

Surmonter les anti-patrons culturels nuisant à DevOps

Jean-Marc Lavoie (SingularIT Solutions)

Sylvie Trudel (UQAM)

Pour mieux comprendre les anti-patterns

- *Bref historique*
- *DevOps: capacités requises*
- *Quelques anti-patterns communs*

Un bref historique



Je n'ai pas besoin de savoir où je vais, j'ai juste besoin de savoir où je suis allé.

Tow Mater (Cars, 2006)

Ceux qui ne peuvent se souvenir du passé sont condamnés à le répéter.
George Santayana (Vie de raison, 1905)

1945

1965

1985

1999

2001

2009

- Premiers ordinateurs numériques
- Premiers bogues



1945

1965

1985

1999

2001

2009

- *Industrialisation des ordinateurs*
- *Stockage*
- *Premiers processus*

Herbert D. Benington, 1956



Analyse

Conception

Code

Vérification

Maintenance

NATO Software Engineering Conference 1968

Analyse

Conception

Code

Installation

Maintenance

1945

1965

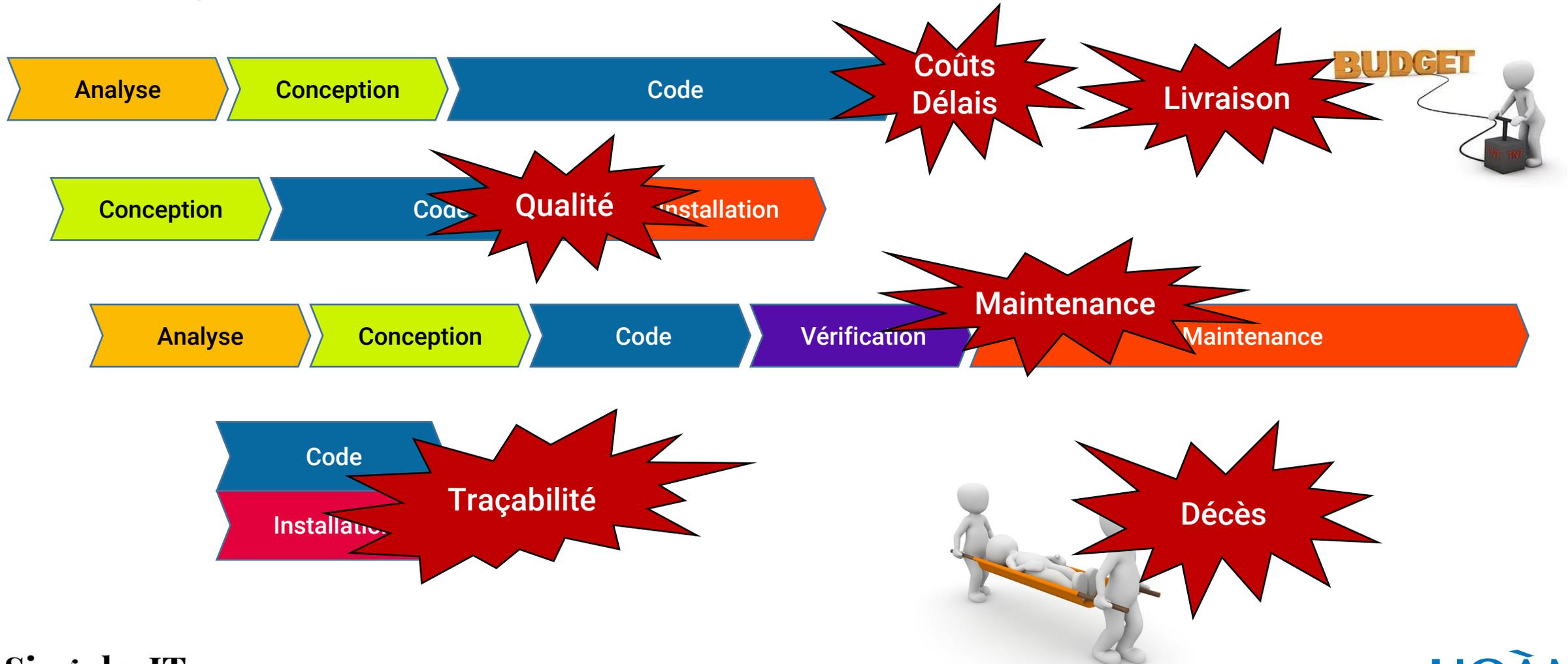
1985

1999

2001

2009

La crise logicielle



1945

1965

1985

1999

2001

2009

Émergence de processus et outils

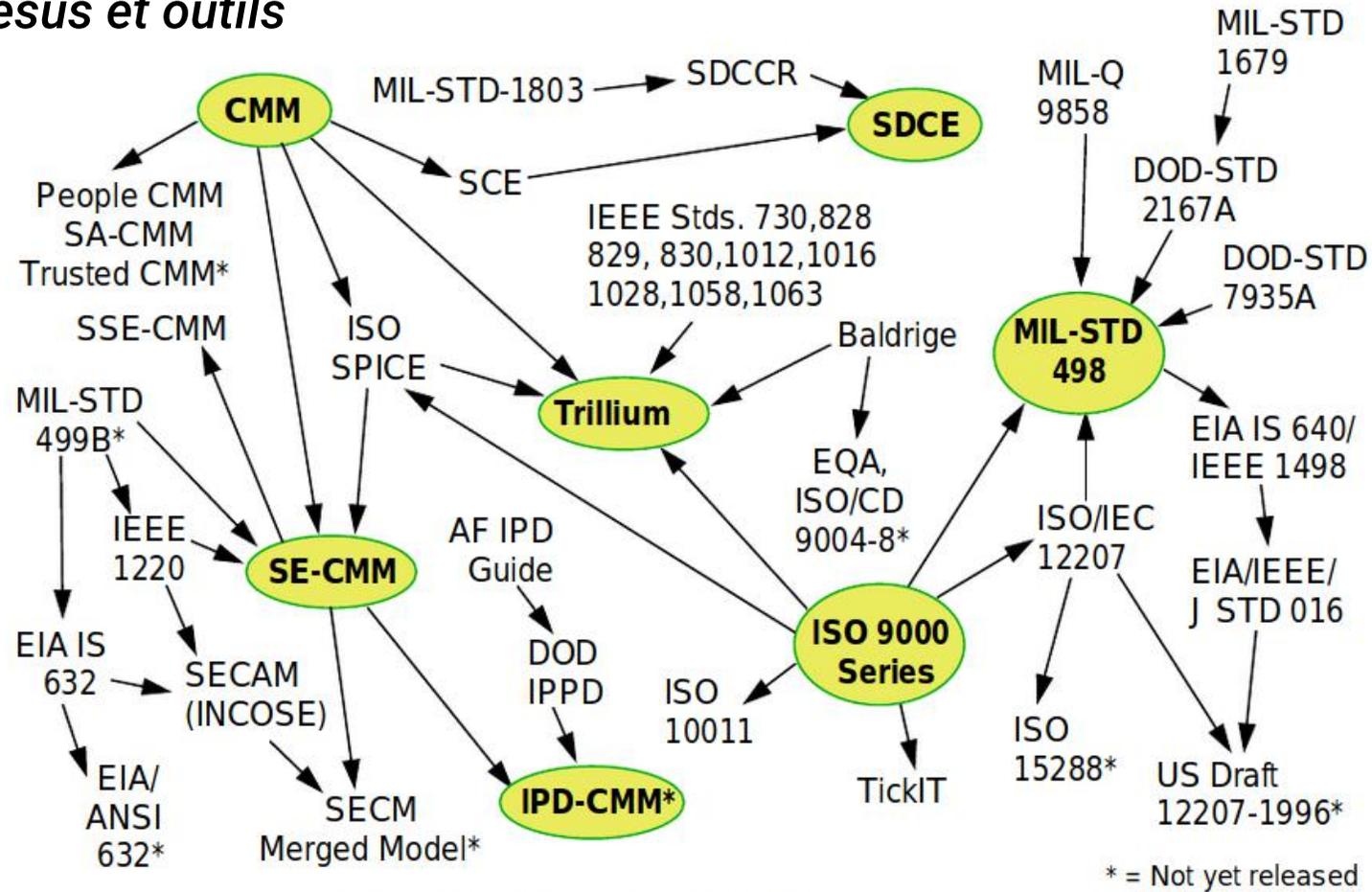
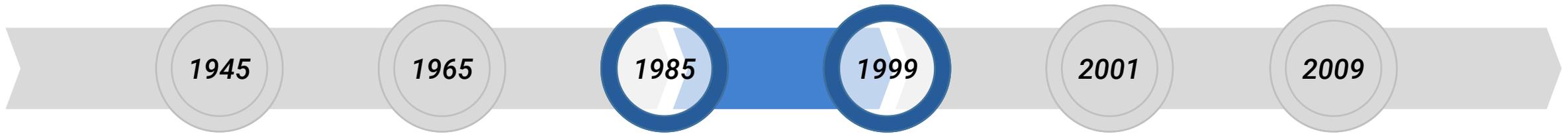
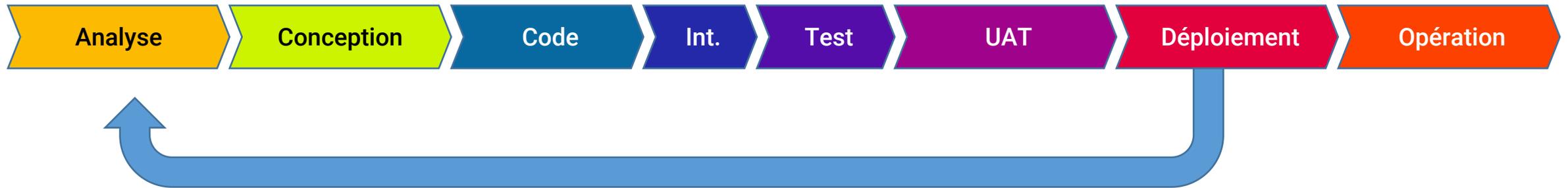


