

# Projet Génie logiciel

IR 3 - 2006-2007

Rémi Forax & Tibault Crucy

forax@univ-mlv.fr, tcrucy@etudiant.univ-mlv.fr

## Le projet

Le but de ce projet est la réalisation d'un logiciel de suivi et de reporting automatique de l'état d'un projet informatique.

## Documents nécessaire à chaque rendu

Pour chaque rendu, en plus des documents spécifiés, les groupes devront fournir :

- les comptes rendus des diverses réunions que votre équipe a tenu.
- le plan de formation de l'équipe.
- réalisation d'un état d'avancement du projet :
  - le planning prévisionnel de la prochaine phase, ainsi qu'une version moins détaillé des plannings des phases suivantes, sous forme de diagramme de gantt avec un découpage par tâche dans le temps, de plus devront figurer la ou les personne(s) assignée(s) à une tâche ainsi que l'avancement à l'intérieur d'une tâche.
  - le planning réalisé de la phase courante ainsi qu'une explication sur :
    - les problèmes rencontrés et solution pour palier à ceux-ci.
    - les écarts au planning (dans les deux sens !!).
  - une fiche récapitulatif par personne de l'équipe de l'ensemble de ses taches assignés indiquant pour chaque tâche le temps prévisionnel et le temps passé ainsi que le nombre total d'heures passé sur le projet.
  - Donc sous forme d'un tableau :
    - récapitulatif du temps passé par personne pour chacune de ses tâches
    - répartition du temps à venir, attribution des tâches futures aux membres de l'équipe
- une fiche de suivi et de notation par personne de l'équipe (chef de projet exclu).

Tous ces documents devront être rangés dans le cahier de projet.

La liste ci-dessous expose les différentes rendez-vous de réalisation du projet ainsi que les documents spécifiques à fournir lors de chaque rendu. Notez que vous avez le droit de vous organiser comme vous le désirez à l'intérieur de votre équipe et que décidez de créer un document avant que celui-ci soit demandé n'est pas forcément une erreur.

## Phase 1 : Présentation du projet/Cahier Des Charges

Le but de cette étape est :

- la présentation de ce document
- la création des chacunes des équipes
- la nomination du chef de projet
- l'établissement du calendrier des différentes rendu du projet

L'équipe devra

- trouver un nom d'équipe
- attribuer à chaque membre de l'équipe un ou plusieurs rôles :
  - chef d'équipe  
(anime les début et fin de réunion, établi ordre du jour de la prochaine réunion, fixe le planning (cf état d'avancement), les objectifs de chacun, indique les plus et moins de chacun et participe à la réunion des chefs)
  - un responsable client  
(interlocuteur qui dialoguera exclusivement avec le client)
  - un responsable de la qualité de la documentation  
(lisibilité, uniformité, template documentaire et secrétariat des comptes rendus)
  - un expert technique  
(référenciel des solutions techniques employés)
  - un expert qualité du code et tests  
(qualité du code, intégration entre les composants, respect du code par rapport au spécifications, couverture des tests)
  - un expert de l'interface graphique  
(ergonomie, accessibilité des fonctionnalités)
  - un gestionnaire du site web interne et outil interne  
(devra mettre en place le site, créer les procédures permettant à chaque membre de l'équipe d'ajouter des documents, de maintenir le site à jour)

Pour le rendu suivant l'équipe devra réaliser un pré-cahier des charges en fonction de la réunion avec le client

(ce document permet au client de vérifier que votre équipe a bien compris et bien noté toutes ses volontés)

**Cahier des Charges** (max 40 pages) contient :

- une description de l'équipe
- (rôle et compétence de chacun, etc)
- glossaire des termes importants/termes du client
- une description du logiciel (au moins 7 pages)
  - recensement des acteurs
  - relation entre les acteurs et le système d'information
  
- une description rapide de l'architecture logiciel
- une description des différents lots/livrables correspondant au logiciel  
(chaque lot devra contenir les macro-fonctionnalités du logiciel)
- contraintes techniques émises par le client

Le CDC, est le document (donc avec une intro, un placement dans le contexte, etc.) servira de base pour la signature du contrat.

## **Phase 2 : Définition Des Outils De Travail**

Le début de la phase 2 commence par la signature du pré-cahier des charges établissant un contrat entre le client et votre équipe.

Pour la suite du projet, si jamais, vous vous apercevez que vous avez proposé au client

quelque chose de pas réalisable (compte tenu des délais, de la technique etc.) vous devrez rédiger une notice de changement ou de modification de fonctionnalité qui devra être accepté par le client comme avenant au cahier des charges.

