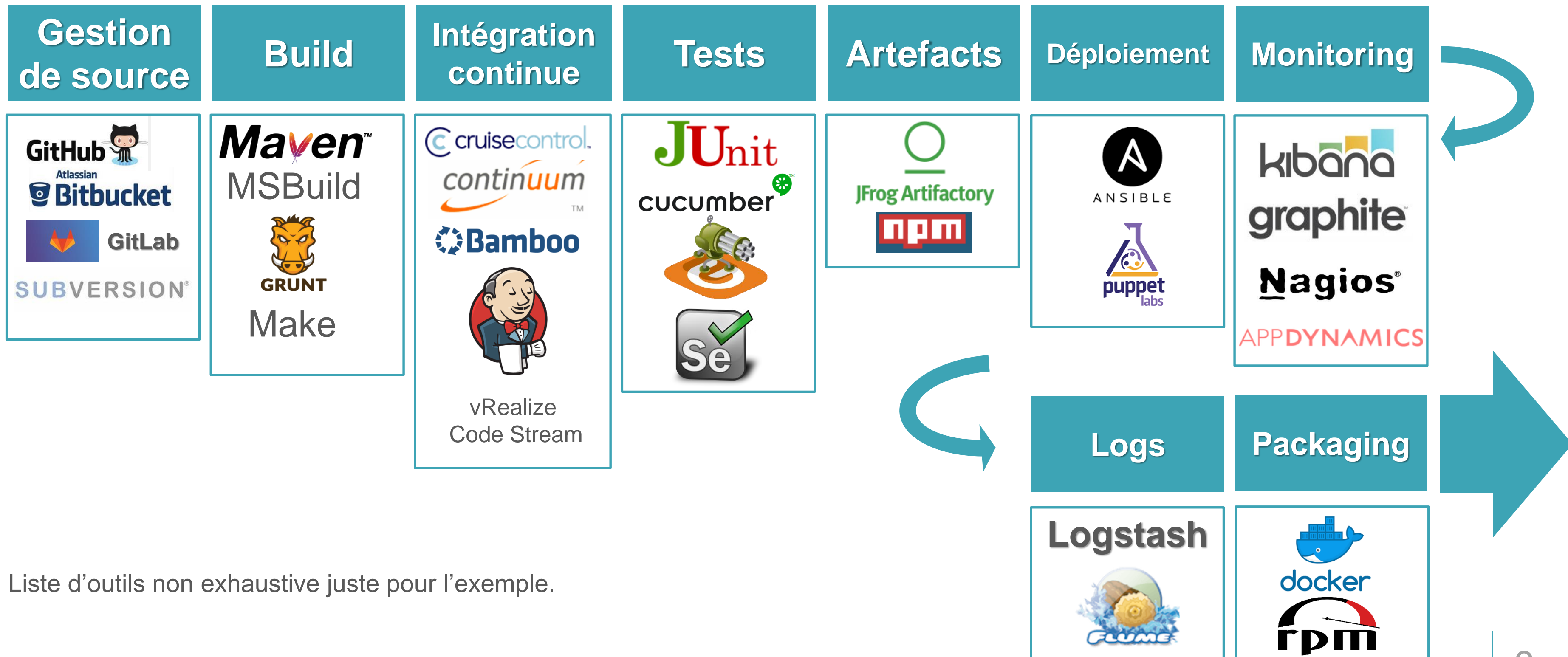


Prévention

- Se focaliser sur le résultat et annoncer dès le début le périmètre couvert par l'outillage et que des compromis seront peut être pris.
- Annoncer la flexibilité des processus et l'importance de cycles de réglages progressifs et continus.
- Impliquer toutes les parties dans le choix de l'outillage et choisir de préférence une boite à outils plutôt qu'une solution packagée par un éditeur

Pipeline DevOps

Au delà de l'automatisation du déploiement, un vrai pipeline d'outils à choisir !



Liste d'outils non exhaustive juste pour l'exemple.

