

# Gestion de Projet Informatique 2000-2010

## Rapport de Planification

DUBUC XAVIER

Première année du grade de master en sciences informatiques  
Faculté des Sciences, Université de Mons  
Av. du champ de Mars 6, 7000 Mons, Belgium  
XAVIER.DUBUC@student.umons.ac.be

23 octobre 2010

**Résumé** Ce rapport est rendu dans le cadre du cours de “Gestion de Projets Logiciels” (dispensé par Monsieur *Tom Mens* en année académique 2010-2011). Le but de ce rapport est d’énoncer les ressources matérielles et humaines nécessaires ainsi que les contraintes de temps, de budget et de fonctionnalité du projet.

## 1 Introduction

### 1.1 Objectifs

Le travail consiste à lire, comprendre et implémenter une méthode récente de recherche du nombre chromatique d’un graphe, la recherche dans des espaces variables, proposée par *Hertz* et al [1]. Pour rappel, le nombre chromatique d’un graphe est le nombre de couleur minimum nécessaires à la coloration de ses sommets de telle manière que des sommets adjacents n’aient pas la même couleur.

### 1.2 Exigences fonctionnelles

Le logiciel devra être capable de réussir les tests effectués sur des instances réputées difficiles (DIMACS challenges [2]) et, si le temps le permet, applicable sur prototype simple de logiciel de planification d’horaire d’examens.

### 1.3 Exigences non-fonctionnelles

Aucune exigence quant à la portabilité mais le projet sera implémenté à JAVA afin de nous assurer cette portabilité tout de même, il va sans dire que les structures de données et algorithmes implémentés devront être optimisés afin de diminuer les temps de calcul.

#### 1.4 Contraintes de temps

*Y a-t-il des contraintes sur l'emploi du temps ? (Dates d'échéance, vacances, examens, autres) ? Si vous le désirez vous pouvez utiliser une table ici pour clarifier l'emploi du temps. Vous pouvez également ajouter une figure.*

Il y a 3 échéances au total,

- le **28 octobre 2010**, date où il faut rendre le rapport de planification qui devra donc être fini avant cette date. Ceci implique que la compréhension du projet, les démarches administratives, ainsi que la planification doivent être terminés avant cette date.
- début **janvier 2011**, date où il faudra rendre le prérapport, ceci implique donc que toutes les recherches et autres préliminaires ou prémises (conception, ...) doivent être finis avant cette date afin que ces résultats soient incorporés au rapport.
- **mai 2011**, date que nous nous fixons pour rendre le dernier livrable, c'est-à-dire le programme implémenté, testé et débuggé, devant rendre le rapport pour le mois de juin, il convient de nous laisser une plage de temps acceptable pour la rédaction de celui-ci.

#### 1.5 Contraintes de budget

Le logiciel n'utilisant que JAVA comme base et les DIMACS challenge n'étant également pas payant, aucune contrainte budgétaire n'est à remarquer.

## 2 Ressources

### 2.1 Les ressources humaines (personnel)

- *Le responsable des projets (Tom Mens)* s'occupe de la validation des choix de projets et de l'établissement du calendrier. C'est également lui qui résoud les éventuels problèmes liés à l'organisation.
- *Le directeur de projet (Hadrien Melot)* oriente l'étudiant vers la documentation nécessaire à l'accomplissement du projet, il s'assure également que ce projet est réalisable dans les délais. Il fait également partie de ceux qui évaluent le travail final après sa remise. *(Il établit également, avec l'étudiant, le cahier des charges)*
- *L'étudiant (Xavier Dubuc)* réalise le travail, il se documente, lit et comprend cette documentation ; écrit des rapports à intervalles fixés, développe l'application en choisissant des outils (ou en utilisant ceux imposés) pour mener le projet à bien. C'est lui le responsable du travail et du rapport final.
- *Le(s) rapporteur(s) (Véronique Bruyère et ?)* complètent l'équipe qui évaluera le travail de l'étudiant après sa remise, cette évaluation sera donc basée sur le travail en lui même, le rapport final ainsi que sa présentation.

