301 - La pensée DevOps

DEVOPS

WIK-DEVOPS-301

Intervenant : Jeremy Trufier <jeremy@wikodit.fr>

ynov

wikodit

WIK-DEVOPS

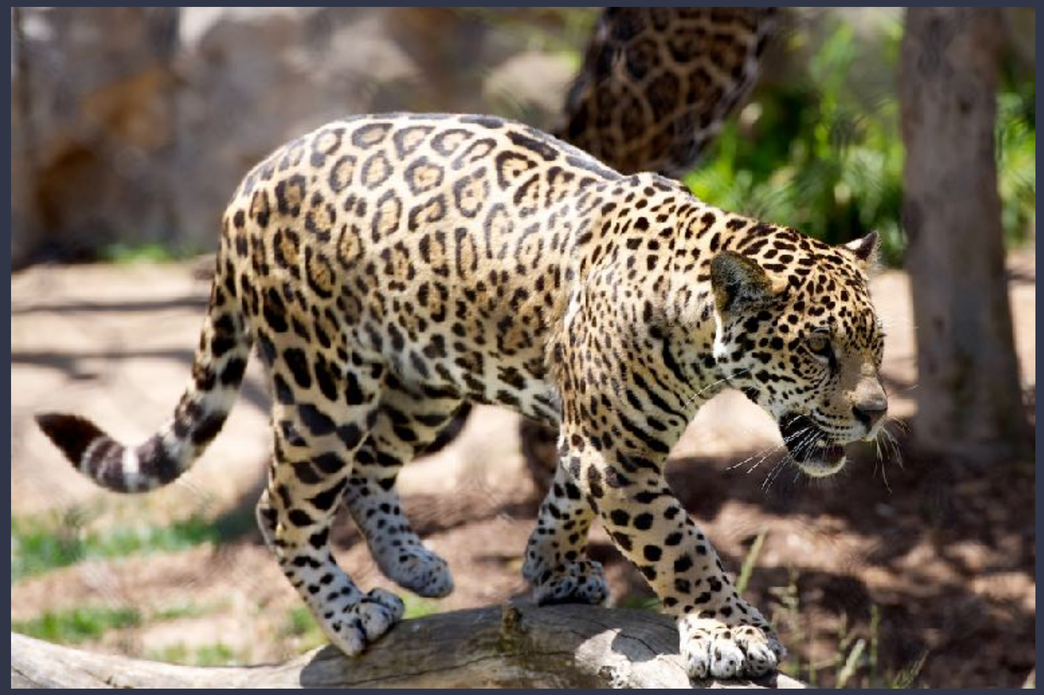
Programme DEVOPS

301 – Introduction, Kaizen et CAMS	1 séance
302 – La culture DevOps	1ch / séance
303 – Automatisation / Déploiement continu	4 séance
304 – Mesure & Monitoring	2 séance
305 – Partage	2 séance

I. Introduction

DevOps est agile

- Remise en cause permanente
- Marketing, R&D, développement, ...
- Le DevOps "agilifie" les équipes de production



Une vision positive de l'échec

- Learn Fast, Fail early
- Pas de succès sans risques
- Savoir évaluer les menaces
- Savoir changer de plan rapidement

La démarche DevOps

- Collaboration agile pour construire la confiance
- L'automatisation pour pérenniser la confiance
- L'industrialisation pour tendre vers le continuous delivery

DevOps et le Cloud

- Évolution vers le PAAS
- Évolution du rôle d'opérationnel
- Centré sur l'application

Infrastructure As Code

DevOps et le Cloud : Évolution de l'architecture

- découper l'application en composant fonctionnels
- définir un modèle du comportement de l'application en production
- définir un modèle de disponibilité pour chaque composant
- Identifier les points et les modes de défaillance

DevOps et mobilité

- les stores applicatifs : hyperconcurrentiel
- applications multi-plateforme
- plus de frontières particulier / pro
- une vraie course au besoins

DevOps et l'innovation : Design Thinking

- Empathie (DO, THINK, FEEL, SAY)
- Définir
- Imaginer
- Prototyper
- Tester

II. Le modèle CAMS

CAMS

- DevOps Café : John Willis & Damon Edwards
- Les pilliers :
  - **C**ulture
  - **A**utomation / Automatisation
  - **M**easurement
  - **S**haring / Partage

Culture

C'est quoi ?

- Culture > Stratégie
- Indispensable pour le succès de la société
- Aucune question, ça fait partie de la culture
- Les outils ne font pas tout

"Culture eats strategy for breakfast"

Culture

Comment ?

