



Manage Yourself

Rapport de planification

Projet de 4ème année informatique

Equipe :

Etienne Alibert,
Florian Barbedette,
Pierre Chesneau,
Mathias Deshayes,
Sevan Hartunians,
Mathieu Poinet.

Encadrant :

Laurence Rozé

Sommaire

Introduction.....	3
Présentation du projet	3
Méthode de planification.....	4
Diagramme de Gantt	5
Description des tâches	6
Interface administrateur	8
Mise à jour incrémentale des règles	8
Apprentissage sur rapport.....	8
Apprentissage sur exemple jouet.....	8
Génération simple de règles	8
Système expert.....	8
Base de règles.....	9
Test système expert	9
Test application téléphone.....	9
Application démon sur téléphone.....	9
Test génération de rapports.....	9
Communication et tests communication.....	9
Intégration et tests	9
Conclusion	10

Introduction

Dans le cadre du projet de quatrième année, il nous est demandé de rendre un dossier de planification initiale. Ce dossier présentera la planification du projet pour les mois à suivre jusqu'à la dernière soutenance. Il sera constitué, pour l'essentiel, d'une brève présentation du projet, ainsi que des explications nécessaires sur la méthode de planification utilisée. Il reportera également une première planification individuelle des grandes tâches à effectuer et contiendra notamment un premier diagramme de Gantt qui résume graphiquement les interactions et recouvrements temporels des tâches définies.

Cette phase de planification est primordiale dans la réalisation d'un projet, permettant de fixer les dates butoirs, de répartir les ressources en fonction des tâches et de garantir le respect des délais.

Présentation du projet

Manage Yourself est un projet de diagnostic et surveillance de plates-formes embarquées, il s'inscrit dans le cadre d'une collaboration entre Telelogos et DREAM. Telelogos est une entreprise éditant des logiciels professionnels qui automatisent, administrent et optimisent les processus d'échanges entre système informatique central et ceux des utilisateurs distants. DREAM est une équipe de recherche de l'IRISA spécialisée dans l'aide à la surveillance et au diagnostic de systèmes évoluant dans le temps.

Le besoin de développer une telle application émane de Telelogos, qui, dans le cadre d'un programme de formation aux technologies d'innovation cherche à développer deux logiciels concepts. Ces logiciels concepts ont pour but de tester la faisabilité de la mise en œuvre de technologies d'innovation et de créer un nouveau pôle d'activité autour de cette technologie si le résultat s'avère concluant. Manage Yourself est l'un de ses logiciels, ayant pour but d'explorer les possibilités dans le domaine du MBM (Mobile Device Management).

