

# Licence 2e année - Semestre 3

## Gestion du projet S3

Vincent Dugat

4 octobre 2018

### Table des matières

<b>1</b>	<b>Introduction générale</b>	<b>1</b>
1.1	But du projet de S3 . . . . .	1
1.2	La gestion de projet . . . . .	2
<b>2</b>	<b>Les livrables (10 pts)</b>	<b>3</b>
2.1	Document de conception (Plan) 3 pts . . . . .	3
2.2	Framaboard . . . . .	4
2.3	Etat d'avancement (Plan et Review) 1 pts . . . . .	4
2.4	Doc de bilan (Review) 3 pts . . . . .	4
2.5	Validation (Review) 3 pts . . . . .	5
<b>3</b>	<b>La programmation (Do) 10 pts maximum</b>	<b>5</b>
<b>4</b>	<b>Ressources</b>	<b>5</b>
<b>5</b>	<b>Synthèse des livrables</b>	<b>6</b>

## 1 Introduction générale

Le projet de S3 démarre au tout début du mois d'octobre et sera validé la première semaine de la rentrée de janvier. Il dure donc trois mois pleins, et représente environ 1000+ vraies lignes de code (les commentaires ne comptent pas).

### 1.1 But du projet de S3

- Faire le lien entre les différents savoirs et compétences acquises dans les UE de L1 et L2.
- Illustrer des principes de ré-utilisation de code, modularité, tests, documentation sur un programme plus ambitieux qu'un exemple de TP forcément plus limité.
- Rendre indispensable une organisation et une méthodologie de développement (préliminaire à l'étude des méthodes industrielles de gestion de projet)

Dans ce but, le projet que vous allez programmer ne se limite pas au code informatique. Il intègre une méthodologie de travail. En clair le projet comporte deux aspects :

- Le projet de programmation proprement dit (l'implémentation) que nous appellerons *partie technique*,
- Le cadre organisationnel que nous appellerons *partie gestion de projet* qui est indépendant du sujet proprement dit.

Le présent document traite de l'aspect organisationnel et donc de la gestion du projet. La partie technique (c'est à dire le détail de ce qu'il faut programmer) est traitée dans un document séparé.

Dans le cadre de la L2 informatique, il y a une U.E. projet par semestre. Pour le S3, **le projet est individuel**. Les semestres postérieurs vous donneront l'occasion de travailler en équipe de plusieurs personnes.

## 1.2 La gestion de projet

Dès qu'un projet de programmation devient complexe, on emploie des techniques de gestion de projet. Ces techniques sont classiques dans l'industrie tous secteurs confondus. Dans le cas du logiciel on utilise des techniques de génie logiciel. Ces techniques seront abordées dans les semestres postérieurs. Pour ce projet de S3, nous allons utiliser un simple principe de management : PLAN, DO, REVIEW (ou PLAN, DO, CHECK, ACT).



FIGURE 1 – Le cycle PDR

- **PLAN :**
  - Réfléchir à ce que vous allez/devez faire (compréhension du problème, spécifications informelles et formelles, prévisionnel)
  - Diviser le travail en sous-parties
  - Que quoi avez-vous besoin : machines, logiciels, environnement de programmation ?
  - De quelles connaissances/savoir-faire avez-vous besoin, allez-vous devoir les apprendre ?
  - Comment allez-vous tester les parties, le logiciel, quel peut être *a priori* le degré de sûreté de votre code ?
- **DO :**
  - Obtenir ce qui manque
  - Apprendre les nouvelles compétences
  - Programmer les différentes parties et les tester en suivant le plan
  - Assembler les parties et tester les assemblages ou le programme final
- **REVIEW :**
  - En cours de développement :
    - Suis-je en train d'avancer vers la résolution du problème de départ ?
    - Ai-je besoin de modifier le plan ? Pourquoi ? Comment ?
    - Si oui, modifier le plan, documenter la modification
  - En fin de projet :

- Ce qui marche, ce qui ne marche pas
- Ai-je atteint les objectifs ? Pourquoi ?
- Ai-je appris des choses ?
- Comment ai-je résolu les problèmes ?
- Suis-je resté en deça de mon savoir-faire ? Ai-je augmenté mon savoir-faire ?
- Me suis-je challengé moi-même ?
- Mon plan de départ était-il juste ? Ai-je dû le changer ? Selon quelle proportion ?
- Ce projet m'a-t-il été utile ?

Afin de communiquer avec les autres parties (client, chef de projet), dans notre cas le tuteur, le responsable d'UE, un certain nombre de documents vont être demandés avec une date butoir. On les appelle les livrables.