Figure 1. The Frameworks Quagmire



Le processus classique



6 mois - 2 ans

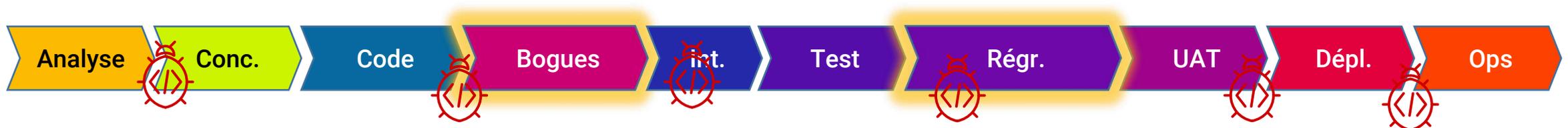
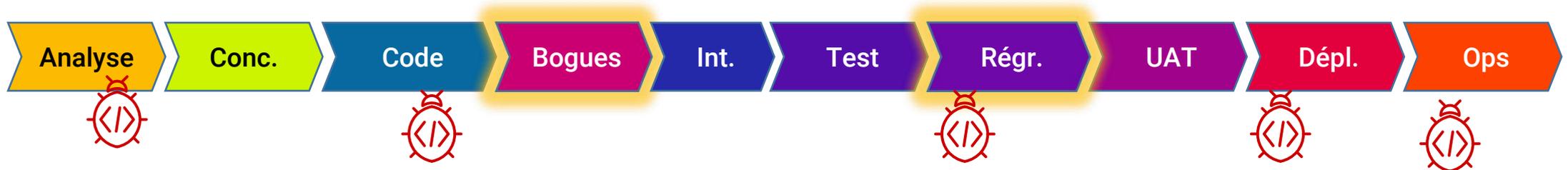
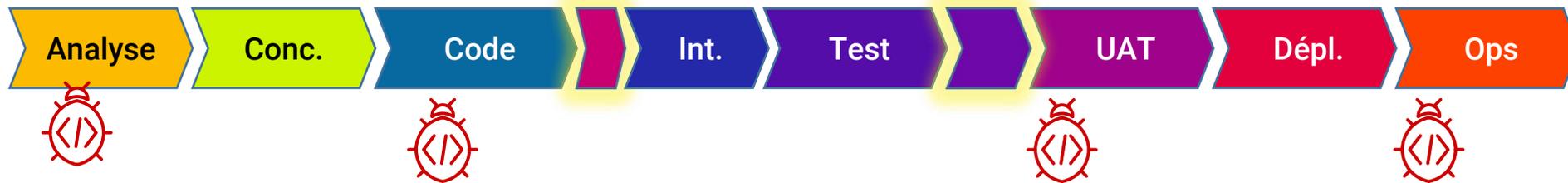
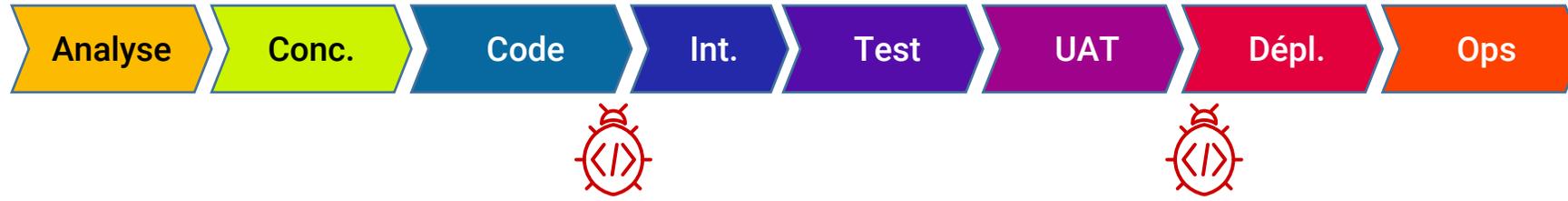
Quel est le problème ?

