

Introduction

Le développement se fait toujours dans un contexte (client, sous-traitants, développeurs, ...) et les participants ne parlent pas tous « l'orienté objets ». L'analyse des besoins doit interpréter correctement les exigences du client, leur connaissance du domaine, et leurs préoccupations et présenter le résultat sous une forme compréhensible par tous.

Les cas d'utilisations permettent de décrire comment les exigences du client peuvent être satisfaites au travers d'interactions entre un utilisateur et le système. C'est un point de vue « externe » qui illustre les grandes fonctionnalités et le comportement attendus du système. Les cas d'utilisation décrivent des activités réalisées par l'utilisateur et le système pour accomplir une tâche, mais pas les détails de réalisation. Cependant le niveau de détail peut varier en fonction de l'auditoire auquel le cas d'utilisation est destiné.

Cas d'utilisation (décider de ce qu'on veut)

Concerne la découverte des besoins fonctionnels (principalement). Que veut faire l'utilisateur avec le système ?

Limites du système

Le système est ce qu'on développe, le reste est à l'extérieur.

Identifier les acteurs et leurs objectifs

Les acteurs principaux ont des objectifs qu'ils atteignent en utilisant le système. Un acteur définit un rôle par rapport au système.

Cela peut être

- une personne,
- un autre système,
- une organisation,
- un programme ...

Rmq : Représentez les acteurs humains avec des symboles « bonhomme » et les autres avec des boîtes stéréotypées « acteur ». Placez les acteurs principaux à gauche dans les diagrammes.

Des questions qui peuvent aider à trouver des acteurs « oubliés » :

- qui démarre/arrête le système ?
- qui assure l'administration ?
- qui gère les utilisateurs et la sécurité ?
- Est-ce que le « temps » est un acteur parce que le système fait quelque chose en réponse à un événement temporel ?
- Y a-t-il un processus de supervision qui relance le système en cas d'arrêt ?
- Qui évalue l'activité du système ou ses performances ?

- Comment sont gérées les mises à jour ?
- Qui surveille les journaux d'activité ? ...

Il peut exister des acteurs dont le rôle est de fournir des services au système qu'on développe.

Rmq : c'est suspect s'il n'y a aucun acteur représentant un système (ou processus logiciel) externe, sauf dans des projets de portée très limitée.

Pour les acteurs principaux, lister leurs objectifs métier.

En général : un objectif métier → un cas d'utilisation.

Exception : Create, Retrieve, Update, Delete X → Manage X (pattern)

Nommer les cas avec Verbe+Complément.

Les cas d'utilisation sont des objectifs pour les acteurs lorsqu'ils utilisent le système. Le système produit un résultat observable et intéressant pour l'acteur impliqué.

Un cas d'utilisation doit être un processus élémentaire du métier/domaine. Ne pas détailler les étapes/sous-fonctions/sous-tâches.

Description des Use Cases

Un cas d'utilisation est principalement un ensemble d'histoires / scénarios qui décrivent de manière textuelle une façon d'utiliser le système. Il doit être compréhensible par tous les participants (clients, développeurs, ...). Les diagrammes de cas d'utilisation sont secondaires, ils servent à poser le contexte et résumer le comportement du système et de ses acteurs.

Plusieurs « styles » sont possibles pour l'expression des cas d'utilisation : boîte noire (acteur-système), bref (résumé), normal (plusieurs scénarios), complet (toutes les étapes et alternatives, pré/post-conditions ...), ...

Écrivez-les cas en vous plaçant au niveau des objectifs de l'acteur et des responsabilités du système, ne vous préoccupez pas de l'IHM, ni des actions concrètes.

Félicitations ! vous avez terminé d'écrire les cas d'utilisation ... et c'est pas parfait !

Glossaire/dictionnaire

Un glossaire peut être utile pour détailler la signification de certains termes métier, ou donner des précisions (unités, ...) sur les données.

Formes de rédaction des Use Cases

Plusieurs formes de rédaction sont possibles, chacune avec ses forces et faiblesses :

« Récit » : pour présenter une vue d'ensemble, synthétique

Faire une partie

Le joueur lance des cochons pour faire des figures et cumuler des points. Le système gère le score des joueurs et le tour de jeu. La partie se termine dès qu'un joueur atteint 100 points.