## 2 Les livrables (10 pts)

Donc ce sont des documents à rendre. La clarté et la lisibilité de vos documents sera prise en compte. Tout écrit confus, superficiel, alambiqué risque d'être peut rétribué. Le but de cette documentation est d'être informative dans une entreprise où les développeurs passent d'un projet à l'autre<sup>1</sup>.

### 2.1 Document de conception (Plan) 3 pts

Vous devez rendre sur Moodle un document préliminaire où vous décrivez :

- La liste des modules que vous pensez pouvoir programmer *a priori* (cf. sujet) dans le temps imparti.  
**Un dépassement des objectifs est possible, ne pas les atteindre devra être argumenté dans le document de bilan.**
- La structure de votre programme : identification des différentes parties, modules, sous-modules, interaction entre les parties,
- Un plan de tests et un échéancier : comment allez-vous tester les modules ? Pouvez-vous *a priori* prévoir les tests des différents modules et des assemblages de modules jusqu'à l'intégration finale ? Quand allez-vous les tester ?
- L'échéancier sous forme de diagramme de Gantt (export Framaboard voir plus loin).

En détail :

#### Le cadre administratif

- Nom,
- Prénom,
- Groupe TP,
- Mail UPS
- Code apogée UE (c'est sur la page Moodle)
- Titre du projet
- Année civile en cours
- Tuteur avec son mail
- Fiche échéancier prévisionnel (export Framaboard)

#### Le cadre de développement

- Sur quelle(s) machine(s) et environnements(s) allez-vous développer ? Est-elle assez puissante, fiable, accessible ?
- Y a-t-il tous les logiciels (compilateurs, éditeur, Doxygen, utilitaires systèmes, etc.) dont vous avez besoin ?

---

1. Dans la vraie vie (professionnelle) on lit plus de code qu'on en écrit. C'est pourquoi on doit programmer de la manière la plus lisible possible car on va être lu. C'est une question d'efficacité générale (de l'entreprise, du labo, du projet libre, de l'asso de développeur, etc.)

- Comment gérer les sauvegardes du code en cas de crash ? (GitHub, dropbox, repl.it, clé USB, NAS, c'est l'admin système qui le fait ?,...) )
- Y a-t-il des softs à installer avant toute chose ?

### Le projet

- Quelles structures de données : matrice, vecteurs, autres, combien, pourquoi, quels types ?
- Combien de fichiers de code, quels rôles, quelles sont les SD partagées, comment ?
- Fichier header : combien, quels rôles ?
- Makefile ou pas ?
- Quel ordre dans le développement des fonctionnalités ? Y a-t-il des dépendances ?
- liste exhaustive des fonctionnalités, des variables globales, des paramètres nécessaires à chaque fonctionnalité

### Gestion du projet

- Quelles étapes principales voyez-vous ? Leurs ordre temporel ?
- Evaluation du temps de développement de chaque étape
- Comment tester chaque étape, fonctionnalité, sous-programme, bout de code ?
- Etes-vous capable de lister les tests avant même de commencer à développer ?
- Quel temps prévoyez-vous pour les tests ?
- Pouvez-vous vous projeter dans le temps et planifier les développements et les tests au point de donner une date de fin pour chaque étape (développement et tests) ? Pour le projet aussi ?
- Comment allez-vous gérer l'avancement du projet ? Les retards éventuels ?

Ce document doit aller à l'essentiel mais pas trop. Le détails des parties, sous-parties et échéancier sera visible sur Framaboard. Inutile de le répéter ici.

## 2.2 Framaboard

Afin de gérer la feuille de route du projet nous allons utiliser un logiciel de gestion de projet : Framaboard. Il s'agit d'un logiciel où vous pouvez décrire l'ensemble des tâches et de leurs sous-tâches de votre projet et leur donner une étiquette comme : en attente, prêt à coder, en cours de développement, en test, validé, en retard, etc. Le logiciel permet des commentaires, de faire un échéancier et votre chef de projet (tuteur) aura accès à votre tableau de bord.

Votre compte Framaboard sera créé par le staff de l'UE (enfin... par moi). Vous serez notifiés (identifiant et mdp) sur votre adresse UPS ou par votre tuteur.

Un tuto sur ce logiciel sera fait en amphi.

<https://projetcondorcet.framaboard.org/?controller=AuthController&action=login>

## 2.3 Etat d'avancement (Plan et Review) 1 pts

Vous devez rendre sur Moodle un état d'avancement à mi-projet qui définit ce qui a été fait, ce qui reste à faire, les retards éventuels et leur causes et les ajustements par rapport au prévisionnel. Cette fiche sera un simple export de votre projet Framaboard **à jour**.

