

## GL2

mardi 28 janvier 2014  
10:23

UML Distilled: Fowler 2004  
Software Engineering: Sommerville 2007

Pierre/Benjamin/Thibault

Créer un git/SVN -> <http://www.commentcamarche.net/faq/13094-utiliser-git-sous-windows>  
<http://msysgit.github.io/>

Intégrer une documentation (rapport technique etc écrit avant le développement)

### Préparation au développement:

\*Analyse des besoins, en particulier des besoins non fonctionnels et fonctionnels.

#### \$ Besoins non-fonctionnels:

Qualités globales que l'on attend du logiciel mais qui ne correspondent pas aux fonctionnalités propres du logiciel.

#### **(Présence très limitée pour le projet GL2)**

Ces besoins sont souvent les plus critiques.

Exemples: \*Besoins en comportement:

Performances, fiabilité, facilité d'utilisation/interface graphique, domaine d'action, portabilité...

\*Besoins organisationnels:

Standards, processus de développement: Méthode d'organisation, Algorithmes...  
Gestion du temps: Plannings, calendrier des livrables.

\*Besoins externes:

Interopérabilité: Interaction avec d'autres systèmes  
Contraintes légales: Code utilisé, open source, GNU/GPL, etc.  
Contraintes éthiques: Acceptabilité par les utilisateurs

Performances acceptables, minimisation des pertes de données, etc. sont des exemples précis de besoins non fonctionnels.

**Il ne suffit pas de simplement énoncer les besoins, il faut quantifier, donner des bornes.**

Par exemple, interface utilisable par des enfants de moins de 5 ans, valeurs min/max des données traitées et des temps de réponse...

Transactions par seconde, temps de réponse, nombre d'octets pris en mémoire, temps moyen avant un échec...

Ces bornes sont déterminées par des prototypes.

#### \$ Besoins fonctionnels:

Fonctionnalités nécessaires au logiciel, ce qu'il doit faire et les services qu'il doit rendre.

Ces besoins appartiennent à deux types, utilisateur ou système.

Exemples: \*L'utilisateur doit pouvoir rechercher dans un ensemble par défaut ou des sous-ensembles de BDD

\*L'utilisateur doit pouvoir garder une trace du résultat de ses requêtes, par exemple par impression(requête utilisateur)

\*Chaque recherche laisse un log (requête système)

La description des besoins fonctionnels doit être:

**Abstraite:** Ne pas influencer l'implémentation

**Claire:** Pouvoir en discuter avec le client

**Précise:** Ils doivent pouvoir servir à détailler les besoins non-fonctionnels.

**Hiérarchisée:** Poser des priorités.

**Ne jamais faire apparaître des éléments liés à des langages ou des styles de programmation.**

=>Ne pas parler du "comment", juste du "quoi"

La description de ces besoins doit être précisée par:

\*La quantification

\*La spécification de tests de contrôle/validation (ex: fixer une taille de données sur laquelle on peut atteindre le seuil de performance raisonnable)

\*La description de contraintes ou difficultés techniques (ici, java imposé)

\*L'énonciation de risques: Prévision de ce qui pourrait mal se passer(ex: Choix techniques sans bonne connaissance(ici, java), ou utilisation d'outils sans maintenance)

\*Leur priorité dans le développement.

En général, la liste de besoins se met en place à l'aide d'itérations successives au fil des discussions avec le client.

#### **Conclusion:**

Etablir une liste des besoins fonctionnels et non fonctionnels, ainsi que leur description et analyse, avec à chaque fois:

-Quantifications

-Tests de contrôle et de validation

-Contraintes et problèmes techniques

-Risques et parades, niveaux de difficulté

-Priorités dans le développement.

\*Elaboration de schémas/représentations graphiques, mettant en avant les besoins et structures du logiciel prévu. (Ex: Diagrammes UML, dessin interface graphique)

\*Planification du développement, calendrier et gestion du temps.

-> Prendre en charge les temps d'apprentissage des technologies utilisées (Java, UML, MVC, etc)

====> Passer 2 semaines sur les besoins, par la suite finaliser le cahier des charges avec les schémas UML, et commencer à coder vers fin Février.

Liste de besoins fonctionnels et non fonctionnels

Description et analyse de ces besoins, avec si nécessaires:

\*Quantifications

\*Priorités dans le développement

\*Tests de contrôle et de validation

\*Contraintes et problèmes techniques

\*Risques et parades, niveaux de difficulté.

Scénarios, représentations, schémas, prototypes papier

Planning DdGantt

#### Réaliser un scénario:

Environnement et acteurs ou le logiciel est requis.

Un scénario induit des besoins fonctionnels, et met en scène les besoins non-fonctionnels.

Ils peuvent être des tests de variation.

Pour justifier les besoins, on peut décrire des ensembles de scénarios.

#### Moyens de représentation:

Représentent le système et ses composants.

Fort efficace, en particulier pour aider à la mise en place des besoins fonctionnels:

-Souvent graphiques

-Souvent multiples

-Prototypes papier, schémas de structure, UML, etc.

UML très utilisé en entreprise en tant que vecteur de communication entre l'ingénieur et le client.

Au minimum:

-Etablir la liste des tâches par rapport aux besoins

-Etablir un calendrier en utilisant l'analyse des besoins et répartir les tâches

-Déterminer les versions et phases successives du programme

(Livrables):

Quelque chose que le client peut voir et tester.

Diagramme de Gantt