User Story

As a {who},
I want {what}
Because {why}

1. Définition

Une User Story est **l'expression d'un besoin utilisateur**. Elle est simplement constituée d'une phrase de description qui exprime un besoin avec une vue utilisateur. Cette phrase sert de support à la discussion, elle est complétée d'un ensemble de conditions d'acceptation qui en définissent le périmètre exact du besoin.

Les objectifs d'une User Story sont les suivants :

- Être un support de communication. Pour cela, elle ne sera pas trop détaillée.
- Être compréhensible par toute l'équipe (technique et fonctionnelle). Pour cela, on n'utilisera pas des termes trop techniques, et on préférera le langage de l'utilisateur à celui du développeur.
- Être un support de pilotage de l'avancement du projet. Pour cela, on préférera les histoires de petite taille.

L'expression d'une User Story respecte en général **le formalisme** suivant :

En tant que *<rôle de l'utilisateur>*
Je veux *<description du besoin>*
Afin de *<objectif de la demande>*

Exemple de User Story « Créer un compte étudiant »

En tant que *Bibliothécaire*
Je veux *créer un compte étudiant*
Afin que *cet étudiant devienne adhérent de la bibliothèque*

Exemple de User Story « Se connecter au réseau social »

En tant que *Membre*
Je veux *me connecter au réseau social du département informatique de l'IUT*
Afin d'*accéder à tous les services proposés par le réseau social*

Ce formalisme permet de s'assurer que les points principaux ont été réfléchis, à savoir :

- Qui est le demandeur principal du besoin ?
- Quelle est la demande ?
- Dans quel but fait-on cette demande ?

2. « Investir » ses User Stories

Pour pouvoir être considéré comme "User Story", un besoin utilisateur doit autant que possible respecter un certain nombre de contraintes rassemblées sous la dénomination mnémotechnique **INVEST.**

Indépendante	Le critère le plus difficile à respecter. Toutes les User Stories du Product Backlog doivent être autant que possible indépendantes les unes des autres. En effet pour garder toute sa souplesse au Product Backlog, nous ne devons pas conditionner la réalisation d'une User Story à l'existence préalable d'une autre User Story.
Négociable	Une User story doit rester ouverte à la discussion jusqu'à la fin de sa réalisation. Ceci veut dire qu'une User Story est une base de discussion, et qu'on s'autorise à la faire évoluer en fonction du besoin afin de trouver la meilleure implémentation possible du besoin initial. Cela en vue de la simplification des besoins techniques et d'une meilleure satisfaction client.
Valorisable	Il doit être possible de donner une valeur métier à toute User Story. Ceci signifie que toute User Story doit être compréhensible et avoir un sens pour un utilisateur final.
Estimable	Le périmètre de la User Story doit être suffisamment défini pour pouvoir la chiffrer (ou l'estimer).
Suffisamment petite (Small)	La granularité de la User Story doit être faible. Il faut que l'équipe soit en mesure d'embarquer un minimum de 4 à 5 User Stories par sprint. En effet en Scrum, le décompte de l'avancement du projet se fait sur des éléments finis. Si l'équipe n'embarque qu'une User Story dans son sprint, et qu'elle la termine à 90%, l'avancement sera considéré comme nul, ce qui n'est pas la vérité.
Testable	Les User Stories doivent être testables par un utilisateur. Ceci peut avoir un sens très différent en fonction du type d'application, mais cela reste très lié au fait que la User Story doit avoir une valeur métier et donc rendre un service à l'utilisateur.

3. Rendre les user stories suffisamment petites

Une user story doit être suffisamment petite pour pouvoir être implémentée lors d'une itération (on pourra d'ailleurs réaliser plusieurs user stories dans l'itération). Elle doit également être suffisamment petite pour pouvoir être estimée en termes de valeur et d'effort. Donc, si une story est trop grosse, il faut la découper en plusieurs user stories.

