

Spécification des besoins logiciels & planification

- Définir avec précision le travail à réaliser : affiner le sujet
- Définir une démarche :
 - étapes,
 - tâches,
 - activités
- Estimer
- Planifier

Spécifier le travail à réaliser

- *« Les clients sont plus prompts à prodiguer des conseils de mise en œuvre qu'à fournir la description exacte de leur problème : le programmeur n'arrive pas à obtenir une spécification de ce qu'il faut faire, mais reçoit maintes indications sur la manière de le faire. »*
Meyer & Baudoin, 1984
- Spécifier, c'est essentiellement répondre à la question
« Quoi ? » ie décrire ce que doit faire le système

Spécifier le travail à réaliser

- Objectif : se mettre d'accord sur ce que doit faire le logiciel
- Communiquer !
 - Échange d'idées avec futurs utilisateurs et futurs concepteurs (faisabilité) : consensus
 - Points de vue différents, complémentaires : ne rien oublier, mais ...
- Spécifications rarement complètes et définitives !
- Le document de spécifications doit définir les caractéristiques principales du logiciel à développer

Spécification des besoins logiciels & planification

- Formaliser les besoins du client
 - Exigences fonctionnelles : quelles fonctions ?
 - Contraintes utilisateur : quelles conditions ?
 - Fonctionnelles ou techniques
- Préparer la stratégie de validation
 - Permettre de s'assurer que le logiciel répond à tous ces besoins
- Planifier la suite du projet

Documents

- Document de spécification des besoins logiciels (DSBL)
 - Spécifications fonctionnelles
 - Fonctions
 - Relations entre elles
 - Description de chacune : entrées, sorties, traitements, données manipulées
 - Spécifications non fonctionnelles
 - Contraintes diverses : fiabilité, performances, règles de gestion ...
 - Interfaces avec le système ou les autres sous-systèmes
 - Logiciel/matériel, logiciel/logiciel

Documents

- Manuel utilisateur (V0)
 - Fonctions du logiciel, du point de vue de l'opérateur
 - Traitements réalisables et enchaînements possibles
 - Procédures de démarrage, arrêt, reprise
 - Présentation des écrans, format des produits en entrée et en sortie
 - ...
 - Peut être complété/remplacé par une maquette de l'IHM

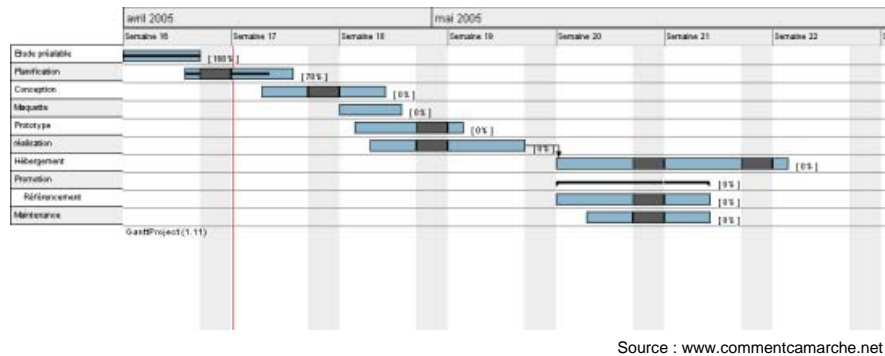
Documents

- Plan de validation du logiciel (PVL)
 - Organisation, conditions de test, moyens à mettre en œuvre
 - Scénarios de test
 - Enchaînements
- Dossier de test de validation du logiciel (DTVL) : recueil des éléments en vue de la recette utilisateur
 - Détail des scénarios de test
 - Entrées
 - Sorties
 - Critères d'acceptation

Documents

- Plan de développement du logiciel (PDL)
 - Tâches à réaliser
 - Activités, produits, responsabilités
 - Notations, méthodes et outils à mettre en œuvre
 - Estimations des charges et délais
 - Modèles d'estimation des coûts et outils associés
 - Planification dans le temps
 - PERT (réseaux de tâches ; optimisation des coûts, charges et délais ; facilite le suivi d'avancement)
 - GANTT (diagrammes barre, issus du PERT ; visualisation de la répartition des tâches dans le temps)

Exemple de Gantt

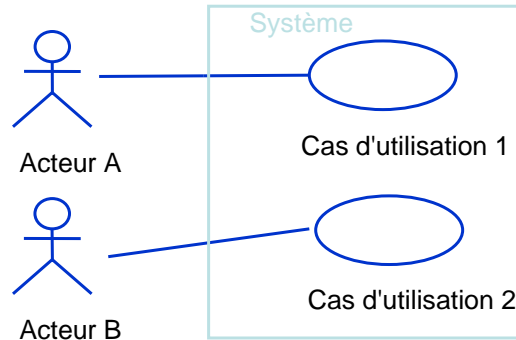


Document de spécification des besoins logiciels (DSBL)

- Rédigé en collaboration avec les futurs utilisateurs
 - Langage compréhensible par des non informaticiens
 - Terminologie du domaine
 - Recours fréquent à des notations graphiques : SADT, UML ...
 - Exemple : modélisation des besoins avec UML

Diagramme des cas d'utilisation UML

- Montre les **relations** entre **acteurs** et **cas d'utilisation** du système



2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

11

Acteur

- **Quelqu'un** ou **quelque chose** qui **interagit directement** avec le système
- **Extérieur** au système
- Participe à **au moins un** cas d'utilisation
- Peut consulter/modifier l'état du système



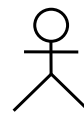
Client



Responsable



Agent de
maintenance



SI
banque

?

