

Génie Logiciel

Dominique Revuz, Philippe Finkel, Philippe Cluzeau

ESIPE

avril 2021

Plan

- 1 Élaboration : présentation
- 2 Stratégie générale
- 3 Éléments de l'élaboration
- 4 Les cas d'utilisation
- 5 MOD
- 6 Ecrire des UC

Rôle de la phase Élaboration

Élaboration ?

- ① la phase d'élaboration se termine avec une idée précise du logiciel à construire
- ② Pour cela différentes techniques vont être mises en place pour décrire au mieux le logiciel.
 - Glossaire
 - Frontière
 - Acteurs
 - Use Case et Scénario
 - Objets du Domaine
 - FQM
 - **Outils de cohérence entre ces outils**

Quelques Idées - Questions

- 1) Le système
- 2) Le dedans et le dehors
- 3) interagit avec qui pourquoi et comment
- 4) Y a-t-il un jargon spécifique au métier
- 5) Y a-t-il des concepts spécifiques au métier
- 6) Que FAIT le système ?

Le contexte

- ① les autres logiciels
 - PizzaCompta
 - le système d'exploitation
 - ERP
 - Environnement urbanisation logiciel
- ② Les activités non informatiques des processus industriels
- ③ Légal
- ④ Organisationnel

UML et Objectory

UML

- ① Une notation pour les éléments concrets de la modélisation
- ② Des diagrammes pour permettre une vue d'ensemble de la modélisation
- ③ Des concepts de conception utiles
- ④ une norme aujourd'hui : *Unified Modeling language* (“fusion” de OOSE, OMT, Booch)

OCL

- ① Object Constraint Language
- ② OCL permet de décrire des invariants dans un modèle, sous forme de pseudo-code
<https://www.omg.org/spec/OCL/>

Ivar Jacobson



Figure 1 – Ivar_jacobson.jpg

“Ivar Jacobson a développé la technique des cas d’utilisation pour capturer les exigences tout en travaillant chez Ericsson”. Un des pères d’UML.

https://www.wikiwand.com/fr/Processus_unifi%C3%A9

Objectory

- 1) Une stratégie d'écriture et d'analyse d'un logiciel
- 2) Des concepts utiles
 - Acteurs
 - Cas d'utilisation
 - Diagrammes
 - Classes
 - Séquence
 - Déploiement
 - des outils (diagrammes) pour vérifier/challenger/corriger.
 - ex : *Entity-control-boundary*. "Il vise à structurer les classes selon leurs responsabilités dans la mise en œuvre de cas d'utilisations".

Stratégies d'élaboration

- ① Trois stratégies d'élaboration que nous allons utiliser successivement et répétitivement pour assurer la cohérence et la complétude de notre modélisation :
 - par les Données
 - par les événements
 - par les Processus
- ② la Cohérence est nécessaire pour ne pas avoir de contraintes contradictoires
- ③ l'absence de complétude dans les phases préliminaires est un risque important dans une démarche projet. Quand une exigence/une spécification/une contrainte architecturale/une contrainte conceptuelle est découverte dans une phase postérieure, cela impose un retour arrière dans les différentes phases; ce qui a un coût de plus en plus astronomique avec le nombre de phases redémarrées.

L'élaboration en pratique

Définitions

- ① Elaborer, verbe transitif : Préparer par un lent travail de réflexion.
- ② Elaborate (eng) : Planned or executed with painstaking attention to numerous parts or details.

Les éléments de l'élaboration

- 1) Les acteurs et le diagramme des acteurs
- 2) Le diagramme d'interaction générale (diagramme contextuel)
- 3) La frontière
- 4) Le diagramme des cas d'utilisation
- 5) Les cas d'utilisation
- 6) Le glossaire
- 7) Les objets du domaine (MOD)
- 8) Le FQM

> **INFO2 : votre liste exhaustive !**

Les 4 Éléments principaux du Modèle d'Élaboration, les autres s'en déduisent logiquement.

- 1 Acteurs
- 2 Use case
- 3 MOD
- 4 Glossaire

Cas d'Utilisation (use Case / UC / CU)

Un cas d'utilisation est un document **narratif** qui décrit la **séquence** d'événements qui permet à un **acteur** de compléter un **objectif** en utilisant le **système**.

