

Projet dirigé

Étude des Besoins, suite

Formaliser les besoins par les user-stories

Plan de la séance

- Retour sur la séance précédente
- Les user-stories, définition
- La propriété INVEST
- Les tests d'acceptation
- Détailler une user-story
- User-stories VS les cas d'utilisation
- Exercices

Les user-stories, définition

- Définition

Une **user-story** ou un **scénario** est une exigence du système à développer formulée en une ou deux phrases dans le langage des utilisateurs pour servir un but.

Sa granularité doit permettre à l'équipe de réalisation d'estimer son coût et de la réaliser entièrement à l'intérieur d'une itération.

- Avantage:

Elles facilitent la démarche en deux temps : générales et grossières au début, elles s'enrichissent ensuite d'exemples, de tests d'acceptation. Elles facilitent la communication, l'ajout ou la suppression de détails.

Les user-stories, définition

- Écriture narrative:

En tant que <rôle>
Je veux <liste de tâches>
Afin de <valeur ajoutée ou résultat>

- Exemples:

En tant qu'étudiant, je veux me connecter à Colnet **pour** consulter ma note en KBE pour le mini-test1

En tant que joueur, je veux connaître mon solde en écu **afin de** pouvoir acheter un item.

En tant qu'étudiant, je souhaite accéder à Colnet **afin de** changer mon horaire de la session en cours.

La propriété INVEST

INVEST: Une story doit respecter la propriété INVEST pour être considérée comme une bonne story:

- I, Indépendante
- N, Négociable
- V, Valuable
- E, Estimable
- S, Suffisamment petit
- T, Testable.

La propriété INVEST

- Indépendante: lorsque le client peut en toute liberté décider de l'ordre dans lequel les scénarios (story) est implémenté sans qu'interviennent des contraintes techniques.
- Négociable: L'équipe de développement n'est pas contrainte par la manière dont sera implémenté le scénario : Elle a la latitude d'imaginer une solution efficace. Éviter de dire :en cliquant sur un bouton, en choisissant telle option etc..
- Valuable : (V pour Valuable , a une valeur ajoutée) un bon scénario ou story décrit une fonctionnalité complète de l'application dont le client apprécie l'intérêt à l'intérieur d'une itération:
- Estimable: Connaitre le coût d'implémentation **(en points)**
- Suffisamment petit: pour pouvoir l'estimer, la story doit être petite. (terminée à l'intérieur d'une itération)
- Testable: la tester pour dire qu'elle est terminée → **prévoir des tests d'acceptation.**

Sur le site dur cours, il y a des exemples que vous devez consulter (pages 22 à 27)

Tests d'acceptation pour « Testable »

Ron Jeffries (co-fondateur de XP pour eXtreme Programming) définit une story par trois composants :

- La carte pour enregistrer son titre;
- La conversation pour la raconter, (détailler la user-story pour pouvoir l'estimer)
- La confirmation pour s'assurer qu'elle est finie : **Le test d'acceptation**

T pour Testable : planifier un test d'acceptation est un excellent moyen pour vérifier que tout le monde a bien compris le scénario ou la story. C'est un excellent moyen de déterminer quand est-ce que la story est terminée.

- Les tests d'acceptation sont les tests qui garantissent que le produit final est conforme aux exigences du client.
- En général, ils sont rédigés sous forme de scénario ou sous forme de liste.

Tests d'acceptation pour « Testable »

- Une story doit être terminée par un test d'acceptation. Un test d'acceptation est la pratique qui permet d'accepter une story à la fin d'une itération. Il consiste en :
 - **Décrire le comportement attendu avec les conditions de satisfaction**
 - Transformer ces conditions en cas de test : **storytest**
 - Écrire le code applicatif qui répond au comportement attendu
 - Passer la storytest sur le code applicatif. En cas d'échec, corriger les tests ou le code. En cas de succès la story est terminée.

Tests d'acceptation pour « Testable »

Exemple 1

Story 1 :

En tant que client, je veux connaître la liste des factures par date afin de connaître les fractures les plus urgentes.

Test d'acceptation : afficher la liste des factures ordonnées par la date de facturation .

Le test tel qu'il écrit implique l'écriture du code qui va permettre d'afficher la liste des factures ordonnées par date.

Tests d'acceptation pour « Testable »

Exemple2,

Story 5 : (sous forme de scénario):

En tant qu'étudiant du CLG, je veux me connecter au site ColNet afin d'accéder à mon dossier scolaire.

Test d'acceptation:

L'identifiant saisi est dans le bon format (on peut préciser le format : ex: email).

Le mot de passe est dans le bon format (on peut préciser le nb caractères max ou format).

L'étudiant est identifié ou non.

Étudiant identifié :

Étant donné que la combinaison de l'identifiant et du mot de passe est bonne, l'étudiant est identifié et la page d'accueil du site ColNet lui est présentée.

Étudiant non-identifié

Étant donné que la combinaison de l'identifiant et du mot de passe n'est pas bonne, un message informant l'étudiant s'affiche et il doit ressaisir les informations de connexion.

Tests d'acceptation pour « Testable »

Exemple 3, Story 30 (sous forme de scénario)

En tant qu'intéressé, je veux m'inscrire à la conférence sur le climat « SOS Terre » qui se déroule à la salle « Talbot » afin de pouvoir y assister.