L'organisation

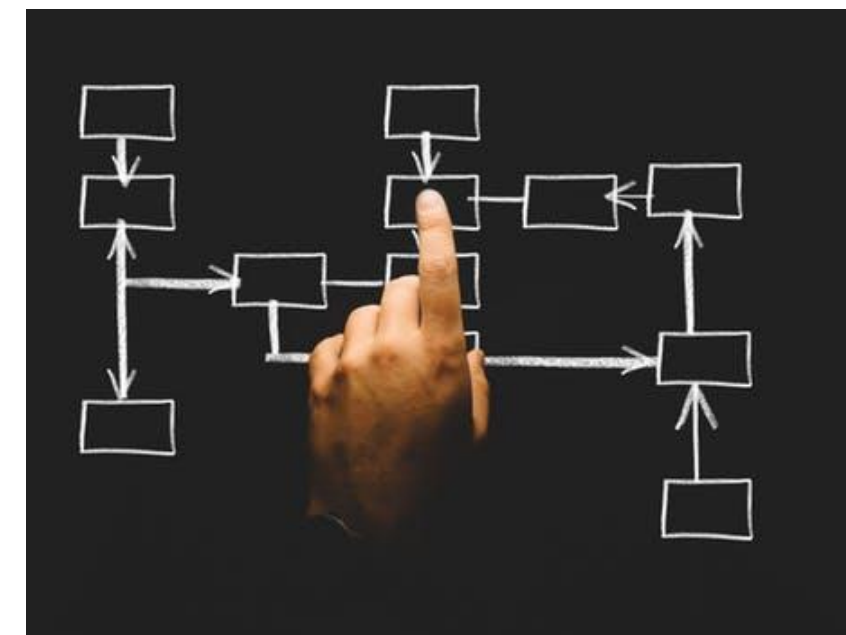
Sabotage

La transition va nécessiter tôt ou tard la création de nouveaux rôles, de nouvelles organisations, des changement de lignes de reporting, des réaffectations de manager et de ressources voire de nouveaux chemins de carrière:





- Il faut alerter les équipes sur les risques liés à ces changements !
- Bien identifier les conflits d'objectifs et les monter en épingle: SLA versus risques liés aux changements
- Il faut mobiliser l'équipe RH ! C'est une requalification du contrat de travail...

Prévention

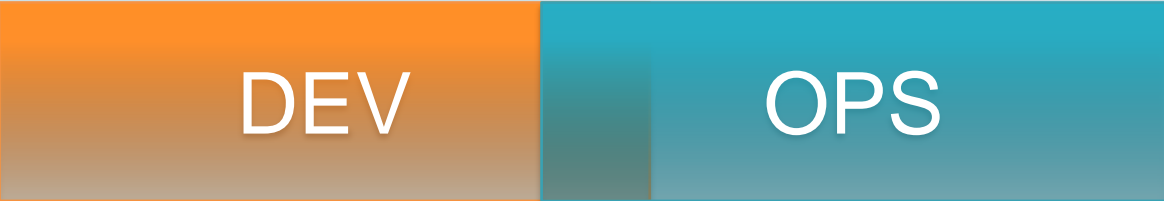
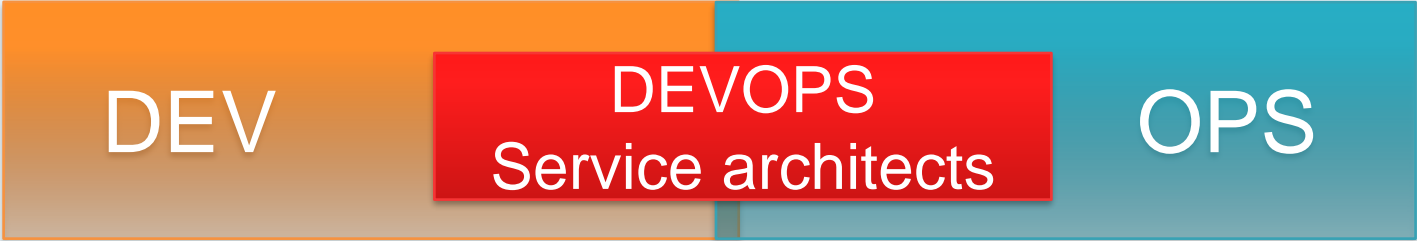


- Préparer la généralisation au delà du projet pilote en impliquant les RH, les managers et en anticipant de potentiels changements d'organisation.
- Réaliser un travail de fond sur les objectifs individuels et d'équipe pour sortir du conflit originel dev-ops; les un focalisé sur les dates de livraison et les autres sur la stabilité du service.