Pour le rendu suivant l'équipe devra :

- créer d'un site web (intranet), wiki etc. permettant d'effectuer un suivi du projet de façon électronique. Ce site devra dans un premier temps regrouper l'ensemble des documents servant à la gestion du projet
- l'équipe devra créer un document définissant le outils de travail, **Définition Des Outils De Travail** (max 30 pages) regroupant :
  - un document décrivant les différents outils de gestion et de suivi de projet utilisés, classés en fonction de la réponse au besoin de votre équipe, en insistant sur les différents outils que vous avez envisager d'utiliser pour répondre au besoin et sur les raisons qui vous ont poussé à choisir un outil plutôt qu'un autre.
  - un pré-rapport technique
    - identifié les catégories de problèmes ou besoins techniques (par exemple, base de données, langage de programmation, bibliothèque pour ...)
    - pour chaque catégorie, une liste de logiciels, bibliothèques utilisables par votre projet et permettant de répondre à chaque besoin technique

Tous ces documents devront être rangés avec ceux déjà produit et mis à jour dans le cahier de projet. L'avancement du projet devra être mis à jour en fonction des réunions, taches etc. effectuées et le déroulement les phases suivantes devront être prévu.

### **Phase 3 : Cahier des charges graphiques**

Le cahier des charges graphiques est un document qui permet au client d'avoir une bonne idée du rendu graphique de l'application qu'il a commandé.

Pour le rendu suivant l'équipe devra établir un **Cahier Des Charges Graphique** (max 40 pages) contenant :

- une charte Graphique :
  - style et police utilisé
  - ergonomie générale de l'application
  - etc.
- un document pour chaque interface graphique décrivant celle-ci par zone
  - séparation de l'interface graphique en différentes zones conceptuellement différentes
  - description pour chaque zones les sous-zones associées et les composants graphiques associés (menu, bouton, texte etc.)
  - Expliquer la dynamique (les actions) de l'application et relié les composants graphiques aux fonctionnalités correspondantes en indiquant précisément les paramètres de celles-ci.

L'ensemble des documents doivent décrire précisément l'interface graphique en français. Vous pouvez faire des copies d'écran uniquement s'il s'agit de copie d'écran d'un prototype que vous aurez créé (pas de patchwork de différents logiciels). Les copies d'écrans devront être à jour par rapport aux textes.

## **Phase 4 : Cahier des charges fonctionnels/ Cahier des charges techniques**

Le début de la phase 2 commence par la signature du pré-cahier des charges établissant un contrat entre le client et votre équipe.

### **Cahier Des Charges Fonctionnelles (80 pages max)**

Ce cahier des charges devra contenir de façon exhaustive l'ensemble des cas d'utilisation (Use Cases) qui devront être livrés lors de la version 1 (V1).

Ce document devra contenir :

- une description de l'équipe
- une description du logiciel
- une description des différents lots/livrables correspondant au logiciel avec pour chaque lot sa date de rendu
- Pour l'ensemble des lots, une liste des objets du domaine
- Pour chaque lot de la V1,
  - pour chaque macro-fonctionnalité :
    - sa priorité de développement
    - la ou les qualités demandées
    - les procédures de mesure ou test permettant de vérifier les qualités (bref FQM)
- créer un diagramme détaillant les paquetages et les relations entre ceux-ci (il faut avoir fait tous les diagrammes de paquetages avant)
- par paquetage, un diagramme détaillant les relations entre les UC d'un même paquetage (et celle des autres paquetages s'il y en a)
- L'ensemble des Use Cases d'un paquetage avec une fiche très détaillée par UC. La fiche devra pour chaque UC indiquer :
  - les acteurs impliqués
  - la priorité du UC
  - les précondition et postcondition
  - le scénario décrit sous forme de discussion entre les acteurs et le système (demande des acteurs, réponse du système) le scénario doit être très détaillé et sans terme d'interface graphique.
  - les éventuels scénarios alternatifs et exceptions

### **Cahier Des Charges Techniques (80 pages max)**

Le cahier des charges techniques est une réponse technique au cahier des charges fonctionnelles. Il indique principalement l'architecture logicielle du projet ainsi que les choix techniques qui ont été effectués.

