

# Gestion de projets agile

IPSSI PARIS BA/BD  
23.3 2023



**onepoint.**

**Emma KERGO**

Chef de projet digital onepoint

[emma.kergo@gmail.com](mailto:emma.kergo@gmail.com)

# Planning détaillé

# Qu'est ce qu'un planning détaillé?

- Le document essentiel réalisé par le chef de projet
- Priorisation des tâches (avec l'ensemble des tâches/fonctionnalités)
- Permet d'avoir une vue d'ensemble du projet détaillée
- Permet de voir si retard dans le planning prévu de base
- Important: toujours voir large dans l'estimation des tâches
- Il peut être réaliser via un logiciel spécifique (Ms Project...) ou via excel
- Il est réalisé après le Macro Planning

# Que doit-on avoir dans un planning détaillé?

- Définir les grandes étapes d'un projet (**exemple site internet**) :
  - Cadrage projet
  - Conception
  - Réalisation / Développement / Tests
  - Mise en ligne
- Lister et détailler toutes les tâches nécessaires au projet
- Hiérarchiser les tâches et les répartir en fonction des étapes du projet
- Mettre les dates de début et de fin pour chaque tâche (+ estimation de temps associées)
- Plusieurs tâches peuvent se réaliser en parallèle les unes des autres
- Il est préférable d'être en jour mais on peut aussi le faire en semaine si projet très long.

# Planning : MS Project

# Qu'est ce que MS Project ?

- **Microsoft Project (MS Project)** est un logiciel Microsoft dédié à la gestion de projets. Il permet aux planificateurs et aux chefs de projets d'organiser et de piloter leur projet, de gérer les ressources, le budget, l'analyse des données etc...
- Les fonctionnalités de MS project :
  - Élaboration de diagramme de Gantt
  - Affectation des équipes
  - Gestion de la charge
  - Gestion des jours fériés
  - Génération automatique de documentation au format word, pdf et HTML

# L'interface MS Project

Projet1 - Project Professionnel (Évaluation)

msprojectpratique@outlook.fr

Fichier Tâche Ressource Rapport Projet Affichage Format

Dites-nous ce que vous voulez faire

Diagramme de Gantt Utilisation des tâches Réseau de tâches Calendrier Planificateur d'équipe Tableau des ressources Autres affichages

Tri Plan Tables Surbrillance: [Non surligné] Échelle de temps: Jours

Zoom Zoom entier Tâches sélectionnées

Chronologie Détails

Nouvelle fenêtre Fenêtre Macros

Mode Tâche Nom de la tâche Durée Début Fin Prédécesseurs Noms ressources

DIAGRAMME DE GANTT

Prêt

Nouvelles tâches : Planifiées manuellement



# Présentation de MS project



## La barre de navigation

Project propose 6 onglets :

- **Fichier** : modifie l'allure de l'interface pour une vue plus épurée, permettant de travailler sur les options globales du système et de gérer les projets (ouverture, enregistrement, exports...)
- **Tâche** : présente toutes les fonctionnalités importantes liées au traitement des tâches : liens, hausser/abaisser, jalons...
- **Ressource** : permet de gérer les ressources (humaines et matérielles) du projet
- **Rapport** : propose un puissant moteur de reporting pour présenter les données et communiquer sur un projet.
- **Projet** : présente des raccourcis intéressants pour gérer les propriétés générales du projet (modifier la date de départ, modifier les heures de travail, champs personnalisés...)
- **Format** : permet de modifier l'apparence des éléments, mais aussi de mettre en avant certains points importants dans un projet (tâches critiques/en retard, chemin de tâche...)

# Les avantages de MS project (1/2)

## **1. MS Project permet la gestion des ressources.**

Il y a différents types de ressources :

- Les ressources travail => qui effectuent les charges de travail
- Les ressources matérielles => qui sont des ressources consommables avec possibilité de leur attribuer un nombre ou une quantité
- Les ressources coût => qui sont des frais associés à certains postes

Ces ressources peuvent être ensuite affectés à différentes tâches du projet.

MS Project permet via un graphe des ressources, d'analyser les plans de charge des ressources.

# Les avantages de MS project (2/2)

**2. Pour chaque ressource, MS project permet de lui ajouter un coût =>** coût unitaire (ressources matérielles) et/ou taux journalier (ressource travail). MS Project peut donc calculer le coût du projet.

**3. MS Project permet d'analyser les données (ex: rapports).** Les informations du projet sont exportables dans **MS Project** pour analyser le travail, les coûts, les temps nécessaires grâce aux tableaux, graphiques, diagrammes...

# Planning : Excel



# Méthodes classiques

(ou méthodes traditionnelles/méthodes prédictives)

# Les spécificités de ces méthodes

- **Découpage linéaire et séquentiel du cycle du projet** : le chef de projet veille à ce que chaque étape du déroulement du projet soit terminée avant de passer à la suivante.
- **Cadrage et planification** : le projet est réalisé en accord avec ce qui a été convenu initialement avec le client (cahier des charges). Tout est cadré en amont, et les besoins utilisateurs parfaitement définis.
- **Peu d'interaction en cours de projet avec le client** : grâce au cadrage décrit ci-dessus, le client n'intervient pas en cours de projet. Le contrôle qualité a lieu à la fin.

# Les inconvénients de ces méthodes

- **Manque de flexibilité** : la méthode classique de gestion de projet ne permet aucun retour en arrière. Cette approche peu flexible se révèle contraignante en cas d'imprévu. Il faut réussir au maximum à anticiper les risques en début de projet.
- **Effet tunnel** : la méthode classique de gestion de projet implique une livraison du produit dans sa version finale. Elle favorise l'effet tunnel (manque de communication et de visibilité entre CP et client), et la déception du client (évolution de ses besoins)





