

# 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

# SCRUM – En bref

- Méthode Agile où l'équipe est en auto-gestion
  - Pas de chef pour surveiller
- L'équipe **s'engage** à livrer régulièrement une nouvelle version du produit pour apporter de la valeur

## Source(s) :

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

# SCRUM – En bref

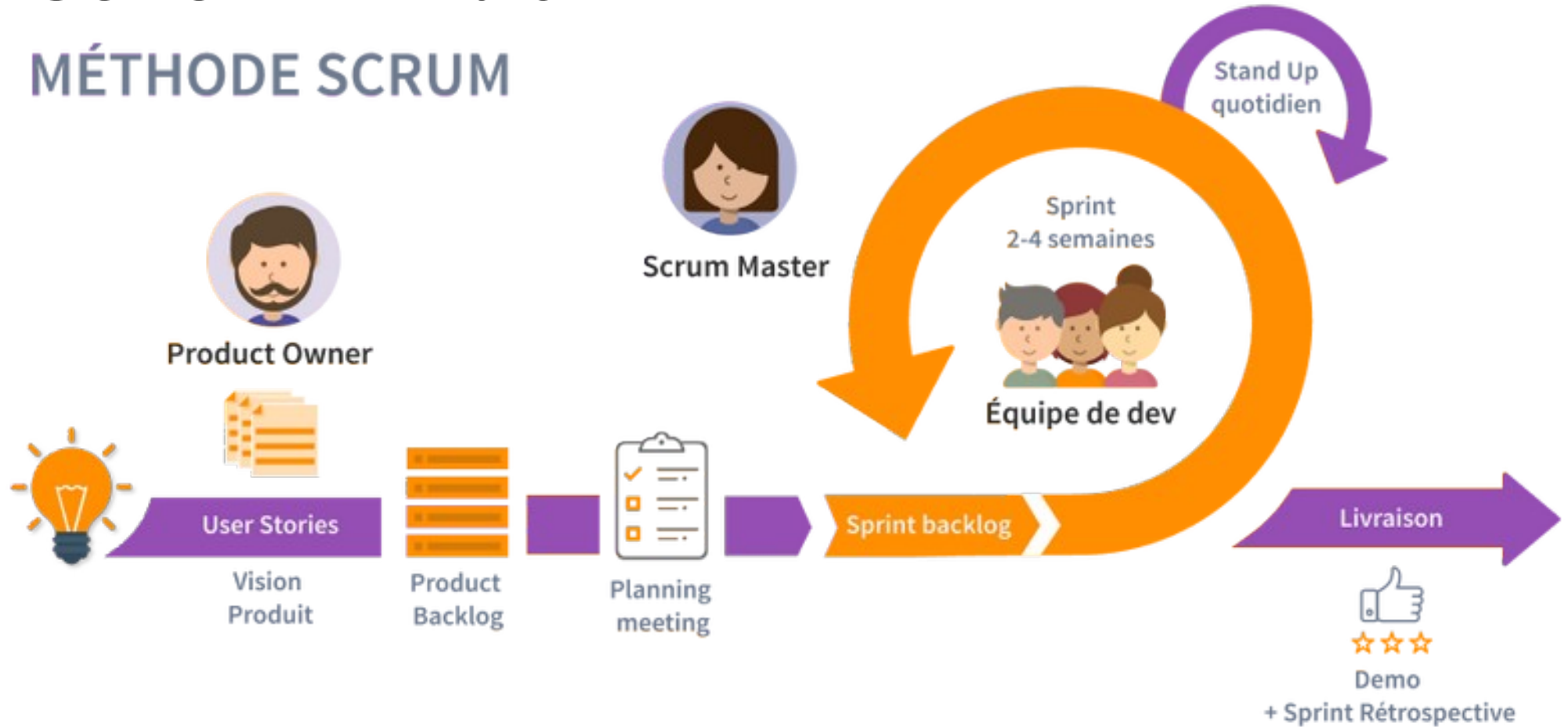
- Trois rôles possibles :
  - Producteur Owner (PO) : définit la vision du produit
  - Développeur : développe le produit
  - Scrum Master : Met en place (et adapte) SCRUM