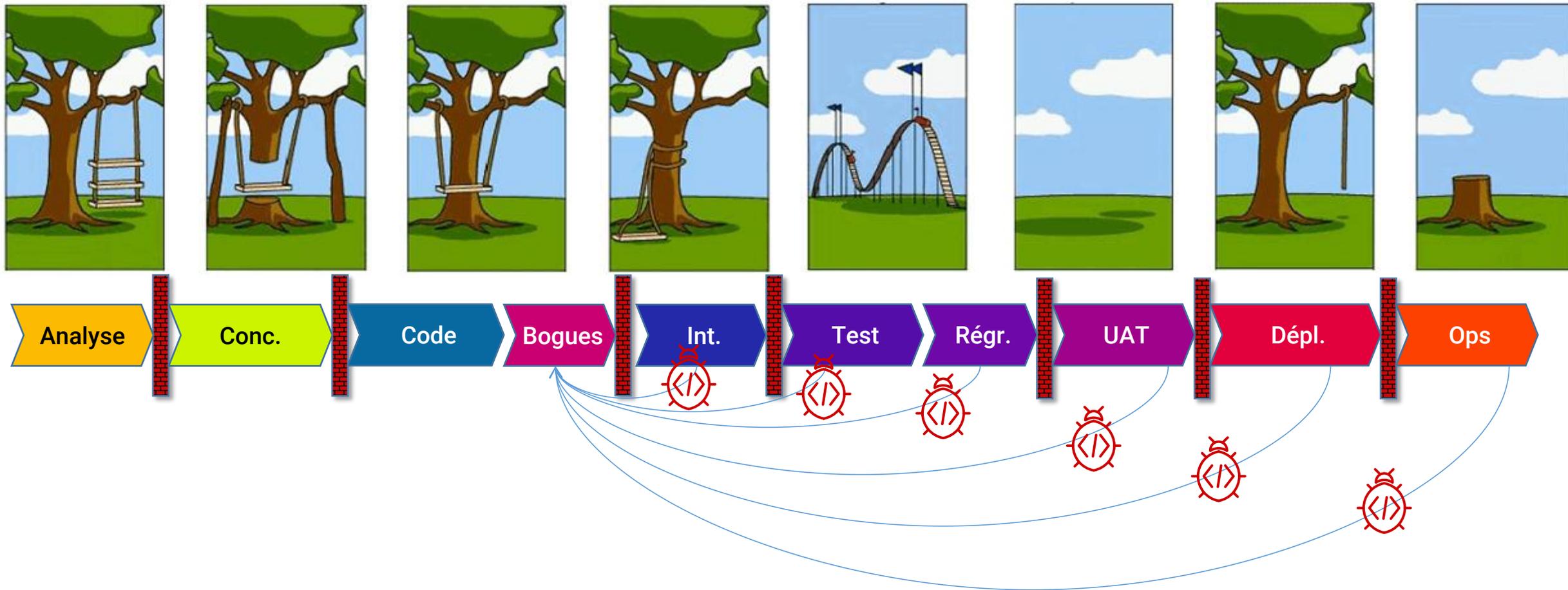


Considérer la régression



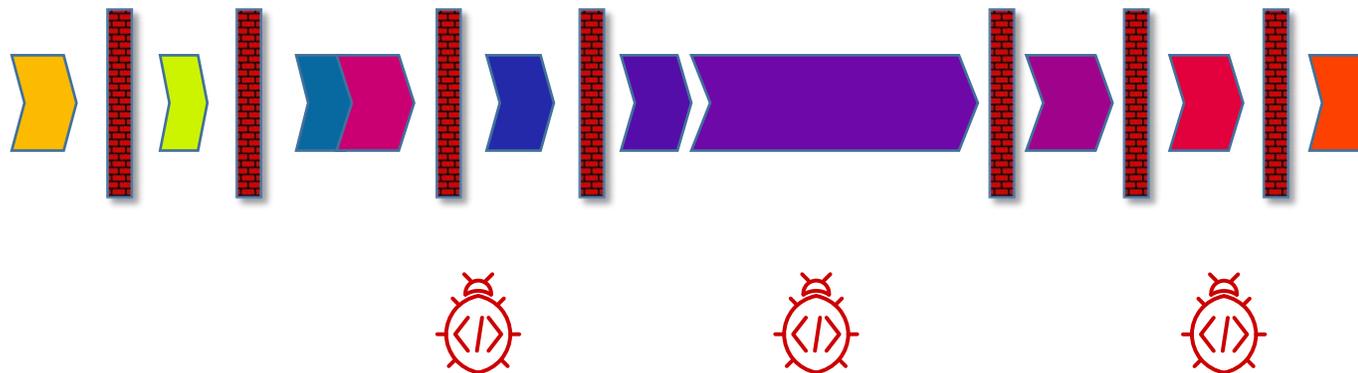
Ajoutez quelques cycles d'évolution





Ajouter quelques murs de confusion (silos)

Impossible à accélérer

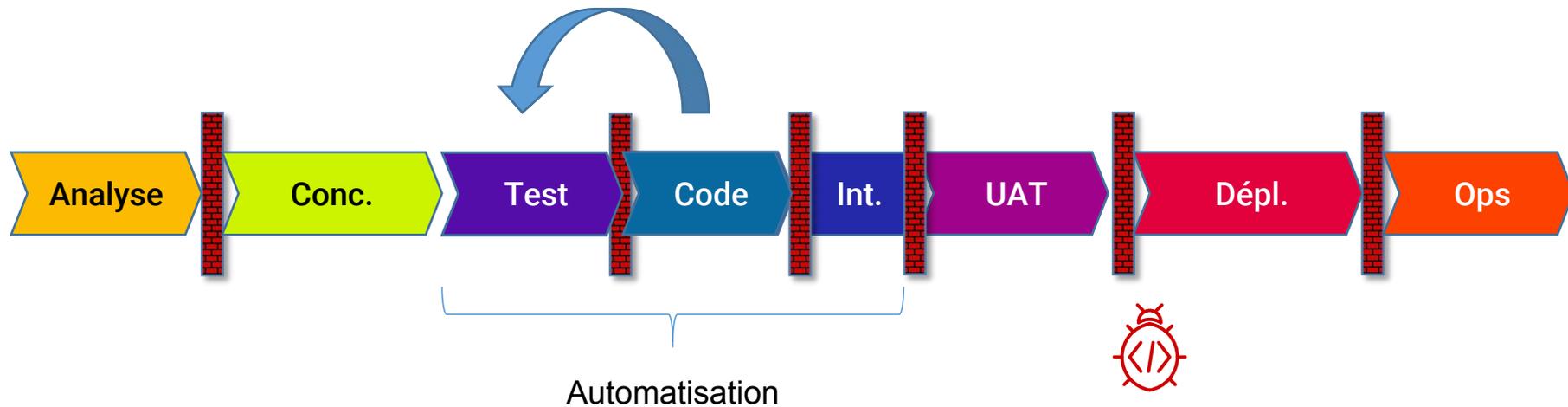


Comment s'en sortir?





Extreme Programming 1999++



1-6 Mois en 1999

1945

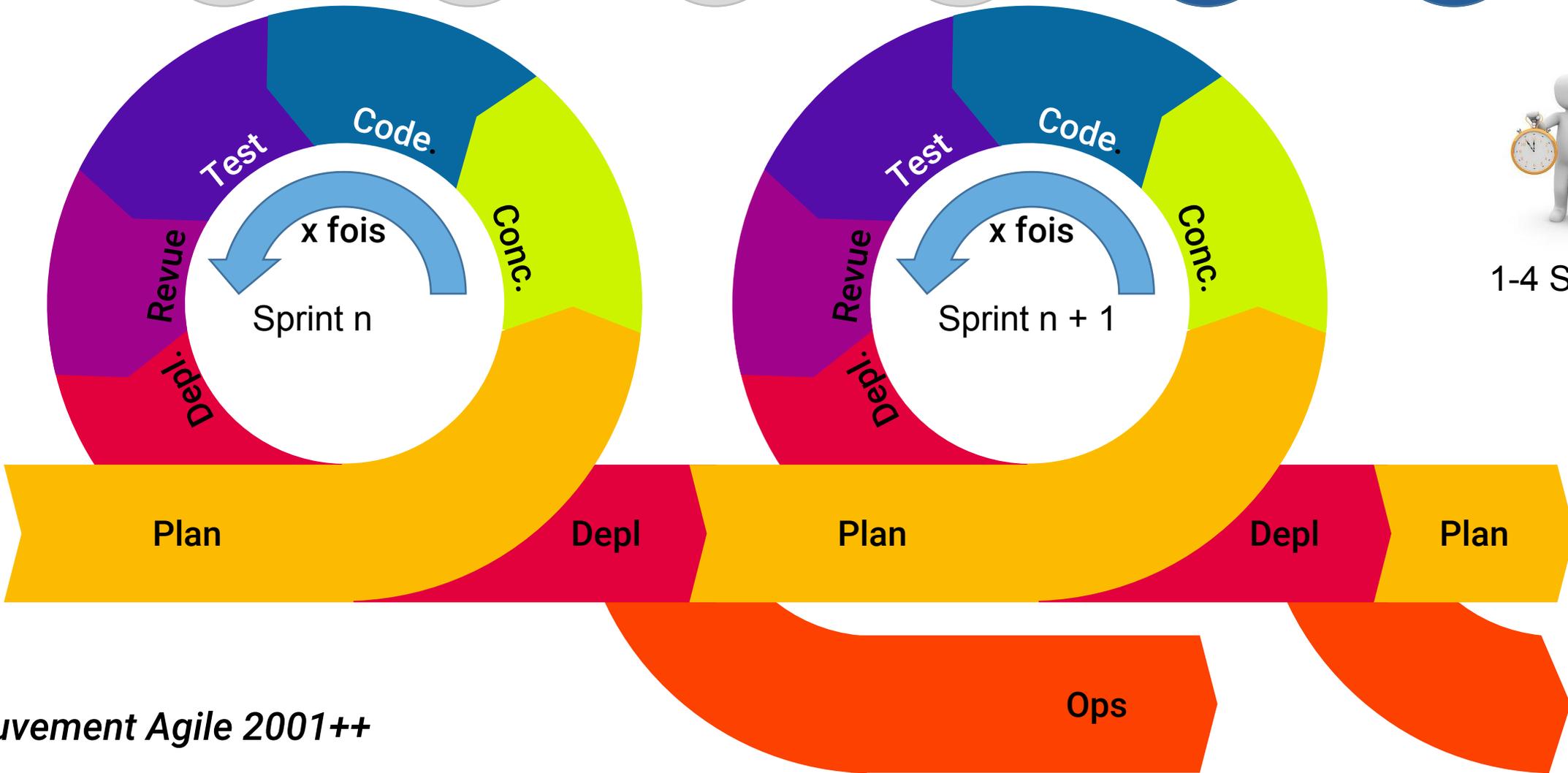
1965

1985

1999

2001

2009



1-4 Sem.