Les cas d'utilisation (ou cas d'usage)

- 1) Le cas d'utilisation est un processus métier.
- 2) Le processus une fois démarré doit atteindre sa conclusion et apporter une plus value à l'entreprise.
- 3) Un UC est une description de bout en bout d'un processus assez large qui contient un certain nombre de transactions et d'étapes.

Le cas d'utilisation est défini par :

- 1) le système est considéré comme une boîte noire.
- 2) les acteurs sont des receveurs ou producteurs de messages.
- 3) le contexte de fonctionnement du UC (les situations significatives).
Prérequis/Pré-conditions. Post-Condition.
- 4) les événements ou signaux pertinents qui en fonctions du contexte déclenchent des actions
- 5) les actions du système en réponse à ces événements

Le cas d'utilisation : template

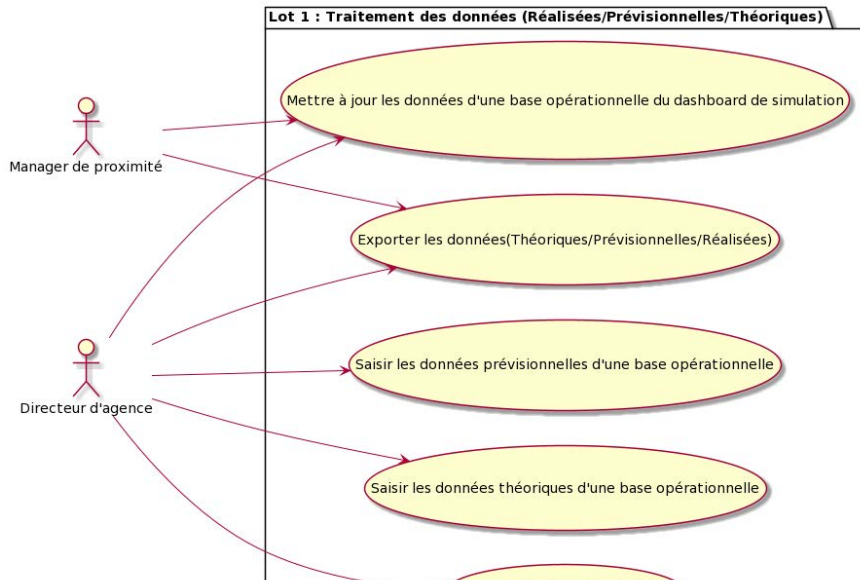
- **Cas d'utilisation** : <UC ID> : <UC name>
- **Description** : ...
 - Une ou deux phrases pour présenter brièvement le UC, incluant l'objectif pour l'acteur principal du UC.
 - N'oubliez pas de mentionner les concepts manipulés.
- **Niveau** : Haut niveau ou détail ?
- **Déclencheur** : Décrire l'événement qui initie ce UC
- **Acteur Primaire**: et son objectif pour ce UC
- **Acteurs secondaires**: et leur objectif éventuel pour ce UC
- **Parties Prenantes concernées**
- **Scenario Nominal** : tout se passe bien, c'est le scénario parfait !
 - c'est là que vous allez raconter l'histoire des interactions entre le système et les acteurs