# Méthode Waterfall

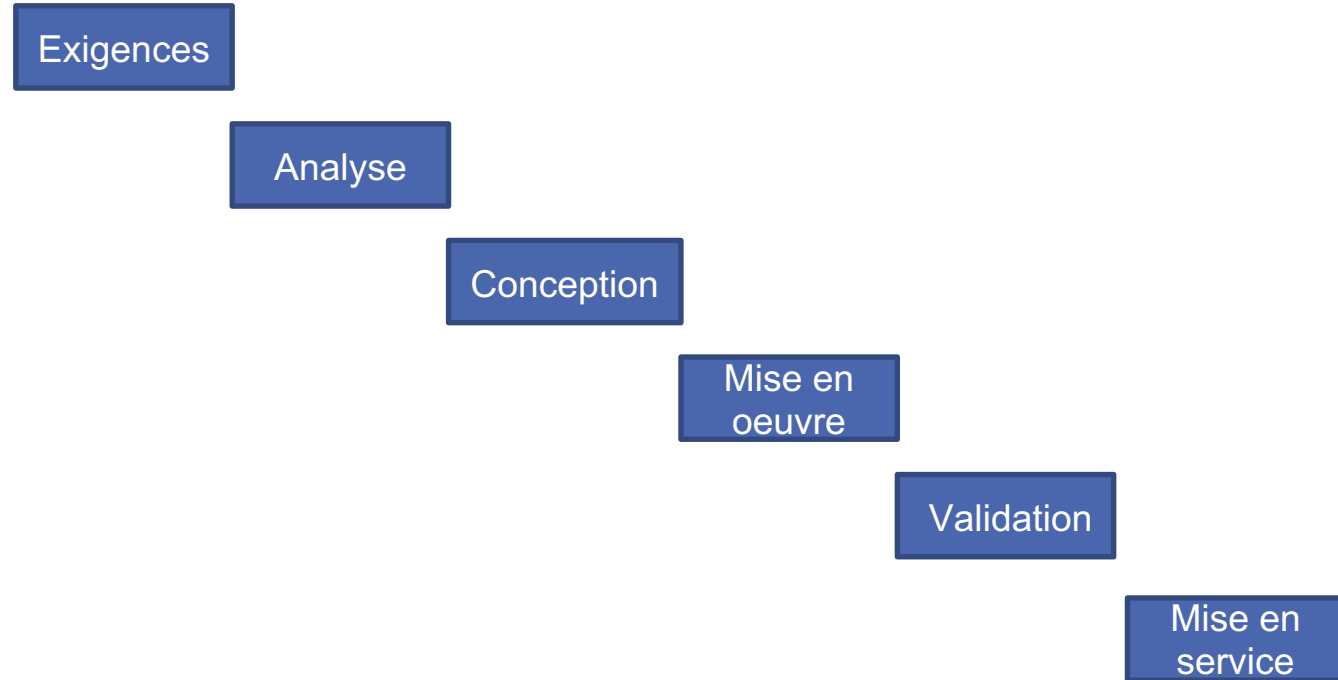
(en cascade)



# Principes

- Cette méthodologie repose sur une succession de séquences.
- Pour rappel, chacune de ces étapes doit être terminée avant de passer à la suivante. Tout retour en arrière n'est pas possible (ou s'avérerait couteux).

# Différentes étapes



# Définition des étapes (1/2)

- **Exigences** : le client exprime ses besoins et objectifs.
- **Analyse** : l'équipe projet analyse les besoins du client et procède aux spécifications fonctionnelles. (CDC rédigé à cette étape et/ou complété)
- **Conception** : l'équipe projet travaille sur les spécifications, afin de définir précisément la manière dont le produit sera conçu.

# Définition des étapes (2/2)

- **Mise en œuvre (=Réalisation)** : il s'agit de l'élaboration du produit en lui-même (Création, mise en place technique, développement...) + **phase de recette interne**
- **Validation (=phase de recette client)** : le produit est testé afin de vérifier que tout fonctionne correctement et ensuite validé par le client.
- **Mise en service (=mise en production)**: le logiciel est mis en production suite à validation du client.



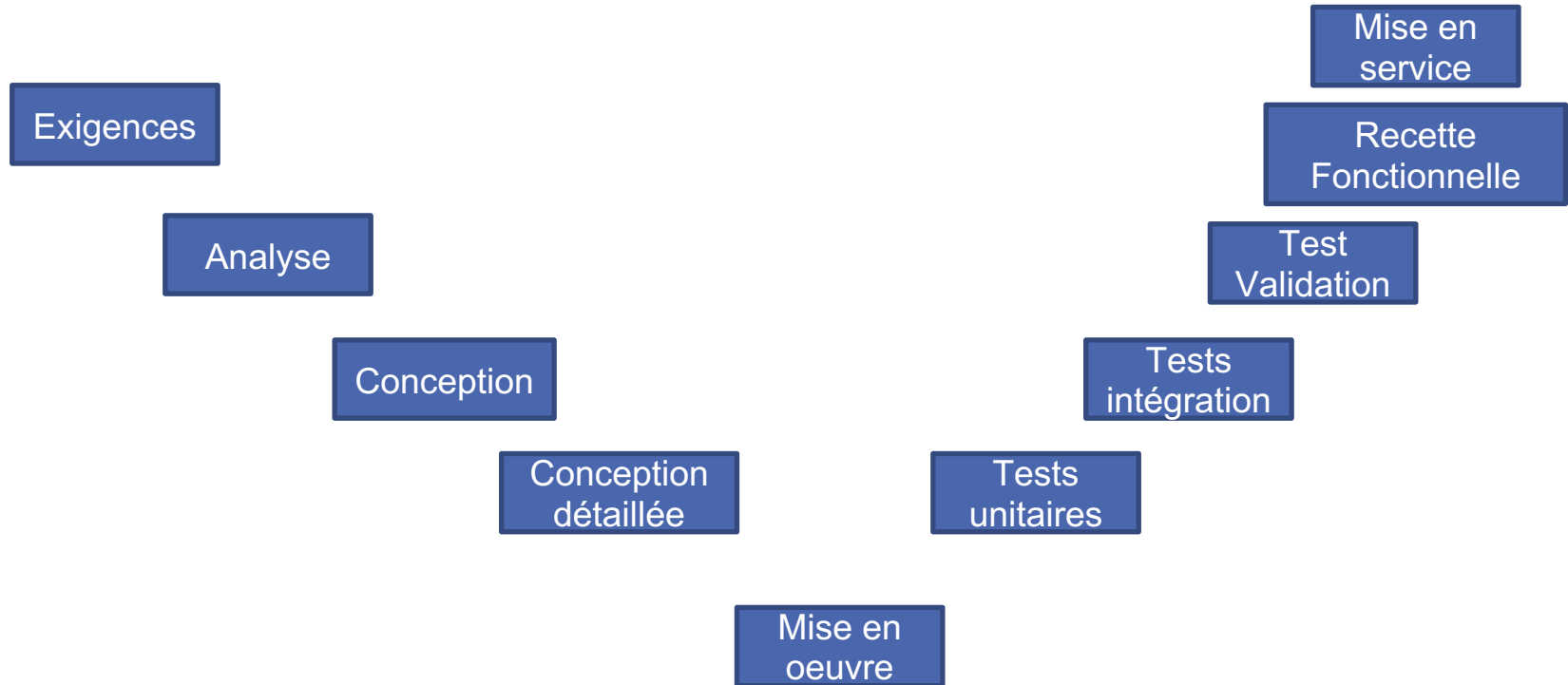
# Méthode Cycle en V



# Principes

- Le Cycle en V s'inspire de la méthode en cascade.
- Reprend le même principe de succession de séquences, à la différence que chaque phase descendante (= la conception du produit) est associée à une phase ascendante (=la validation du produit).
- Nécessite un excellent cadrage en amont.