Il est fortement conseillé d'utiliser le logiciel comme cadre de développement au jour le jour (et pas seulement quand vous y pensez et parce que c'est demandé).

## 2.4 Doc de bilan (Review) 3 pts

A la fin du projet, vous indiquerez dans un bref document :

- Ce qui a effectivement été fait, ce qui fonctionne et ce qui ne fonctionne pas dans cette version (une simple liste), les bug repérés mais non résolus (un simple tableau avec un pourcentage peut suffire à faire le job).
- Une estimation à la louche du pourcentage de votre plan qui s’est révélé juste (un nombre et deux lignes de commentaires au plus),
- Une estimation du degré de fiabilité de votre programme,
- Un bilan personnel : ce qui vous a posé des difficultés, comment les avez-vous résolues (ou pas), ce que vous avez appris au cours de ce projet, tant sur le plan organisationnel que sur le plan technique. Cette partie inclut une analyse d’échecs éventuels et de la recherche leurs causes. Il s’agit d’un simple débriefing de l’opération.

## 2.5 Validation (Review) 3 pts

En fin de projet votre programme sera validé. Cette étape est souvent appelée recette dans l’industrie du logiciel :

[http://fr.wikipedia.org/wiki/Test\\_d'acceptation](http://fr.wikipedia.org/wiki/Test_d'acceptation)

*En informatique, le test d'acceptation (ou recette) est une phase de développement des projets, visant à assurer formellement que le produit est conforme aux spécifications (réponse donnée à un instant « t » aux attentes formulées). Elle s'inscrit dans les activités plus générales de qualification. Cette étape implique, en la présence effective des différents acteurs du projet, maîtrise d'œuvre et maîtrise d'ouvrage, le déroulement rigoureux de procédures de tests préalablement décrits, et l'identification de tout écart fonctionnel ou technique.*

**Conséquence : absence à la recette = abi à la partie code.**

Typiquement cela se passe en deux temps :

1. vous devez faire la démonstration du fonctionnement de votre programme à votre enseignant tuteur (éventuellement en projection vidéo devant le groupe),
2. dans un deuxième temps vous devez répondre à ses questions techniques et organisationnelles. Dans cette partie, le tuteur est libre de demander une modification du programme en live ou l’exécution du programme sur un jeu de test qu’il fournira.

## 3 La programmation (Do) 10 pts maximum

Le sujet de la partie technique est à télécharger sur Moodle. Il est divisé en modules de difficultés variables. Chaque module rapporte potentiellement un certain nombre de points. Vous devez indiquer dans le document de conception un choix de modules de que vous pensez pouvoir mener à bien. Vous devez tout faire dans un module pour qu’il vous soit crédité.

Tout faire dans un module signifie :

- Avoir implémenté l’ensemble des fonctionnalités demandées en respectant les spécifications et avoir testé et validé ce code (50% des points)
- Avoir respecté les règles de programmation propre (clean code) et commenté ce code avec Doxygen (50% des points)

Dans ce cas, et dans ce cas seulement, vous pouvez espérer l’ensemble des points du module.

## 4 Ressources

Chaque étudiants aura un enseignant tuteur qui sera son référent pour le projet. Bien sûr, vous pouvez fouiller le web, la BU, vos relations, travailler à plusieurs, mais, **aucun plagiat ne sera toléré. Votre**

**code est personnel.**

Les tuteurs organisent des réunions avec les étudiants qu'ils suivent. C'est l'occasion de faire des bilans d'avancement, de poser et résoudre des problèmes, de demander des conseils. Les tuteurs peuvent être contactés par email, ou peuvent vous contacter. Le tuteur joue le rôle du client, partiellement du chef de projet, et bien sûr de l'enseignant qui va évaluer votre travail.

Un projet informatique (à la fac, dans l'industrie, etc.) est un challenge technique mais aussi une communication avec vos pairs et votre tuteur. Il sera grandement apprécié de venir aux réunions et répondre aux mail.

**Rappel : Dans tous les cas, ne pas venir à la validation, même en ayant déposé un code sur Moodle, sera considéré comme ABI.**

## 5 Synthèse des livrables

### Début de projet

- document technique de conception
- projet privé Framaboard à créer

### Tout le long du projet

- Framaboard doit être mis à jour en continu

### Milieu de projet

- Export Framaboard de l'état d'avancement

### Fin de projet

- déposer la doc de bilan.
- déposer le code
- mettre à jour la documentation Doxygen

`https://projetcondorcet.framaboard.org/?controller=AuthController&action=login`