Antipatterns d'organisation

Pattern			
	Créer un silo devops		<p>Une équipe devops constitué pleine de personnes devops. Donc encore un autre silo qui va défendre son approche « pure » de devops contre les développeurs et les opérations à l'ancienne.</p> <p>Pas d'impact en terme de culture, de collaboration, de partage. Même en terme d'automatisation, risque de rejet de la part des autres équipes.</p>
	NoOps		<p>« On n'a plus besoin d'ops de nos jours, on verra plus tard quand on déploiera dans le cloud ».</p> <p>La vélocité de l'équipe de dev va s'effondrer quand elle va attaquer les déploiements: manque de compétences (monitoring, PCA/PRA), va devoir reprendre certains aspects de l'architecture et du code, et va finalement réaliser que l'opération d'un service ne se résume pas qu'à l'utilisation d'une infrastructure.</p>
	Equipe tools limité à la CI		<p>Une équipe coté dev est chargée d'industrialiser le pipeline de développement (« <i>continous integration</i> »), et va augmenter le rythme des releases sans se préoccuper des besoins des opérations (déploiement, configuration).</p> <p>Résultat, les releases vont s'empiler coté opérations et le délai de mise en production augmenter.</p> <p>De plus, le peu d'implication des opérations dans la conception des applications risque d'entraîner des A/R entre les opérations et les devs.</p>
			<p>Embaucher des « devops » et les affecter à son équipe opérations ne permet en rien de créer une collaboration entre les deux équipes.</p> <p>Pas de changement en terme de culture, éventuellement une amélioration de l'automatisation coté opérations.</p>

Patterns d'organisation

Pattern	
 <p>Collaboration idéale (graal)</p>	<p>Deux équipes avec leur compétences, spécialisées sur leur domaines mais surtout partageant les mêmes objectifs. Nécessite un excellent niveau de maturité sur le plan culturel pour ne pas régresser sur l'aspect collaboration et prendre la responsabilité collective de la conception de l'application. Le management si il est distinct doit avoir compréhension excellent du métier de l'autre équipe.</p>
 <p>Service Architects</p>	<p>Les opérations se concentrent sur l'intégration de l'infrastructure adaptée aux développeurs (on-premise et/ou cloud). Une équipe spécialisée transverse composée de services architectes, accompagne la conception, fournit l'expertise opérations aux développeurs (conception, monitoring, etc) et assure l'intégration des projets en production avec les ops. Permet de fournir de la valeur rapidement et peut servir de première étape avant une collaboration totale.</p>
	<p>Une équipe (parfois constituée d'intervenants extérieurs) assure le rapprochement entre les dev et les ops. Elle a pour vocation d'accélérer la transition vers une vraie collaboration et se veut temporaire.</p>
	<p>Dans ce modèle, l'équipe DepOps récupère les livrables applicatifs en ayant injecté en amont un cahier des charges exploitation aux dev. Elle conçoit le service de production, assure son déploiement et son exploitation. Les opérations se concentrent sur la fourniture de l'infrastructure.</p>

Les métriques et KPI

Sabotage

Focaliser l'attention sur des métriques non pertinentes:

- S'alarmer du nombre de d'erreurs de déploiement
- Pointer l'investissement versus l'opération concrète réalisée (100 jours hommes pour copier des fichiers en production ??)



Prévention

Mettre en avant les objectifs du point de vue client:

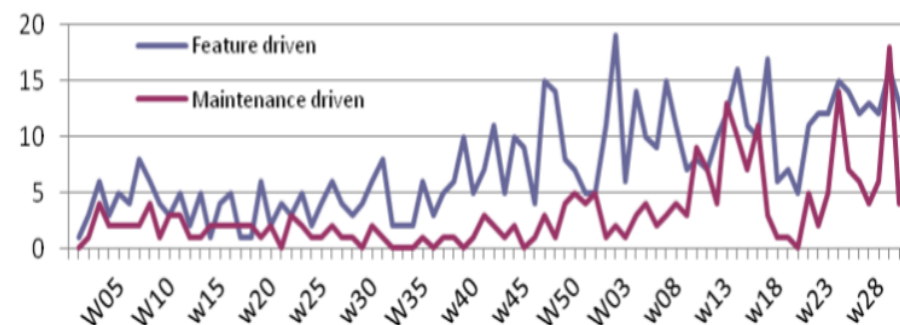
- Délai de mise à disposition des versions en production
- Durée de déploiement,
- Augmentation de la fréquence des déploiements
- Répétabilité sans effort

Mais garder des métriques internes et les utiliser pour améliorer le système et justifier l'effort global.