Pour découper une user story (voire un cas d'utilisation) en plusieurs stories plus petites, il existe plusieurs stratégie de décomposition :

1. Une user story par scénarios :

   Si une grosse user story représente un cas d'utilisation qui regroupe beaucoup de scénarios, il ne semble pas raisonnable de développer tous ces scénarios dans une seule et même itération. Il est possible de décomposer notre grosse user story en plusieurs petites stories, une pour chacun des scénarios du cas d'utilisation. Une story pour le scénario nominal où tout se passe bien (et on arrive au but par le chemin normal), une ou des stories pour les scénarios alternatifs (où on arrive au but voulu mais par des chemins détournés), une ou plusieurs stories pour les scénarios d'échec (où on n'atteint pas le but du cas d'utilisation).

   - Exemple de la bibliothèque

   Le cas d'utilisation « Prolonger un emprunt » peut éventuellement être décomposé si on estime qu'il est trop gros.
   On pourra avoir la user story qui correspond au le scénario nominal où la prolongation de l'emprunt est acceptée et où tout se passe bien.
   Et la user story qui correspond au scénario d'erreur où la prolongation de l'emprunt est refusée parce que le livre est déjà réservé.

   Bref des scénarios il y en a beaucoup, on ne pourra sans doute pas tous les décrire. Il faudra faire des choix en ne décrivant que ceux qui nous semblent essentiels. Attention tout de même à ne pas omettre des scénarios qui semblent évidents mais qui ne sont pas forcément évidents pour tout le monde.

2. Une user story par séquence d'actions :

   Si une user story représente un seul scénario mais qu'il est très (trop) long en terme d'interactions, tellement long qu'on va avoir du mal à l'estimer et à le développer en une seul itération, on la découpe en plusieurs stories indépendantes, chacune d'entre elles regroupera un sous-ensemble des séquences d'actions du scénario original.

   - Exemple de la bibliothèque

   Le scénario « Enregistrer un emprunt » peut éventuellement être décomposé si on estime qu'il est trop gros.
   On pourra avoir la user story où on vérifie que le livre est disponible, la user story où on vérifie que l'adhérent a le droit d'emprunter un nouveau livre (il n'a pas déjà atteint son quota maximum d'emprunts) et la user story où on enregistre l'emprunt.

3. Une user story par opération :

   Si une user story représente un cas d'utilisation qui regroupe plusieurs scénarios nominaux (où tout se passe bien) mais effectuant des opérations différentes sur une entité données, il est possible de décomposer ce cas d'utilisation en plusieurs user stories, une pour chaque des opérations qui sont réalisées sur l'entité.

   - Exemple de la bibliothèque

   Le cas d'utilisation « Gérer les comptes étudiant », englobe les scénarios où le bibliothécaire crée un compte, où il supprime un compte, où il modifie un compte (en UML on parle parfois de cas d'utilisation CRUD).
   Il est possible de décomposer ce cas d'utilisation en trois user stories : la user story créer un compte étudiant, la user story supprimer un compte étudiant, et enfin, la user story modifier un compte étudiant.

4. Une user story pour chaque forme d'opération :

   Si une user story représente un cas d'utilisation qui regroupe plusieurs scénarios qui réalisent la même chose mais qui ont une forme différente (en UML on parle parfois de cas polymorphes) ou nécessitent des données différentes. Il est possible de décomposer ce cas d'utilisation en plusieurs user stories, une pour chacune des formes.

   - Exemple de la bibliothèque

   Le cas d'utilisation « Rechercher un livre », peut englober des scénarios où le l'adhérent réalise des recherches en fonctions de données en entrée qui sont différentes. Par exemple une recherche par mots-clés, une recherche par catégorie, une recherche par auteur, ...
   On peut donc écrire trois user stories : rechercher un livre par mot mots-clés, rechercher un livre par catégorie, rechercher un livre par auteur.

5. Une user story par type d'entrée ou sortie :

   Si une user story représente plusieurs scénarios qui réalisent la même chose avec les mêmes données mais dont les données en entrée ou en sortie peuvent avoir des formats différents. Il est possible de décomposer ce scénario en plusieurs user stories, une pour chacune type.

   - Exemple de la bibliothèque

   Le cas d'utilisation « Consulter l'historique » peut donner des données en sorties de deux types différents. Soit les données sont affichées sur une page du site, soit dans un pdf. On peut donc écrire deux user stories : afficher l'historique sur une page et afficher l'historique dans un pdf.

6. Une user story par type acteur :

   Si une user story est partagée par plusieurs acteurs mais que pour chacun de ces acteurs l'interface graphique du scénario diffère, on peut la décomposer en plusieurs user stories (une par acteur).