Mouvement Agile 2001++

1945

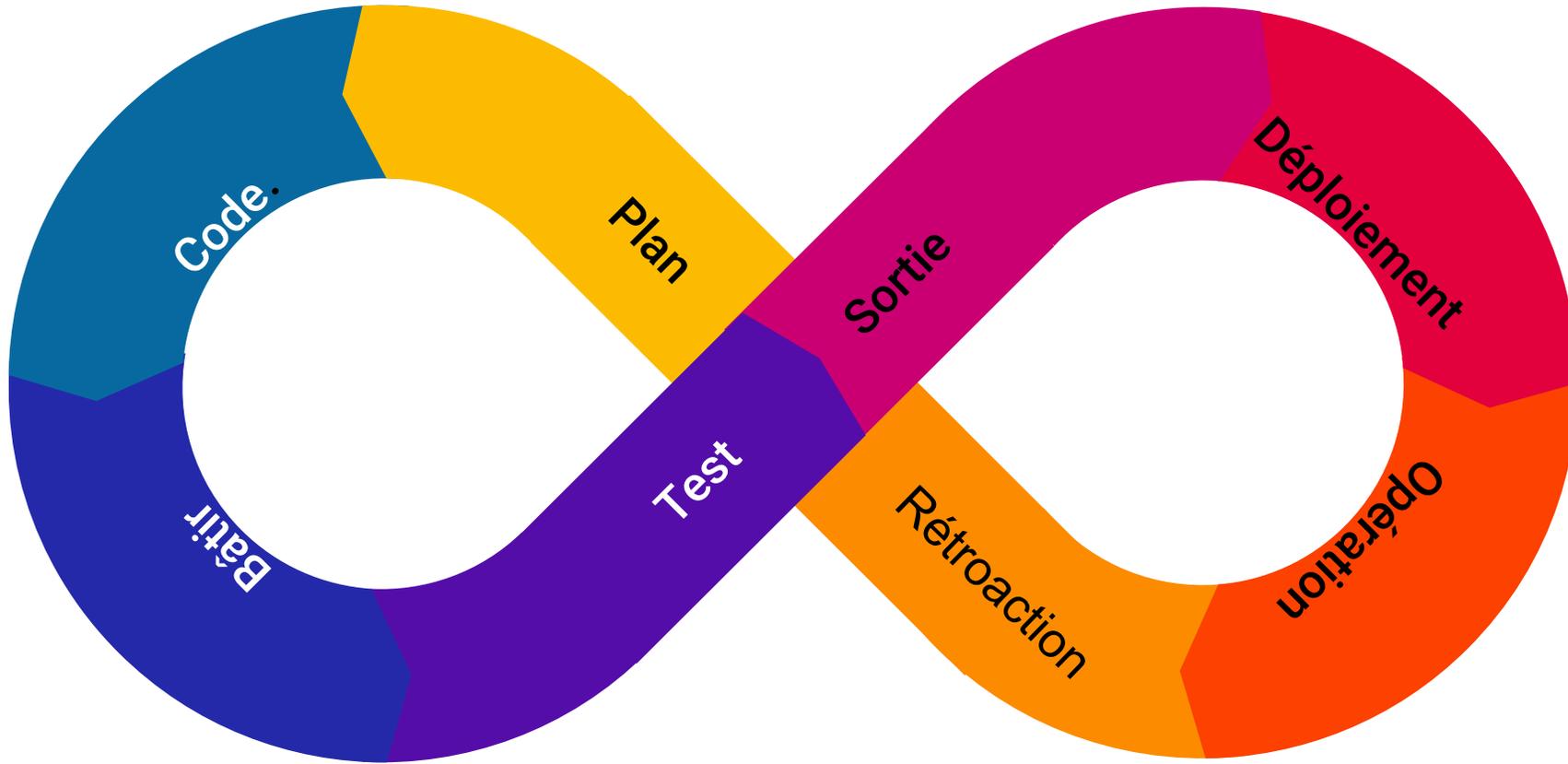
1965

1985

1999

2001

2009

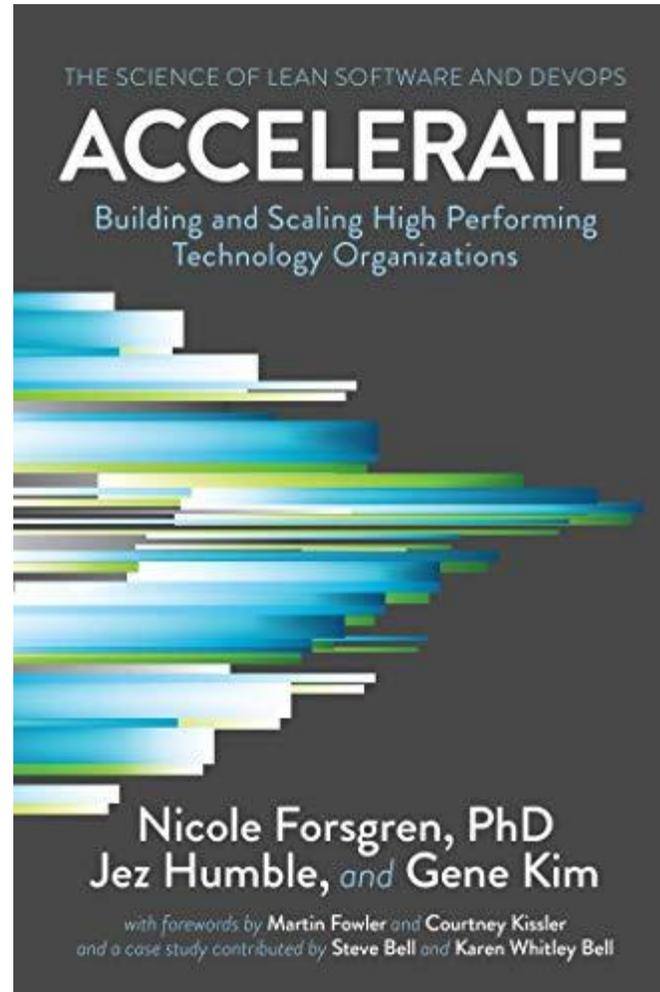


Continu

Mouvement DevOps 2009++

Le DevOps ca vaut le coût?





Comment se mesure la performance de livraison de logiciel?

Performance de livraison de logiciel

Délai de réalisation

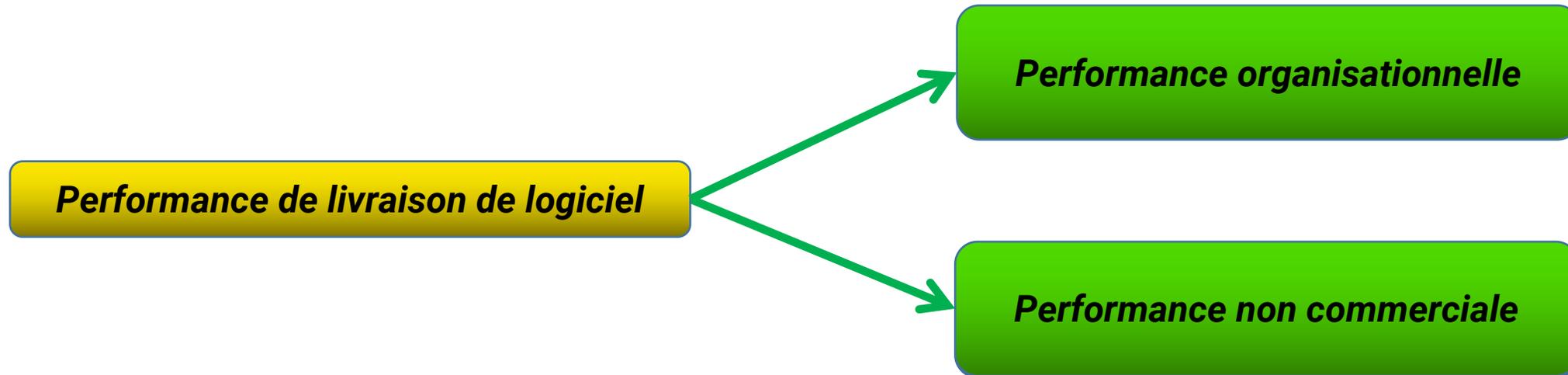
Fréquence de déploiement

Délai moyen de remise en service

Pourcentage de défaillances



Quel est l'impact sur l'organisation?



Par où commencer?