Quelques exemples de métriques



Nombre de déploiements ou changements par semaine



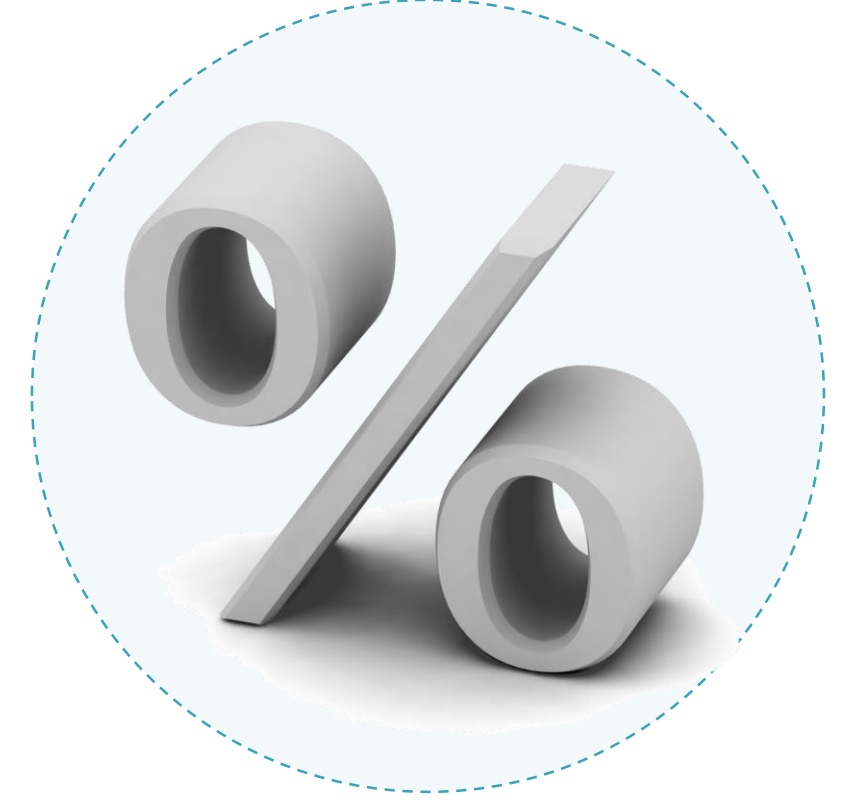
Volume des changements hebdo.

Cela peut être le nb de lignes de code, le nb de composants impliqués, voire le nombre de user stories intégrés.



Délai de mise en production.

Mesurer le délai nécessaire entre la fin de l'écriture d'un patch et la disponibilité en production.



Pourcentage de déploiements ratés.

Les échecs et erreurs sont normaux (qui ne fait jamais de bug?) mais il faut surveiller l'évolution du pourcentage.