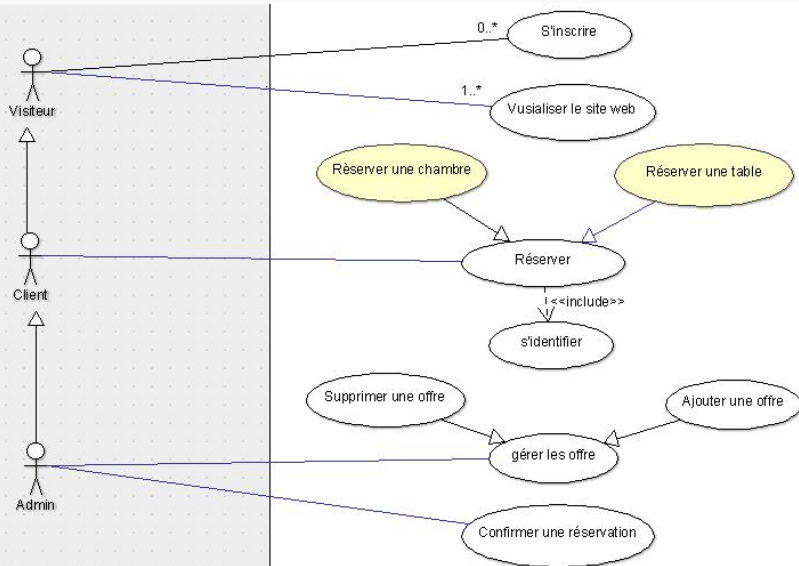
- **Pre-conditions** : Listez les conditions nécessaires pour que ce cas d'utilisation puisse avoir lieu
- **Extensions / Scénarii alternatifs** : “Document alternate flows and exceptions to the main success scenario. Extensions are branches from the main scenario, and numbering should align with the step of the success scenario where the branch occurs.”
 - Indiquez dans quel point du scénario nominal le chemin alternatif démarre et où il reprend.
- **Post-Conditions**
 - Conditions de succès : “Describe the end condition of the Use Case where the Primary Actor’s goal is satisfied”
 - Minimal Guarantees : “Describe the guarantee or assurance that this Use Case provides to all Actors and Stakeholders to protect their interest regardless of whether the Use Case ends with success or failure.”
 - Condition finale en cas d’échec : “Describe the end condition that results if the Primary Actor fails to accomplish his goal.” (ex: le DAB rend la CB. . .)

Et penser aussi (mais pas pour tous les UC) :

- Frquence “Indicate how often the use case is expected to occur. This information aids designers and developers in understanding capacity requirements.”
- Besoins Spéciaux : “Describe any additional factors that impact the execution of the use case. These could be environmental, regulatory, organizational or market-driven in nature.”
- Performance
- Security
- Usability / Accessibility
- Other ...
- Problèmes et étapes suivantes : “Note any issues related to the definition of this use case that will require clarification prior to development. Also list any follow-up work that needs to be done prior to sign-off on the use case.”
 - définir ce qui pose problème dans la description **actuelle** du cas d'utilisation.
 - vous pouvez ajouter ici un lien vers une issue...

Le Diagramme des Cas d'Utilisation





Axes de Conception des UC

Axes de Conception des UC

- ① Deux axes de conception des cas d'utilisation qui vont nous permettre de maîtriser l'effort à fournir pour faire une réponse adaptée au besoin du projet.
- ② Le niveau de Détail
 - Haut niveau
 - Détaillé
- ③ Le niveau d'abstraction
 - Essentiel
 - Réel

Pousser le niveau de détail

- ① Plus de détail => plus de TRAVAIL
- ② On commence toujours par une description HAUT NIVEAU
- ③ Le niveau de détail est proportionnel au besoin de conception préalable, dans le cas où le client ne connaît pas ou mal son besoin, il ne faut pas trop détailler et utiliser une méthode plus agile.
- ④ Si au contraire nous sommes dans une situation où le coût de développement est grand et que l'on souhaite une conception avancée, le niveau de détail sera très important, (le coût des ingénieurs par rapport au coût de production est faible).

Pousser le niveau d'abstraction

- ❶ Le niveau d'abstraction est directement proportionnel au risque de refaire un logiciel sur le même type de prestation.
- ❷ Il permet aussi de se faire une meilleure idée du logiciel à construire. Si l'on a bien fait l'analyse de ce qui est ESSENTIEL dans le logiciel, la qualité du produit final en est grandement améliorée. En particulier il est fréquent que les logiciels ont trop de fonctionnalités inutiles qui complexifient l'apprentissage et souvent font baisser l'ergonomie.
- ❸ L'opposé d'un UC Essentiel est un UC Réel, c'est à dire sans aucun essai d'abstraction (effectivement plus facile à définir pour les opérateurs mais...). Curseur classique entre risque d'overdesign ou manque d'ambition/vision : réglage difficile ...

Définition et Matérialisation de la Frontière

- ❶ La Frontière est la limite du logiciel
- ❷ En effet au cours de l'analyse des besoins et de l'écriture des exigences il est nécessaire de parler du contexte de développement avec le client.
- ❸ Il est nécessaire de faire la part de ce qui fait partie du logiciel et de ce qui n'en fait pas partie. L'écriture des spécifications devra donc définir cette limite
- ❹ Le diagramme des Cas d'Utilisation permet de dessiner une frontière où l'on voit les Cas d'utilisation qui sont de la responsabilité de l'équipe de développement et ceux qui sont de la responsabilité du client ou d'une autre équipe.