5 principes clés



Qualité



Petits lots



Automatisation



Amélioration continue



Responsabilité

24 capacités classées en 4 catégories



Techniques



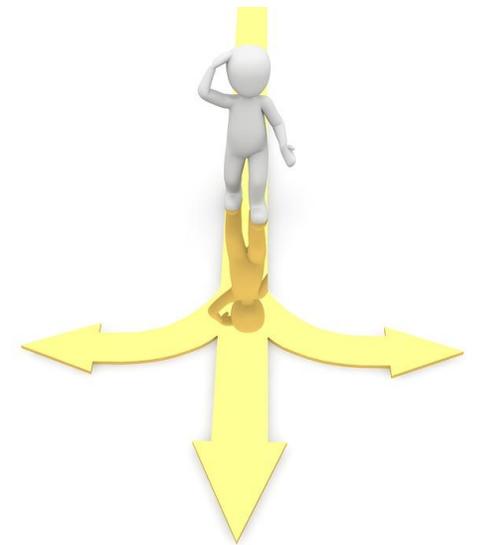
Lean Management



Développement de produit Lean



Leadership transformationnel



Capacités techniques



 **Automatisation des tests**

 **Automatisation du déploiement**

 **Développement sur le tronc**

 **Décalage à gauche de la sécurité**

 **Architecture découplée**

 **Équipes autonomes**

 **Intégration Continue**

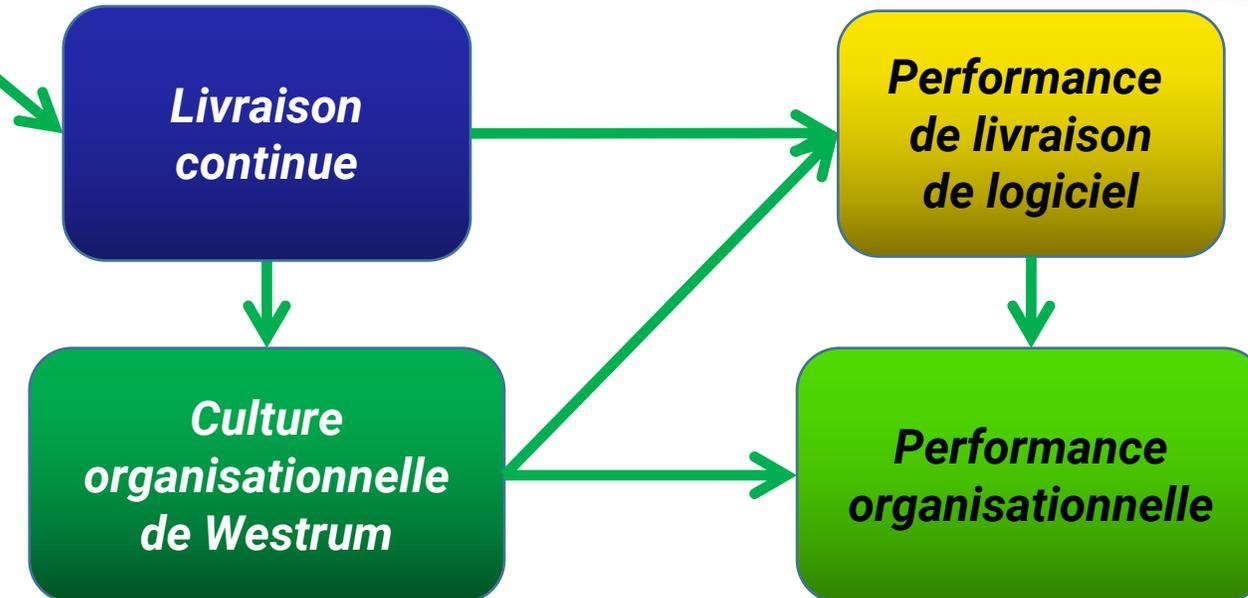
 **Contrôle des versions**

 **Gestion des données de test**

 **Surveillance**

 **Notification proactive**

Capacités techniques



Lean management



Limite des travaux en cours



Gestion visuelle



Rétroaction de la production



Approbation simple des modifications

**Culture
organisationnelle
de Westrum**

**Performance
de livraison
de logiciel**



Développement de produit Lean



Travailler en petits lots



Rendre le flux de travail visible



Recueillir et appliquer la rétroaction des clients



Expérimentation de l'équipe

Culture organisationnelle de Westrum

Performance de livraison de logiciel



Leadership transformationnel

Technique

Développement de produit Lean



Vision



Communication inspirante



Stimulation intellectuelle



Meneur supporteur



Reconnaissance personnelle



Anti-Patrons Culturels

Ce n'est qu'un sous-ensemble...

Anti-Patron Culturel

- *Solution qui semble bonne*
 - *En usage depuis plusieurs années*
 - *Ne convient plus ou pas à la réalité actuelle*

Anti-Patron No 1: Project Owner (au lieu du Product Owner)



Create your own at [Storyboard That](#)

Anti-Patron No 1: Project Owner (au lieu du Product Owner)

Conséquences

- *Solutions prédéfinies*
- *Retient la livraison*
- *Dissociation de l'équipe et de l'application à la livraison*



Expérimentation de l'équipe



Travailler en petits lots



Rétroaction des clients



Rétroaction de la production



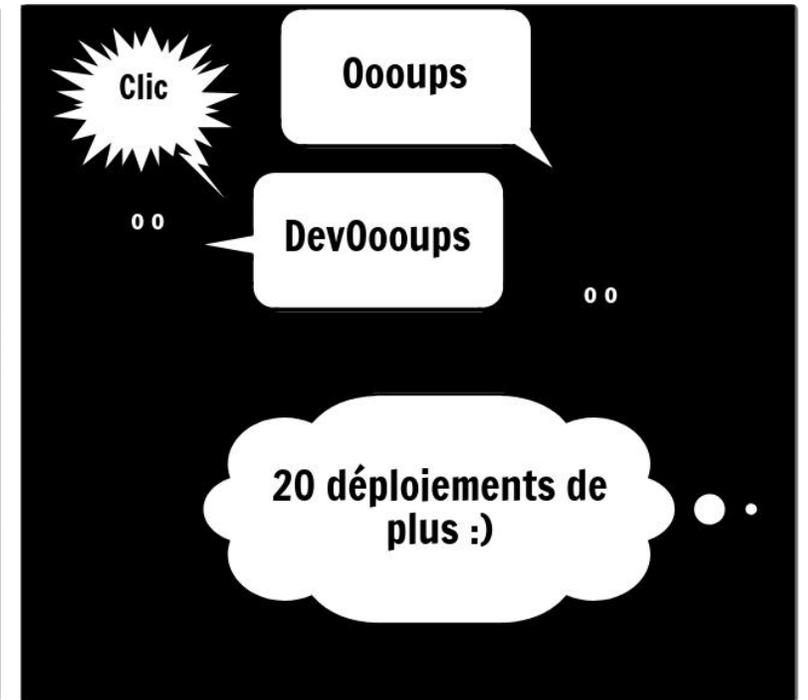
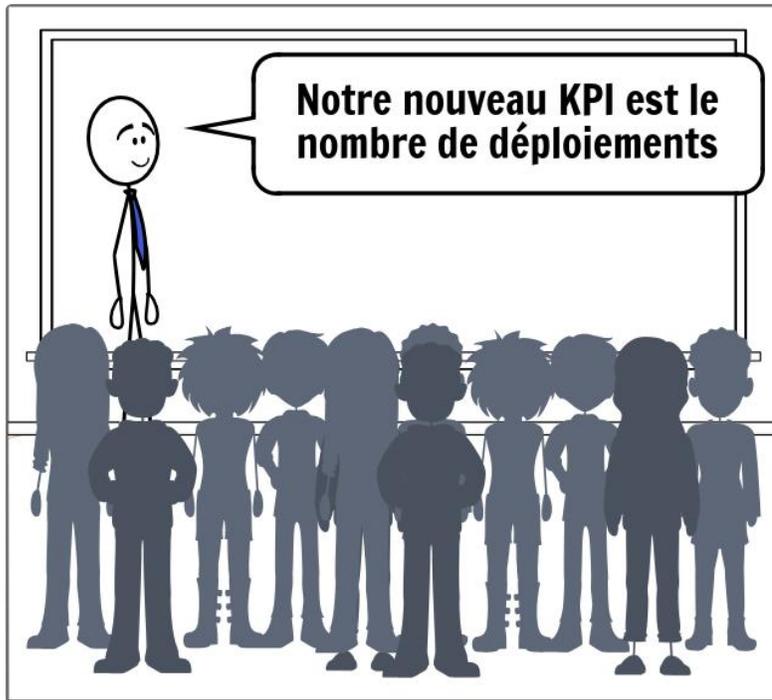
Responsabilité

Anti-Patron No 1: Project Owner (au lieu du Product Owner)

Solutions

- *Travailler sur la création d'un produit qui évoluera*

Anti-Patron No 2: Vite vite vite



Create your own at [Storyboard That](#)