## Source(s) :

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

# SCRUM – En bref

## MÉTHODE SCRUM



Source(s) :

- <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-French.pdf>
- <https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes>

# SCRUM – En bref

- Découpage du projet en sprint
  - Unité de temps qui doit se conclure par la livraison d'une unité de valeur du produit
- L'équipe est plus engagée et consciente du produit, grâce aux (User) Stories

## Source(s) :

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



# (User) Story

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

Source(s) :

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

# XP – eXtreme Programming

- Méthode Agile pensée pour le dev logiciel
- Ensemble de processus (contraignants) visant à développer un logiciel de qualité
- Prône 5 valeurs
  - Certaines ont été reprises dans le manifeste Agile

## Source(s) :

- <https://www.youtube.com/watch?v=L3FylT7prMo>
- [https://fr.wikipedia.org/wiki/Extreme\\_programming](https://fr.wikipedia.org/wiki/Extreme_programming)

**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

- (Autre) Rituel de SCRUM
- Parfois appelé "backlog grooming"
- Réunion où sont présentées et estimées les Stories à venir
- 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](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

**Si vous n'arrivez pas à exprimer une fonctionnalité sous forme de Story.  
Deux possibilités :**

- **Pas assez claire**
- **Pas assez pertinente**



**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

Créer un utilisateur

Ajouter un produit

Ces tâches peuvent être développées indépendamment

Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes

# Acronyme INVEST

**N pour Négociable.** Les devs ne sont pas bloqués dans la solution

Story : En tant qu'utilisateur, je souhaite noter un produit afin d'encourager l'achat

Noter sur 5

Noter sur 10

Noter sur 100

Ici, la note maximale n'est pas imposée, l'équipe de devs peut donc faire un choix.

Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes

# Acronyme INVEST

**V pour Valuable.** Apporte de la valeur au produit

Permettre la sauvegarde  
de paniers



Changer la couleur  
de l'icône de notifs



Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes

# Acronyme INVEST

**E pour Estimable.** On "sait" combien de temps ça prend à réaliser

Retirer les icônes  
de l'ancienne version



Rendre le site plus beau



Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes

# Acronyme INVEST

En tant qu'utilisateur,  
je souhaite noter un produit  
afin d'encourager l'achat



En tant qu'utilisateur,  
je souhaite noter un produit  
sur 5 étoiles  
afin d'encourager l'achat



La story de droite n'est pas négociable.  
La solution est imposée.

Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes

# Acronyme INVEST

**S pour Small.** Ça doit rentrer dans **un** sprint

Ajouter un système de tri



Développer un site



Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes

# Acronyme INVEST

**T pour Testable.** On doit pouvoir s'assurer le développement répond au besoin

Gestion du panier



- Je peux ajouter un élément
- Je peux retirer un élément
- Je peux changer la quantité
- Je peux mettre un élément de côté
- ...

Source(s) :

- <https://www.youtube.com/watch?v=PO4vDYkezgY> - Moins de 15 minutes



# 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>

# (User) Story

- Rien ne vous empêche d'ajouter une description, un contexte à la Story
  - Plus le ticket sera clair, plus faibles seront les probabilités d'un contre-sens lors du développement

Source(s) :

- [https://fr.wikipedia.org/wiki/R%C3%A9cit\\_utilisateur](https://fr.wikipedia.org/wiki/R%C3%A9cit_utilisateur)

# Job Story

- Alternative aux User Story
- Centré sur une action
  - L'User Story est centrée sur l'utilisateur
  - Aide à comprendre la motivation de l'utilisateur
- Incite à l'innovation

Source(s) :

- <https://www.slideshare.net/agabrillagues/job-stories-penser-autrement-l'expression-du-besoin>

**Une Job Story doit répondre à la question :**  
**Pour quelle(s) tâche(s) les utilisateurs**  
**"embauchent" votre produit ?**

Source(s) :

- <https://www.slideshare.net/agabrillagues/job-stories-penser-autrement-l'expression-du-besoin>

# Job Story

- **Quand [situation/contexte]**
  - Qu'est-ce qui est en train de se passer ?
- **je veux [quoi]**
  - La motivation de l'utilisateur
- **afin de [résultat]**
  - Quel problème on a résolu ?