Organisation et UC

Prérequis : listes exhaustives

- 1) Acteurs
- 2) Use Case
- 3) Fonctions

Écriture en mode haut niveau de tous les cas d'utilisation du Système

Cf la discussion sur Fonctionnalité / User Story / Use Case...

Diagrammes des UC

Une synthèse graphique. simple et terriblement utile

Relations entre UC

- include
- extent

Versions détaillées pour les UC les plus risqués

Un travail proportionnel au risque.

Un très grand risque : un niveau de détail absolu et une approche REEL, sans abstraction

Ordonner les UC (après hiérarchisation)

Vous en tirerez des éléments essentiels pour votre roadmap / lotissement.

Qui participe a la recherche des cas d'utilisation

Utilisateurs finaux

- 1) Le terme Utilisateur final doit être entendu à deux niveaux.
- 2) Les utilisateurs qui interagissent directement avec le logiciel (ex: développeurs sous Unix, Conseiller financier dans une banque) et qui ont besoin d'une formation pour utiliser le logiciel. Acteurs (directs) du logiciel.
- 3) Les utilisateurs dont les besoins sont couverts par l'interaction avec un utilisateur qui utilise le logiciel pour fournir le service (acteur indirect).
- 4) Pour définir les exigences il est important de penser aux utilisateurs finaux, que ce soit des acteurs directs ou indirects de la solution.
- 5) Ce qui distingue en général ces deux types d'utilisateurs est la nécessité d'une formation.

Sponsors

- ① Ceux qui soutiennent le projet financièrement ou matériellement.
- ② Pensez qu'un projet peut être financé uniquement par du travail éventuellement bénévole.
- ③ Ne négligez pas l'importance de sponsor dans le lancement d'un projet, leur nombre et leur importance peuvent faire la différence dans certains cas et apporter des éléments d'arbitrage sur des questions de tout ordre.

Stakeholders / Parties Prenantes

- ① Ceux qui décident.
- ② Ceux pour lesquels il y a un enjeu dans le logiciel.
- ③ Ceux qui ont une part de risque dans le projet et donc un intérêt dans la réussite du projet.
- ④ On différencie en général des rôles (MOA,MOE,MOAD,MOED)
 - MOA: Maîtrise d'OuvrAge
 - MOE Maîtrise d'OEuvre
 - MOAD: Maîtrise d'Ouvrage Déléguée
 - MOED: Maîtrise d'œuvre déléguée
- ⑤ L'acheteur et le Vendeur

Analyste ?

- ④ La création de Cas d'utilisation Haut niveau se fait par élicitation de l'analyste à partir de la définition des exigences.

> *“En gestion de projet et notamment de projet informatique, « éliciter » est l'action de comprendre et de modéliser les processus du maître d'ouvrage (MOA) pour les partager avec l'équipe chargée de la mise en œuvre du projet (MOE).”*

Qui participe à l'écriture des cas d'utilisation ?

- 1) Travail de l'équipe de spécification
- 2) Analyste

Comment s'organiser pour créer les UC

Première Itération Acteurs : Objectifs

- ① Mode Brainstorming
- ② typiquement avec des Utilisateurs / Sponsors / Stakeholders
- ③ Il faut répondre à la question :
 - pourquoi l'acteur utilise le système ?
 - Pour atteindre un objectif !
- ④ Validation
 - La validation se fait quand la créativité des participants est asséchée
 - Un travail d'homogénéisation des termes doit être entrepris
 - Une première priorisation des UC peut être faite ici avec le point de vue des stakeholders

Définir la Stratégie d'écriture

- ① Haut Niveau
- ② Détaillé
- ③ Essentiel
- ④ Réel

Construction des Cas d'utilisation

- ① La construction d'un UC est faite en deux étapes
- ② le scénario nominal qui permet de valider d'une part l'intérêt de l'UC et la réalisation de l'objectif de l'acteur principal
- ③ Les scénarii alternatifs qui permettent d'explorer l'ensemble des situations non nominales. Ex :
 - Éléments manquants
 - Problèmes systèmes (DD plein)
 - Cas plus complexes

Création des Scénarii Nominaux

- ❶ Ici nous cherchons à démontrer que l'objectif de l'acteur peut être atteint
- ❷ Il faut commencer à modéliser le système en même temps (MOD, "architecture")
- ❸ Première validation du UC sur le SN

Création des Scénarii alternatifs

- ❶ Le scénario nominal est repris du début à la fin et pour chaque possibilité de choix ou d'erreur de fonctionnement, un nouveau point de sortie est créé.
- ❷ Chacun des points de sortie est résolu.
- ❸ Dans le cas où le point de sortie est associé à des variations importantes de comportement, cela amène à des réflexions sur les liens entre UC (granularité, include, extension)

Ou comment décider que le travail des cas d'utilisation est terminé.