# Différentes étapes





# Définition des étapes (1/3)

- **Phase de conception :**

**Exigences** : le client exprime ses besoins et objectifs.

**Analyse** : l'équipe projet analyse les besoins du client et procède aux spécifications fonctionnelles. (CDC rédigé à cette étape et/ou complété)

**Conception** : l'équipe projet travaille sur les spécifications , afin de définir la manière dont le produit sera conçu.

**Conception détaillée** : l'équipe projet entre plus dans le détail de la conception du logiciel. Par exemple : définition de bouts de code, ou d'une documentation relative à chaque brique fonctionnelle.

# Définition des étapes (2/3)

- **Mise en œuvre (=Réalisation)** : il s'agit de l'élaboration du produit en lui-même (Création, mise en place technique, développement...)

# Définition des étapes (3/3)

- **Phase de validation :**

**Tests unitaires** : Chaque fonctionnalité (mais aussi l'ensemble de ce qui a été mis en place) est testée par l'équipe projet.

**Tests d'intégration** : Tests effectués sur l'ensemble du produit (par l'équipe technique) pour vérifier que le système fonctionne sur tous les composants.

**Tests validation** : Tests de performance et de régression.

**Recette fonctionnelle** (=validation): avant la mise en production (MEP), le client test et verifie que le produit correspond bien à ses exigences.

**Mise en service** : Mise en production dès validation du client.

# La méthode agile

# Historique

- Depuis 1986 : Nombreuses méthodes de développement dites évolutives offrent une alternative plus efficace que les méthodes traditionnelles en cascade ou cycle en V
- En 2001: Création du mouvement agile et du manifeste agile.

# Présentation

- Approche itérative (Sprints) et collaborative, capable de prendre en compte les besoins initiaux du client et ceux liés aux évolutions.
- Terme « agile » = capacité
- Client au centre
- Permet d'accélérer le développement
- L'agilité reconnaît que le monde est changeant. Il s'agit de développer la capacité à détecter de nouveaux contextes et à s'y adapter rapidement et au moindre coût.

# Valeurs manifeste agile

Individus et interactions

Logiciels opérationnels

Collaboration clients

Adaptation aux changements

**PLUS QUE**

Les processus et outils

Documentation exhaustive

Négociation contractuelle

Suivre un plan

# Principes sous-jacents du manifeste agile

1. Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée
2. Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client
3. Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts
4. Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet
5. Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
6. La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.



# Principes sous-jacents du manifeste agile

- 7. Un logiciel opérationnel est la principale mesure d'avancement
- 8. Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- 9. Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
- 10. La simplicité –c'est-à-dire l'art de minimiser la quantité de travail inutile –est essentielle.
- 11. Les meilleures architectures, spécifications et conceptions émergent d'équipes auto organisées.
- 12. À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

# Avantages de cette méthode

- Meilleure communication
- Meilleure visibilité
- Meilleure contrôle de qualité
- Meilleure détection des risques
- Motivation et confiance de l'équipe
- Contrôle des coûts

# Quelques noms de méthodes agiles

- Extreme Programming (XP) : pousser à l'extrême des principes de management simples. Principalement dans le cas d'importante programmation et dans le jeux vidéos
- DSDM : assez rassurant dans le mode de fonctionnement car analyse micro et macro en amont (étude de faisabilité) et implication importante de tous les acteurs dans la phase de développement
- Crystal : Se base sur la communication et la collaboration. Moins de process. Plutôt fait pour des activités moins techniques.
- Scrum : Plus général. Touche tous les domaines d'activités.
- FDD : assigner chaque fonctionnalité à un développeur en particulier pour permettre plus de cohérence dans le design, et une meilleure performance que si tout le monde « touche à tout ».



# La méthode SCRUM

<https://scrumguides.org/>



# Principes

- Approche dynamique et collaborative
- Rapidité de livraison d'un prototype pour l'évaluation client
- Collaboration étroite entre le client et l'équipe
- Fonctionnement par sprint/itération de 2 à 4 semaines

# La méthode Scrum c'est :

## 3 piliers

- Transparence
- Inspection
- Adaptation

## 5 valeurs

- Engagement
- Courage
- Focus
- Ouverture
- Respect

# La Team Scrum

## **Product Owner**

*Vision client  
Produit final  
Garant du Backlog et le rédige*

## **Scrum Master**

*Facilitateur  
Garant de la méthode  
Mets en place les process*

## **Developers**

*Réalise le projet Equipe Auto  
organisée et motivée*

# Le product owner

## Ne doit pas:

- Influencer l'estimation des fonctionnalités et des tâches à réaliser
- Décider du contenu de chaque sprint sans consulter l'équipe
- Affecter les tâches de développement aux membres de l'équipe

## Sa mission

- Représente le client et le métier
- Possède la vision du produit à réaliser
- Responsable du backlog et du périmètre fonctionnel
- Définit les priorités
- Clarifie le besoin fonctionnel si besoin
- Définit les critères d'acceptation avec les developers
- Valide les développements (User stories)



# Le scrum master

## Ne doit pas:

- Décider du contenu de chaque sprint à la place de l'équipe
- Définir les priorités sur le contenu du projet
- Affecter les tâches de développement aux membres de l'équipe
- Décider des actions d'amélioration à la place de l'équipe

## Sa mission

- S'assure que les pratiques Scrum sont correctement appliquées
- S'assure qu'un processus d'amélioration continue est en place
- Aide à lever les obstacles rencontrés par l'équipe
- Facilite la collaboration
- S'assure que le progrès de l'équipe Scrum est visible et mis à jour

# Les developers

## **Ne doit pas:**

- Définir les priorités sur le contenu du projet
- Masquer ou ignorer les problèmes
- Fuir ses responsabilités en terme d'engagement

## **Sa mission**

- En charge de la réalisation
- Auto-organisée
- Fait les estimations
- Planifie ses activités pendant le sprint
- Respecte les priorités données par le Product Owner
- S'engage sur le contenu de chaque sprint
- Dispose de l'ensemble des compétences nécessaires à la réalisation

# Autres définitions

# Backlog produit

- Contient toutes les exigences/tâches/fonctionnalités qui définissent le produit
- Chaque user story est discuté avec le client, le métier, les utilisateurs et les developers
- La valeur métier d'une user story définit son importance pour le client
- La complexité d'une user story définit la difficulté de sa réalisation par les developers.
- Le product owner priorise les user stories en fonction de la valeur métier, de la complexité et du risque.

# Backlog sprint

- Liste des activités à faire durant le sprint
- Chaque Sprint produit un incrément (produit partiel testable)
- Il reflète le plan de l'équipe pour atteindre l'objectif du sprint
- Il est la propriété des developers

# Qu'est ce qu'une User Story ?

- Reflète une conversation que l'on peut avoir avec un utilisateur et de la demande qu'il pourrait formuler concernant une fonctionnalité.
- Décrit une fonctionnalité et porte la valeur de cette dernière.
- Contient assez d'informations pour que les développeurs puissent estimer.