   - Exemple de la bibliothèque

   On a distingué la user story « Consulter son compte » pour l'étudiant et l'enseignant car l'interface graphique est différente.

4. Comment rédiger les tests d'acceptation des user stories

Exemple de user story « Ajouter un compte étudiant »

En tant que *Bibliothécaire*
Je veux *créer un compte étudiant*
Afin que *cet étudiant devienne adhérent de la bibliothèque*

Exemple de tests d'acceptation :

- *Le bibliothécaire est connecté au site.*
- *Le bibliothécaire accède au formulaire de saisie d'un nouveau compte étudiant.*
- *Le bibliothécaire saisit le numéro étudiant, le nom, le prénom, l'adresse et l'adresse mail du nouvel adhérent.*
- *Le bibliothécaire clique sur le bouton « ajouter ».*
- *Le système vérifie que les données sont valides.*
- *Le numéro étudiant, le nom et l'adresse mail sont obligatoires.*
- *Si certaines données ne sont pas valides, le système affiche un message en indiquant les données non valides.*
- *Le système affiche le message « Le compte a été créé ».*
- *L'étudiant reçoit un email avec ses identifiants de connexion (adresse email, mot de passe).*

Si les développeurs ne sont pas suffisamment expérimentés, on peut détailler les tests d'acceptation.

Exemple plus détaillé de tests d'acceptation :

- *Le bibliothécaire est connecté au site.*
- *Le bibliothécaire accède au formulaire de saisie d'un nouveau compte étudiant.*
- *Le bibliothécaire saisit le numéro étudiant, le nom, le prénom, l'adresse et l'adresse mail du nouvel adhérent.*
- *Le bibliothécaire clique sur le bouton « ajouter ».*
- *Le système vérifie que les données sont valides.*
  - ⇒ *Le numéro étudiant, le nom et l'adresse mail sont obligatoires.*
  - ⇒ *Les champs obligatoires doivent être marqués par une étoile *.*
  - ⇒ *Le numéro étudiant est composé d'une lettre et de onze chiffres.*
  - ⇒ *Le nom et le prénom ne contiennent pas de chiffres.*
  - ⇒ *Le nom doit contenir entre 2 et 40 caractères.*
  - ⇒ *L'adresse est composée au minimum d'un nom de rue, d'un code postal et d'une ville.*
  - ⇒ *Le code postal est composé de cinq chiffres.*
  - ⇒ *La ville ne peut pas contenir de chiffres.*
  - ⇒ *L'adresse mail doit avoir le format attendu (une partie locale, un @ et un nom de domaine).*
- *Si certaines données ne sont pas valides, le système affiche un message en indiquant les données non valides et les données saisies dans le formulaire ne sont pas effacées.*
- *Le système affiche le message « Le compte a été créé ».*
- *L'étudiant reçoit un email avec ses identifiants de connexion (adresse mail, mot de passe).*

Exemple de tests d'acceptation plus formalisé
On peut utiliser le format GWT (Given When Then), en français Etant donné Quand Alors. Ce format permet d'automatiser les tests d'acceptation dans un environnement tel qu'Eclipse.

User Story « Se connecter au réseau social »

En tant que Membre
Je veux me connecter au réseau social du département informatique de l'IUT
Afin d'accéder à tous les services proposés par le réseau social

Exemples de tests d'acceptation :
Etant donné Le membre n'est pas connecté
Et Le membre est sur la page d'accueil du site
Quand Le membre demande à se connecter
Alors La page de connexion est affichée

Etant donné Le membre n'est pas connecté
Et Le membre est sur la page de connexion
Quand Le membre saisit son identifiant
Et Le membre saisit son mot de passe
Et Le membre clique sur « connexion »
Alors La page d'accueil de connexion du site s'affiche
Et L'identifiant du membre est affiché à droite dans le bandeau

5. Prioriser

a. De 1 à 9
b. MoSCoW (MSCW) - En français c'est le FISPE
   M: Must have
   S: Should have
   C: Could have
   W: Want to have but won't have.
c. Suite de Fibbonacci
d. Nombre premier.
e. Et encore d'autre.
...
p.