



# B3

## Méthodologie Agile

Thibault Vinchent

# Méthodologie Agile

## Programme des 13 heures

- Cours 1 :
  - Bases de l'agilité
  - Agile avancé
- Cours 2 :
  - Atelier : Création d'un projet en mode Agile : par équipe de 3, 4 max

# Agile

Introduction : le manifeste Agile : <https://manifesteagile.fr/>

Valoriser :

Les individus et leurs interactions, de préférence aux processus et aux outils,  
Des solutions opérationnelles, de préférence à une documentation exhaustive,  
La collaboration avec les clients, de préférence aux négociations contractuelles,  
La réponse au changement, de préférence au respect d'un plan.

# Agile

## Cadre d'utilisation

Utilisé pour tout type de projet,

Du FBI aux agences de pub en passant par le génie civil,

Très utilisés dans les projets informatiques.

# Agile

## Cadre d'utilisation

S'utilise sans certification, il faut juste connaître les méthodes et le jargon

“Scrum, c'est comme le poker, vous apprenez les règles en 10 minutes, mais vous mettez du temps pour devenir vraiment bon”.

Agile

Valeurs de l'agile

Collaboration

Autonomie

Transparence

# Agile

## Le cadre de travail

Scrum: framework pour implémenter la méthodologie agile (mise en application d'agile)

# Agile

## L'équipe

Scrum master: lead dev qui s'assure que la méthodologie scrum est respectée

Product owner: moa internalisé, représente les intérêts du client

Différence avec une équipe classique: MOE, MOA

# Agile

## L'organisation

Daily scrum ou daily stand up journalier:

15 min max (en général le matin à la 1ere heure) pour faire part de:

- son avancement (ce qu'il a fait la veille)
- de ses points de blocage (ses demandes d'aide éventuelles)
- le reste devra être développé en dehors de la réunion (qui ne doit pas durer plus de 15 minutes généralement)
- de ses tickets à venir

# Agile

## L'organisation

Poker planning: pour chaque tâche à effectuer, chaque membre de l'équipe affecte le niveau de difficulté qu'il estime le sien à réaliser la tâche.

- Ce niveau de difficulté se fait avec des cartes de fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21

# Agile

## Cycle de travail

- Sprint: de 2 heures à 1 mois max, définition des tâches qui doivent être accomplies (différence avec le cycle en V)
- Rétrospective (à la fin du sprint): retour d'expérience et amélioration possible

# Agile

## Outils

- Trello: backlog: liste de taches / cahier des charges **PRIORISÉ**
- Git: code collaboratif

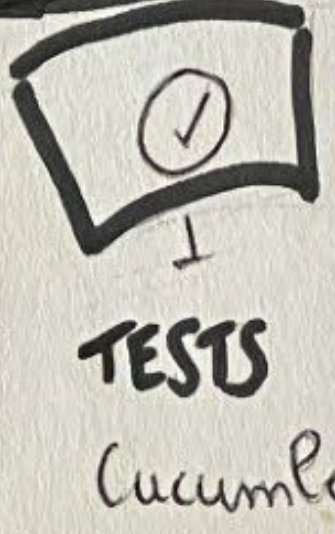
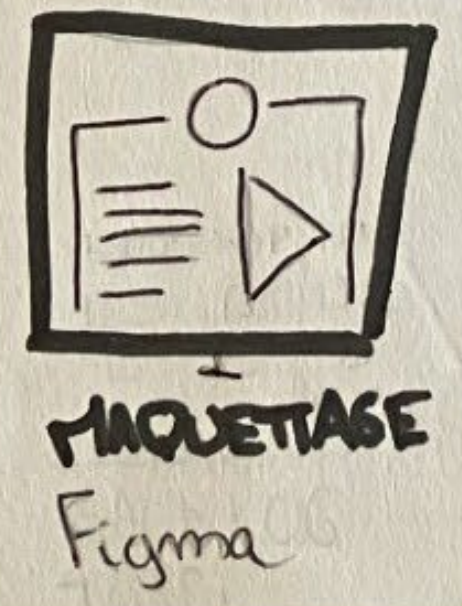
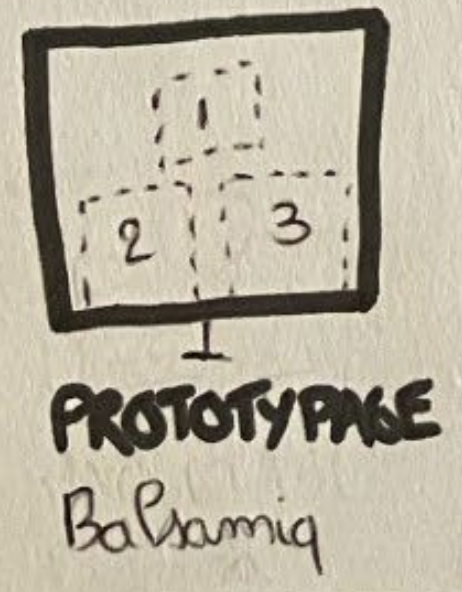
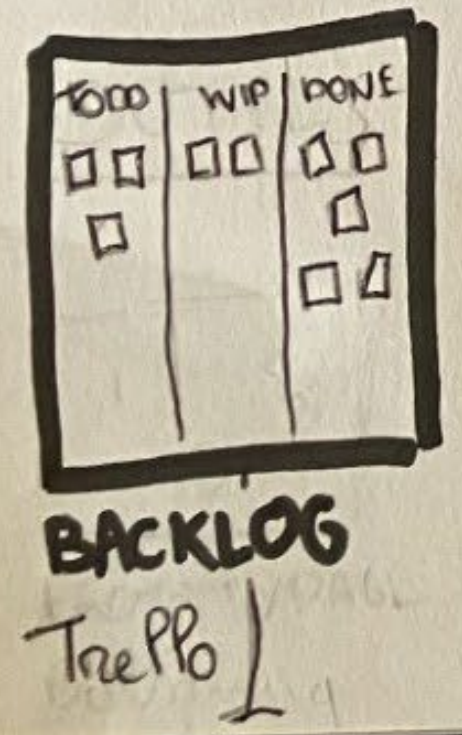
# Methode **AGILE**

