

Gestion de projet

MMI 2 – TP#1 S3

Danielo **JEAN-LOUIS**

But du cours

- Découvrir la méthode Agile
- Découvrir les méthodes de travail issues de la méthode Agile

Apprentissages critiques ciblés

- AC24.02 | Mettre en place ou développer un back office
- AC25.01 | Gérer un projet avec une méthode d'amélioration continue par exemple une méthode agile

(User) Story

- Issu d'eXtreme Programming (XP)
- Récit Utilisateur en français
- Description simple et **petite** d'un besoin
- Représente une tâche pour un **développeur**

Source(s) :

- <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-French.pdf>

Pour rappel, dans SCRUM, un développeur n'est pas forcément une personne qui code

(User) Story

- **Doit apporter une valeur au produit**
 - Si ça n'apporte pas à une valeur au produit → Poubelle
- Rédigé par le Product Owner (PO)
 - Affinée, complétée avec l'équipe lors du **backlog refinement**

Source(s) :

- <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-French.pdf>

Backlog Refinement

- Rituel de SCRUM
- Anciennement appelé "backlog grooming"
- Réunion où on discute des Stories pour s'assurer que tout est compris
- Affinement des Stories si problème

Source(s) :

- <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-French.pdf>

(User) Story

- Structuré de la façon suivante :

**« En tant que [persona],
je veux [quoi]
afin de [pourquoi] »**

Source(s) :

- <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-French.pdf>

(User) Story

- **En tant que [persona]**
 - Pour qui on développe ceci ?
- **je veux [quoi]**
 - L'intention du persona
- **afin de [pourquoi]**
 - Quel problème on résout ?

Source(s) :

- https://fr.wikipedia.org/wiki/R%C3%A9cit_utilisateur

Bonne ou mauvaise Story ?

En tant qu'acheteur, je veux pouvoir payer afin de finaliser mes achats et les recevoir

Mauvaise Story

- Comment je paie ?
 - Paypal, CB, chèque... ?
- Comment sont gérés les numéros de carte ?

Mauvaise Story

En tant qu'acheteur, je veux pouvoir payer avec PayPal afin de finaliser mes achats et les recevoir

Bonne ou mauvaise Story ?

En tant qu'utilisateur, je veux chercher par
prénom d'auteur

Mauvaise Story

- **Pourquoi je dois permettre ceci ?**
 - S'il n'y a pas de raison, c'est certainement que c'est inutile

Correction

En tant qu'utilisateur, je veux chercher par auteur
afin de trouver plus rapidement le contenu qui
m'intéresse

Bonne ou mauvaise Story ?

En tant qu'enseignant, je souhaite connaître mon
emploi du temps **afin que** je sache quand venir

Il existe d'autres gabarits autres que :
« *En tant que [persona], je veux [quoi] afin de [pourquoi]* »

Mais celui-ci est le plus répandu

**Pour faire une Story simple (et bonne), on peut
s'arranger pour qu'elle respecte INVEST**

Acronyme INVEST

- **I pour Indépendant.** On peut développer ce ticket de façon autonome
- **N pour Négociable.** Les devs ne sont pas bloqués dans la solution
- **V pour Valuable.** Apporte de la valeur au produit

Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY>

Acronyme INVEST

- **E pour Estimable.** On "sait" combien de temps ça prend à réaliser
- **S pour Small.** Ça doit rentrer dans un sprint
- **T pour Testable.** On doit pouvoir s'assurer qu'on a réussi

Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY>

Pratiquons ! - User Stories (Partie 1)

Pré-requis :

- Avoir la ressource **ressources/redaction-user-stories**

A télécharger ici :

<https://download-directory.github.io/?url=https%3A%2F%2Fgithub.com%2FDanYellow%2Fcours%2Ftree%2Fmain%2Fgestion-de-projet-s3%2Ftravaux-pratiques%2Fnumero-1%2Fressources%2Fredaction-user-stories>

**Les Stories, c'est bien.
Avec des Critères d'Acceptation, c'est mieux.**

Critères d'acceptation – AC / CA

- Permettent de valider une Story
 - "T" d'INVEST. T = Testable
- Expression dans un langage simple