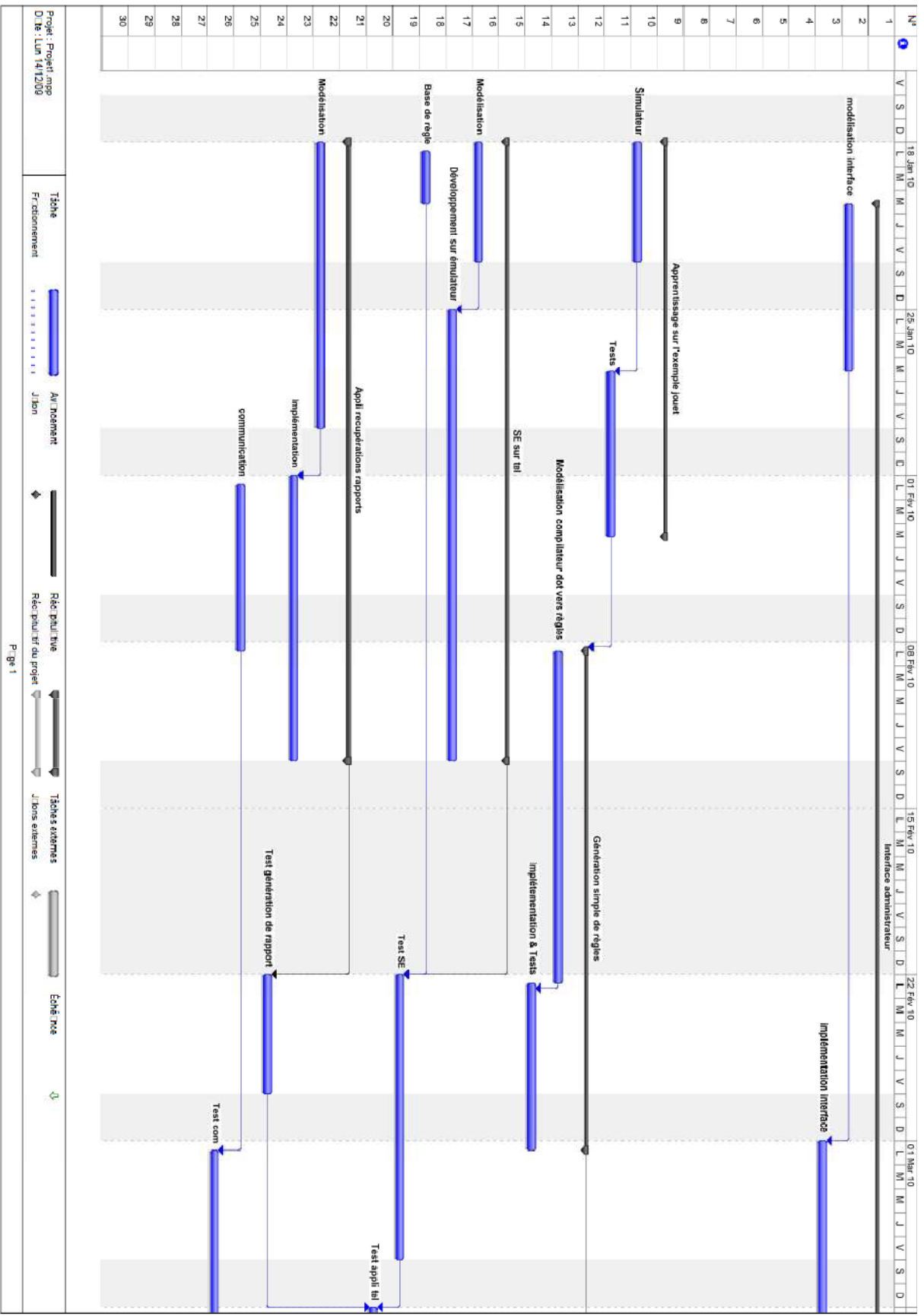
Méthode de planification

Afin de planifier le restant du projet, et ceci jusqu'à la fin de l'année, nous avons utilisé le logiciel vu en cours : Microsoft Project. MS Project est un logiciel de gestion de projet édité par Microsoft. Il permet de planifier les projets et les ressources, et d'assurer le suivi des projets pendant leur réalisation. Il permet au chef de projet d'assurer une gestion de projet professionnelle, cohérente, et ainsi de garantir le respect des délais.

Dans le cadre de notre projet, la planification sous MS Project s'avère quelques peu plus compliquée. En effet, les heures consacrées par chacun au projet sont variables selon les semaines et les jours. C'est pourquoi nous avons dû effectuer quelques ajustements tels qu'établir la durée par défaut en semaine, ou encore ne pas spécialiser chaque membre du groupe. Nous avons donc créé une ressource générale représentant l'ensemble du groupe et affecté ces ressources aux différentes tâches de la planification du projet.

Cette planification nous a donc permis d'établir un premier diagramme de Gantt ainsi qu'un tableau de tâches représentant notre planification initiale prenant en compte les périodes estimées des tâches à effectuer.

