

# INVEST

## Ecrire une bonne US

Adrien CAUBEL

# L'acronyme

# Définition

« Les caractéristiques d'une story peuvent se résumer avec l'acronyme INVEST »

- Indépendante
- Négociable
- Valuable
- Estimable
- Small
- Testable

# Indépendante

# Indépendante

Indépendance par rapport à quoi ? (à votre avis)

« L'indépendance signifie qu'une user story peut être développé, testé et éventuellement livré seul. Par conséquent, elle apporte de la valeur de manière indépendante »

Apporter de la valeur, l'erreur est d'avoir des US techniques (donc dépendantes)  
Il faut une indépendance technique

# Indépendante : exemple

En tant qu'administrateur, je peux définir les règles de sécurité des mots de passe du consommateur afin que les utilisateurs soient tenus de créer et conserver des mots de passe sûrs.

En tant que consommateur, je suis tenu de respecter les règles de sécurité des mots de passe définies par l'administrateur afin de maintenir un niveau de sécurité élevé pour mon compte.

Est-ce que j'ai une dépendance ?

# Indépendante : exemple

En tant qu'administrateur, je peux définir le délai d'expiration du mot de passe de manière à ce que les utilisateurs soient obligés de changer leur mot de passe périodiquement.

En tant qu'administrateur, je peux définir les caractéristiques de force du mot de passe afin que les utilisateurs soient obligés de créer des mots de passe difficiles à pirater

Est-ce que j'ai une dépendance ?

- Chaque story est autonome et peut être développée, testée et déployée de manière indépendante.
- Chaque story apporte sa propre valeur et elles ne sont pas obligées d'être implémentées dans un même sprint.

# Indépendance

- L'indépendance est donc **au niveau technique**

En tant qu'utilisateur  
Je veux accéder à mon compte

L'accès à un compte  
Se fait de manière sécurisée (auth)

Fonctionnellement ça n'aurait pas de sens de se connecter de manière non sécurisée.

⇒ Ca ne serait pas un produit qu'on livrerait au client

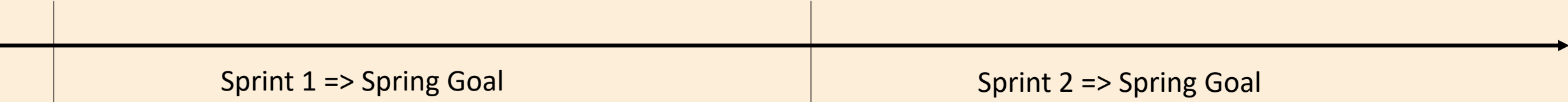
⇒ MAIS de manière purement technique on peut les construire dans n'importe quel ordre



# Indépendance

En tant qu'utilisateur  
Je veux accéder à mon compte

L'accès à un compte  
Se fait de manière sécurisée (auth)



Sprint 1 => Spring Goal

Sprint 2 => Spring Goal

Note importante : en soit que les US soient indépendantes c'est pas une fin. Il faut plutôt se concentrer sur le sprint (apporter de la valeur). Si lors d'un Sprint des US sont dépendantes ce n'est pas « la mort »

1. Vision produit
2. Sprint Goal
3. Puis influencer le backlog et les US

# Négociable

# Negociable

Négocier quoi ? (à votre avis)

« Une user story n'est pas un contrat mais une discussion sur les exigences à développer, tester et déployer. »

- Une US laisse de la liberté par rapport à son implémentation  
On ne dit pas à l'avance comment faire => laisser les ingénieurs techniques le faire
- Flexible dans son périmètre si on se rend compte qu'on ne va pas finir

# Valeur

# Valuable

« L'objectif de l'agilité est de délivrer un produit de haute qualité dans une contrainte de temps et de ressource. »

- Chaque user-story doit apporter de la valeur à l'utilisateur, au client ou aux parties prenantes.
- Orienté utilisateur d'où le « En tant que [Utilisateur] »

# Valuable : exemple

Refactorer le système de log des erreurs

En tant que consommateur,  
Je peux recevoir un message d'erreur clair et  
cohérent à n'importe quel endroit du produit,  
Afin que je puisse de savoir comment résoudre  
le problème

Devient

**Estimable**

# Estimable

« L'user story contient le nécessaire pour que l'équipe de développement puisse l'estimer »

- Si l'équipe n'est pas capable de donner une estimation cela signifie qu'elle est trop grosse et doit être redécoupée
- Est-ce qu'on a bien compris cette story ??



# Estimable : exemple

En tant que qu'acheteur  
Je veux pouvoir payer  
Afin de finaliser mes achats

Est-ce estimable ? (à votre avis)

- Payer comment ? CB
- Doit-on gérer CB => RGPD
- Doit-on avoir une double sécurisation ?
- On utilise paypal => appel d'api ?

- On a bien compris le besoin :
  - Mais il faut faire d'autres discussion avant de donner une estimation

Small

# Small

« Une story doit être assez petite pour être développée en une itération. »

- facilite l'estimation
- apporte de la valeur à la fin du sprint
- permet d'avoir un feedback spécifique
- si on s'est planté alors ça ne sera que sur une petite partie.