# Comment créer une user story?

**EN TANT QUE (C)**  
**JE SOUHAITE/VEUX (A)**  
**AFIN DE/QUE (B)**

- (A) est une fonctionnalité
- (B) est le bénéfice de la fonctionnalité
- (C) est la personne ou rôle qui va en bénéficier

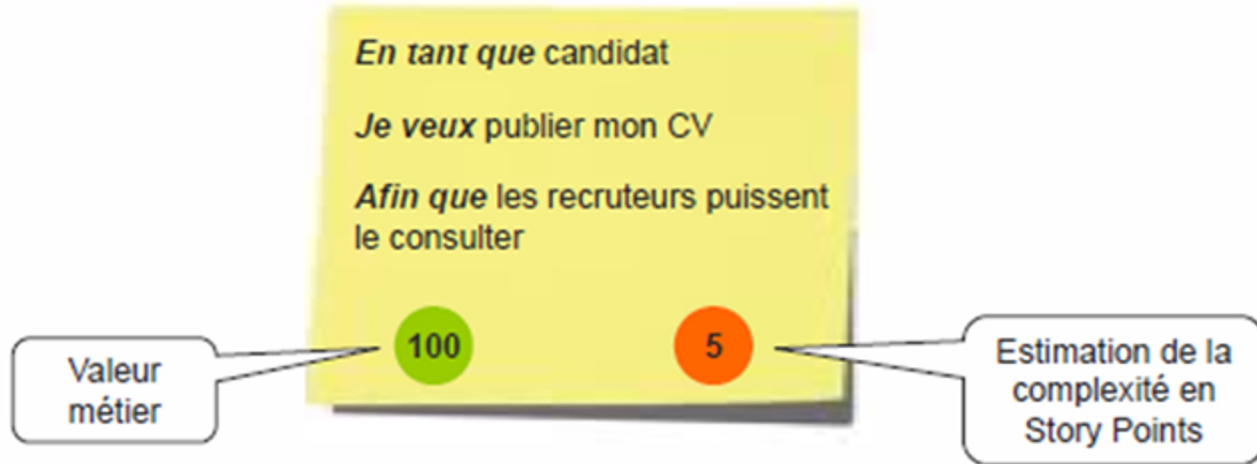
# Une user story est:

- Courte et simple
- Comprise par tous
- Écrite du point de vue de l'utilisateur
- Porteuse d'un bénéfice clair et univoque
- La description d'une seule fonctionnalité (et non une liste)



# Vie d'une user story

Sa valeur est estimée par le client/utilisateur et sa complexité par l'équipe de réalisation



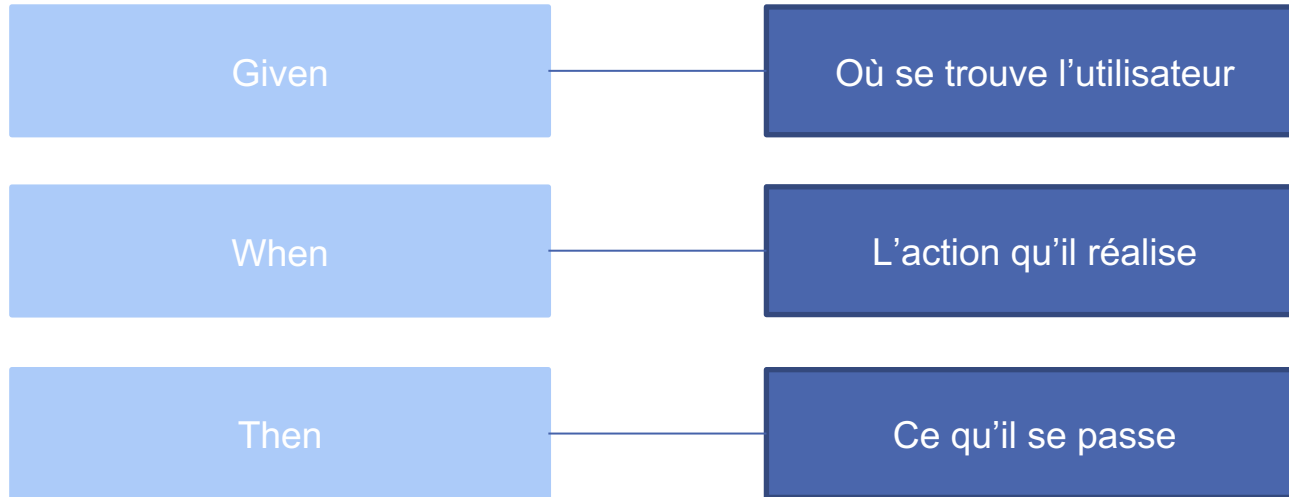
# Critères d'acceptations

- Chaque US a des critères d'acceptations qui sont co-écrits par l'équipe
- Scénarios de tests
- Pas plus de 5-6 critères pour une US

**ETANT DONNE  
QUAND  
ALORS**

# Vie d'une user story

- Les critères d'acceptations prennent le plus souvent la forme de la matrice «Given–When–Then»



# Exemple User Story

**User story 1** : En tant qu'utilisateur je souhaite m'authentifier afin d'accéder au site

## Description :

- A gauche dans encadré bleu canard : Texte d'introduction du site
- Au milieu : Encadré avec informations à saisir Mail et mot de passe
- Mot de passe oublié : Au clic, une pop in s'affiche demandant le mail de l'utilisateur + CTA "valider" / Ensuite réception d'un mail permettant de changer le mdp.
- CTA "je m'identifie": Au clic, accès à la Home page de la plateforme

## Critères d'acceptances:

<p>ÉTANT DONNÉ QUE : je renseigne les identifiants d'un user Super-Admin : test-admin@email.fr, Mot2Passe</p> <p>QUAND : je sou mets le formulaire d'authentification</p> <p>ALORS : je suis authentifié</p>	<p>ÉTANT DONNÉ QUE : je renseigne mal les identifiants d'un user : test@email.fr, Mot2Passe (soit un user inexistant)</p> <p>QUAND : je sou mets le formulaire d'authentification</p> <p>ALORS : l'authentification échoue</p>
--	--

# Exemple User Story

**User story 2 :** En tant qu'utilisateur ayant perdu son mot de passe je souhaite avoir un nouveau mot de passe afin d'accéder au site

## Description :

- Au clic sur "mot de passe oublié" , une pop in s'affiche demandant le mail de l'utilisateur + CTA "valider"
- Ensuite réception d'un mail permettant de changer le mdp.