Critères d'acceptation – AC / CA

- Peuvent s'écrire sous le format :
"Étant donné que [état précédent/contexte],
quand [action] alors [résultat]"...
- ou une liste classique

Pratiquons ! - User Stories (Partie 2)

Pré-requis :

- Avoir la ressource **ressources/redaction-user-stories**

A télécharger ici :

<https://download-directory.github.io/?url=https%3A%2F%2Fgithub.com%2FDanYellow%2Fcours%2Ftree%2Fmain%2Fgestion-de-projet-s3%2Ftravaux-pratiques%2Fnumero-1%2Fressources%2Fredaction-user-stories>

Feature

- Grosse fonctionnalité du produit
 - Se place au-dessus d'une epic



Feature

- Exemple :
 - Feature : Gestion des vidéos
 - Epic : Suggérer des vidéos
 - Story : Afficher des vidéos en fonction des préférences

Epic

- Ensemble de stories
- Si pas importante, inutile de la découper

Prioriser les US ?

Deux méthodologies

Matrice de KANO – Méthode 1

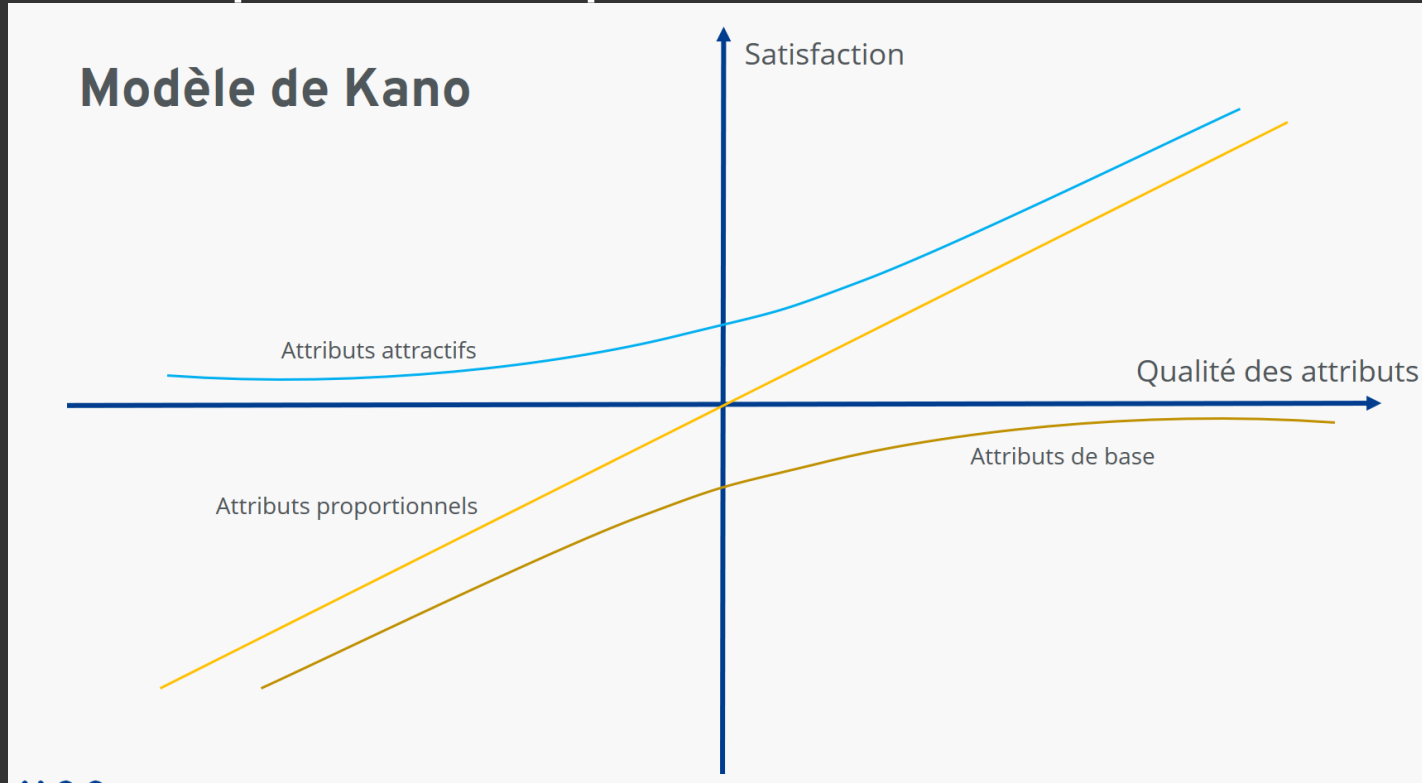
- Vise à définir cinq types de fonctionnalités :
 - **Obligatoires** : L'utilisateur doit l'avoir
 - **Attractives** : "nice to have", si c'est là tant mieux, si c'est absent, l'utilisateur ne le remarque pas
 - **Linéaires** : *game changer*. Apportent un vrai plus. Plus la fonctionnalité est aboutie, plus le client devrait être satisfait et inversement