- Service ownership
- S'occuper de la dette technique
- Testing destructif
- Équipes multi-fonctions
- Agile

La culture ne se copie pas mais se comprend et peut être adaptée !

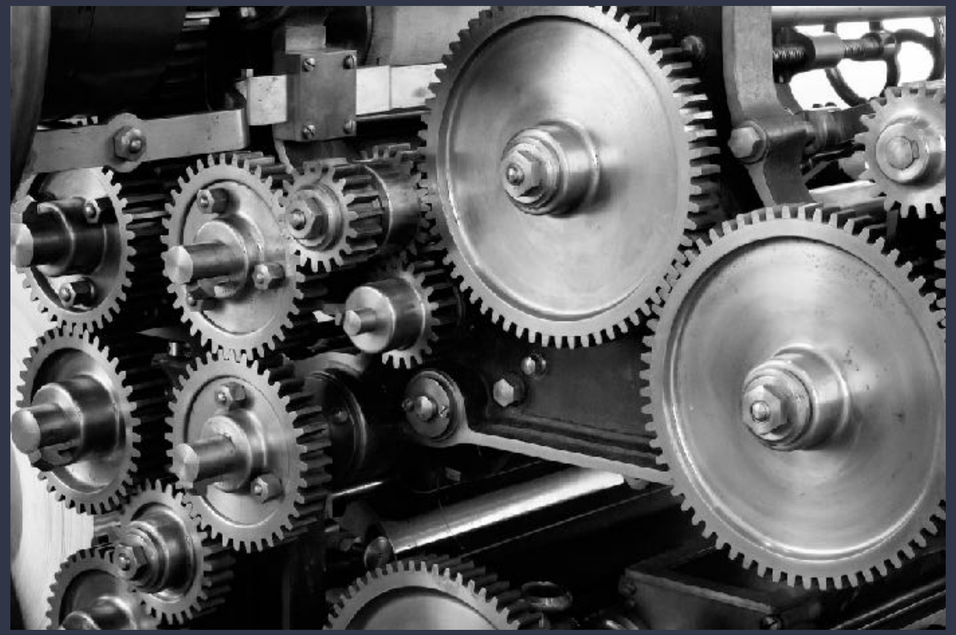
Automation

Pourquoi ?

- DevOps = Rapide : Produit rapide et fonctionnel
- Embaucher plus de gens ?
  - Couteux
  - Pas forcément plus productif
  - Erreurs humaine

Automation

Comment ?



- Le travail manuel devient automatique
- Consistance, Rapidité, Sécurité
- Infrastructure as Code (Configuration Management)
- CD / CI pipeline
- Orchestration Engine

Measurement

Pourquoi ?

- Comment savoir si on s'améliore ?
- Est-ce que ce qu'on applique est utile ?
- Est-ce qu'aujourd'hui et mieux que hier ?
- Est-on sur le bon chemin ?

"If we have data, let's look at data. If all we have are opinions, let's go with mine." – Jim Barksdale, Netscape CEO

Measurement

Quoi ?

- Productivité
- Monitoring : Statistiques, temps de réponse, ...
- Business : Acquisition, revenue, clients qui payent, ...
- Support : combien, coût, ...

"If it move, measure it !"

Measurement

Comment ?