Quelles Questions

- ① A-t-on trouvé tous les cas d'utilisation ?
- ② Quand commence-t-on la modélisation avec les use case ?

Les réponses simples

- ① Si ni les utilisateurs ni les sponsors ni les Stakeholders n'arrivent à proposer de nouveaux UC
- ② Dès le premier cas d'utilisation la modélisation commence (DR: “la première crêpe est toujours ratée” ... pas grave)

Exemples

Actions de l'acteur	Réponses du système
1.Ce UC commence quand un client arrive à la caisse	
2.Le caissier enregistre chaque article. Si il y a plusieurs articles identiques leur nombre peut être entré par le caissier.	3. déterminer le prix de l'article mettre à jour la liste des articles. Le prix et la description de l'article courant sont affichés
4. le caissier indique que l'ajout d'articles est terminé	5. Calcul et affichage du total.
6. le caissier annonce le total.	
7. le client fourni un paiement en liquide supérieur ou égal au total.	
8. le caissier entre le montant du liquide fournit	9. Affichage de la monnaie a rendre. Imprime un reçu.
10. Le caissier range le liquide dans la caisse, en retire la monnaie. Fournit la monnaie et le reçu au	11. enregistre la vente.

Modèle des Objets du Domaine

Introduction

- 1) Les Objets du domaine sont un glossaire graphique que l'on manipule avec le client et dans l'équipe d'élaboration.
- 2) Ce diagramme d'objet (pas de concept informatique) permet de décrire très rapidement les concepts manipulés par les acteurs et le système
- 3) L'intérêt premier de cet outil de modélisation est son pouvoir d'expression.
- 4) Diagramme de "concepts"
- 5) le MOD est informatif avant tout
- 6) peu coûteux sauf si l'on y perd son temps...
- 7) et très utile pour *secouer* ses use cases

Le modèle conceptuel ne contient pas d'objets

- ① Le MC doit montrer
 - les concepts du domaine
 - les associations entre concepts
 - les attributs des concepts
- ② Les concepts sont un outil de décomposition
- ③ Ils permettent de réduire la complexité ou du moins de simplifier la manipulation des systèmes complexes
- ④ Il faut mieux avoir un modèle conceptuel trop riche (détaillé) que trop pauvre.
- ⑤ Le silence est plus dangereux que le bruit.
c'est aussi un outil d'échange et de pédagogie !

Construire le MOD

- 1) Faire la liste des Candidats concepts
- 2) Les dessiner en un modèle conceptuel
- 3) Ajouter les associations
- 4) Ajouter les attributs

Stratégies

- 1) Principe du Cartographe :
 - Utiliser les Noms existants
 - Exclure les éléments non pertinents
 - Ne rien ajouter !
- 1) Privilégier les Concepts aux attributs

Quelques Concepts

- ① Physique tangible
- ② spécifications, conceptions, descriptions de choses
- ③ lieux
- ④ transactions
- ⑤ éléments de transactions
- ⑥ Rôles de personnes
- ⑦ conteneurs
- ⑧ éléments dans un conteneur
- ⑨ appareils
- ⑩ Noms de Concepts abstraits
- ⑪ Organisations
- ⑫ événements
- ⑬ processus
- ⑭ règles et politiques
- ⑮ catalogues
- ⑯ documents financiers de travail, contractuels, légaux
- ⑰ instruments financiers
- ⑱ manuels livres

Groupes Nominaux

- ① Une méthode pour la recherche de concepts est d'associer un concept ou un attribut à chaque groupe nominal trouvé dans les descriptions du domaine (de la description des UC), les documents de référence fournis par le client, etc.
- ② Mais c'est dangereux du fait de l'ambiguïté de la langue...
- ③ Mais fait avec précaution cela peut être très utile. Cela produit de la donnée qui peut aider pour l'élicitation

