

User Stories

Une User Story est une description simple d'un besoin, exprimée par un utilisateur, pour déterminer les fonctionnalités à développer. Elle vise une fonctionnalité précise et décrit comment l'utilisateur interagira avec le système une fois celui-ci terminé

Une User Story possède un titre (court), qui permet de la distinguer des autres sans risque de se tromper. Elle s'articule ensuite en quatre parties: **Description**, **Tests d'acceptance**, **Tâches** et **DoD**.

Description

La description énonce le besoin lui-même, mais aussi **qui** a ce besoin et avec quelle **intention**.

On utilise généralement la structure suivante pour formuler la description:

En tant que ...
Je veux ...
Pour ...

Ne soyons pas dogmatiques, cette structure n'a pas besoin d'être conservée à la lettre ! Elle doit servir avant tout de rappel à ce qui doit figurer dans la story.

Exemple 'à la lettre' :

En tant qu'utilisateur,
je veux m'authentifier auprès du système
pour pouvoir accéder à mes données.

Variante 1, où on considère l'intention du login comme évidente:

En tant qu'utilisateur,
je veux m'authentifier auprès du système

Variante 2, où on n'utilise pas la structure "En tant que ...":

Les utilisateurs peuvent s'authentifier sur le système

Tests d'acceptance

Les tests d'acceptance sont le résultat de l'analyse fonctionnelle de la story. Ce sont eux qui décrivent comment le système va fonctionner pour répondre aux attentes du client, en racontant comment le système se comporte quand l'utilisateur agit sur lui.

Commençons par une définition utile ici (mais pas que):

Une assertion est une phrase qui dit quelque chose que l'on soutient comme vraie

Exemple d'assertion: "Je suis rentré chez moi avant minuit hier soir"

Un test d'acceptance dans le contexte d'une user story est une assertion concernant le système fini. On s'imagine qu'on a terminé le développement de notre produit et on formule une assertion au sujet d'un utilisateur en train de l'utiliser

Exemple: "Dans ma page de profil, quand je clique sur ma photo de profil, elle s'agrandit pour prendre tout l'espace disponible."

Dans un test d'acceptance, l'assertion doit contenir trois éléments:

1. Le contexte de départ
2. L'action effectuée par l'utilisateur
3. La réaction du système. Attention: il doit s'agir de quelque chose que l'utilisateur peut constater (exemple à ne **pas** faire: "~~... le nouveau nom est stocké dans la base de données~~")

Astuces pour vous simplifier l'écriture des tests:

- On peut faire référence à des maquettes, autant pour le contexte que pour le résultat. Cela vous économisera beaucoup de texte
- Mettez-vous à la place de l'utilisateur final et rédigez vos tests en "je" ! C'est plus facile à écrire

Une user story est complète et prête à être codée quand elle a (en général) avoir entre 2 et 10 tests d'acceptance. Pourquoi ces chiffres ? parce que s'il n'y a qu'un test, cela veut dire que la story est trop petite (=pas ambitieuse) et si elle en a plus que 10, elle est trop complexe.

--- ATTENTION ---

Il est de la plus haute importance de valider les tests d'acceptance avec le client avant d'aller plus loin!!!

Si vous ne le faites pas, vous courez le risque de faire un travail inutile.

Voici quelques exemples de tests bien formulés

Sur la page d'accueil du site (maquette 1),
quand je clique le bouton 'Login',
le formulaire d'authentification s'ouvre (maquette 2)

Dans le formulaire d'authentification avec des champs vides (maquette 2)
quand le clique 'Ok'
le formulaire reste ouvert et des messages d'erreur sont affichés (maquette 3)

Dans le formulaire d'authentification avec des champs contenant des identifiants incorrects
quand le clique 'Ok'
le formulaire reste ouvert et des messages d'erreur sont affichés (maquette 4)

Dans le formulaire d'authentification avec des champs contenant des identifiants valables quand le clique 'Ok'
Je retourne à l'accueil. Mon nom d'utilisateur et un bouton de déconnexion sont visibles (maquette 5)

Tâches

La liste des tâches est le résultat de l'analyse technique. Elle décrit le travail à effectuer pour arriver au point où on pourra exécuter chacun des tests d'acceptance sur le système.

Pour chaque tâche, estimez et notez le temps dont vous pensez avoir besoin pour sa réalisation.

Exemple de liste suite à l'analyse technique:

- Créer le formulaire d'authentification (15min)
- Script de modification de base de données pour ajouter le champ 'password' (15min)
- Traitement du formulaire bien rempli (10min)
- Traitement du formulaire mal rempli (10min)

Durant le développement, il est très probable que vous allez identifier des tâches auxquelles vous n'avez pas pensé durant l'analyse technique. J'appelle cela des "surprises". Ajoutez-les à la suite de la liste, avec un tag "surprise".

Lorsqu'une tâche est terminée, notez le temps qu'elle vous a effectivement pris. Si ce temps est très différent de votre estimation, ajoutez un tag:

- "facile" si le temps effectif est moins que la moitié du temps estimé
- "durdur" si le temps effectif est plus du double du temps estimé

Ainsi, votre liste de tâches à la fin de la story pourrait ressembler à ceci:

- Créer le formulaire d'authentification (15min/10min)
- Script de modification de base de données pour ajouter le champ 'password' (15min/20min)
- Traitement du formulaire bien rempli (10min/10min)
- Traitement du formulaire mal rempli (10min/30min)**durdur**
- Mise à jour du MLD dans la documentation (5min/5min)**surprise**

Moins vous aurez de **surprise**, **durdur** et **facile** à la fin de vos stories, meilleur vous serez !

DoD

La DoD (Definition of Done) est une liste de points qui doivent être vérifiés pour que l'on puisse déclaré avoir terminé le travail sur la story.

Exemple:

- Tests d'acceptance validés en présence du client
- Documentation technique mise à jour
- Commits Git publiés (push)
- Revue de code effectuée avec un collègue