## Critères d'acceptances:

<p>ÉTANT DONNÉ QUE : je renseigne un email utilisateur existant en base de données (et en mesure de réceptionner des emails : emma.kergo@gmail.com)</p> <p>QUAND : je sou mets le formulaire</p> <p>ALORS : un message de confirmation s'affiche : "Un e-mail a été envoyé sur emma.kergo@gmail.com pour vous permettre de réinitialiser votre mot de passe"</p> <p>ET : un email est réceptionné sur l'adresse email renseigné (ex : emma.kergo@gmail.com)</p>	<p>ÉTANT DONNÉ QUE : je renseigne un email utilisateur inexistant en base de données</p> <p>QUAND : je sou mets le formulaire</p> <p>ALORS : un message d'erreur s'affiche : "Cet e-mail ne correspond pas à un compte utilisateur"</p>
---	---

# Exemple User Story

**User story 3** : En tant qu'utilisateur je souhaite me déconnecter afin de ne plus avoir ma session active

## Description :

- Depuis le menu une icône de déconnexion permet de se déconnecter
- Après déconnexion le user est systématiquement redirigé sur la page de login, toutes les autres pages du site doivent être inaccessibles

## Critères d'acceptances:

ÉTANT DONNÉ QUE : je suis authentifié	ÉTANT DONNÉ QUE : je ne suis pas authentifié
QUAND : je clique sur l'icône de déconnexion	QUAND : je charge une url d'une page du site exemple : /accueil
ALORS : je suis redirigé vers le login	ALORS : je suis redirigé vers le login

# Exemple User Story

**User story 4 :**En tant qu'utilisateur je souhaite avoir un message d'erreur afin de savoir si je me trompe dans les données que j'indique au login

## **Description :**

- Si l'utilisateur s'est trompé sur son email et/ou mot de passe un message d'erreur s'affiche.

## **Critères d'acceptances:**

ÉTANT DONNÉ QUE : je renseigne mal les identifiants d'un user : toto@email.fr, Mot2Passe (soit un user inexistant)

QUAND : je sou mets le formulaire d'authentification

ALORS : un message d'erreur s'affiche "identifiants incorrects"

ET : l'authentification échoue

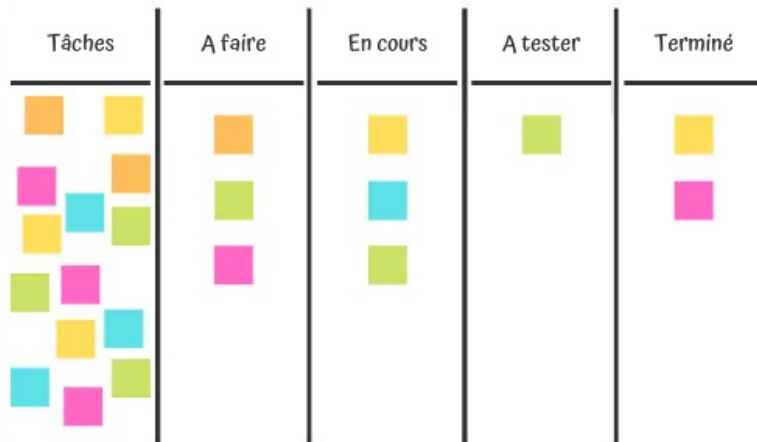
ET : je reste sur la page de login

Une User Story est prête à être embarquée dans un sprint lorsqu'elle est «INVEST» :

Initiale	Signification	Description
I	<b>Independant</b>	L'US doit être indépendante des autres User story pour faciliter sa planification dans un sprint.
N	<b>Negociable</b>	L'US est négociée, discutée lors du planning poker, lors des planifications de Sprints, lors des développements.
V	<b>Valuable</b>	L'US apporte de la valeur métier à l'utilisateur.
E	<b>Estimable</b>	L'équipe doit pouvoir estimer l'US en points de complexité. Si elle ne peut pas, l'US doit être précisée.
S	<b>Small</b>	L'US doit être assez petite pour être terminée en un seul sprint
T	<b>Testable</b>	L'US doit être testable. Les critères d'acceptation sont définis en même temps que l'US.



# User stories : Présentation sous forme de tableau Kanban



- Situé sur le lieu de travail de l'équipe.
- Mis à jour de façon quotidienne
- Permet à l'équipe de se synchroniser et d'avoir la visibilité de tous les sujets en cours.
- Basé sur du flux tiré, c'est uniquement l'équipe qui a la possibilité de faire avancer les tickets.
- Permet de visualiser rapidement d'éventuels blocages.

# Outils de gestion de projet Agile

- **Trello:** Réalisation d'un tableau Kanban personnalisé avec choix des colonnes (et des noms aux choix), avec couleurs et assignation des tâches => <https://trello.com>
- **Notion :** Plateforme complète pour ajouter vos wikis, documents et suivre vos projets. Vous pouvez créer tableau kanban, planning mais aussi intégrer les documents de vos projets => <https://www.notion.so/fr-fr>
- **Jira :** Réalisation d'un tableau Kanban personnalisé avec couleurs et assignation des tâches, ainsi que possibilité de créer des sprints. La version gratuite limite le nombre de comptes => <https://www.atlassian.com/fr/software/jira>
- **Kanban Tool :** un tableau Kanban en ligne. Il propose des cartes Kanban, des couloirs, des couleurs et plus encore pour garantir la visibilité de votre projet. La version de base de l'outil est gratuite pour un maximum de deux utilisateurs et deux tableaux => <https://kanbantool.com/fr/>

# Exemple Trello (1/4)

The screenshot shows a Trello board interface with a blue header and a large blue background. The header includes a navigation bar with 'Tableau', 'TEST', and a star icon. Below this is a sub-header with 'Espace de travail de emmakergo2', a lock icon, 'Privé', and an 'Inviter' button. On the right side of the header are buttons for 'Power-ups', 'Automatisation', 'Filtre', and 'Afficher le menu'.

The board is organized into columns. The first column is labeled 'BACKLOG PRODUIT' and contains four cards with purple and orange headers. The subsequent four columns are labeled 'SPRINT 1', 'SPRINT 2', 'SPRINT 3', and 'SPRINT 4', each with a '+ Ajouter une carte' button. The final column is labeled 'EN COURS' and also has a '+ Ajouter une carte' button.