Anti-Patron No 2: Vite vite vite

Conséquence

- *Met de coté les 5 principes clés*



Qualité



Petits lots



Automatisation



Amélioration continue



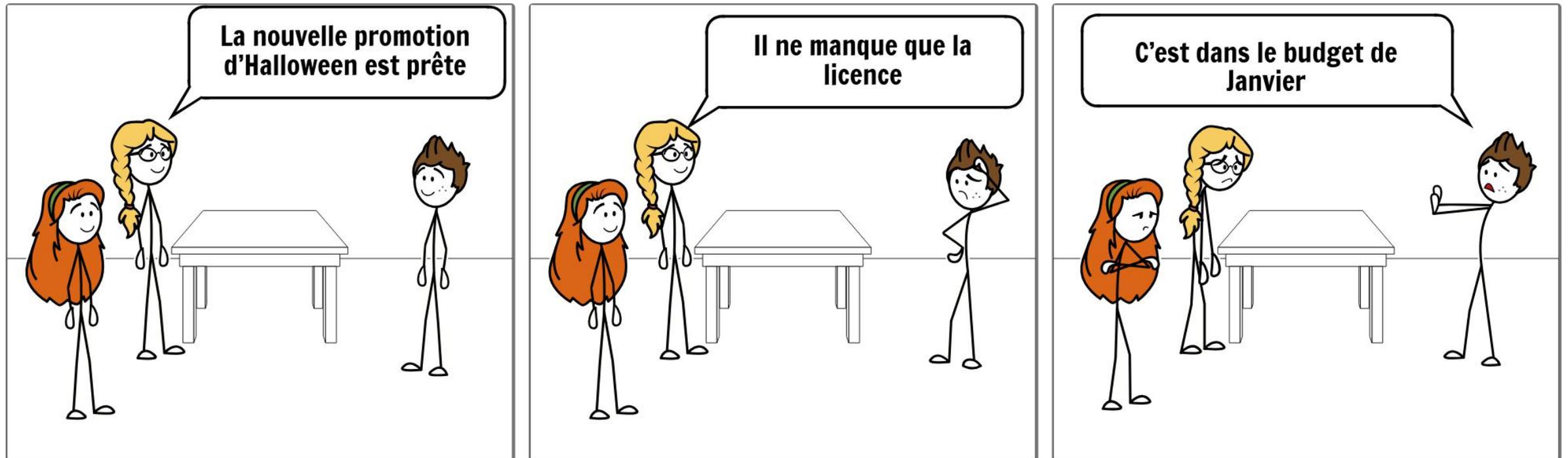
Responsabilité

Anti-Patron No 2: Vite vite vite

Solutions

- *Implémenter les 4 indicateurs clés*
- *Évaluer régulièrement les capacités DevOps*
- *Former les gestionnaires*

Anti-Patron No 3: Planification par budget (au lieu de par valeur)



Create your own at [Storyboard That](#)

Anti-Patron No 3: Planification par budget (au lieu de par valeur)

Conséquences

- *Le contrôle est parfois excessif*
- *Budget désaligné avec les besoins évolutifs*
- *Affecte les capacités inter-silo*
- *Mal appliqué, peut limiter la transparence*



Équipes autonomes



Expérimentation de l'équipe



Limite des travaux en cours



Gestion visuelle



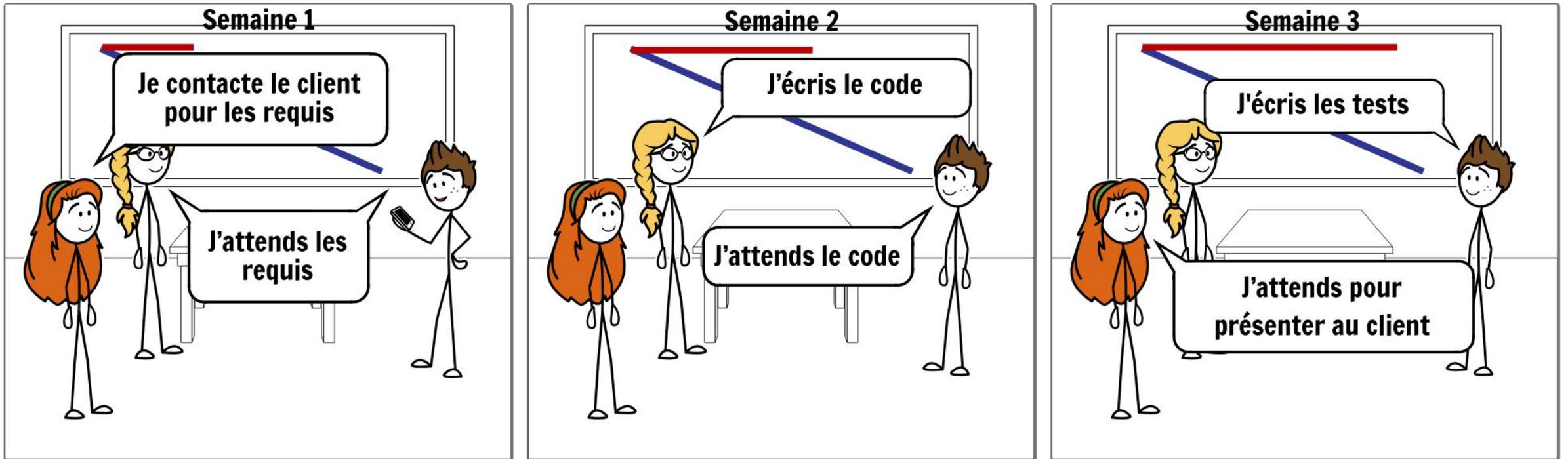
Flux de travail visible

Anti-Patron No 3: Planification par budget (au lieu de par valeur)

Solutions

- *Adopter des budgets leans*
 - *Financer la chaine de valeur*
 - *Ajuster l'horizon d'investissement*
 - *Employer un budget participatif*

Anti-Patron No 4: Mini Waterfall



Create your own at [Storyboard That](#)

Anti-Patron No 4: Mini Waterfall

Conséquences

- *Ajoute des transferts*
- *Retarde la rétroaction*
- *Validation déborde du Sprint*



Responsabilité



Rétroaction des clients



Flux de travail visible



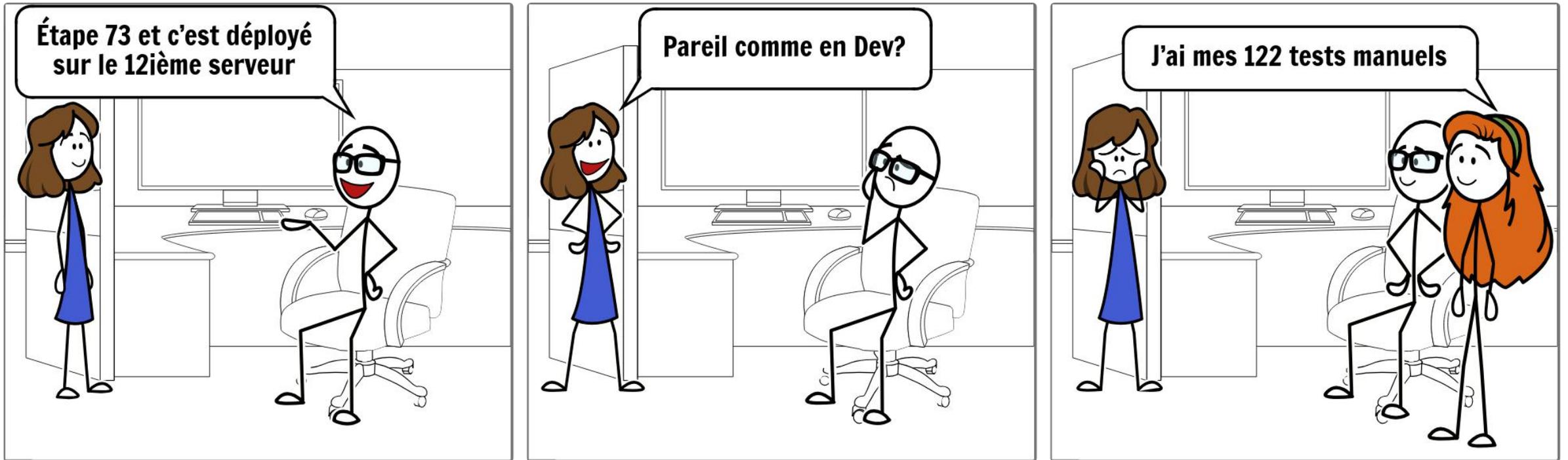
Limite des travaux en cours

Anti-Patron No 4: Mini Waterfall

Solutions

- *Intégrer le client dans l'équipe*
- *Décalage à gauche des test (ATDD, BDD, TDD)*
- *Envisager Kanban*

Anti-Patron No 5: L'artisan



Create your own at [Storyboard That](#)

Anti-Patron No 5: L'artisan

Conséquences

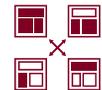
- *Limite l'automatisation*
- *"Snowflake Server" ¹*



Automatisation des tests



Automatisation du déploiement



Gestion des données de test



Contrôle des versions

Anti-Patron No 5: L'artisan

Solutions

- *Formation pratique*
 - *Capacités techniques*
- *Coaching*
 - *Sur la vision DevOps*

Anti-Patron No 6: Externaliser vers des artisans



Create your own at [Storyboard That](#)