## VALEURS

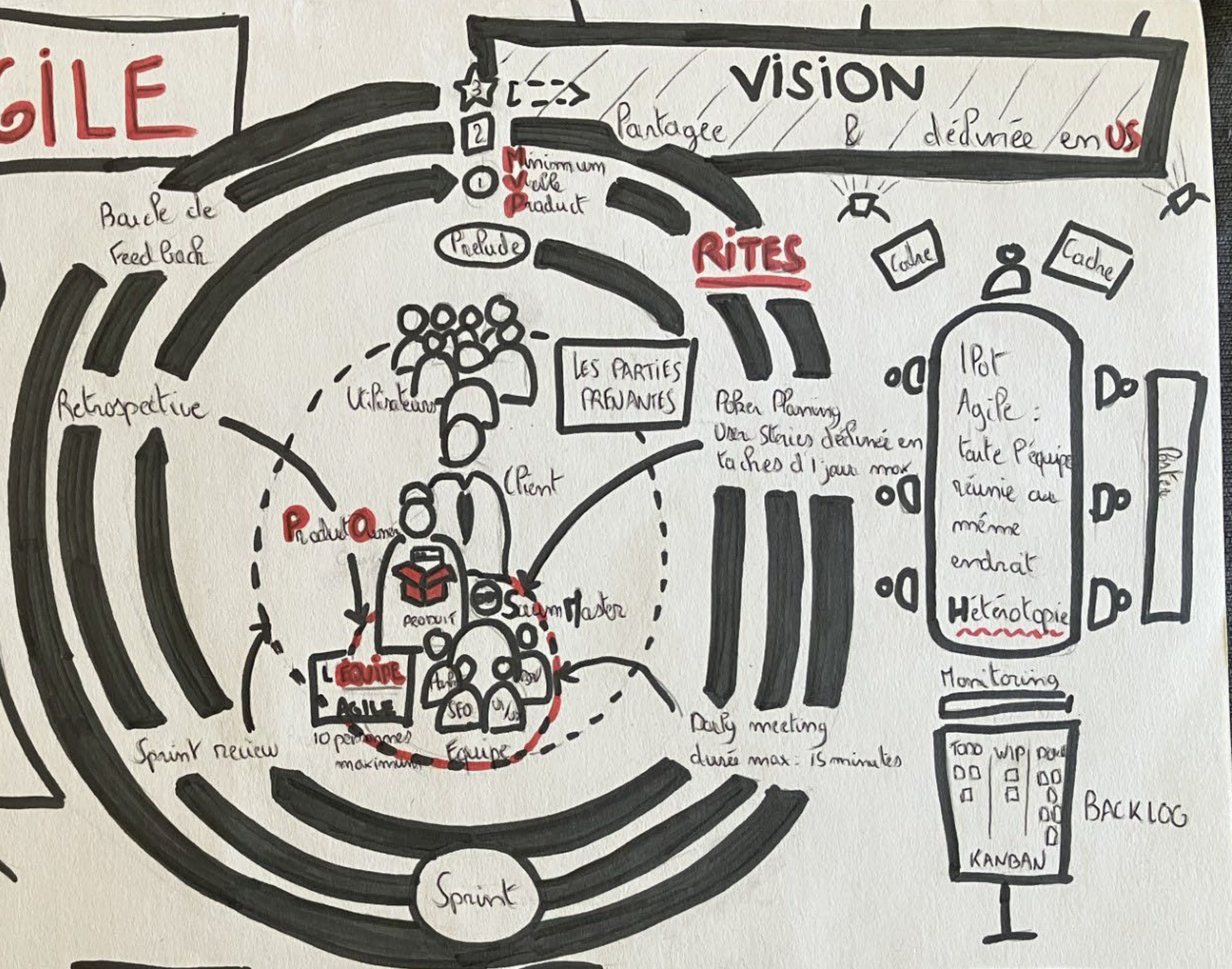
- Collaboration
- Autonomie
- Transparence

Evaluation de la performance collective plutôt qu'individuelle: calcul de la vélocité

## OUTILS



# Goweb.



**VISION**

Partagée & déclinée en US

Minimum Viable Product

Prelude

**RITES**

LES PARTIES PREJANTES

Sprint Planning  
User stories déclinées en tâches d'1 jour max

Pot Agile :  
toute l'équipe réunie au même endroit  
Hétérotopie

Monitoring

BACKLOG

KANBAN

Sprint

Sprint review

Retrospective

Boucle de Feed Back

Product Owner

Scrum Master

LES EQUIPES AGILES

10 personnes maximum

Equipe

Stakeholders

Client

Utilisateurs

Stakeholders

Cache

Cache

Cache

Cache

Cache

Cache

Cache

Cache

Cache

# Agile avancé

## L'organisation Agile

Reprise du schéma. (Kanban, Backlog, Sprint, Poker planning, Daily meeting, Retrospective, MVP). Ajout de :