**BACKLOG PRODUIT**

- AUTHENTIFICATION** / En tant qu'utilisateur je souhaite m'authentifier afin d'accéder au site  
0/2
- MDP OUBLIE** / En tant qu'utilisateur ayant perdu son mot de passe je souhaite avoir un nouveau mot de passe afin d'accéder au site  
0/2
- MESSAGE ERREUR** / En tant qu'utilisateur je souhaite avoir un message d'erreur afin de savoir si je me trompe dans les données que j'indique au login  
0/1
- DECONNEXION** / En tant qu'utilisateur je souhaite me déconnecter afin de ne plus avoir ma session active  
0/2

**US 5**

**US 6**

**US 7**

**US 8**

**US 9**

**SPRINT 1**

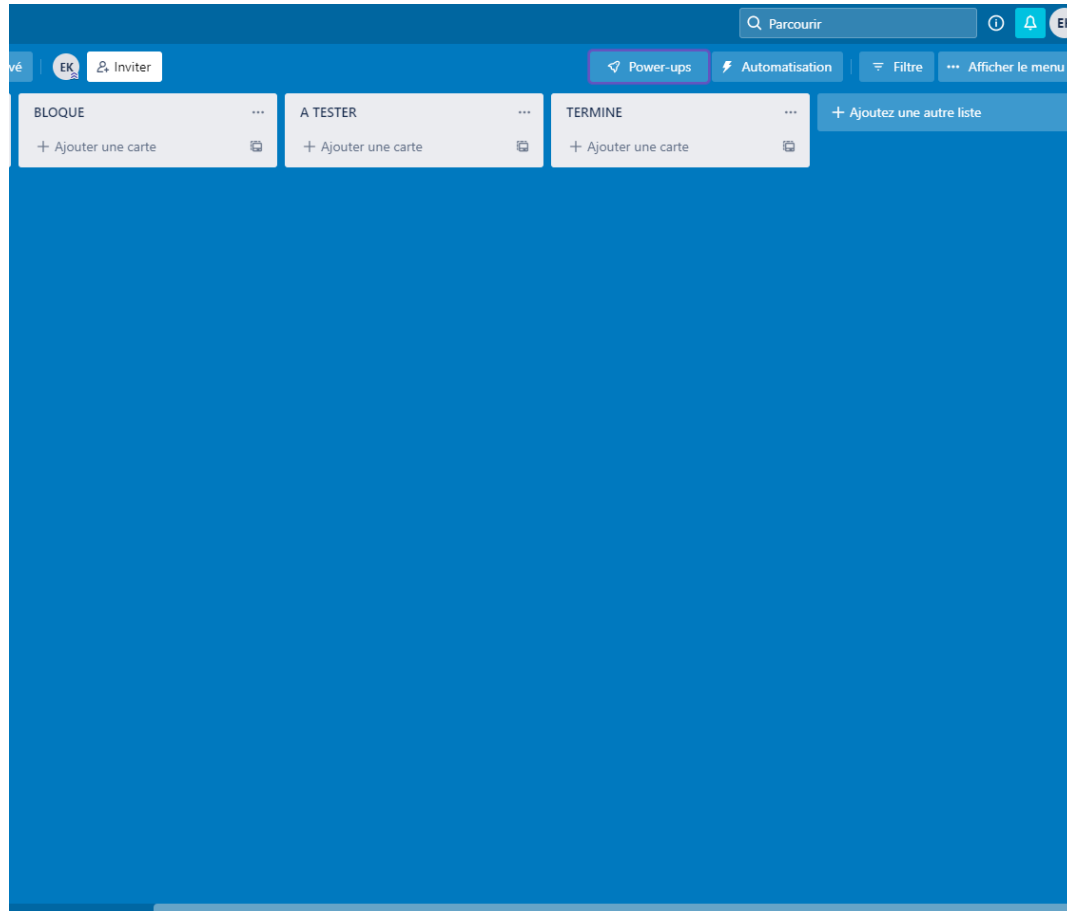
**SPRINT 2**

**SPRINT 3**

**SPRINT 4**

**EN COURS**

# Exemple Trello (2/4)



# Exemple Trello (3/4)

The screenshot displays a Trello board with a blue background and a white header bar. The board is organized into six columns, each with a title and a 'plus' icon for adding cards. The columns are: BACKLOG PRODUIT, SPRINT 1, EN COURS, BLOQUE, A TESTER, and TERMINE. The SPRINT 1 column is currently active and contains three cards. The first card is titled 'AUTHENTIFICATION' and describes a user story about logging in. The second card is titled 'MDP OUBLIE' and describes a user story about password recovery. The third card is titled 'MESSAGE ERREUR' and describes a user story about error messages during login. Each card has a progress bar at the top, a list of labels, and a 'plus' icon for adding more labels. The BACKLOG PRODUIT column contains a list of user stories (US 7 to US 15) and a 'plus' icon for adding more cards. The EN COURS, BLOQUE, A TESTER, and TERMINE columns are currently empty.

**BACKLOG PRODUIT**

- DECONNEXION / En tant qu'utilisateur je souhaite me déconnecter afin de ne plus avoir ma session active
- US 7
- US 8
- US 9
- US 10
- US 11
- US 12
- US 13
- US 14
- US 15
- + Ajouter une carte

**SPRINT 1**

- AUTHENTIFICATION / En tant qu'utilisateur je souhaite m'authentifier afin d'accéder au site
- MDP OUBLIE / En tant qu'utilisateur ayant perdu son mot de passe je souhaite avoir un nouveau mot de passe afin d'accéder au site
- MESSAGE ERREUR / En tant qu'utilisateur je souhaite avoir un message d'erreur afin de savoir si je me trompe dans les données que j'indique au login
- US 5
- US 6
- + Ajouter une carte

**EN COURS**

+ Ajouter une carte

**BLOQUE**

+ Ajouter une carte

**A TESTER**

+ Ajouter une carte

**TERMINE**

+ Ajouter une carte

# Exemple Trello (4/4)

📄 AUTHENTIFICATION / En tant qu'utilisateur je souhaite m'authentifier afin d'accéder au site

Dans la liste SPRINT 1

Étiquettes

+

Description

Modifier

A gauche dans encadré bleu canard : Texte d'introduction du site  
Au milieu : Encadré avec informations à saisir Mail et mot de passe  
Mot de passe oublié : Au clic, une pop in s'affiche demandant le mail de l'utilisateur + CTA "valider" / Ensuite réception d'un mail permettant de changer le mdp.  
CTA "je m'identifie": Au clic, accès à la Home page de la plateforme

Valeur métier : 100  
Complexité : 21

☒ Critères d'acceptances

Masquer les tâches cochées

Supprimer

50%

☒ ÉTANT DONNÉ QUE : je renseigne les identifiants d'un user Super Admin : test\_admin@email.fr, Mot2Passe QUAND : je soumetts le formulaire d'authentification ALORS : je suis authentifié

☐ ÉTANT DONNÉ QUE : je renseigne mal les identifiants d'un user : test@email.fr, Mot2Passe (soit un user inexistant) QUAND : je soumetts le formulaire d'authentification ALORS : l'authentification échoue

Ajouter un élément

Activité

Afficher les détails

EK

Écrivez un commentaire...

EK

Emma Kergo 14 mars à 16:16

nok critère 2

Modifier - Supprimer

Ajouter à la carte

Membres

Étiquettes

Checklist

Dates

Pièce jointe

Image de couvert...

Champs persona...

Ajoutez des menus déroulants, des champs de texte, des dates et plus encore à vos cartes.

Commencer l'essai gratuit

Power-Ups

Ajouter des Power...