Source(s) :

- <https://www.slideshare.net/agabrillagues/job-stories-penser-autrement-l'expression-du-besoin>

# Job Story - Exemple

Quand il est l'heure de manger  
et qu'il y a du monde à la maison,  
je veux des pizzas  
afin de manger rapidement

En tant qu'utilisateur,  
je souhaite manger des pizzas  
afin de manger rapidement

A gauche, une job story (plus de contexte).  
A droite, une user story (moins de contexte).

## Source(s) :

- <https://www.slideshare.net/agabrillagues/job-stories-penser-autrement-l'expression-du-besoin>
- [https://wikiagile.cesi.fr/index.php?title=Les\\_Job\\_Stories\\_offrent\\_une\\_alternative\\_viable\\_aux\\_User\\_Stories](https://wikiagile.cesi.fr/index.php?title=Les_Job_Stories_offrent_une_alternative_viable_aux_User_Stories)

# Job Story - Exemple

Quand je doute sur  
l'achat un produit  
je souhaite connaître  
l'avis des acheteurs  
afin de valider mon achat

En tant qu'utilisateur,  
je souhaite noter un produit  
afin d'encourager l'achat

A gauche, une job story (plus de contexte).  
A droite, une user story (moins de contexte).

Source(s) :

- <https://www.slideshare.net/agabrillagues/job-stories-penser-autrement-l'expression-du-besoin>

# Job Story - Exemple

Quand je suis ravi de mon achat,  
je souhaite le noter  
afin d'en encourager à l'achat

En tant qu'utilisateur,  
je souhaite noter un produit  
afin d'encourager l'achat

A gauche, une job story (plus de contexte).  
A droite, une user story (moins de contexte).

Source(s) :

- <https://www.slideshare.net/agabrillagues/job-stories-penser-autrement-l'expression-du-besoin>



**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
- **Ne doivent contenir que des points fonctionnels**
  - Le résultat doit être visible

# Critères d'acceptation – AC / CA

- Peuvent s'écrire sous le format :  
"Étant donné que [état précédent/contexte], quand [action] alors [résultat]"
  - Plusieurs actions (quand) et résultats (alors) peuvent être simultanées
- ou une liste d'éléments
  - A privilégier, par exemple, quand on veut définir les règles de validation de champs de formulaire

# Critères d'acceptation – AC / CA

## Gestion du panier

En tant qu'utilisateur,  
je souhaite savoir ce que  
je m'apprête à acheter  
de gérer mon budget

- Étant donné que je suis sur une fiche produit quand j'ajoute le produit alors le panier a ce produit
- Étant donné que je suis dans le panier quand je décrément la quantité d'un produit et que la valeur passe à zéro alors ce produit disparaît du panier

# 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>

**Du mal à rédiger vos critères d'acceptation ?  
Pensez SMART.**

# Acronyme SMART

**S pour Spécifique.** Le critère doit suffisamment précis pour être compris

**M pour Mesurable.** Il existe un moyen de valider le critère

**A pour Atteignable.** On doit pouvoir le réaliser

- Il existe des variantes pour le A

Source(s) :

- [https://fr.wikipedia.org/wiki/Objectifs\\_et\\_indicateurs\\_SMART](https://fr.wikipedia.org/wiki/Objectifs_et_indicateurs_SMART)

# Acronyme SMART

**R pour Réaliste.** Doit réalisable et ne pas démotiver

**T pour Temporaire.** Doit être réalisé dans un temps défini

Source(s) :

- [https://fr.wikipedia.org/wiki/Objectifs\\_et\\_indicateurs\\_SMART](https://fr.wikipedia.org/wiki/Objectifs_et_indicateurs_SMART)



## Critères d'acceptation – AC / CA

- Permettent de définir les tests unitaires
  - ça concerne plutôt ceux qui codent