Diagramme de Gantt



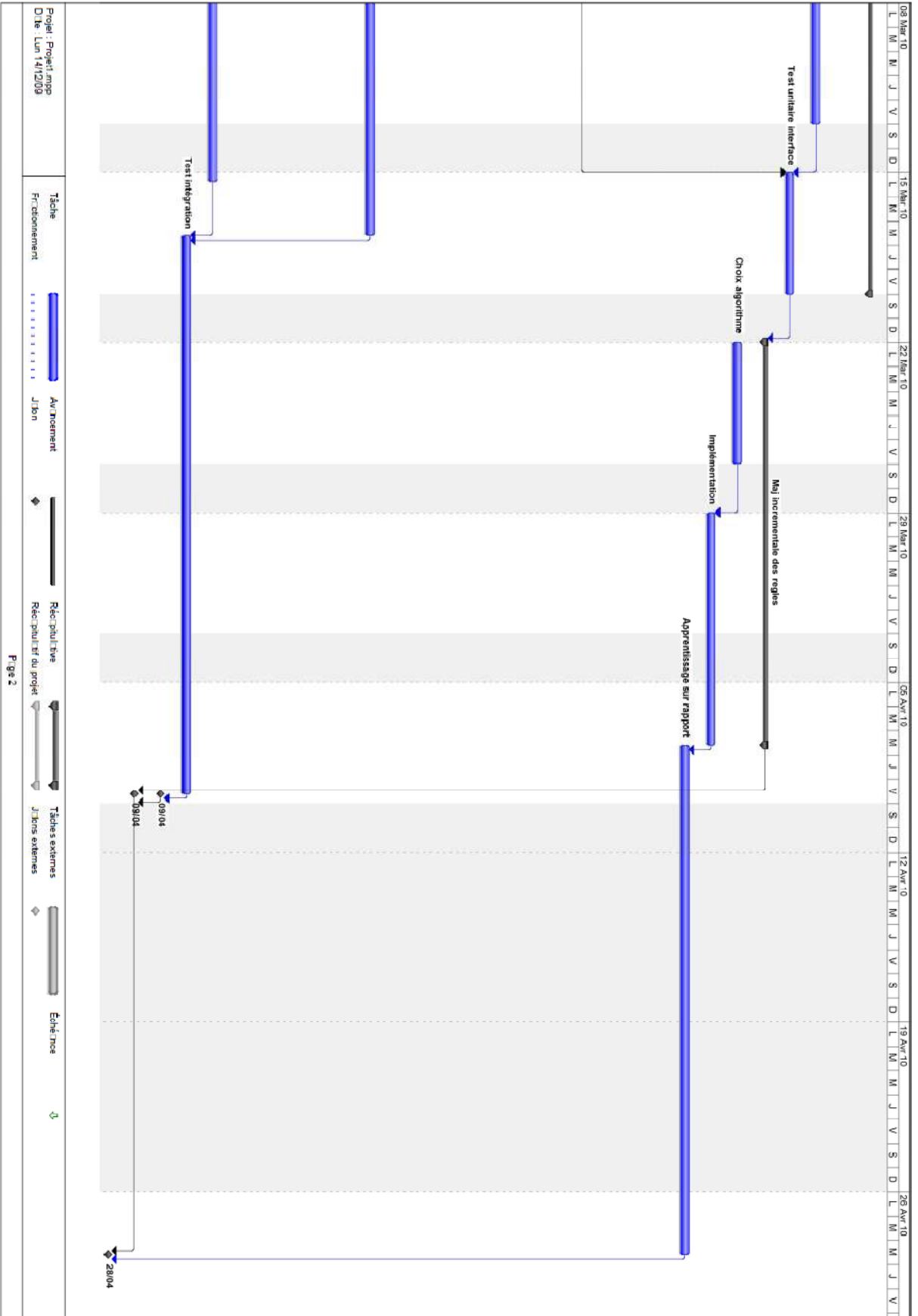


Figure 1 - Diagramme de Gantt

Tableau de tâches

La planification initiale décrite ci-avant nous a permis d'établir un tableau des tâches à effectuer avec les dates correspondantes et les ressources utilisées :