Test d'acceptation : Inscription acceptée ou refusée.

Inscription acceptée :

Étant donné l'utilisateur Primogene connecté à la conférence « SOS Terre » qui se déroule à la salle Talbot d'une capacité de 200 places et dont le nombre d'inscrits actuels est 152. Quand Primogene s'inscrit à « SOS Terre » alors, l'inscription est acceptée, un message de confirmation est envoyé à Primogene et le nombre d'inscrits sera augmenté de 1.

Inscription refusée:

Étant donné l'utilisateur Patoche connecté à la conférence « SOS Terre » qui se déroule à la salle Talbot d'une capacité de 200 places et dont le nombre d'inscrits actuels est 200. Quand Patoche s'inscrit à « SOS Terre » alors, l'inscription est refusée, un message de refus est envoyé à Patoche

Les tests d'acceptation

Exemple 4: Story 3(sous forme de scénario)

En tant que joueur connecté, je souhaite ajouter un item à mon panier afin de pouvoir l'acheter

Critères d'acceptation:

Étant donné le joueur Barackuda connecté à Darquest, lorsqu'il choisit d'ajouter l'item « Hache de fer » à son panier avec une quantité x , le montant du panier est mis à jour.

Exemple 5 (sous forme de liste)

En tant que joueur connecté, je souhaite effectuer une recherche d'items afin de pouvoir l'acheter ou non.

Critères d'acceptation:

- Résultat unique: afficher l'item avec ses détails: nom, prix, efficacité ...
- Pas de résultat: inviter le joueur à faire une autre recherche
- Plusieurs résultats : afficher la liste d'items ordonnée par prix

Tests d'acceptation pour « Testable »

À essayer:

- Se servir des tests pour communiquer: la communication est facilitée entre les niveaux métier et développement. On parle de spécifications par l'exemple.

À éviter :

- Tester une story dans le sprint suivant son développement : ne pas tester et corriger une story dans le sprint où elle est développée **est un disfonctionnement** qu'il faut éviter. Cela diminue la productivité des développeurs qui doivent se replonger dans la story, cela rend la planification difficile. Pas satisfaisant pour l'équipe.

Détailler une User-story

En général, l'énoncé d'une user-story n'est pas suffisant **pour pouvoir l'estimer** (en temps ou en point), il faudra donner des détails pertinents

Exemple

En tant que joueur potentiel, je veux créer un compte afin de devenir membre

Test d'acceptation

- Compte créé.
- Courriel de confirmation envoyé au joueur

Détails:

1. Créer le formulaire d'inscription
2. Valider toutes les entrées
3. Vérifier que l'Alias du joueur n'existe pas.
4. Enregistrer proprement le formulaire
5. Confirmer la création du compte par courriel pour le joueur en lui indiquant que le compte avec Alias « alias » es crée

Étude des besoins, formaliser les besoins: conclusion

User Story	Use case
C'est une brève description d'une fonctionnalité telle que vue par l'utilisateur.	Représente une séquence d'action qu'un système peut accomplir en interagissant avec les acteurs du système
Mode orale et collaboratif	Mode écrit et souvent distant
Grande lisibilité vu sa simplicité	Pourrait souvent manquer de lisibilité (volume)
Format écrit court laissant part à de belles discussions orales	Format écrit très riche en information(postcondition, scénario ...) peu de place à l'oral
Utilisée pour spécification des exigences mais surtout pour estimation de la planification	Utilisé seulement en tant que spécification
Émergence rapide au travers des ateliers de collaboration	Long travail d'analyse et de formalisation

Étude des besoins, formaliser les besoins: conclusion

User Story	Use case
Implémentée et testée en une itération seulement	Implémenté et testé en plusieurs itérations
Contient des test d'acceptation (au dos de la carte)	Ne contient pas les cas de test qui en découle.
Grande lisibilité vu sa simplicité	Pourrait souvent manquer de lisibilité (volume)
Difficile à lier les unes aux autres: Absence de vue globale	Liaison et vue globale faciles au travers du diagramme des cas d'utilisation qui lient les use case
Associée au méthodes agiles comme SCRUM, XP	Associée généralement au Processus Unifié
Très facile à maintenir	Plus difficile à maintenir

Étude des besoins, conclusion

Le recueil des besoins est une tâche complexe qui implique des acteurs qui ne parlent pas le même langage, qui ont des points de vue différents sur le produit à développer.

L'objectif est donc de faciliter la communication et la coopération entre le client et l'équipe de réalisation.

- Il faut qu'une vision soit partagée par TOUS sur l'objectif, la portée et les enjeux du produit.
- Il est primordial que le client s'implique activement dans le recueil des besoins.
- Les besoins évoluent, les priorités changent il faudra valider avec le client régulièrement.
- Les besoins doivent être priorisés. Il existe des techniques (MoSCoW) pour le faire.
- il faut formaliser les besoins afin de garder leur trace, les retrouver etc... Il existe des techniques pour le faire: la norme IEEE, les cas d'utilisation et **les users-stories**, etc.. Ce qui est important est d'avoir un **référentiel UNIQUE** support à la communication entre les acteurs et sur lequel l'équipe de développement doit s'appuyer.

Étude des besoins



CONCLUSION



QUESTIONS ??