Automatisation

Ajouter un bouton

Actions

Déplacer

Copier

Créer un modèle

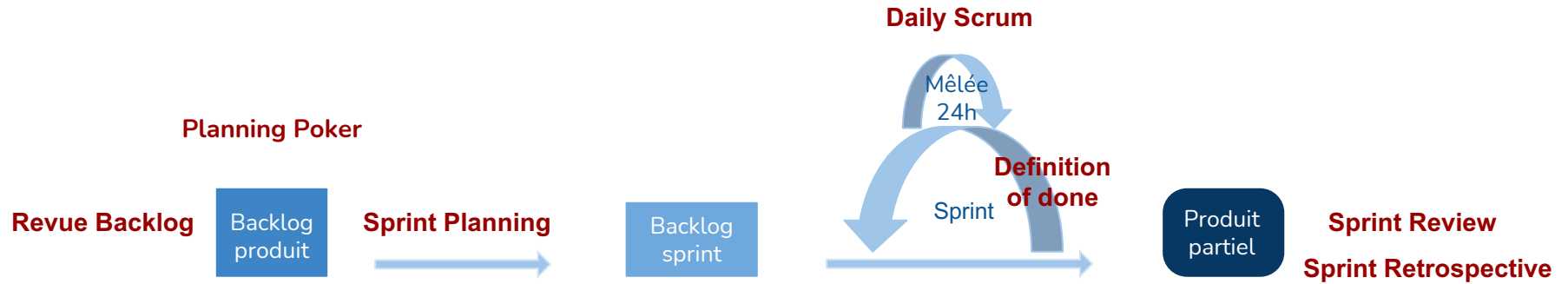
Suivre

Archiver

Partager

# Les réunions SCRUM

# Vue d'ensemble Scrum





# Planning poker – Estimation User Story

- Estimation de la complexité par l'équipe de réalisation
- Les US du backlog sont estimées unitairement à l'aide de la suite de Fibonacci :  
**0, 1, 2, 3, 5, 8, 13, 21, 34, 55**
- Cette suite illustre la relation entre la complexité d'une tâche et l'incertitude à l'estimer.
- Plus la **complexité** d'une tâche est grande plus **l'incertitude** est grande.
- Cette estimation peut se faire via à un jeu de carte, ou via site internet spécialisé :  
(<https://planningpokeronline.com>)

# Sprint planning

- Premier rendez-vous du Sprint. L'équipe sélectionne des user stories et s'engage sur leur réalisation pendant l'itération.
- Le Product Owner indique les fonctionnalités prioritaires les plus importantes.
- **Durée:** 2H par semaine de Sprint. Si sprint de 2 semaines, alors la réunion sera de 4h.

## Participants :

L'équipe Scrum

Des experts (optionnels)

## Entrées :

Le Product Backlog priorisé

Les retours du Sprint précédent

La capacité prévue

## Sorties :

Le Sprint Backlog estimé

Les objectifs du sprint

# Daily scrum

- Réunion tous les matins
- **Chaque membre de l'équipe répond de façon synthétique à ces 3 questions**
  - 1.Ce que j'ai fait hier ?
  - 2.Ce que je fais aujourd'hui ?
  - 3.A quels problèmes je fais face ?
- **Durée:** 15 minutes par jour

## Participants:

L'équipe de dev

Le Product Owner(optionnel)

Le Scrum Master (optionnel)

## Entrées:

Le Sprint Backlog

Les réalisations de la veille

## Sorties:

Le Sprint Backlog mis à jour

Les problèmes à résoudre

# Definition of done

- La «Definition of Done» permet une validation claire des User Stories
- Document de référence permettant de lever toute ambiguïté sur **ce qui doit être fait pour qu'une user story soit terminée**
- Utilisée par le Product Owner pendant la réunion de fin de sprint pour accepter ou refuser les exigences
- Peut être enrichie d'un sprint à l'autre chaque fois que des manques sont identifiés par l'équipe

# Sprint review ou démo

- L'objectif de cette réunion est de présenter le produit réalisé au cours du Sprint, de discuter des difficultés rencontrées et de la roadmap à venir
- *On démontre un produit fonctionnel*
- **Durée:** 4h max

## **Participants:**

L'équipe Scrum

Des utilisateurs

Toute autre partie prenante

## **Entrées:**

Le Sprint Backlog

## **Sorties:**

Les objectifs prévisionnels du prochain sprint

+ retours sur le Sprint Backlog

# Sprint retrospective

- La rétrospective est un moment d'échange où l'équipe peut faire un bilan sur le Sprint passé.
- **Durée:** 3h max

## **Participants :**

L'équipe Scrum

## **Entrées :**

Des connaissances sur les relations, les procédures et les outils de l'équipe

Des idées d'amélioration

## **Sorties :**

Un plan d'action pour le prochain sprint

# Revue de backlog

- L'objectif de cette réunion est d'affiner le Product Backlog et de prendre en compte les nouvelles User Stories prioritaires.
- **Durée:** 4h