# 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
- A découper que si on compte travailler prochainement dessus

**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

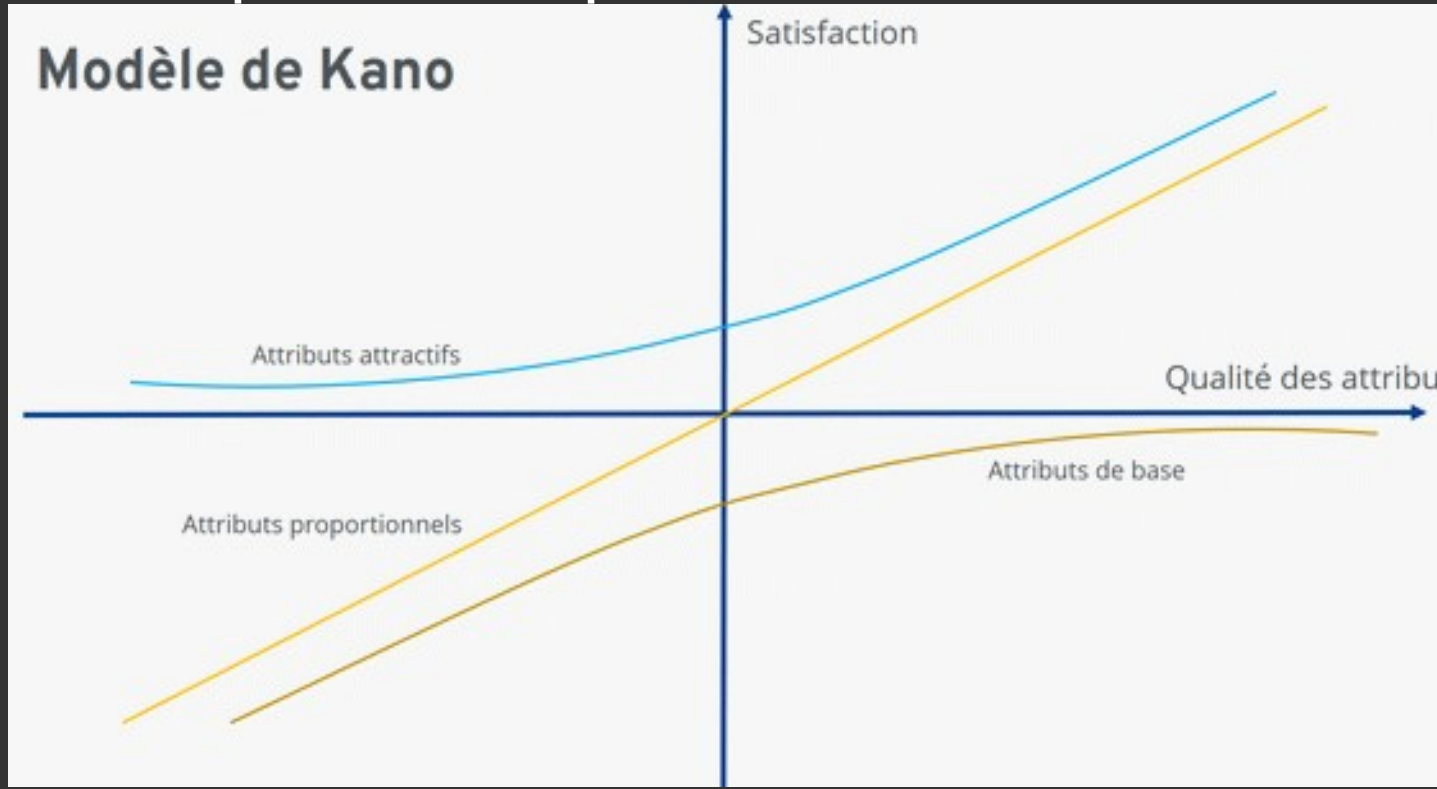
# Matrice de KANO – Méthode 1

- 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 (DoD)

- Notion globale à toutes les Stories (dans l'ensemble)
- **Différent** des critères d'acceptation
  - Le DoD complète les CA

# Notion de fini – Definition of Done (DoD)

- 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 externes sont prêtes
  - Tout le monde l'a comprise (la Story)
  - ...

Source(s) :

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

**Comme les Stories, la Définition of Ready (DoR)  
et Définition of Done (DoD) ça s'affine.  
Elles peuvent changer durant le cycle de vie du  
projet.**

***Garbage in, garbage out.***

*(Entrée pourrie, sortie pourrie)*

**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 ?**