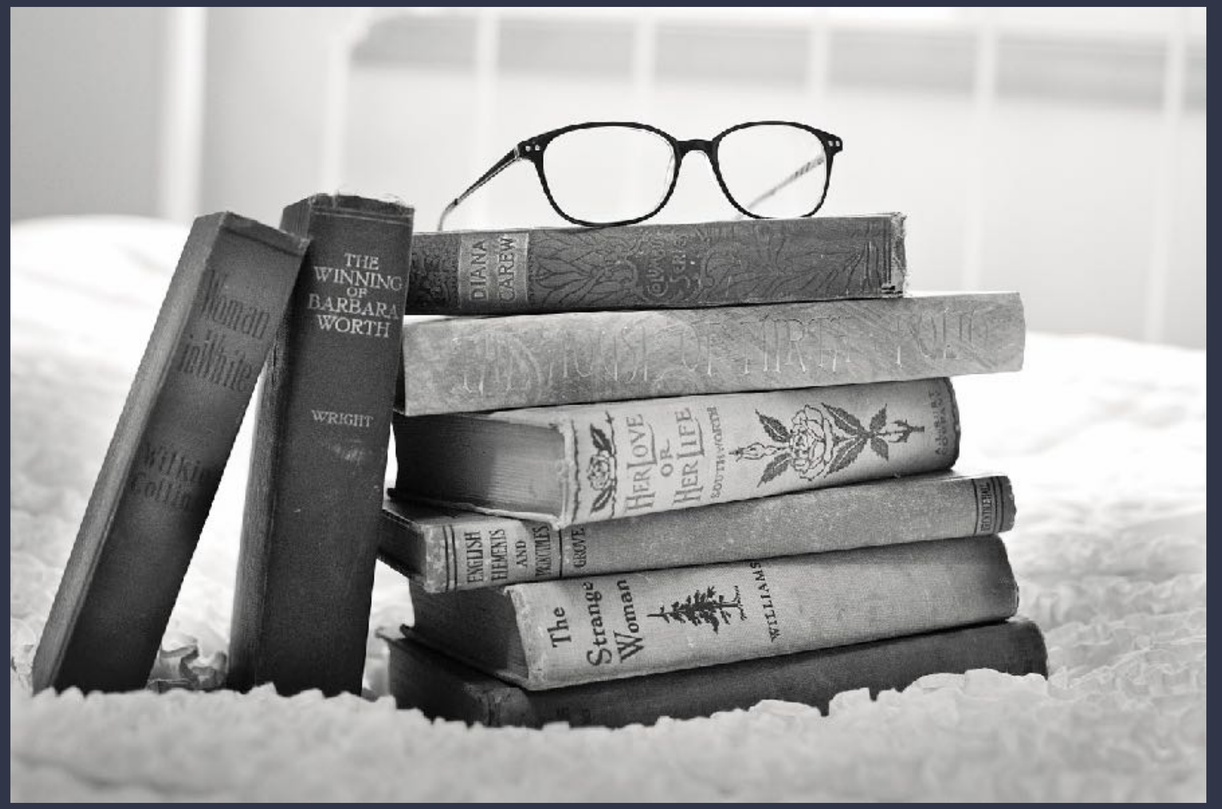
- Soft de monitoring (réseau, système)
- Analytics
- "Information radiators"
- Graphiques Agiles

Sharing
Pourquoi ?

- Early feedback
- Visibilité en temps réel (What)
- Transparence : même objectif (Why)
- Transfert de connaissance

Sharing
Comment ?

- Daily standup
- Rétrospective => Kaizen
- **Documentations !!!!!**
- Tech talk, conférences
- ChatOps



III. Le principe de Kaizen

Inspiration : Lean Startup

- Cycle de commercialisation de produit
- Design itératif
- Intégration du retour client et utilisateur
- Tester le marché

L'objectif de l'entreprise

- Donner au client ce qu'il veut
- Quand il le veut
- Et au meilleur prix !

Le problème

- Cassure entre le monde du dev et de l'ops
- Le cauchemar du management
- Les limites du Plan-Do-Study-Act
- Chacun travaille individuellement

En bref : Aucune visibilité !

Kaizen

Changement Bon

- Amélioration continue
- Approche scientifique
- Équipes impliquées et engagée
- Les changements infimes sont très important
- Chaque acteur propose des optimisations

Amélioration constante du flux de travail

Objectif : le client

Comment l'appliquer ?