Anti-Patron No 6: Externaliser vers des artisans

Conséquences

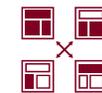
- *Limite l'automatisation*
- *Transfert de responsabilités*
- *Logiciel développé par un tiers*  *Performance¹*



Automatisation des tests



Automatisation du déploiement



Gestion des données de test



Contrôle des versions



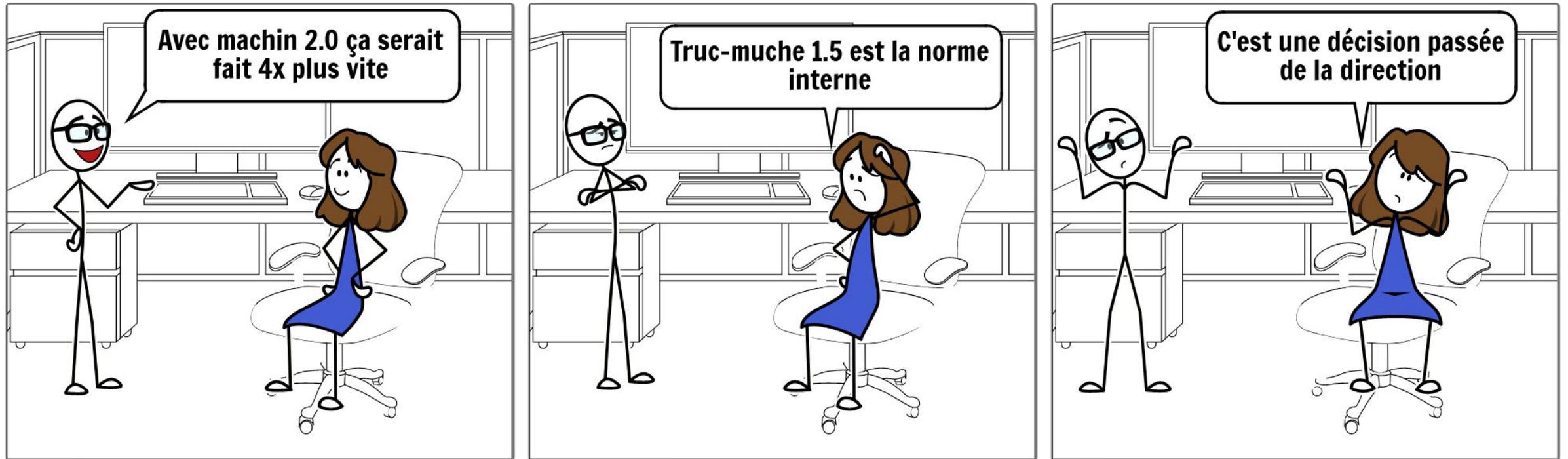
Responsabilité

Anti-Patron No 6: Externaliser vers des artisans

Solutions

- *Spécifier par contrat les tests et l'automatisation*
- *Développer les projets critiques à l'interne*

Anti-Patron No 7: Hiérarchie de contrôle



Create your own at [Storyboard That](#)

Anti-Patron No 7: Hiérarchie de contrôle

Conséquences

- *Communication entre départements*
- *Décisions bureaucratiques plus lentes*
- *Adaptation au changement difficile*
- *Reconnaissance va aux gestionnaire*
- *Autorité forte*



Limite des travaux en cours



Approbation simple



Vision



Reconnaissance Personelle



Meneur supporteur

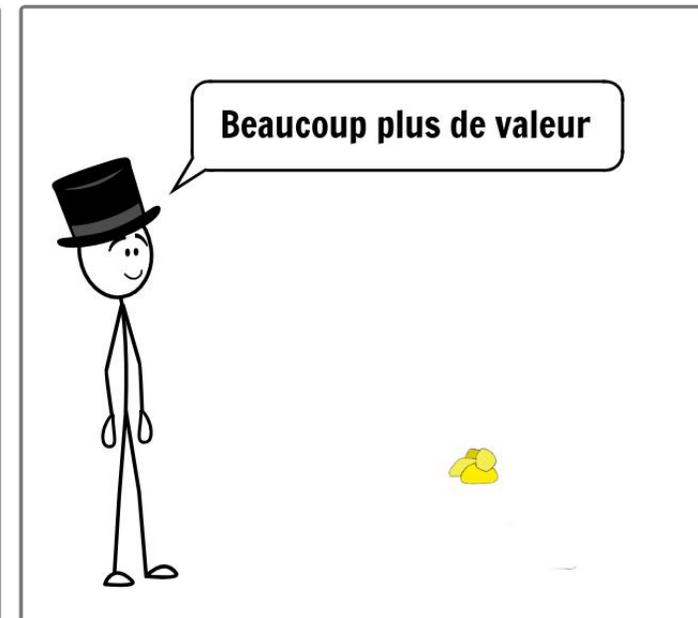
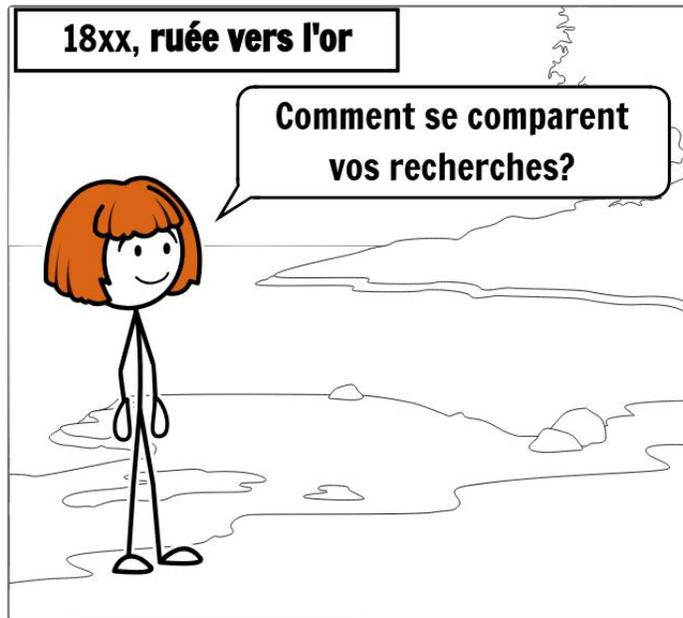