Associations du MOD

- 1) Une association est une relation entre concepts qui met en valeur une connexion intéressante ou significative
- 2) Deux types d'association :
 - Need to Know : elle participera sans doute au système
 - Comprehension-only : elle est là pour aider à la compréhension

Quelques Associations

- ① A partie physique de B
- ② A partie logique de B
- ③ A contenu physiquement dans/sur B
- ④ A contenu logiquement par B
- ⑤ A est une description de B
- ⑥ A élément de la transaction B
- ⑦ A est membre de B
- ⑧ A partie de l'organisation B
- ⑨ A utilise B
- ⑩ A gère ou dirige B
- ⑪ A communique avec B
- ⑫ A est en relation avec une transaction B
- ⑬ A est connu/enregistré/noté/capturé par B
- ⑭ A est proche de B
- ⑮ A est possédé par B
- ⑯ A est une transaction en relation avec une transaction B

Rôle dans une association

- 1) On identifie les deux éléments d'une association par deux rôles
- 2) Le rôle est défini par
 - un nom
 - une multiplicité
 - une navigabilité
 - une description textuelle si nécessaire

Attributs du MOD

- ❶ Rappel : toujours pas d'élément informatique
- ❷ Rappelez vous que l'on discute de ce modèle avec des utilisateurs/clients/parties prenantes
- ❸ Un attribut est une donnée qui est logiquement un élément constructif d'un objet.
- ❹ On place dans le modèle conceptuel tous les attributs pour lesquels il existe un besoin de mémoire (dans les UC par ex)
- ❺ Un reçu contient une date et une heure d'où les attributs date et heure dans le concept vente.
- ❻ Un usager d'hôpital à un numéro de sécu (ou pas)

Attributs

- ① Quand choisir d'ajouter un attribut plutôt qu'un concept et réciproquement ?
- ② Un attribut est constitutif (le concept ne fonctionne pas sans)
- ③ Une caisse dans le supermarché n'est pas un attribut
- ④ Destination et Vol ?

Attributs et type primitifs

- ❶ Ne pas décomposer jusqu'aux types primitifs de votre langage !
- ❷ Utiliser des types construits simples qui sont auto documentés
- ❸ Date, Adresse, NSS, ISBN etc

Attribut ou Association

- 1) Attribut ou association le choix dépend de ce que vous voulez modéliser.
- 2) Exemple: Paiement -> Montant -> Monnaie

Glossaire

- 1) **Rappelez vous que vous devez mettre a jour votre glossaire !**
- 2) Restfull access to the glossary ?

Écrire les cas d'utilisation

Lisibles !

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

Largeur puis profondeur

Commencer à visiter l'ensemble des cas d'utilisation avec un premier niveau d'analyse. Puis progressivement avancer dans les niveaux 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 Points de sortie et cas d'erreurs.
- NIVEAU 4: La gestion des extensions est détaillée.

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'ordonnancement (ou leur absence si l'ordre des étapes est non valide)
- Posez vous la question de pourquoi pour trouver l'objectif de niveau suivant (plus fn).

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
 - allers-retours et cohérence avec le MOD.
- Niveau 3: Précision pour chaque champs du type, taille, et validateurs.

Gestion des documents

Chaque cas d'utilisation doit avoir un **identifiant unique**, et **stable**.

Et un nom, qui lui pourrait légèrement évoluer.

Vos documents doivent naturellement être versionnés (git).

Faites des références précises entre document et pas juste vers le document en entier !
(référence vers un objet précis du MOD, vers une étape numérotée d'un UC, etc...).

Une démarche “pas à pas”

Suite à des demandes répétées de “comment il faut faire”, voici une démarche “pas à pas” proposée par Alistair Cockburn.

https://fr.wikipedia.org/wiki/Alistair_Cockburn

1. Le système

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

Assurez-vous que le périmètre est mis à jour, en particulier les deux listes “dedans” et “dehors”.

2. Acteurs

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

3. Objectifs

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

Proposition : un fichier par acteur avec la liste de ses objectifs.

4. Capture des UC

Capturez (identifier et définir) les cas d'utilisation les plus vastes / externes qui correspondent aux besoins fondamentaux de l'acteur (qui englobe tous ses 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.

5. Résumé des UC

Construire le résumé des cas d'utilisation.