### 2.2 Les ressources logicielles

L'application sera développée en **JAVA**, cette application ne nécessite aucune librairie supplémentaire ni de programmes complémentaires. **JAVA** étant *OpenSource*, aucun coût n'est donc à imputer, comme spécifié dans la section des contraintes de budget.

### **2.3 Les ressources matérielles**

La machine de développement sera celle de l'étudiant (*ASUS X64J, Intel Core i3-350M (2,26 GHz), 4Gb DDR 3*) doté d'un dual boot Windows 7even/Ubuntu 10.04 (probablement 10.10 prochainement) permettant de tester la compatibilité de l'application sous Windows et Linux (pour tester la portabilité).

### 3 Analyse des risques

#### 3.1 Identification des risques

Risque	Probabilité	Sévérité
Difficultés d'implémentation des structures	basse	sérieuse
Mauvaise compréhension d'un concept	basse	sérieuse
Problème de gestion du temps	modérée	tolérable
Produit final défectueux	modérée	sérieuse

TABLE 1. Analyse des risques.

L'implémentation des structures et la compréhension des concepts étant la base même du projet, une quelconque erreur dans ces 2 domaines peuvent entraîner une perte de qualité certaines ainsi qu'un retard considérable (par exemple si une structure est mal implémentée, le produit final sera plus lent et si un concept est mal compris, l'algorithme ne fonctionnera pas et il faudra le débogger). Heureusement ces risques sont de probabilité faible car l'implémentation des structures (graphes) a déjà été fait les années précédentes et ne devrait pas poser de problème. La compréhension des concepts, au vu des concepts utilisés qui sont relativement simples ne devraient pas poser de problème non plus.

La gestion du temps sera pour le moins délicate, car nous aurons à gérer tous les autres travaux qui nous seront demander en plus de celui-ci et un retard est vite arrivé dans ces cas-là ; ce risque reste tolérable car on peut toujours rattraper un retard (tant qu'il n'est pas trop conséquent).

Le produit final défectueux quant à lui posera pas mal de souci également, vu qu'il y a beaucoup de concept à implémenter, le débogage ne sera pas aisé, que ce soit parce qu'il ne fonctionne pas ou parce qu'il est trop lent.

#### 3.2 Gestion des risques

1. *Risque de difficultés d'implémentation des structures*, il convient, en vue de minimiser ce risque, de bien se documenter sur les structures employées ainsi que des meilleures manières de les implémenter. Ne pas hésiter à poser des questions si un choix n'est pas facile à prendre. Si malgré tout le risque est là, il faudra revoir toute l'implémentation de ces structures afin de résoudre le problème occasionné.

2. *Risque de mauvaise compréhension d'un concept*, il convient de bien lire la documentation et de résumer chaque concept de manière à être capable de l'expliquer et de l'appliquer. Si malgré tout le risque est là, il conviendra de relire la documentation attentivement, en chercher de la nouvelle et/ou en parler avec le directeur de projet ou quelqu'un spécialisé dans ce domaine.
3. *Problème de gestion du temps*, il convient d'effectuer également une planification des autres projets et travaux à effectuer pour les autres cours afin de concorder avec celui-ci, il convient également de faire passer ce projet-ci avant les autres en cas de conflit. Si malgré tout le risque est là, il faudra travailler plus vite ou plus souvent pour rattraper le retard encourru.
4. *Produit final défectueux*, il convient de tester le programme assez souvent afin de ne pas avoir un énorme debug à faire au terme du projet. Si malgré tout le risque est là, une période de temps "ultimes corrections" est prévues en cas de besoin, celle-ci empêtrant un peu sur la rédaction du rapport final (voir diagramme de GANTT).

## 4 Répartition du travail

### 4.1 Work Breakdown Structure

ID	Tâche	Dates	Responsable	Autre remarques
T1	Établir le cahier de charges	18 octobre	L'étudiant et le directeur	peut être sujet à modification
T2	Rédiger le rapport de planification	22 octobre	L'étudiant	doit être rendu pour le 28 octobre
T3	Faire la conception de l'application	Novembre - Début Décembre	L'étudiant	
T4	Rédaction du pré-rapport	Décembre	L'étudiant	
T5	Relecture du pré-rapport	Début janvier	Le directeur	
T6	Implémentation et Tests	Janvier-Mai	L'étudiant	
T7	Validation du projet	Début Mai	Le directeur	
T8	Rédaction du rapport final	Mai	L'étudiant	
T9	Relecture du rapport final	Fin Mai - Début Juin	Le directeur	
T10	Dépôt du projet	Juin	L'étudiant	
T11	Défense orale du projet	Juin	L'étudiant (et le directeur)	

TABLE 2. Tableau des tâches.