Exploiter la courbe du changement

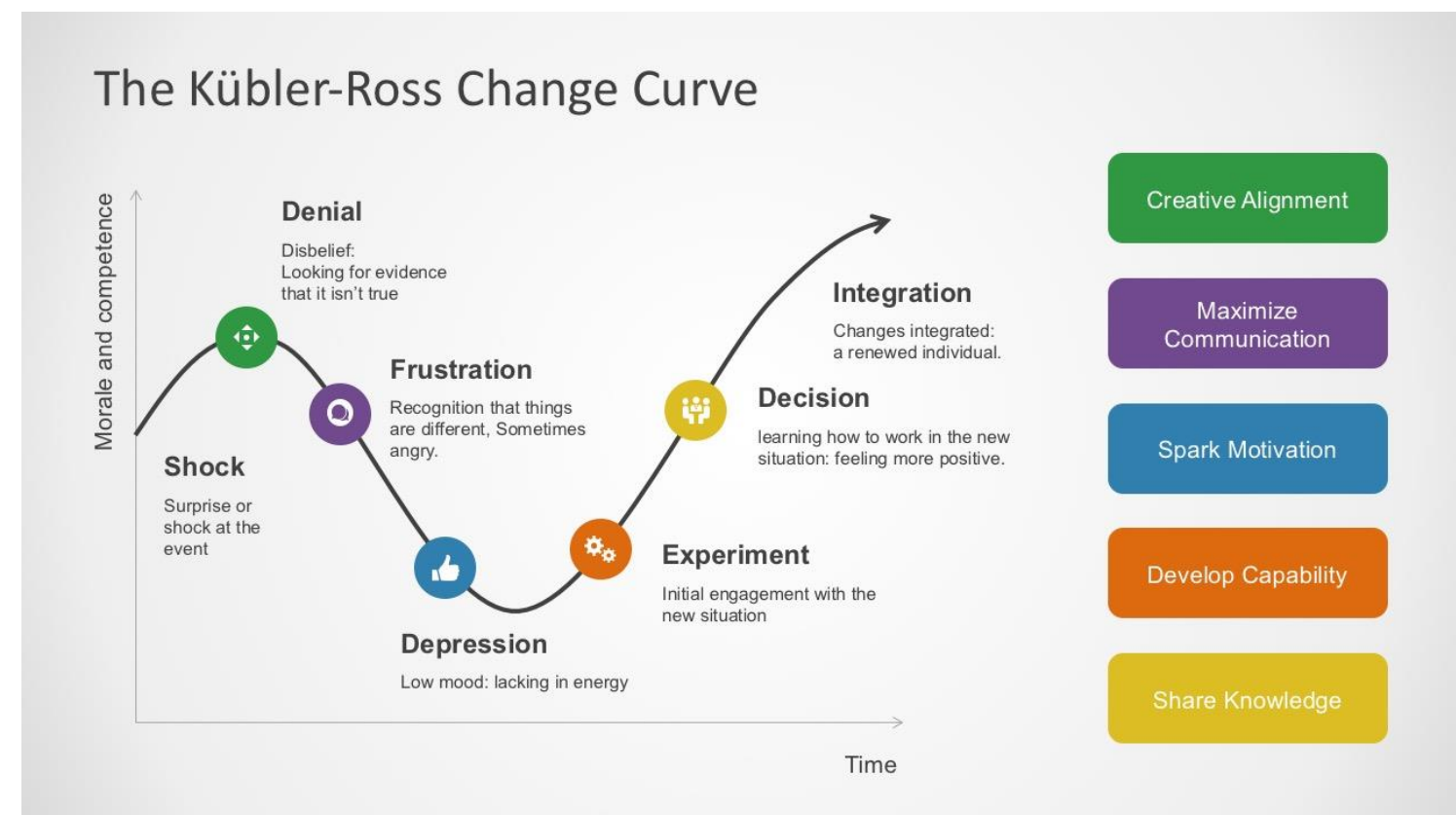
Sabotage

- Le changement entraîne forcément à un moment une perte de productivité et une perte de confiance et de motivation: c'est à ce moment qu'il faut agir...
- Jouer sur le fait que c'est beaucoup plus confortable de garder ce que l'on connaît
- Demander des réunions de ré-évaluation de la transition avec le plus de monde possible



Prévention

- Faire des projets pilotes / de démonstration
- Prendre des volontaires pour ces projets
- Accueillir, apprendre et rendre transparent les couacs
- Bien préparer le projet de transition



Transition: ingrédients

Vision

Vision en rapport avec la stratégie de l'entreprise (définir des objectifs SMART: par ex un déploiement par jour dans un an), ne pas se bloquer sur une organisation cible.

Compétences

Développer à la fois les compétences techniques (avec quoi: maîtrise des outils) mais aussi organisationnelles (pourquoi, comment) et pour tout le monde !

Motivation

Priorité globale, reconnaissance et célébration des succès et des efforts

Ressources

Du temps, de l'argent, des compétences extérieures, du temps pour des conférences ...

Plan d'action

Partager la vision aux équipes, s'assurer du support du top management, laisser décider les équipes des itérations et des objectifs associés (carpaccio)

Transition DevOps



Dévaloriser le projet pilote

Sabotage

Ce que l'on craignait est arrivé; un projet pilote a réussi ☹. Mais rien n'est perdu....