User Story : fonctionnalité correspondant à un besoin client

Epic : rassemblement de US

Grooming : affiner et prioriser les User Story

Backlog : recueil des User Story priorisées

# USER STORY CARD

AS A <USER>



I WANT TO <ACTION>



SO I CAN <ACHIEVE THE GOAL>



# Pilotage de projet numérique

## User-Story Map: Mobile App Feature for Depositing Checks

NNGROUP.COM NN/g

### 1. Activities:

High-level tasks users can do in the digital product



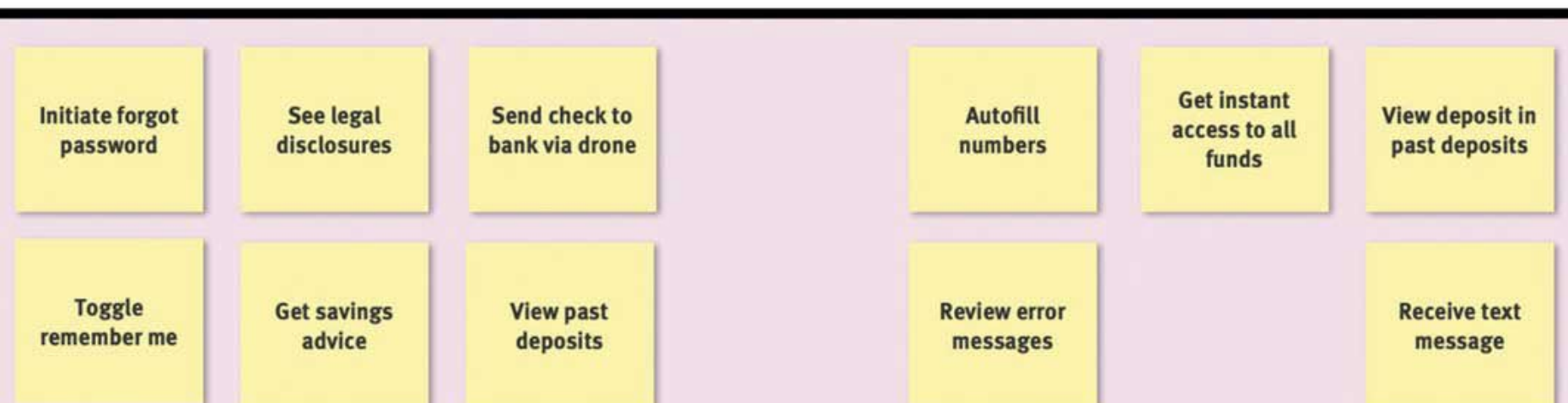
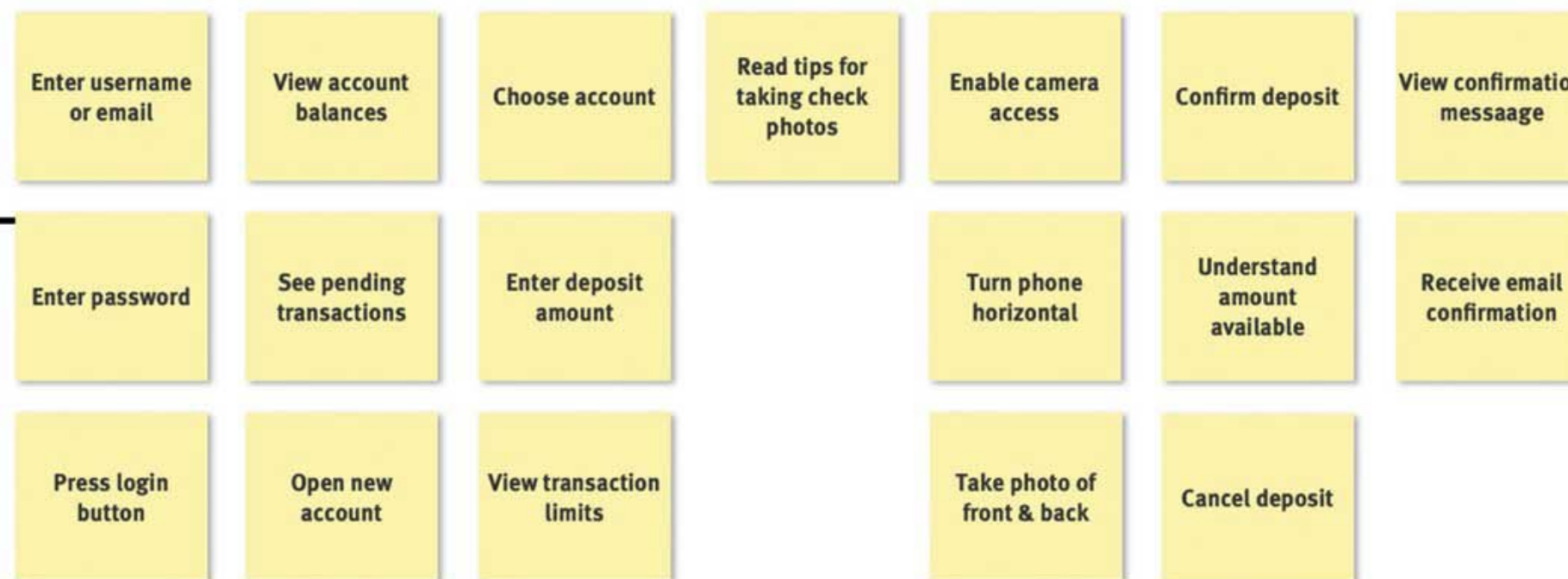
### 2. Steps:

Steps users go through to complete the activity above

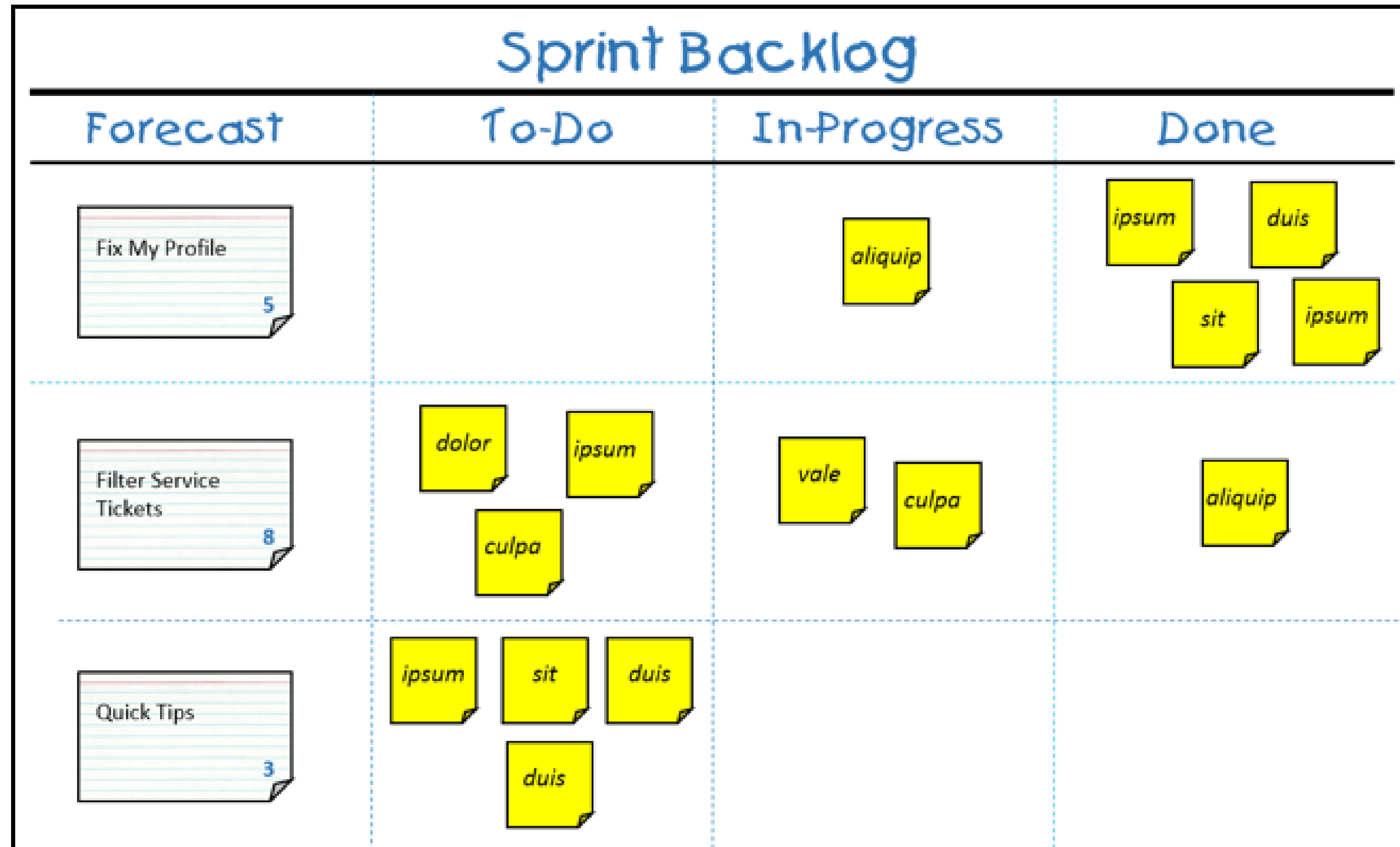


### 3. Details:

Granular, discrete interactions to complete the step above



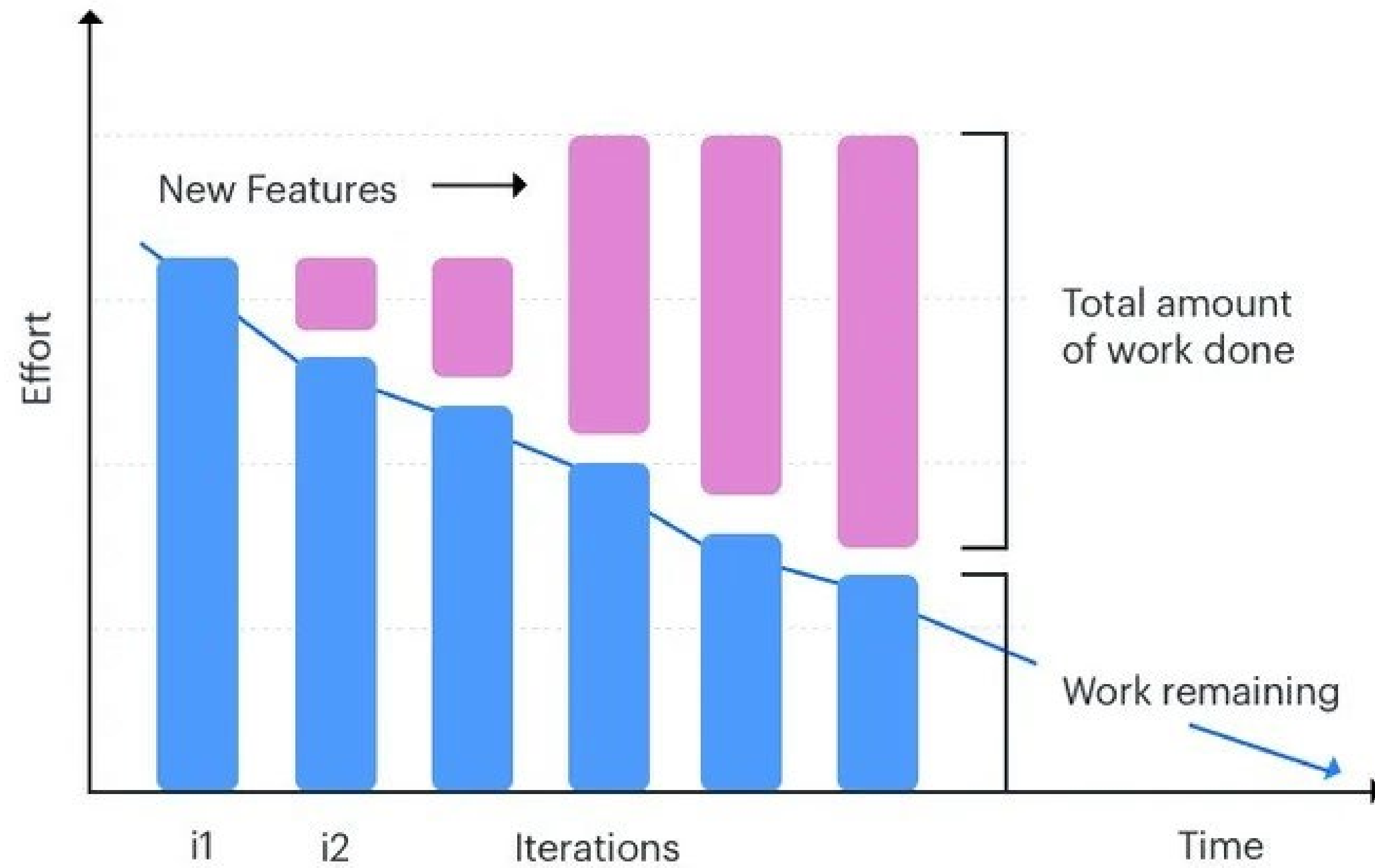
# Pilotage de projet numérique



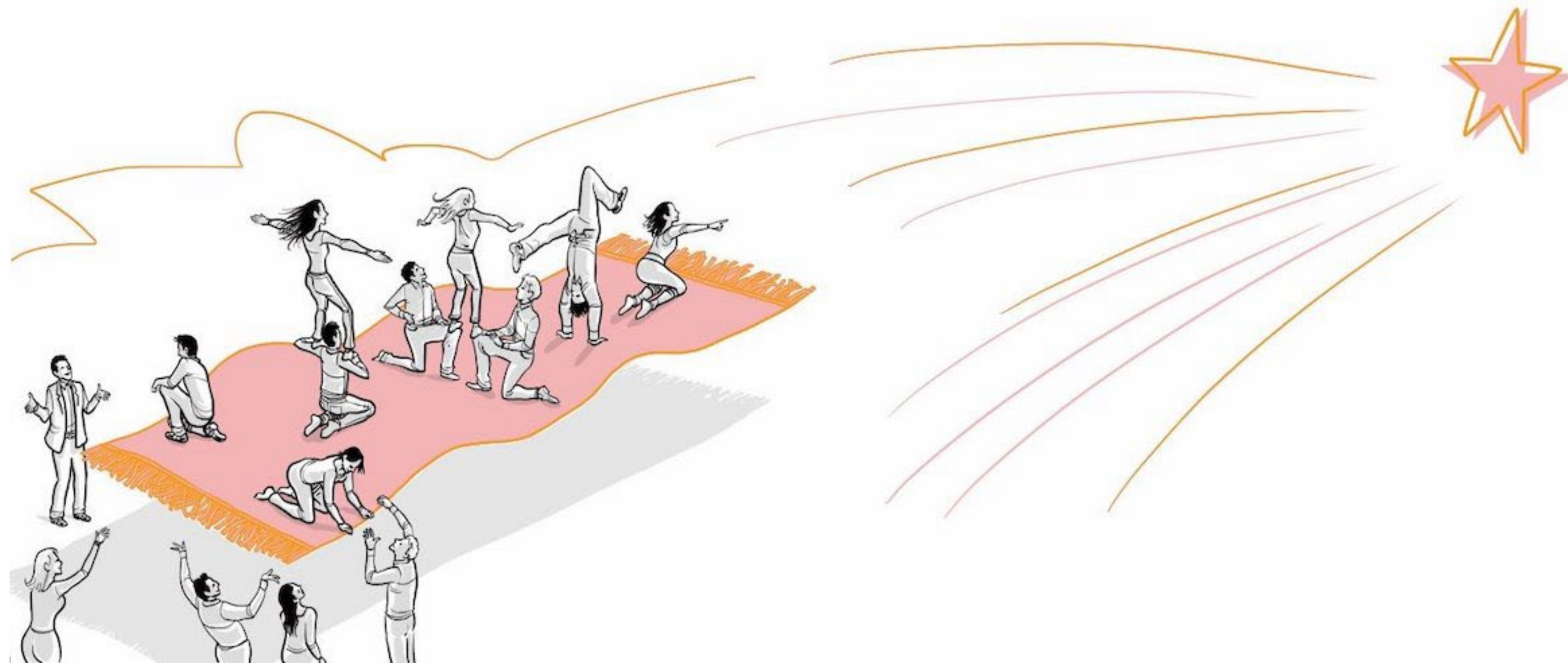
# Pilotage de projet numérique



# Pilotage de projet numérique



# Pilotage de projet numérique



# Agile avancé

## Des US « INVEST »



# Agile avancé

## Des tâches avec des objectifs « SMART »



# Agile avancé

## Matrice des