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

Rechercher tous les événements qui traversent la frontière et pourraient être associés aux cas d'utilisation.

En particulier chercher les événements qui sont déclenchés par le temps.

6. Raconter des histoires

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

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

7. Affinez / complétez

Pour chaque Use case, capturez les intérêts des parties prenantes, les préconditions et les garanties. (cf template de UC).

Le système doit assurer les préconditions et garantir les intérêts.

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

8. Scénarios alternatifs

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

Erreurs, impossibilités, chemins alternatifs, choix.

9. Détailler les scénarios alternatifs

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)
- succès alternatif (sortie)
- échec (sortie)

10. Factorisation (un peu)

Extraire des interactions complexes dans des sous-cas d'utilisation (uses).

Identifiez des sous-cas d'utilisation communs.

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

11. Secouer !

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

- lisibilité
- complétude
- garantir les intérêts des parties prenantes

Encore un bout de glossaire !

- Acteurs principaux du système : S'ils sont absents, le système n'a pas de raison d'être.
- Les acteurs secondaires du système ne font que participer à des cas d'utilisation et n'ont pas d'objectif par rapport au système (ou moins important... définition discutable). Ex: l'administrateur des comptes, s'il n'y a pas d'utilisateurs son rôle disparaît.
- Acteur principal d'un cas d'utilisation : c'est celui qui a un objectif satisfait par le cas d'utilisation.
 - Règle de bon sens : S'il n'y a pas d'acteur principal pourquoi y a-t-il un cas d'utilisation ?
- Acteur secondaire/support d'un use case : acteur qui intervient dans le cas d'utilisation mais qui n'est pas l'acteur principal :-)
- Point d'entrée : une façon de démarrer le cas d'utilisation
 - point d'entrée alternatif : l'utilisateur souhaite faire une 2ème opération sur le DAB
- Point de sortie : une façon de terminer le cas d'utilisation
 - point de sortie alternatif : l'utilisateur annule la transaction.
- Extension des situations rares ou exceptionnelles qu'il faut traiter dans le cas d'utilisation
 - ex: la carte de crédit est dans la liste de cartes volées.
- Événement : l'interface externe du système est sollicitée. Une information passe de l'extérieur à l'intérieur du système

Mesurer ?

Expliciter les Exigences de façon mesurable

N'est Connu que ce qui est mesurable

Lord KELVIN disait clairement que : *Sans l'aptitude de mesurer, on ne peut progresser. Le progrès de nos systèmes de mesure est également le progrès de la science*

FQM - Fonction / Qualité / Mesure

- Exigence : Une condition ou une capacité à laquelle le système doit se conformer.
- Croiser les **Fonctions** et les **Qualités** nous permet d'identifier les deux formes d'exigences, le quoi et le comment.
- Les **Mesures** permettent de valider que l'objectif qualité est atteint.

Les fonctions du système

Le système doit faire <X>.

Catégorie des Fonctions	Sens
Évidente/service	s'exécute et l'utilisateur en est conscient
Cachée/technique	s'exécute et est nécessaire au fonctionnement
Excitante/estime	peu d'effet sur les coûts, mais aide la vente

Exemple

REF	Fonction	Catégorie
1.1	Enregistre la vente en cours c.a.d. les articles achetés	évidente
1.2	Calcul du total courant + tva	évidente
1.3	Mise à jour de l'inventaire	cachée
1.4	Enregistrer les ventes terminées.	cachée
1.5	Procédure d'identification du caissier.	évidente

Qualités Voulues

- Facile d'utilisation
- Robuste
- Portable
- Temps de réponse
- Interface intuitive
- Scalable (supporte la montée en charge)
- Nombre de clics

la difficulté va être d'exprimer une qualité de manière mesurable !

Qualités spécifiques au système

- Des réponses pertinentes :-)
- Meilleur classement / évaluation (scoring)
- ...

Croiser les fonctions avec les qualités, et expliquer comment le mesurer.

REF	Fonction	Catégorie	Qualité	Mesure
1.9	Affiche la description et le prix de l'article	évidente	Temps de réponse	5s Max
2.4	Enregistrer les demandes de crédits	cachée	Interface	Formulaires en couleur
			robuste	Doit être enregistré dans les 24h
			Temps de réponse	10s max

les FQM doivent aider à préparer la recette fonctionnelle.