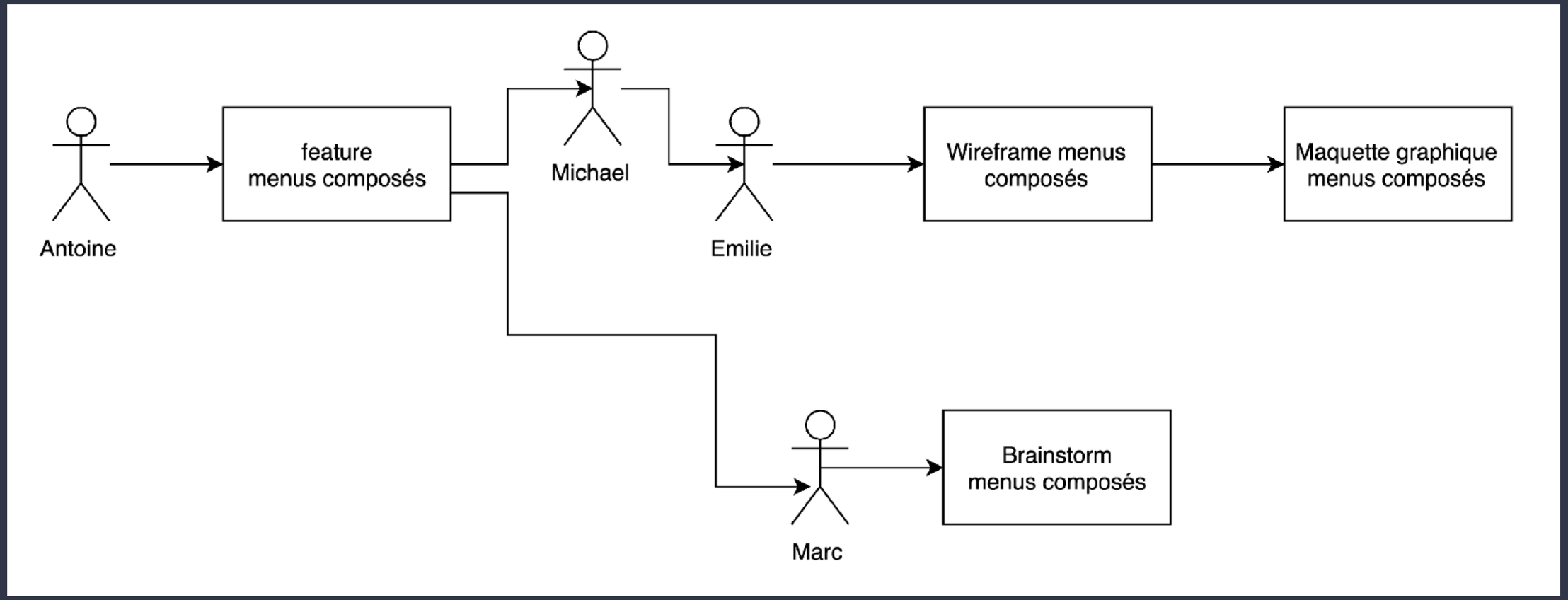
1. Modéliser tous les process

- Temps de gestion
- Temps de traitement
- Nombre d'acteur
- Scrape Rate : rapport de travail utile / inutile

Vision d'ensemble claire

Comment l'appliquer ?

1. Modéliser tous les process



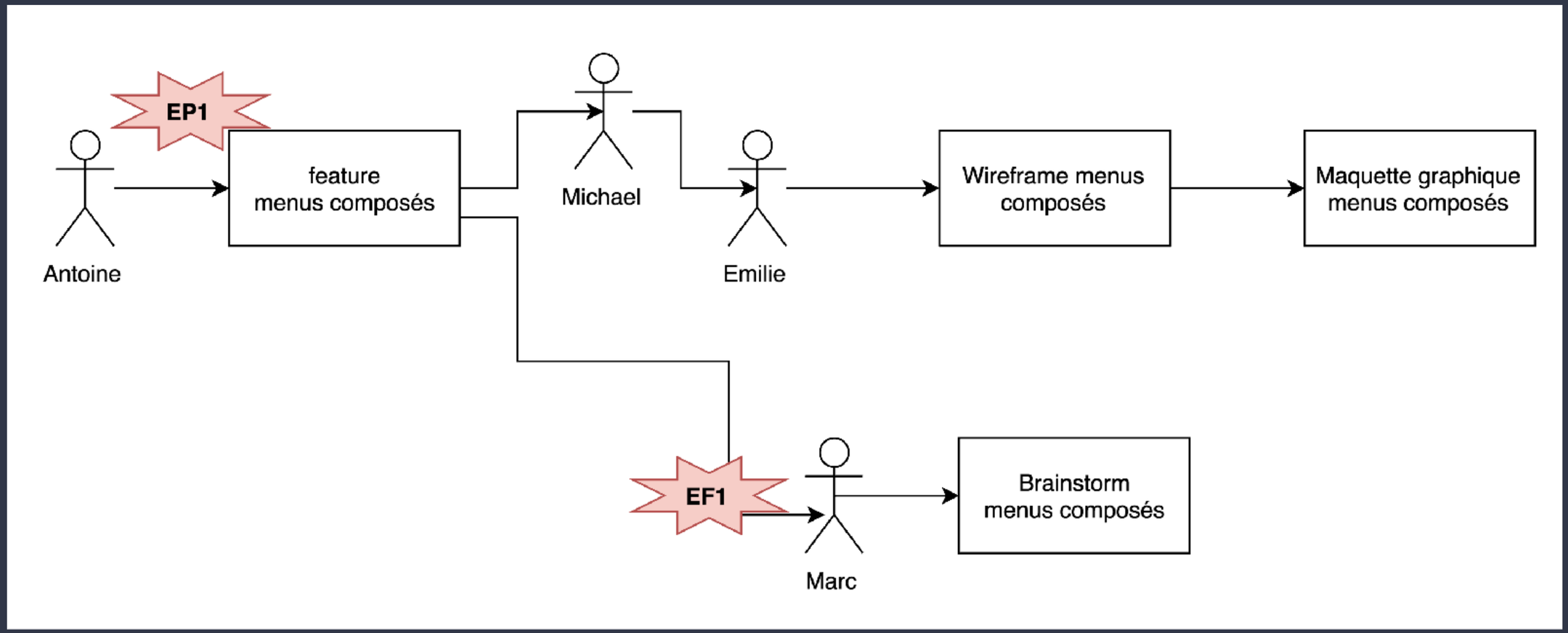
Comment l'appliquer ?