Matrice de KANO – Méthode 1

- Vise à définir cinq types de fonctionnalités :
 - **Indifférentes** : La présence ou non de la fonctionnalités ne change rien pour l'utilisateur
 - **A double tranchant** : Peut provoquer la satisfaction ou l'inconvénient dépendamment de l'utilisateur

Matrice de KANO – Méthode 1

- Représentée par la courbe suivante :



- **Obligatoires** : à droite sur l'axe x
- Attractives : courbe en bleu
- Linéaires en jaune
- Indifférentes : à gauche sur l'axe X
- A double tranchant

Source(s) :

- <https://www.ionos.fr/startupguide/gestion/modele-kano/>

On peut réaliser un questionnaire pour les utilisateurs potentiels pour encore plus affiner

**A noter que certaines fonctionnalités peuvent
changer de statut.**

Par exemple, une haute autonomie pour un smartphone est passé d'attractif à obligatoire ou encore la commande vocale qui peut être vue comme à double tranchant

Méthode MoSCoW – Méthode 2

- Plus simple à mettre en place que la Matrice de Kano
- Quatre types de fonctionnalités :
 - Must : à faire (Urgentes)
 - Should : Importantes
 - Could : Peuvent être faites
 - Won't : A faire si on a du temps

Pratiquons ! - User Stories (Partie 3)

Pré-requis :

- Avoir la ressource **ressources/redaction-user-stories**

A télécharger ici :

<https://download-directory.github.io/?url=https%3A%2F%2Fgithub.com%2FDanYellow%2Fcours%2Ftree%2Fmain%2Fgestion-de-projet-s3%2Ftravaux-pratiques%2Fnumero-1%2Fressources%2Fredaction-user-stories>

Notion de fini – Definition of Done

- Notion globale à toutes les Stories (dans l'ensemble)
- **Différent** des critères d'acceptation
- Permet de définir quand une Story est considérée comme étant terminée. Exemple

Notion de fini – Definition of Done

- Permet de définir quand une Story est considérée comme étant terminée. Exemple :
 - La Story est accessible pour le client
 - Les textes sont traduits
 - Tous les critères d'acceptation sont vérifiés
 - ...

Notion de prêt – Definition of ready

- Permet de définir quand une Story **peut être mise dans le sprint backlog**. Exemple :
 - Tous les critères d'acceptation sont là
 - Les dépendances sont prêtes
 - Tout le monde l'a comprise (la Story)
 - ...

Source(s) :

- <https://www.youtube.com/watch?v=C9XoZQdnKbA>

Garbage in, garbage out.

**C'est ce qui arrive quand on met n'importe quoi comme
User Story dans le sprint backlog...**

En bref – (User) Story

- Description simple et **petite** d'un besoin
- Doit être terminé à la fin **d'un** sprint
 - Sinon on découpe
- Alimente le backlog

Pour terminer, sachez qu'une Story peut avoir des sous-tâches quand elle implique plusieurs corps de métiers.

Questions ?