risques

### Exemple de matrice des risques 5x5

#### Impact

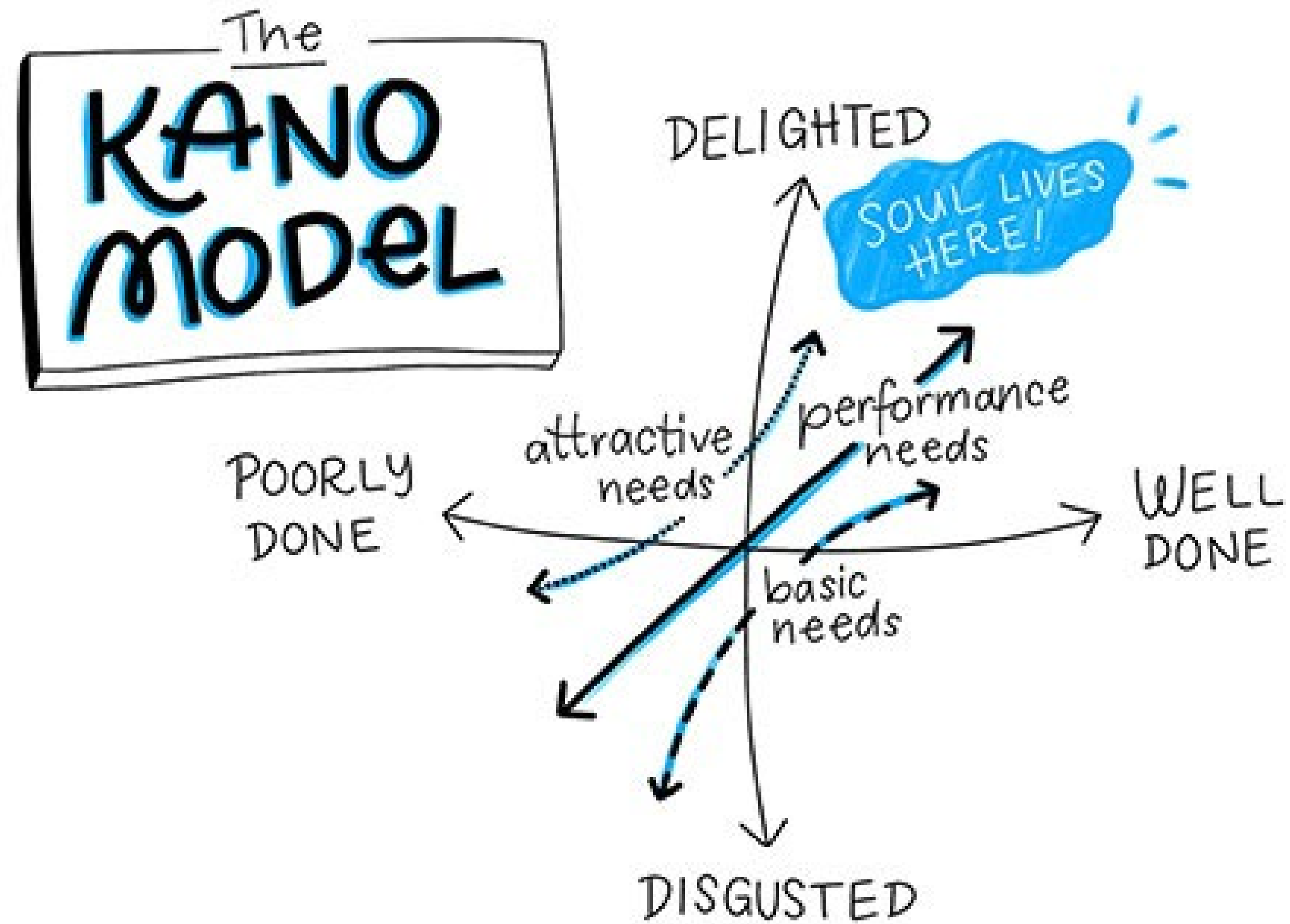
Quelle serait la gravité des conséquences si le risque se produisait ?

Probabilité  
Quelle est la probabilité que le risque se produise ?

	Insignifiante 1	Mineure 2	Significative 3	Majeure 4	Sévère 5
5 Presque certain	Moyen 5	Élevé 10	Très élevé 15	Extrême 20	Extrême 25
4 Probable	Moyen 4	Moyen 8	Élevé 12	Très élevé 16	Extrême 20
3 Modéré	Faible 3	Moyen 6	Moyen 9	Élevé 12	Très élevé 15
2 Improbable	Très faible 2	Faible 4	Moyen 6	Moyen 8	Élevé 10
1 Rare	Très faible 1	Très faible 2	Faible 3	Moyen 4	Moyen 5

# Agile avancé

## Le modèle de Kano



# Agile avancé

## La méthode MoSCoW

### MoSCoW prioritization



#### **MUST HAVE**

All the requirements that are necessary for the successful completion of the project.



#### **SHOULD HAVE**

Requirements that are important for project completion but not necessary.



#### **COULD HAVE**

Requirements that are nice to have, but have a much smaller impact when left out of the project.