ATTENTION : si on découpe ce n'est pas pour que ça rentre dans un sprint; ça ne doit pas être la raison  
**Découper == pour maximiser de la valeur où on se focus sur ce qui a le plus de valeur aujourd'hui et on fera la suite plus tard**

# Testable

# Testable

« Tester permet de vérifier le travail réalisé »

Si une story n'est pas testable cela signifie :

- qu'elle est trop grosse ou trop complexe
- qu'elle dépend d'une autre story dans le backlog

Cela se traduit par le « Afin que » => mettre une fin visible

# UseCase vs US

- <https://www.qualitystreet.fr/2009/02/16/user-story-vs-use-case-soyez-agile/>
- <https://www.qualitystreet.fr/2007/03/30/user-stories-use-cases-les-differences/>

# US vs UC : définitions

- Une « **User story** » est une exigence du système à développer, formulée en une ou deux phrases dans le langage de l'utilisateur.
- Un « **Use case** » modélise un service rendu par le système. Il représente un ensemble de séquences d'actions qu'un système ou toute autre entité peut accomplir en interagissant avec les acteurs du système.

# UC vs US : objectifs

- **US** : Rôle vers un unique But (pas une séquence d'actions)
- **UC** : Beaucoup plus riche
  - Titre
  - Lié à des acteurs
  - Propose un résumé
  - Scénario nominal
  - Cas d'erreur



# UC vs US : dépendances

- **US** : Indépendantes techniquement
- **UC** : Visibles via Diagramme des cas d'utilisation

# UC vs US : taille

- **US** : Traité en 1 itération (sprint)
- **UC** : Peut être traité sur plusieurs itérations (scénario nominal sur une, scénarios alternatifs sur une autre) en fonction des risques à lever.

# UC vs US

USER STORY	USE CASE
Est une brève description d'une fonctionnalité telle que vue par l'utilisateur (Définition)	Représente une séquences d'actions qu'un système ou toute autre entité peut accomplir en interagissant avec les acteurs du système (Définition)
Format écrit <b>court</b> , laissant la part belle à la <b>discussion orale</b>	Format écrit <b>très riche en informations</b> (pré conditions, Événement déclencheur, scénario principal, alternatives ...). Peu de place à l'oral
Est une <b>partie d'un Use Case</b> (le scénario principal <b>ou</b> une alternative)	Est la somme d'un scénario principal, et de diverses alternatives (variations, cas d'erreur ...)
Utilisée certes en tant que spécification mais <b>surtout pour l'estimation &amp; la planification</b>	Utilisé seulement en tant que spécification
Emergence rapide au travers d'ateliers de travail collaboratifs	<b>Long travail d'analyse et de formalisation</b>
<b>Grande lisibilité</b> du fait de sa simplicité	Manque de lisibilité même au sein d'un Template
<b>Mode ORAL et COLLABORATIF</b>	<b>Mode ECRIT ET DISTANT</b>
Très facile à maintenir (format fiche, court, indépendant)	Difficile à maintenir (Doc. Word de 150 pages)
Implémentée et testée <b>en une itération obligatoirement</b>	Implémenté et testé en une ou plusieurs itérations
Ecrite facilement par un Utilisateur ou un Client (accompagné dans sa démarche)	<b>Souvent rédigé par des User Proxies</b> (AMOA, Analyste ...), rarement par le client
<b>Contient des tests d'acceptation</b> (au dos de la carte) = > implication des testeurs	Ne contient pas les cas de test qui en découlent => pas d'implication des testeurs
Difficile à lier les unes aux autres. Absence de vue globale	<b>Liaison et vision globale facilitée</b> : Sous Use Case, conditions préalables, Diagramme des cas utilisation
Associée historiquement à <b>eXtreme Programming et aux méthodes Agiles</b>	Associé historiquement au Processus Unifié
Un auteur de référence : <b>Mike Cohn</b>	Un auteur de référence : <b>Alistair Cockburn</b>

# UC vs US

USER STORY	USE CASE
Est une brève description d'une fonctionnalité telle que vue par l'utilisateur (Définition)	Représente une séquences d'actions qu'un système ou toute autre entité peut accomplir en interagissant avec les acteurs du système (Définition)
Format écrit <b>court</b> , laissant la part belle à la <b>discussion orale</b>	Format écrit <b>très riche en informations</b> (pré conditions, Événement déclencheur, scénario principal, alternatives ...). Peu de place à l'oral
Est une <b>partie d'un Use Case</b> (le scénario principal <b>ou</b> une alternative)	Est la somme d'un scénario principal, et de diverses alternatives (variations, cas d'erreur ...)
Utilisée certes en tant que spécification mais <b>surtout pour l'estimation &amp; la planification</b>	Utilisé seulement en tant que spécification
Emergence rapide au travers d'ateliers de travail collaboratifs	<b>Long travail d'analyse et de formalisation</b>
<b>Grande lisibilité</b> du fait de sa simplicité	Manque de lisibilité même au sein d'un Template
<b>Mode ORAL et COLLABORATIF</b>	<b>Mode ECRIT ET DISTANT</b>
Très facile à maintenir (format fiche, court, indépendant)	Difficile à maintenir (Doc. Word de 150 pages)
Implémentée et testée <b>en une itération obligatoirement</b>	Implémenté et testé en une ou plusieurs itérations
Ecrite facilement par un Utilisateur ou un Client (accompagné dans sa démarche)	<b>Souvent rédigé par des User Proxies</b> (AMOA, Analyste ...), rarement par le client
<b>Contient des tests d'acceptation</b> (au dos de la carte) = > implication des testeurs	Ne contient pas les cas de test qui en découlent => pas d'implication des testeurs
Difficile à lier les unes aux autres. Absence de vue globale	<b>Liaison et vision globale facilitée</b> : Sous Use Case, conditions préalables, Diagramme des cas utilisation
Associée historiquement à <b>eXtreme Programming et aux méthodes Agiles</b>	Associé historiquement au Processus Unifié
Un auteur de référence : <b>Mike Cohn</b>	Un auteur de référence : <b>Alistair Cockburn</b>