2. Identifier les Deadly Waste (lean)

- Defects / rework - Travail qui ne fonctionne pas ou pas correctement
- Extra Process - Activité en dehors du scope du projet / pas de valeur ajoutée
- Extra Features - Feature inclues trop tôt ou inutiles
- Task Switching - Trop de travail en parallèle
- Partialy done - Travail qui devait être fini, mais ne l'est pas
- Motion / Manual - Intervention manuelle sans valeur ajoutée / déplacements peu utile
- Waiting - En attente de paiement, d'informations supplémentaires, etc...
- Heroic - Prise d'initiative sans valeur ajoutée ou dont personne n'est au courant

Comment l'appliquer ?

2. Identifier les Deadly Waste (lean)



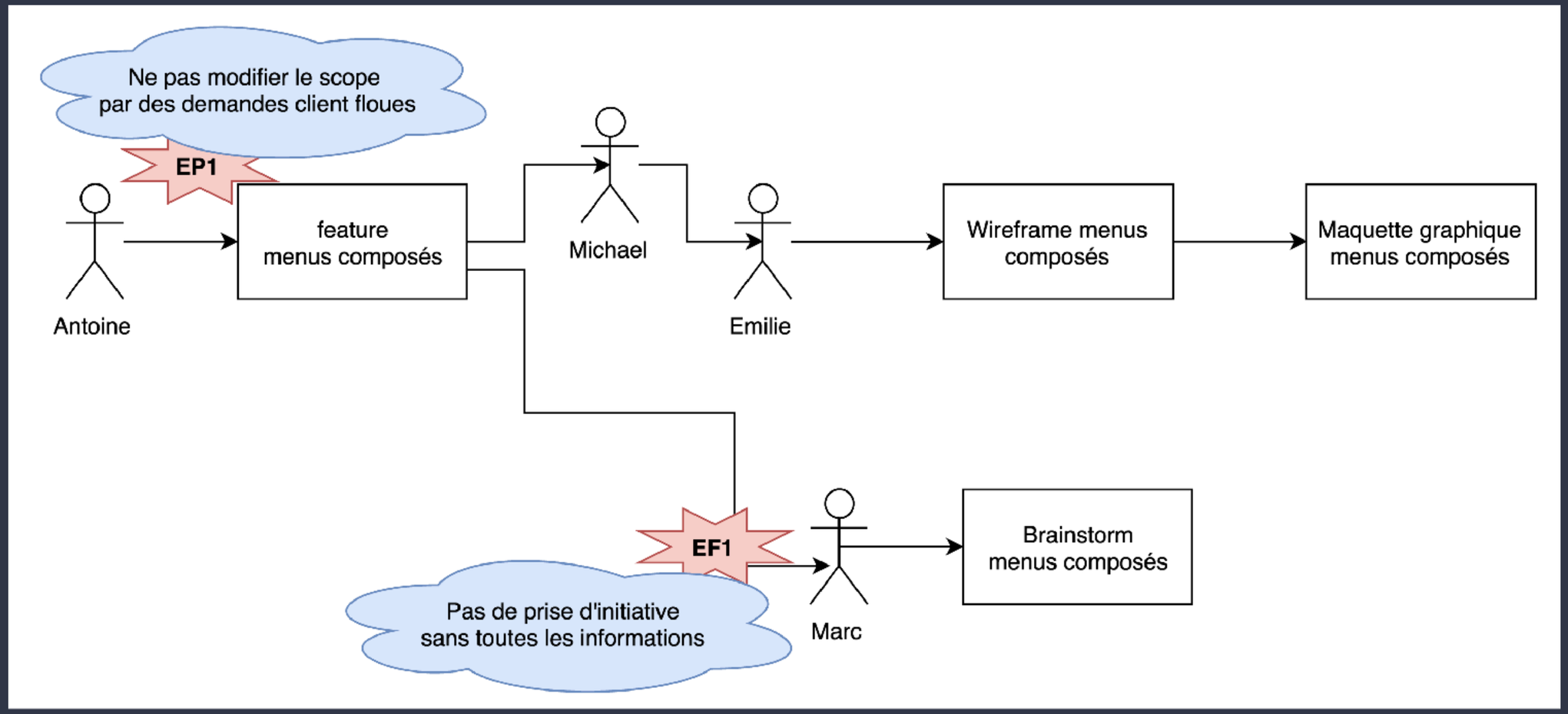
Comment l'appliquer ?