## **Participants :**

L'équipe Scrum

## **Entrées :**

Le Product Backlog

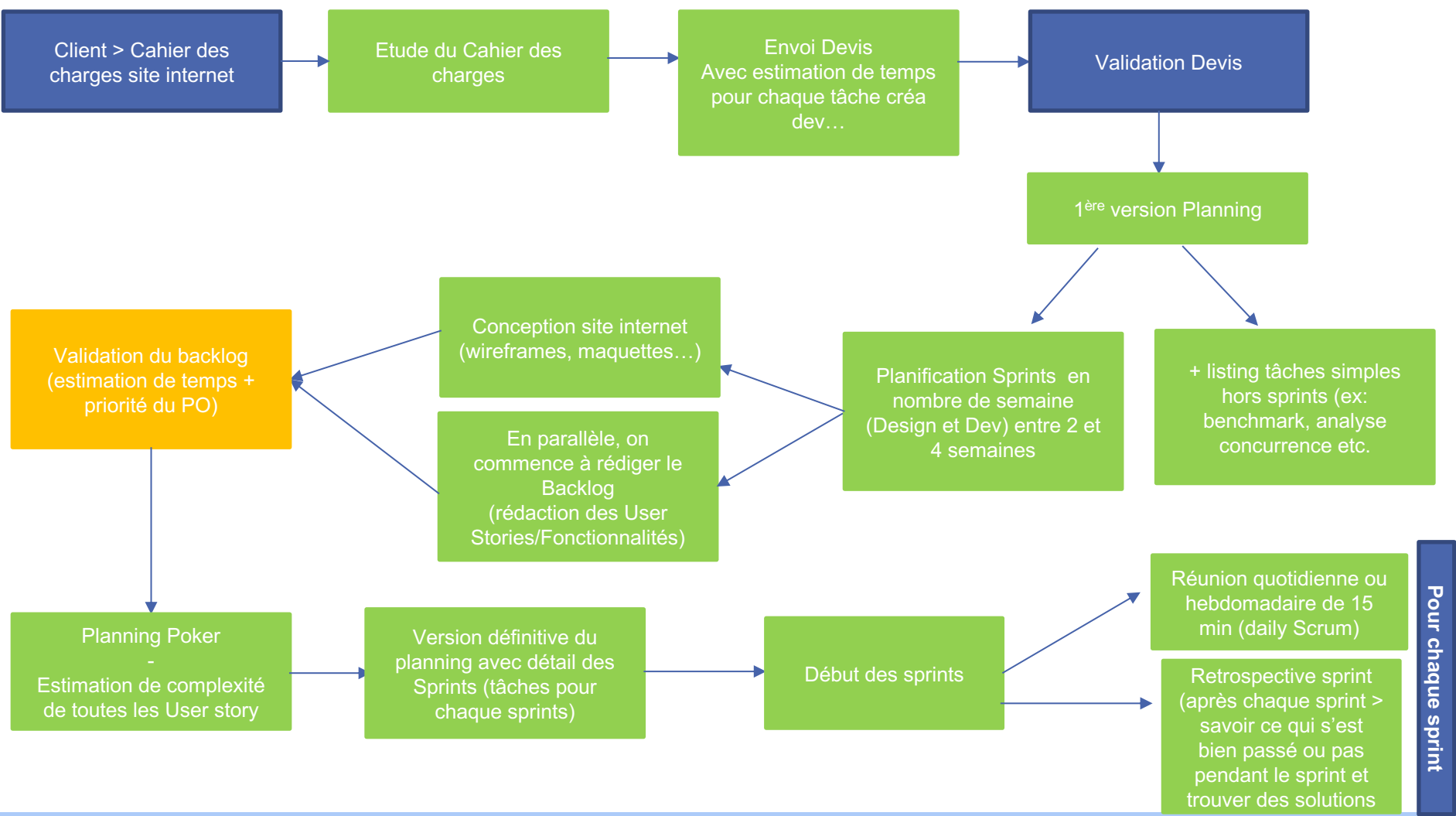
## **Sorties :**

Le Product Backlog ajusté, détaillé et estimé

# Contractualisation agile

- En France, contrat au forfait sur les gros projets
- **Meilleure solution:** forfaitisation de chaque itération
- **Autre possibilité:** troc d'exigence => réalisation d'une fonctionnalité imprévue en échange retrait d'une autre fonctionnalité moins importante (mais avec le même coût).





# Cas si retours client

- 2 possibilités (A acter avec le client) :
  - On prévoit des sprints spécifiques pour les retours après avoir bien avancé sur les fonctionnalités importantes à développer
  - Prise en charge des retours au sprint suivant / et adapter le nombre d'US à réaliser comme il faut intégrer les retours

# Cas ajouts de fonctionnalités par le client

- 2 possibilités (A acter avec le client):
  - Si on ajoute des nouvelles fonctionnalités (sans troc de fonctionnalités) , on ajoute de ou des sprints supplémentaires
  - Si on ajoute des nouvelles fonctionnalités (avec troc de fonctionnalités) , nous n'ajoutons pas de sprints supplémentaire on échange juste les fonctionnalités à ajouter et à supprimer.

# CAS PRATIQUE

# CAS PRATIQUE (3/3)

Après avoir réaliser la note de cadrage et le cahier des charges de votre projet, voici les nouvelles étapes à réaliser sur la partie planning et méthodologie :

1. Lister l'ensemble des fonctionnalités que vous proposez en fonction des besoins de votre projet (Minimum 15 fonctionnalités). Me lister en plus des fonctionnalités les tâches Admin. Bien différencier ce qui sera fait côté Dev et côté Admin.
2. Préparer un backlog complet sur le logiciel de gestion de projet votre choix (trello, notion, jira, etc...). Le backlog comprendra l'ensemble des fonctionnalités/tâches de votre plateforme (celles demandées en point 2). Pour chaque fonctionnalité faire une User Story, une description et des critères d'acceptances. Pas besoin de User story pour les tâches, ne pas oublier de les détailler.
3. En fonction du logiciel de gestion de projet choisi, expliquez moi pourquoi celui-ci.
4. Proposer une organisation de travail spécifique pour la réalisation de votre projet (sur la base de ce que nous avons vu sur le cours en agile). Il faut m'expliquer comment vous souhaitez vous organiser tout au long de votre projet => **NE PAS FAIRE UN COPIER COLLER DU COURS.** Vous pouvez parler de votre backlog, de comment vous l'avez créé, détailler vos sprints (etc...). Vous pouvez illustrer par un schéma mais bien le détailler à l'écrit.
5. Réaliser le planning détaillé de votre projet sur le support de votre choix (ne pas oublier de bien le détailler).

**!! M'ajouter sur votre backlog en fonction du logiciel que vous aurez choisi + me remettre le lien dans votre document**

# Préparation de l'oral de groupe de Jeudi 20 avril après-midi

- Préparer une présentation power point pour l'oral + **bien préparer votre oral !**
- Oral de 15 min par groupe (**10 min présentation 5 min échanges**)
- Choisir votre créneau de passage:

[https://docs.google.com/spreadsheets/d/10BC7zeO0QZgVjHu\\_BiOdo953XmKI7rh2kNjc8L\\_ZDrE/edit?usp=sharing](https://docs.google.com/spreadsheets/d/10BC7zeO0QZgVjHu_BiOdo953XmKI7rh2kNjc8L_ZDrE/edit?usp=sharing)

- Dans cette présentation, vous devrez :
  1. Présenter votre équipe et votre projet (sur la base de votre note de cadrage et de votre cahier des charges)
  2. Présenter succinctement les fonctionnalités/tâches Admin et Dev
  3. Présenter comment vous souhaitez vous organiser pour ce futur projet (votre organisation de travail et ne pas oublier de présenter le backlog que vous aurez réaliser)
  4. Présenter votre planning (attention qu'il soit bien lisible pour la présentation)
  5. M'expliquer comment vous vous êtes repartis les tâches pour le bon déroulement du cas pratique
  6. Votre retour sur le projet de groupe – ce qui s'est bien passé et moins bien passé

# Points importants sur le cas pratique et l'oral

- Bien relire tous vos documents (**Attention aux fautes**)
- Faire des phrases claires et courtes
- Avoir une présentation lisible et agréable à l'œil
- Bien préparer votre oral, respecter les 10 min de présentation

# Après votre l'oral

- **M'envoyer par mail ([emma.kergo@gmail.com](mailto:emma.kergo@gmail.com)) :**
  1. La présentation power point de votre oral
  2. Votre note de cadrage sous format PDF
  3. Votre cahier des charges sous format PDF
  4. Un document complet avec l'ensemble de vos fonctionnalités/tâches à réaliser en fonction du profil (BD ou BA), le lien de votre backlog, la proposition de méthodologie de travail.
  5. Un planning lisible sous format PDF ou excel