« Scénario » : pour décrire une séquence d'événements

Scénario « Tour de jeu standard »

- 1 Le joueur lance les cochons.
 - 2 Le système indique la figure réalisée.
 - 3 Le système indique les point cumulés obtenus pendant ce tour.
 - 4 Le joueur termine son tour
- Optionnel :
- Le joueur poursuit son tour et reprend à l'étape 1
- 5 Le système actualise le score du joueur
 - 6 Le système désigne le joueur suivant

« Dialogue » : pour insister sur les interactions acteur-système

Acteur : Joueur	Système
Lance les cochons	
	Indique la figure réalisée et les points cumulés obtenus pendant ce tour
Termine son tour Optionnellement, lance à nouveau les cochons	
	Actualise le score du joueur

Les différentes formes ne sont pas exclusives : en tout début de projet, on se contentera de définir les acteurs, les noms des cas d'utilisation et leur description sommaire (forme « récit ») ; puis au fur et à mesure de l'avancement et du besoin, on détaillera (formes « scénario » ou « dialogue »).

Ce qui suit montre une possibilité de description détaillée d'un cas d'utilisation, avec un niveau de détail suffisant pour servir de base à la conception.

Description complète

Version	Description	Modifié par	Date
1.0	First draft	Stein	11/11/2014

Objectif	<i>Les joueurs veulent perdre leur temps en faisant une partie de Jeu de Cochons. Ils accumulent des points en réalisant des figures lors des lancers de cochons. Le premier joueur totalisant 100 points a gagné.</i>
Préconditions	<i>Aucune</i>
Acteur principal	<i>Joueur</i>
Déclencheur Métier	<i>Les joueurs veulent faire une partie</i>
Conditions de fin (réussite)	<i>Un joueur gagne en ayant un score de 100 points</i>
Conditions de fin (échec)	<i>Tous les joueurs ont abandonné la partie</i>
Scénario nominal	<i>1. Le système demande le nom des joueurs. 2. Le système choisit le premier joueur. 3. Le système affiche le tour du joueur 4. Le joueur fait un lancer de cochons. 5. Le système affiche les figures obtenues avec les cochons. 6. Le système affiche les points obtenus pour le tour. Le scénario reprend à l'étape 4 à moins que le joueur décide d'arrêter ou qu'il obtienne un Cochon Nul ou un Bon Jambon. 7. Le système met à jour le score du joueur. 8. Le système détermine le joueur suivant. Le scénario reprend à l'étape 3 jusqu'à ce qu'un des joueurs atteigne un score de 100 points.</i>
Post Condition	<i>Un des joueurs a atteint un score de 100 points ou tous les joueurs ont abandonné la partie.</i>
Scénario Alternatif CochonNul	<i>A l'étape 4, le joueur obtient la figure Cochon Nul. Il perd les points accumulés pendant ce tour. Il ne peut pas faire de nouveaux lancers pour ce tour. Le scénario nominal continue à l'étape 7.</i>
Scénario Alternatif Bon jambon	<i>A l'étape 4, le joueur obtient la figure Bon Jambon. Il perd tous les points obtenus jusque là dans cette partie. Il ne peut pas faire de nouveaux lancers pour ce tour. Le scénario nominal continue à l'étape 7.</i>

Scénario Alternatif *Abandonner*

Pendant son tour, (étape 3) un joueur peut décider d'abandonner la partie.

Le système supprime le joueur de la partie.

Le scénario nominal continue à l'étape 8.

Règles Métier

La détermination du premier joueur est aléatoire. Les joueurs jouent à tour de rôle jusqu'à ce que tous aient joué. Pour la suite de la partie, on recommence à partir du premier joueur, dans le même ordre.

Tous les joueurs commencent avec un score nul.

Valeur des figures :

Trotteur 5 points

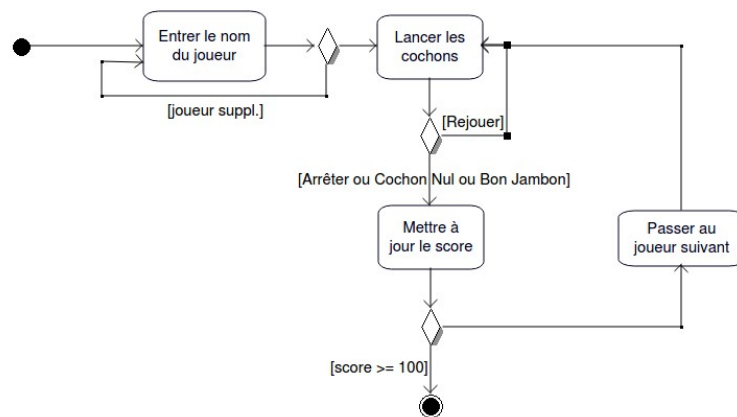
Tournedos 5 points

GroinGroin 10 points

Bajoue 15 points

...

Diagramme d'activité



Maquette