- Coût d'adaptation et ROI de la généralisation à tous les projets ?
- Pousser la généralisation des outils et processus issus du projet pilote tels quels sans adaptation à tout le monde



Prévention

- Définir et partager en amont du projet pilote les objectifs et les critères de succès; ainsi que les étapes ultérieures.
- S'en servir pour obtenir plus d'adhésion
- Ne pas crier victoire, c'est juste un projet pilote
- Bien remettre le projet pilote dans une dynamique plus générale d'amélioration continue et de déclinaison au cas par cas pour tous les projets...

Chercher les petites bêtes

Sabotage

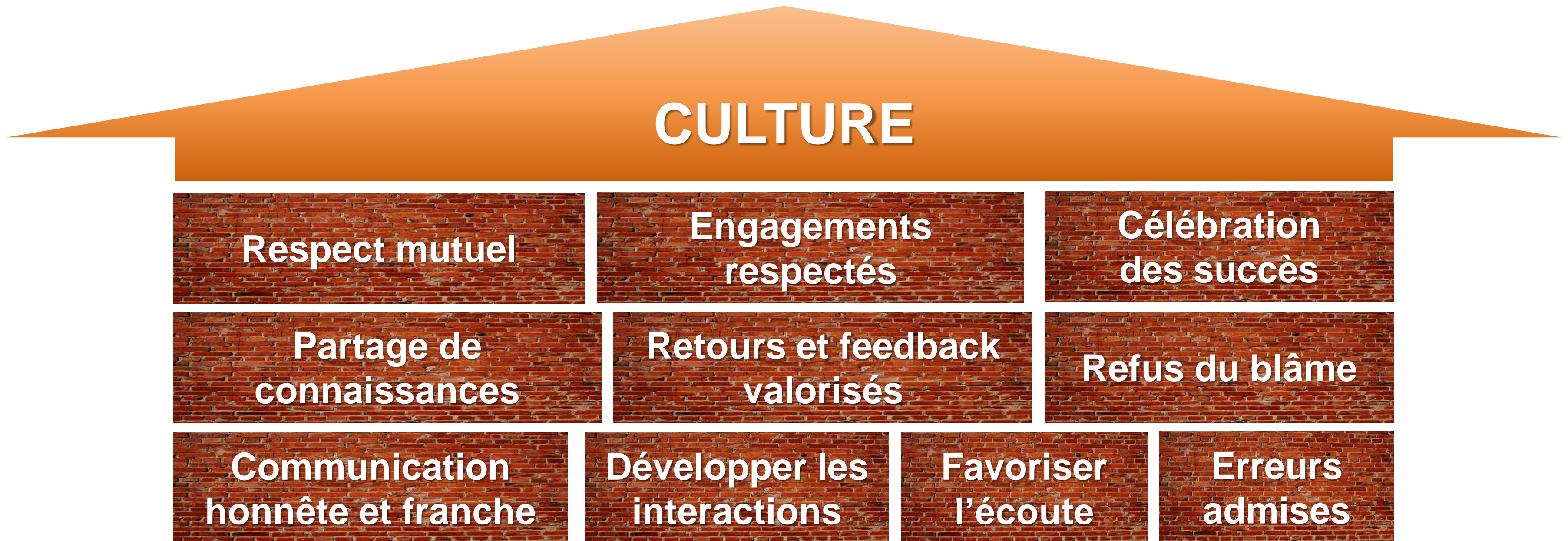
- Insister pour inclure dans la transition le traitement de points à la limite du projet: titre des postes, la gestion fine des droits, l'information des clients, etc
- Lister tout les problèmes non réglés et grossir leur importance. Comment seront - ils traités après la transition ?



Prévention

- Etre honnête sur les objectifs, les compromis et le périmètre de la transition
- Mettre en place une organisation et des moyens de manière à poursuivre après la transition avec des activités d'amélioration, de correction, de formation.