3. Réfléchir aux contres-mesures

- Dans le meilleur des cas : Applicable dès aujourd'hui
- Applicable pour le projet en cours
- Éventuellement infimes (Baby-Steps)
- Stratégique ou seulement recommandations

Comment l'appliquer ?

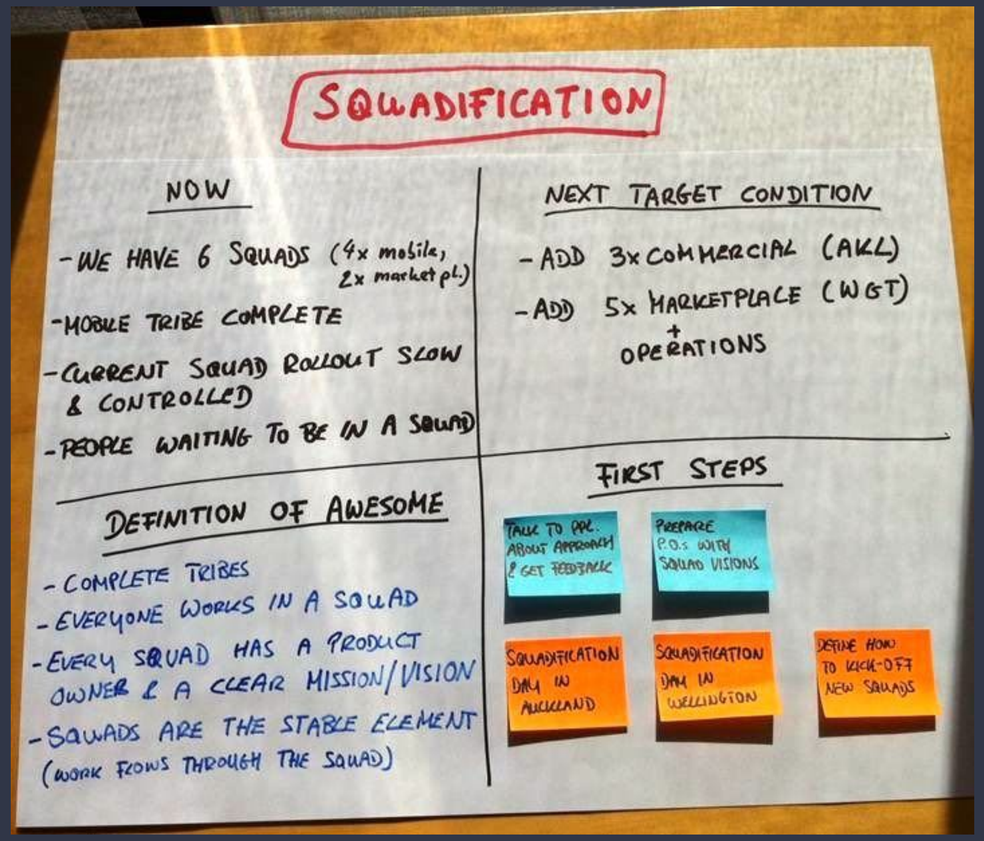
3. Réfléchir aux contres-mesures



Allez plus loin

4. Suivre et orienter

- Créer des fiches de challenges
- Model : Toyota Kata
- Implémentation itérative



Les bons comportements

- Learn Fast, Fail Early
- Organiser le travail autour de tâches petites et précises
- Améliorer la visibilité
- Automatisation

TD : La technique de rétrospective Kaizen

Félicitations !!

Cours WIK-DEVOPS-301 burned :)