Équipes autonomes

Anti-Patron No 7: Hiérarchie de contrôle

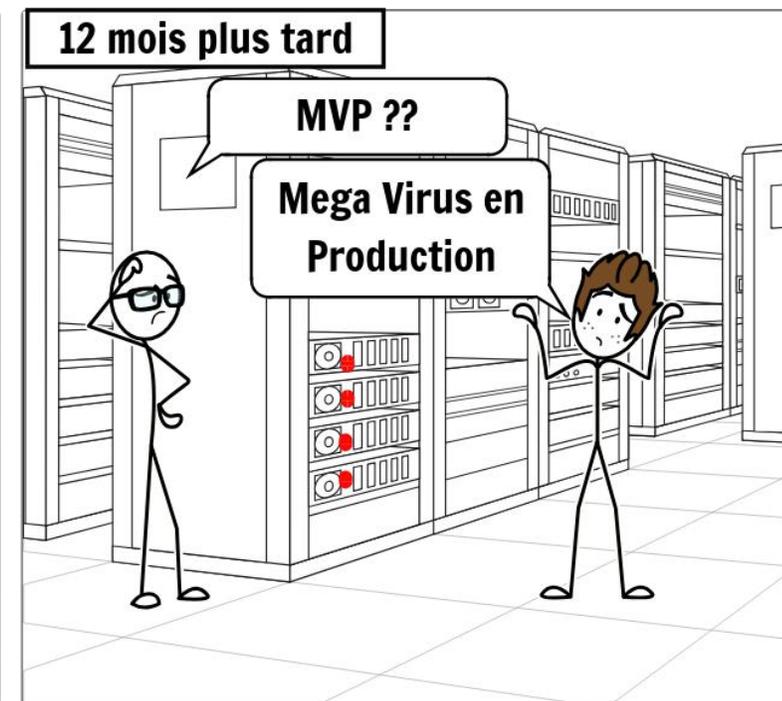
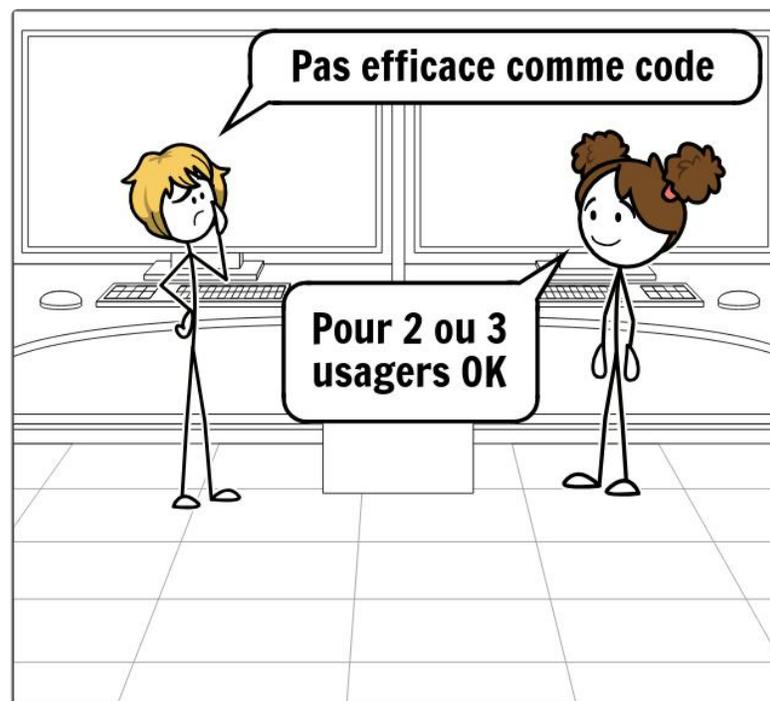
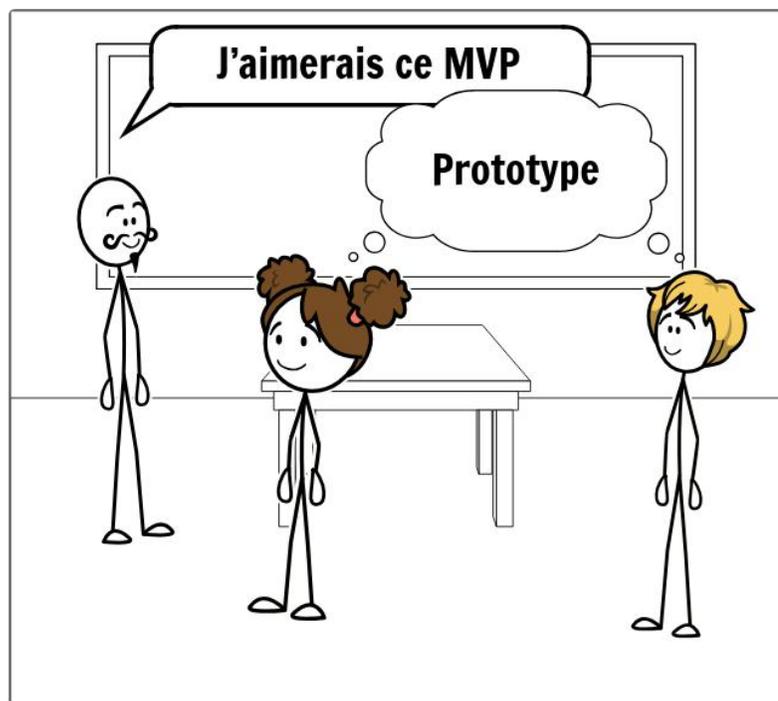
Solutions

- *Équipes multidisciplinaires*
- *Donner la récompense selon la valeur d'affaire produite*



Create your own at [Storyboard That](#)

Anti-Patron No 8: Le MVP va en prod (pour y demeurer)



Create your own at [Storyboard That](#)

Anti-Patron No 8: Le MVP va en prod (pour y demeurer)

Conséquences

- *Mauvais usage du MVP*
- *Escamotage des 5 principes clés*

 *Rétroaction des clients*

 *Expérimentation de l'équipe*

 *Rétroaction de la production*

 *Qualité*

 *Petits lots*

 *Automatisation*

 *Amélioration continue*

 *Responsabilité*

Anti-Patron No 8: Le MVP va en prod (pour y demeurer)

Solutions

- *Employer un MVP pour expérimenter*
- *Employer un MMP pour la production*

Anti-Patrons Culturels

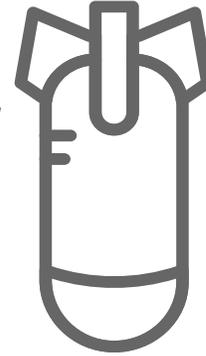
1. *Project Owner (au lieu du Product Owner)*
2. *Vite vite vite*
3. *Planification par budget (au lieu de par valeur)*
4. *Mini Waterfall*
5. *L'artisan*
6. *Externaliser vers des artisans*
7. *Hiérarchie de contrôle*
8. *Le MVP va en prod*

Conclusion

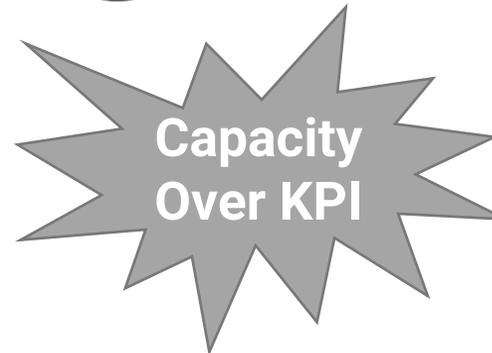
- *Aux 8 premiers anti-patrons, on ose ajouter*

- *La fausse Agilité*

- *Scrum...*



- *Miser sur les capacités supportant DevOps*



- *Mesurer correctement les résultats*



- *5 principes clés à surveiller*



Pour la suite

<https://devopsantipatterns.ninja/fr>



Questions



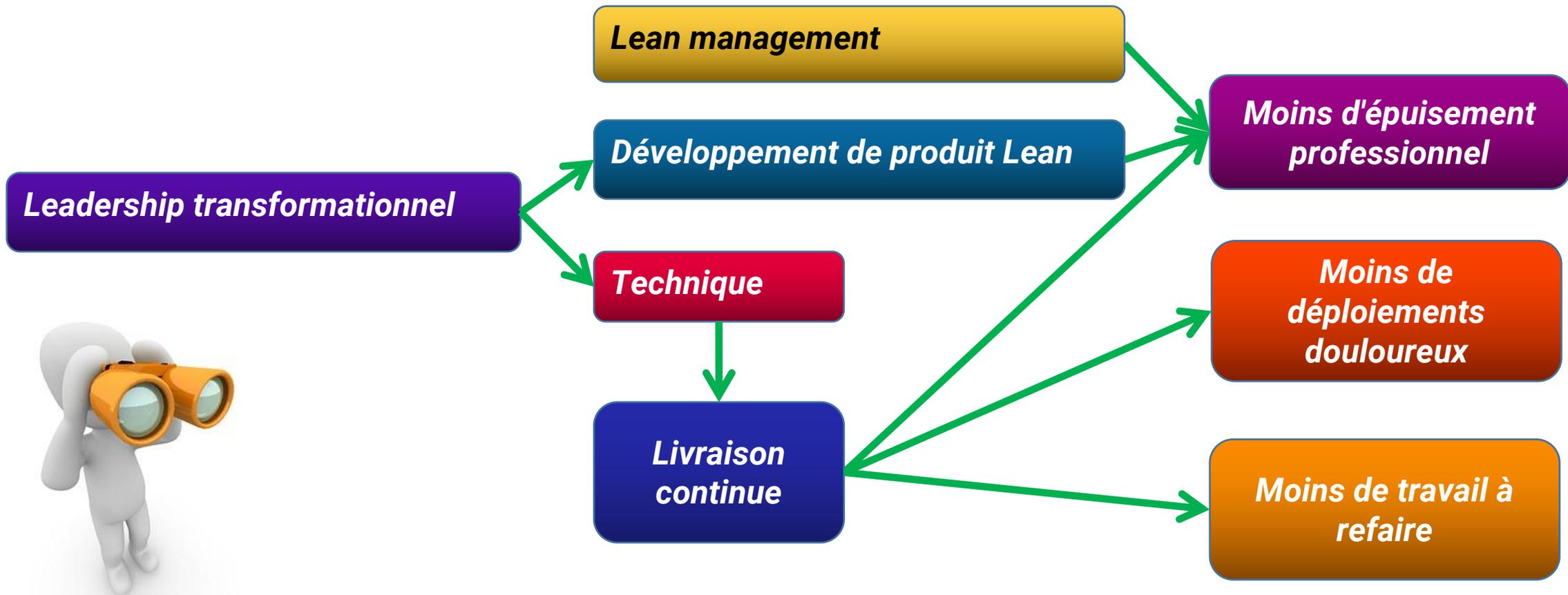
Annexes

Informations supplémentaires

Bénéfices collatéraux



Pérennité



Appartenance organisationnelle