Celui-ci devra contenir:

- une étude technique :  
Cette étude reprend le pré-rapport technique en indiquant les solutions techniques retenues
  - une description de l'architecture logicielle, les bibliothèques, logiciels et plateformes retenue
  - pour chaque catégorie du DDODT, une étude comparative des briques logiciel s'appuyant sur des exemples utilisables pour le projet ainsi que les raisons du choix de la brique logiciel.
    - Justification des choix retenus
    - Implication sur l'architecture de la solution retenue
- les choix de design effectués et les design patterns utilisés (ceux utilisés dans votre logiciel et avec des exemples illustrant leur utilisation dans votre logiciel)
- les UCs générique (parties communes à plusieurs UC utilisant en règle générale des briques logiciels particulières)
- une description du découpage de l'application par paquetage Java avec pour chacun son utilité
- un diagramme de classe pour chaque paquetage Java avec une description du rôle de chaque classe ainsi que l'ensemble des tests à effectuer pour une classe
- Une mise à jour du CDCF, celui-ci devra contenir en plus :
  - un diagramme de séquence UML par Use Case indiquant les appels de méthodes entre les objets du domaine, les patterns et les objets des briques logicielles.
- une mise à jour de l'état d'avancement du projet incluant en sus une prévision précise de quel personne de l'équipe doit développer ou tester quelle classe et ou paquetage.

## **Phase 5 : Développement et Test du prototype (V1)**

Cette phase correspond à la création du code de l'application et des tests validant la qualité du code produit et la non-régression des anciennes fonctionnalités à chaque fois qu'une nouvelle est ajoutée.

Pour la version 1 comme pour les versions futures, l'équipe devra désigner :

- un responsable fonctionnelle
- un responsable technique
- un responsable graphique
- un responsable du code
- un responsable des tests

Il est demandé que ces responsabilités soit attribué à des personnes différentes entre chaque version.

L'équipe doit effectuer la :

- Création des paquetages
- Création des classes
- Rédaction de la documentation développeur de chaque classe

- Création des tests unitaires de chaque classe et gérer de la couverture des tests
- Validation des classes créées par rapport aux tests
- Rédaction de fiche de bug
- Mise en place du paramétrage de l'application (fichiers de configuration)
- Rédaction d'un document de déploiement

Noter qu'il n'y a pas d'intégration.

**La recette** : lors de la démonstration au client, l'équipe devra fournir des fiches de recette permettant au client de valider chaque fonctionnalité une par une. De plus, si le client demande de nouvelles fonctionnalités ou des améliorations notables, un document permettant de définir des avenants devra être prévu à cet effet.

## **Vi: Pour les versions suivantes :**

Le CDCF et CDCT devra être complété avec les nouvelles fonctionnalités: et donc compléter les documents en décrivant :

- l'ensemble des macro-fonctionnalités (FQM) ajoutées
- la spécification graphique correspondante
- l'ensemble des Uses Cases
- un diagramme de séquence par UC (appel de méthodes entre les objets du domaine, les patterns et les objets des briques logicielles)
- l'ensemble des paquetages Java (diagramme des classes + explication de texte) issus des diagrammes de séquence.
- l'ensemble des tests réalisés (une fiche par test)

Pour chaque bug, dans la fiche de bug

- à qui le bug a été assigné
- l'analyse du pourquoi du bug
- le test correspondant au bug (pour éviter les régressions)
- la façon de corriger celui-ci

## **La version Finale (V5)**

Ce dernier rendu se soldera par la démonstration au client de la version finale du logiciel. En plus du travail habituel sur la version, l'équipe devra fournir :

- un manuel utilisateur (ici 2 car il y a deux interfaces graphiques)

## **Qualités des documents à fournir**

Chaque document devra posséder :

- un template documentaire reprenant le logo de l'équipe et son nom
- un titre
- un numéro de version
- une date et un auteur lors de la création puis pour chaque modification
- une date et un auteur pour chaque validation
- des pages numérotées
- un sommaire
- une introduction pour situer le contexte (le document par rapport aux autres)
- un plan clair

Essayer de toujours avoir des textes clairs et précis. Tous les diagramme devront être à la norme UML et chaque diagramme devra avoir un commentaire explicatif à coté. Pensez au lecteur !

L'ensemble des documents à fournir est regroupé dans le **cahier de projet**, ce cahier sera complété au fur et à mesure de la création des documents, le chaier de projet devra toujours être à jour.

## Calendrier

Lors d'une recette, pour la validation par le client, les documents doivent être regroupés dans le cahier de projet sont à fournir au format PDF eux-même contenus dans une archive ZIP au plus tard le jeudi à minuit.

<b>Date</b>	<b>Réunion et documents à fournir</b>
25 septembre 2006	Présentation du projet et création des équipes
10 octobre 2006	
17 octobre 2006	CDC
24 octobre 2006	DDODT
31 octobre 2006	CDCG (client lourd + client léger)
14 novembre 2006	CDCF+CDCT
05 décembre 2006	V1
18 décembre 2005	V2
16 janvier 2007	V3
06 février 2007	V4
27 février 2007	Livraison version 1.0 (V5)