### 4.2 Étapes clés

Date	étape clé	Livrables
30 septembre 2010	Entretien d'acceptation du projet	Engagement
28 octobre 2010	Réunion d'inspection de la planification de projet	Rapport de planification
Janvier 2011	Réunion d'inspection de l'état d'avancement du projet	Pré-rapport
Mars 2011	Réunion d'inspection de l'implémentation des structures	Code JAVA
Avril 2011	Réunion d'inspection de l'implémentation des algorithmes	Code JAVA
Mai 2011	Réunion d'inspection du produit final	Code JAVA
Juin 2011	Remise du logiciel (après corrections éventuelles)	Code JAVA final

TABLE 3. Tableau d'étapes clés.

## 5 Ordonnancement

### 5.1 Diagramme GANTT

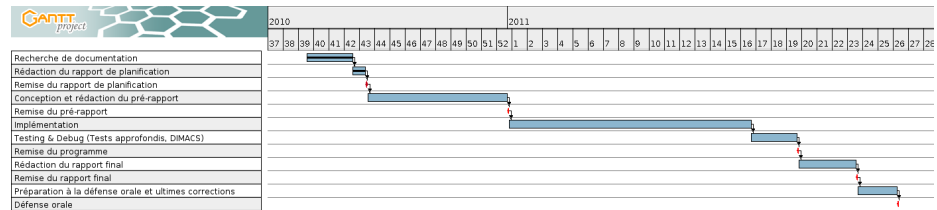


FIGURE 1. Diagramme GANTT.

### 5.2 Diagramme PERT

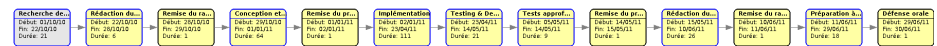


FIGURE 2. Diagramme PERT.

(voir TABLE 4)

### 5.3 Analyse de l'ordonnancement

L'ordonnancement est basé sur le modèle de la cascade d'eau, l'étudiant étant seul pour travailler sur ce projet, il est impossible de mettre des tâches en parallèles, le chemin critique est donc l'unique chemin possible.

### 5.4 Surveillance

Nous pouvons utiliser la technique de la ligne de temps sur le diagramme de GANTT car on a peu de tâches, tâches qui sont spécifiques, de plus de cette façon nous couvrons toute l'échelle de temps.

Tâche	Durée	Effort	Earliest Start	Latest Start	Slack Time	Free Float
Recherche de documentation	21	0,7	01/10/10	01/10/10	0	0
Rédaction du rapport de planification	6	0,2	22/10/10	22/10/10	0	0
Remise du rapport de planification	1	0,03	28/10/10	28/10/10	0	0
Conception et rédaction du pré-rapport	64	2,13	29/10/10	15/11/10	16	0
Remise du pré-rapport	1	0,03	01/01/11	01/01/11	0	0
Implémentation	111	3,7	02/01/11	01/02/11	30	6
Testing & Debug (Tests approfondis, DIMACS)	21	0,7	23/04/11	30/04/11	6	6
Remise du programme	1	0,03	14/05/11	20/05/11	6	6
Rédaction du rapport final	26	0,86	15/05/11	21/05/11	6	6
Remise du rapport final	1	0,03	10/06/11	16/06/11	6	6
Préparation à la défense orale et ultimes corrections	18	0,6	11/06/10	17/06/10	6	0
Défense Orale	1	0,03	29/06/10	29/06/10	0	0

TABLE 4. Informations temporelles importantes sur les tâches.



### 5.5 Références et liens

- [1] Hertz, A., Plumettaz, M., Zufferey, N. *Variable space search for graph coloring*, Discrete Applied Mathematics 156 (2008) 2551-2560
- [2] DIMACS, <ftp://dimacs.rutgers.edu/pub/challenge/graph/>