Saboter le changement de culture



Communauté devops

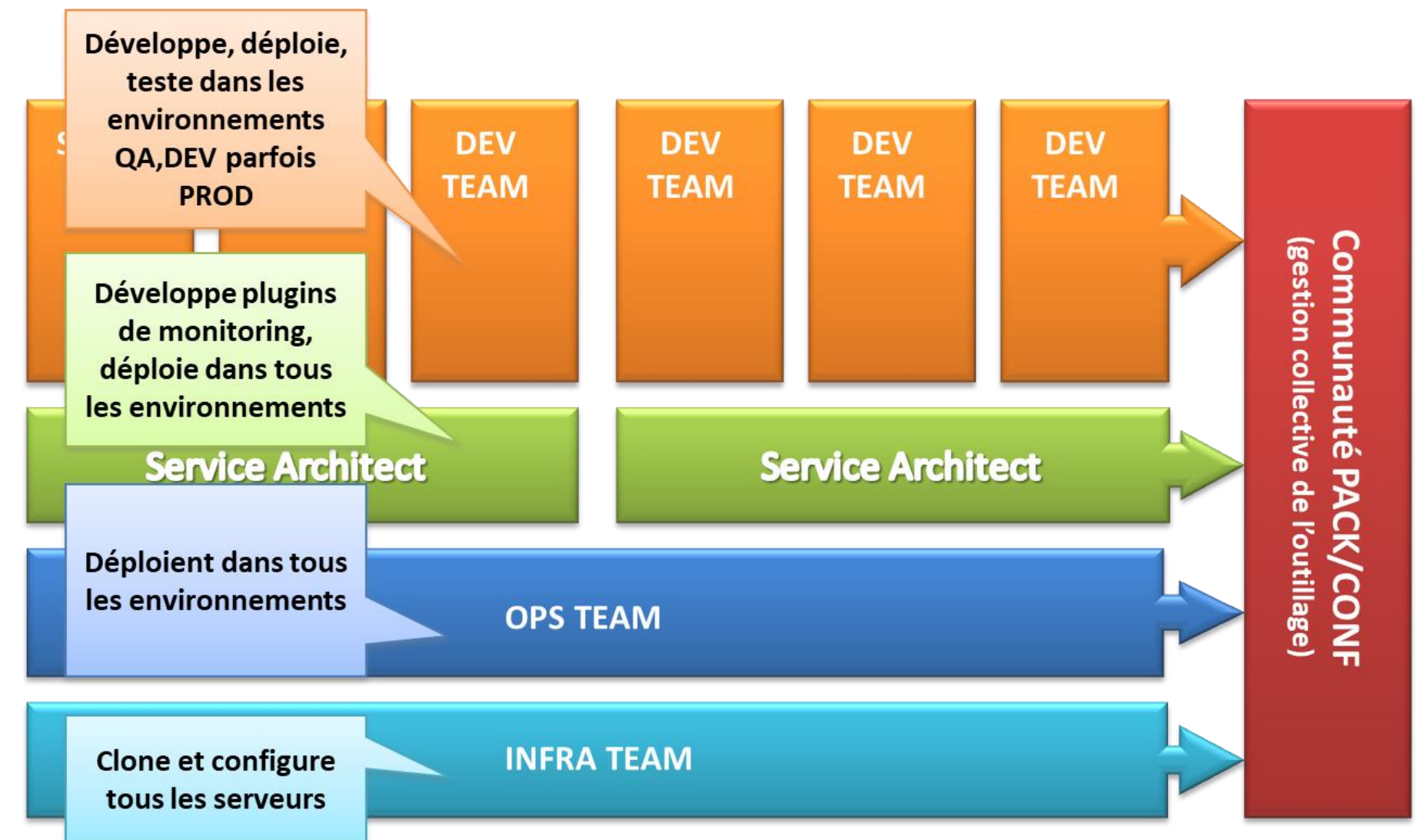
Le pouvoir des organisations non-hiérarchiques..

Comment concilier le besoin de poursuivre le travail après une transition ou un projet pilote sans mobiliser et spécialiser une équipe dédiée ?

Wenger développe le concept de communautés de pratique comme un groupe de personnes qui travaillent ensemble et qui sont en fait conduites à inventer constamment des solutions locales aux problèmes rencontrés dans leur pratiques professionnelles. Après un certain temps et au fur et à mesure que ces personnes partagent leurs connaissances, leurs expertises, ils apprennent ensemble.

Permet de:

- Mobiliser des ressources de manière variable et/ou temporaire
- Contribuer en fonction de ses compétences et besoin
- Fournir un cadre de suivi de la transition



Conclusion



SECURITE



OUTILLAGE



ORGANISATION



METRIQUES



VISION



SUPPORT



CULTURE



STRATEGIE

MERCI !

DES QUESTIONS?

