

Ecrire les cas d'utilisation

Dominique Revuz 2020

Lisibles !

Des cas d'utilisation décontractés et lisibles sont plus utiles que des cas d'utilisations empoulés ou illisibles.

Largeur puis profondeur

Commencer a visiter l'ensemble des cas d'utilisation avec un premier niveau d'analyse. Puis progressivement avancer dans les niveau d'analyse.

NIVEAU 1: L'acteur principal du cas d'utilisation et son objectif.

NIVEAU 2: Le résumé, les variables de base et le scénario nominal détaillé.

NIVEAU 3: Les extensions, Points d'entrée et Point de sortie et cas d'erreurs.

NIVEAU 4: La gestion des extensions est détaillé.

L'écriture du scénario nominal (ou des extensions)

Pour chaque étape (numérotée):

- Explicitiez un objectif atteint
- Décrivez/ capturez l'intention de l'acteur et pas les détails d'interface.
- L'acteur donne une information, valide une condition, met à jour un état.
- Indiquez entre chaque étape les éléments d'ordonancement (ou leur absence si l'ordre des étapes est non valide)
- Posez vous la question de pourquoi pour trouvez l'objectif de niveau suivant (plus fin).

Description des données

Niveau 1: Le nom de la donnée est indiqué dans le scénario nominal.

Niveau 2: dans une deuxième partie du document les champs sont explicités.

Niveau 3: Précision pour chaque champs du type, taille, et validateurs.

Gestion des documents

Chaque cas d'utilisation doit avoir un identifiant text et un identifiant numérique unique (stable).

Vos documents doivent être versionnés (git).

La version doit être visible dans l'entête du document.

Faites des références précises et pas le document en entier !

UC#12/DataDescription/Carburateur et pas UC#12.

Mise en perspective

Porté de la conception	Niveau de l'Objectif
Organisation (Boite noire)	Résumé de très haut niveau
Organisation (Boite blanche)	Résumé
Système (Boite Noire)	Objectif utilisateur
Système (boite blanche)	sous fonction
Composant	détail de conception (Trop bas)

Une démarche “pas à pas”

Suite a des demandes répétées de “comment il faut faire” voici une démarche “pas à pas” proposée par Alistair Cockburn. Pour les b2+ le document original est sur [gl2020](#) dans références : [UseCaseQuestions](#).

1 Le système :

Ecrire la portée et la frontière du système: Le périmètre.

Assurer vous que le périmètre est mis à jour, en particulier les deux listes dedans et dehors.

Et dans l'outil de version tagger quand un élément change de liste.

Listez (réunion créative) les acteurs primaires humain et non humains sur l'ensemble de la vie du système.

Produisez une liste **exhaustive** des Objectifs de chaque acteurs.

Un fichier *objectifs* ou les pairs acteur-objectif sont disponibles. Moi j'aime csv mais c'est pas le cas de tout le monde.

<>.

Capturez (identifier et définir) le cas d'utilisation le plus vaste / externe qui correspondent au besoin fondamental de l'acteur (qui englobe tout ces besoins).

Et ceci pour tout les acteurs principaux (faites les étapes suivantes et revenez ici pour chaque acteur).

Identifiez ainsi les éléments qui ont de la valeur pour l'acteur.

Construire le résumé de cas d'utilisation.

Ajouter soustraire fusionner mixer les différents objectifs dans les cas d'utilisations.

Rechercher tous les événements qui traversent la frontière et pourront être associés au cas d'utilisation. En particulier chercher les événements qui sont déclenchés par le temps.

C'est le moment de raconter le cas d'utilisation, le texte que vous fournissez ici doit permettre de comprendre globalement la suite d'évènements qui mène à un succès (l'acteur a atteint son objectif !).

Ceci doit être un dialogue entre le système et l'acteur.

Pour chaque Use case capturez les intérêts des parties prenantes, les préconditions et les garanties. (voir le fichier patron de fichier usecase).

Le système doit assurer les préconditions et garantir les intérêts.

Reprendre les étapes précédentes <> pour assurer les garanties des parties prenantes.

Pour chaque cas d'utilisation listez (réunion créative) les situations d'extension.

Erreurs, impossibilités, chemins alternatifs, choix.

Pour chaque élément de la liste d'extension écrire le dialogue système-acteur correspondant. Chaque extension doit se terminer par un des trois cas: - retour au scénario nominal (alternative)

- success alternatif (sortie)
- echec (sortie)

Extraire des interactions complexe dans des sous-cas d'utilisation (uses). Identifier des sous-cas d'utilisation communs.

Attention ajouter des cas d'utilisation ajoute un cout au projet.

Relire, améliorer, ajouter, fusionner, soustraire jusqu'à satisfaction des trois critères:

- lisibilité
- complétude
- garantir les intérêts des parties prenantes

Glossaire

Acteurs principaux: Si ils sont absent le système n'a pas de raison d'être. Les acteurs secondaire ne font que participer a des cas d'utilisation et n'ont pas d'objectif par rapport au système (l'administrateur des comptes, si il n'y a pas d'utilisateurs son rôle disparaît). **Acteur principal d'un cas d'utilisation** est celui qui à un objectif satisfait par le cas d'utilisation.

> Règle de bon sens: Si il n'y a pas d'acteur principal pourquoi y a t'il un cas d'utilisation ?

Acteur secondaire/support acteur qui intervient dans le cas d'utilisation mais qui n'est pas l'acteur principal.

Points d'entrée une façon de démarrer le cas d'utilisation alternative. L'utilisateur souhaite faire une deuxième opération sur le DAB. **Point de sortie** une façon de terminer le cas d'utilisation alternative. L'utilisateur annule la transaction. **Extension (Cas alternatifs)** des situations exceptionnelles qu'il faut traiter dans le cas d'utilisation (la crate de crédit est dans la liste de cartes volées).

évènements: L'interface exexterne du système est sollicité. Une information passe de l'exterieur à l'intérieur du système par une