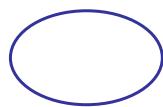
2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

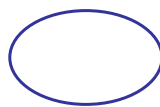
12

Cas d'utilisation

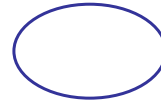
- Pour un acteur donné
- Façon d'utiliser le système
- **Modèle de comportement** : suite d'interactions entre l'acteur et le système
- Produit un **résultat observable intéressant** pour l'acteur



Retirer



Réinitialiser



Tester
reconnaissance
de carte

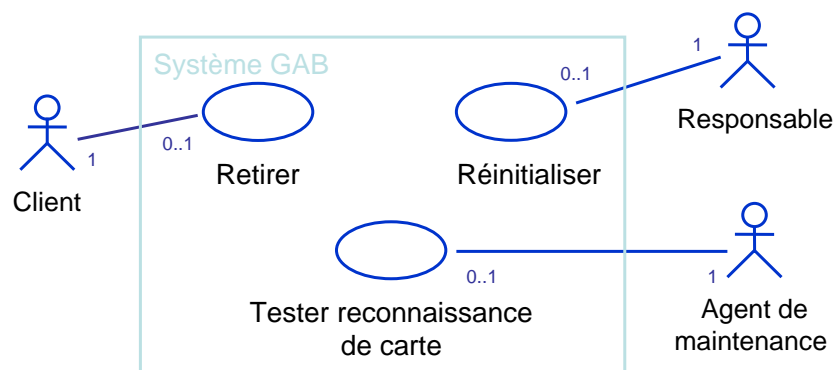
2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

13

Association

- Montre l'**interaction** entre acteur et cas d'utilisation
- Peut porter des indications de multiplicité



2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

14

Cas d'utilisation

- Décrit un **ensemble de scénarios** reliés par un objectif utilisateur commun
- Un **scénario** est une exécution particulière d'un cas d'utilisation
 - nominal
 - alternatif
 - aux limites
 - d'erreur

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

15

Description textuelle structurée

- | | |
|--|---|
| 1. Nom du cas | 3. Alternatives |
| 2. Présentation
Acteur initiateur : ...
But du cas : ...
Pré conditions : ...
[Dire quel doit être l'état du système pour que le cas puisse se déclencher]
Post condition en cas de succès : ...
Cas d'utilisation inclus : ... [optionnel] | Nom de l'alternative
Condition : A l'étape x du scénario principal, ...
x.1 ...
x.2
x.3 le cas se termine (ou le cas se poursuit en ...) |
| 3. Scénario principal aboutissant au succès
1. Ce cas d'utilisation commence quand l'acteur initiateur ...
2. Le système demande ...
3. L'acteur fournit ...
4. Le système ... | 4. Besoins non fonctionnels
Fréquence : ...
Fiabilité : ...
Performance : ...
Règles de gestion : ...
5. Points ouverts
[Lister tous les points actuellement non résolus] |

2011-2012

Henri Massié / L2 Informatique / UE Projet logiciel

17

Exemple

1. Nom du cas : Retirer	
2. Présentation	
Acteur initiateur : Client	
Autres acteurs : SI_Banque	
But du cas : Retirer une certaine somme en espèces	
Pré conditions : Etre titulaire d'au moins un compte, disposer d'au moins une carte bancaire	
Post condition en cas de succès : le client a obtenu la somme demandée, son compte a été débité de la même somme	
Cas d'utilisation inclus : Identification	
3. Scénario principal aboutissant au succès	
1. Ce cas d'utilisation commence quand l'acteur initiateur insère sa carte	
2. Le système demande le code	
3. L'acteur fournit le code	
4. Le système vérifie le code	
5. Le système demande le montant	
6. L'acteur fournit le montant	
7. Le système demande si l'utilisateur veut un reçu	
8. L'acteur répond oui	
9. Le système demande à la banque)	
10. La Banque valide	
11. Le système éjecte la carte	
12. L'acteur prend sa carte	
13. Le système imprime le reçu	
14. L'acteur prend le reçu	
15. Le système fournit les billets	
16. L'acteur prend ses billets	
2011-2012	
Henri Massié / L2 Informatique / UE Projet logiciel	
	4. Alternatives
	Nom de l'alternative : erreur de code
	Condition : A l'étape 4 du scénario principal, le système détecte une erreur de code
	4.1 Le système affiche un message d'erreur
	4.2 Le cas reprend en 2
	Nom de l'alternative : opération refusée
	Condition : A l'étape 10 du scénario principal, la banque refuse
	10.1 Le système affiche un message d'erreur
	10.2 Le cas se termine
	5. Besoins non fonctionnels
	Fréquence : ...
	Fiabilité : ...
	Performance : ...
	Règles de gestion : ...
	6. Points ouverts
	Retrait max par jour et par semaine ?
	qui vérifie : GAB, Banque ? GAB ⇒ règles de gestion, Banque