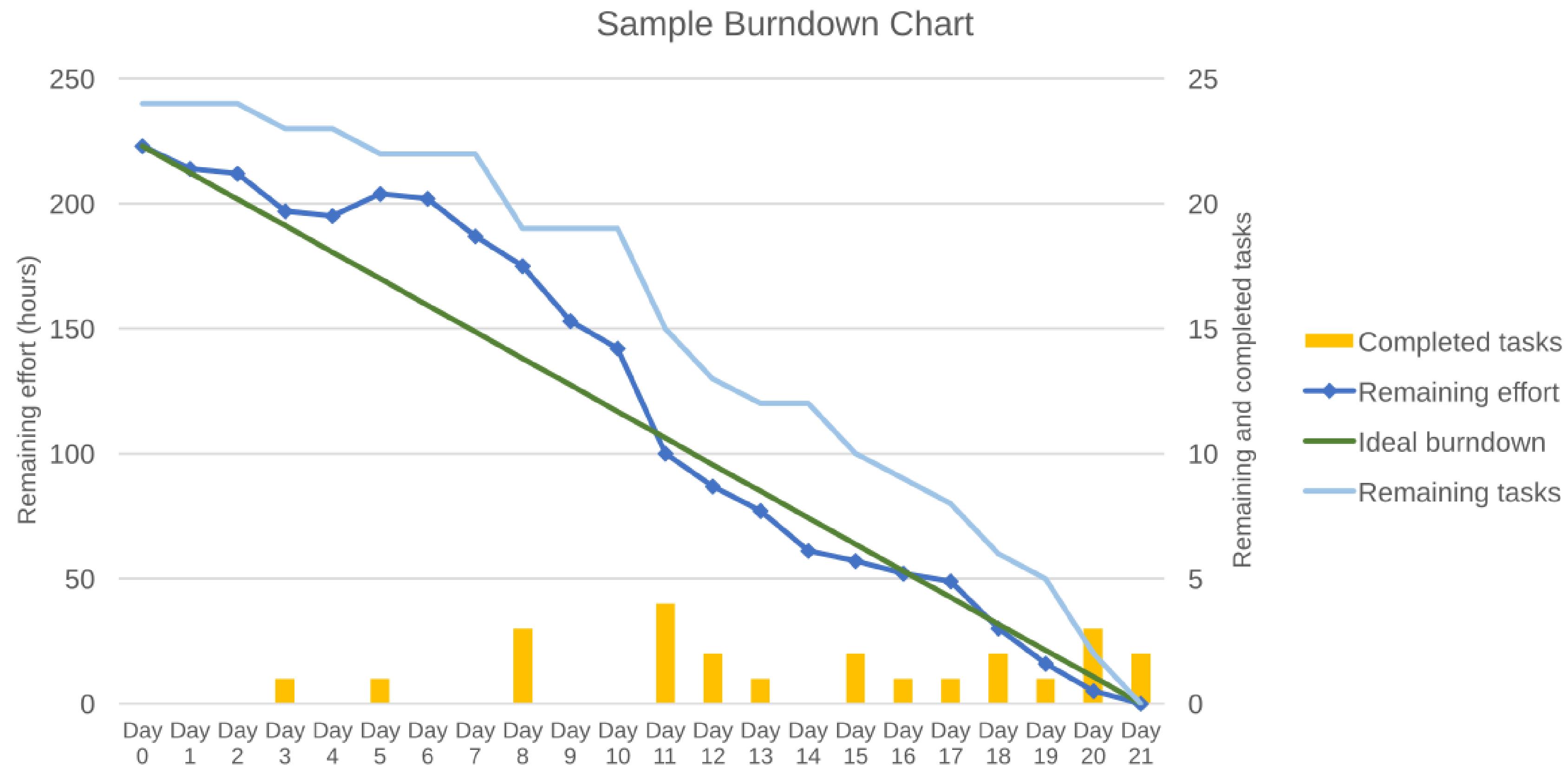


#### **WILL NOT HAVE**

All the requirements that have been recognized as not a priority for the project's timeframe.

# Agile avancé

## La burndown chart



# Agile avancé

## Extreme Programming (XP)

L'originalité de la méthode est de pousser les concepts Agile à l'extrême :

- la [revue de code](#) sera faite en permanence (par un [binôme](#)) ;
- les tests seront faits systématiquement avant chaque mise en œuvre ;
- le code sera retravaillé tout au long du projet ([refactoring](#) ou remaniement du code) ;
- la solution la plus simple sera toujours celle qui sera retenue ;
- des métaphores seront définies et évolueront en concomitance ;
- les modifications seront faites plusieurs fois par jour ;
- des cycles de développement très rapides faciliteront l'adaptation au changement.

# Agile avancé

## La méthode SCRUM

### Différence et similitude avec la méthode Devops

#### 1. Objectifs et portée

Agile Scrum : Vise à optimiser le développement d'un produit en organisant le travail en sprints courts et en favorisant la collaboration entre les développeurs et le Product Owner.

Agile DevOps : Cherche à unifier le développement (Dev) et les opérations (Ops) pour automatiser, tester, et déployer le logiciel plus rapidement et en continu.

# Agile avancé

## La méthode SCRUM

Différence et similitude avec la méthode Devops

### 2. Portée organisationnelle

Scrum : Se concentre sur l'équipe de développement et l'organisation du travail avec des rôles définis (Scrum Master, Product Owner, Dev Team).

DevOps : Va au-delà du développement et intègre les équipes d'exploitation (Ops), sécurité (Sec), et qualité (QA) pour fluidifier tout le cycle de vie du logiciel.

# Agile avancé

## La méthode SCRUM

Différence et similitude avec la méthode Devops

### 3. Cycle de vie et fréquence des livraisons

Scrum : Fonctionne par itérations (généralement de 1 à 4 semaines), avec des livraisons incrémentales à chaque sprint.

DevOps : Privilégie le déploiement continu (CI/CD), permettant des mises en production fréquentes et automatisées (plusieurs fois par jour si nécessaire).

# Agile avancé

## La méthode SCRUM

### Différence et similitude avec la méthode Devops

#### 4. Automatisation et outillage

Scrum : Moins axé sur les outils techniques, il repose davantage sur la gestion agile des tâches et la collaboration.

DevOps : Utilise massivement des outils d'automatisation (CI/CD), monitoring, infrastructure as code (IaC) pour accélérer et fiabiliser le cycle de déploiement.

# Agile avancé

## La méthode SCRUM

Différence et similitude avec la méthode Devops

### 5. Gestion des rôles

Scrum : Définit des rôles précis (Scrum Master, Product Owner, Dev Team).

DevOps : Ne définit pas de rôles spécifiques, mais encourage des profils polyvalents (DevOps engineers, SREs) et une culture de responsabilité partagée.

# Agile avancé

## La méthode SCRUM

Scrum est une méthodologie agile de gestion de projet, idéale pour organiser le développement et la collaboration.

DevOps est une approche technique et organisationnelle qui vise à accélérer le déploiement et l'exploitation des logiciels.

Les deux sont complémentaires : une équipe Scrum peut adopter DevOps pour améliorer son pipeline d'intégration et de déploiement continu (CI/CD).

# Agile avancé

## Une petite métaphore pour se représenter tout ça

Imaginons que travailler sur un projet en agile avec Scrum, c'est comme préparer un gâteau en équipe, étape par étape, pour qu'il soit délicieux à chaque bouchée.

L'équipe de pâtisseries :

Dans notre équipe, il y a :

Le Product Owner (PO) : C'est celui qui décide quel gâteau on fait (au chocolat, à la fraise, etc.) et ce qu'il doit avoir pour plaire aux invités.

Le Scrum Master : C'est un peu le chef d'orchestre. Il ne fait pas le gâteau, mais il aide tout le monde à bien travailler ensemble sans se marcher sur les pieds.

Les pâtisseries (l'équipe) : Ce sont les cuisiniers. Ils mélangent les ingrédients, cuisent et décorent le gâteau.

# Agile avancé

## Une petite métaphore pour se représenter tout ça

La recette :

-Avant de commencer, le PO fait une liste de tout ce qu'il veut dans le gâteau : une génoise, une crème, des décorations... On appelle ça le Backlog.

-Cette liste est triée. Les choses les plus importantes sont en haut, comme la génoise (parce qu'on ne peut pas avoir de gâteau sans ça !).

Les étapes de cuisson (les Sprints) :

-Le travail est divisé en petits morceaux appelés sprints. Un sprint, c'est comme se dire : « Cette semaine, on fait la génoise ! »

-Chaque sprint dure un temps fixe (par exemple, 2 semaines). À la fin de ce temps, une partie du gâteau doit être prête et mangeable (pas besoin d'un gâteau complet, mais ce qui est prêt doit être bon !).

Le plan (Sprint Planning) :

-Au début de chaque sprint, l'équipe regarde le Backlog et choisit ce qu'elle va faire.

Par exemple : « Cette semaine, on fait la génoise et la crème. »

# Agile avancé

## Une petite métaphore pour se représenter tout ça

Travailler en équipe :

-Chaque jour, il y a une petite réunion rapide (comme un check-up, on l'appelle le Daily Scrum).

Chacun dit :

-Ce qu'il a fait hier : « J'ai mélangé les œufs et le sucre. »

-Ce qu'il va faire aujourd'hui : « Je vais mettre le gâteau au four. »

Si quelque chose bloque : « Je n'ai pas de moule ! »

Goûter (Sprint Review) :

À la fin du sprint, tout le monde goûte ce qui a été fait.

Le PO vérifie : « Hmm, la génoise est moelleuse, parfait ! Mais on ajoutera plus de vanille la prochaine fois. »

# Agile avancé

## Une petite métaphore pour se représenter tout ça

Faire mieux la prochaine fois (Rétrospective) :

-L'équipe se réunit et discute :

« Qu'est-ce qui a bien marché ? »

« Qu'est-ce qui a été difficile ? »

« Comment on peut être encore meilleurs pour le prochain sprint ? »

Petit à petit, un gâteau délicieux :

-À chaque sprint, on ajoute une couche au gâteau : la génoise, la crème, les décorations...

-À la fin, le gâteau est prêt, tout le monde est content, et les invités se régalent !

# Agile

## Atelier : projets

Réalisation d'un projet au choix, volontairement très simple, en équipe de 3, 4 max :

- Trombinoscope PROMO
- Location de matériel EPSI
- Réservation de salles EPSI
- Good maps (plus complexe)

# Agile

## Atelier : critères de notation

/3 : Respect de la méthodologie Agile

/2 : Organisation (gestion des imprévus, des conflits etc)

/2 : Dynamique du groupe

/3 : Utilisation d'outils (Trello, Git ou autres)

/6 : Qualité de la réalisation finale

/2 : Mise à profit des compétences des nouveaux arrivants (commit le 19/03 sur 1 point)

/2 : Rendu complet

# Agile

## Atelier : étapes

Création d'US (CDC) priorisé

Découpage en sprint de 2h

Intégration des US sur Trello ou autre

Poker planning

Sprint 1, 2, 3 avec envoi des travaux à l'issue de chacun (via git)

Feedback / Bilan final

Présentation du travail final devant la promo

# Agile

## Atelier : évolution vers l'aspect réseau

### **Trombinoscope interactif avec architecture client-serveur**

Serveur web, API REST, protocole HTTP/HTTPS.

### **Messagerie interne intégrée au trombinoscope**

Communication en temps réel, WebSocket ou MQTT.

### **Trombinoscope distribué en peer-to-peer (P2P)**

Architecture décentralisée, protocoles P2P, partage de données.

### **Système de présence dynamique sur réseau local**

Détection des utilisateurs en ligne, ping, broadcast/multicast UDP.

### **Authentification centralisée et gestion d'accès**

Annuaire LDAP ou OAuth, sécurité réseau.

### **Tableau de bord réseau avec monitoring intégré**

Capture et analyse de trafic réseau, Wireshark, visualisation temps réel.