N°	Nom de la tâche	Durée	Début	Fin	Prédécesseurs	Noms ressources
1	Interface administrateur	7,5 sm	Mer 20/01/10	Ven 19/03/10		
2	modélisation interface	1 sm	Mer 20/01/10	Mer 27/01/10		developpeurs[200%]
3	implémentation interface	2 sm	Lun 01/03/10	Ven 12/03/10	2	developpeurs[200%]
4	Test unitaire interface	1 sm	Lun 15/03/10	Ven 19/03/10 12:3		developpeurs[200%]
5	Maj incrementale des regles	2,5 sm	Lun 22/03/10	Mer 07/04/10	4	
6	Choix algorithme	1 sm	Lun 22/03/10	Ven 26/03/10		Responsable
7	Implémentation	1,5 sm	Lun 29/03/10	Mer 07/04/10	6	developpeurs[200%];Responsable
8	Apprentissage sur rapport	1 sm	Mer 07/04/10	Mer 28/04/10	7	developpeurs[200%];Responsable
9	Apprentissage sur l'exemple jouet	2,5 sm	Lun 18/01/10	Mer 03/02/10		
10	Simulateur	1 sm	Lun 18/01/10	Ven 22/01/10		developpeurs[200%];Responsable
11	Tests	1 sm	Mer 27/01/10	Mer 03/02/10	10	developpeurs[200%];Responsable
12	Génération simple de règles	2 sm	Lun 08/02/10	Lun 01/03/10	11	
13	Modélisation compilateur dot vers règles	1 sm	Lun 08/02/10	Lun 22/02/10		developpeurs[200%]
14	Implémentation & Tests	1 sm	Lun 22/02/10	Lun 01/03/10	13	developpeurs[200%]
15	SE sur tel	4 sm	Lun 18/01/10	Ven 12/02/10		
16	Modélisation	1 sm	Lun 18/01/10	Ven 22/01/10		developpeurs[200%]
17	Développement sur émulateur	3 sm	Lun 25/01/10	Ven 12/02/10	16	developpeurs[200%]
18	Base de règle	0,5 sm	Lun 18/01/10	Mer 20/01/10		developpeurs[200%]
19	Test SE	2 sm	Lun 22/02/10	Ven 05/03/10 18:15		developpeurs[200%]
20	Test appli tel	1,5 sm	Lun 08/03/10	Mer 17/03/10 19:24		developpeurs[200%]
21	Appli recuperations rapports	4 sm	Lun 18/01/10	Ven 12/02/10		
22	Modélisation	2 sm	Lun 18/01/10	Ven 29/01/10		developpeurs[200%]
23	Implémentation	2 sm	Lun 01/02/10	Ven 12/02/10	22	developpeurs[200%]
24	Test génération de rapport	1 sm	Lun 22/02/10	Ven 26/02/10	21	developpeurs[200%]
25	communication	1 sm	Lun 01/02/10	Lun 08/02/10		developpeurs[200%]
26	Test com	2 sm	Lun 01/03/10	Lun 15/03/10	25	developpeurs[200%]
27	Test intégration	1,5 sm	Mer 17/03/10	Ven 09/04/10 26:20		developpeurs[400%]
28	Version 1	0 sm	Ven 09/04/10	Ven 09/04/10 27		
29	Version 2	0 sm	Ven 09/04/10	Ven 09/04/10 28:5		
30	Livraison	0 sm	Mer 28/04/10	Mer 28/04/10 29:8		

Figure 2 - Tableau de tâches

Description des tâches

Interface administrateur

Cette interface située sur le serveur permettra de visualiser, créer, supprimer, et de modifier les différentes règles présentes dans le système. L'administrateur aura la possibilité d'assigner l'action corrective à chaque règle. Côté planification, cette étape s'organisera suivant trois phases : la modélisation, l'implémentation et les tests unitaires de l'interface.

Mise à jour incrémentale des règles

La mise à jour incrémentale des règles est effectuée après l'apprentissage. Ses données sont

- les anciennes règles étiquetées,
- les nouvelles règles à étiqueter.

En sortie le module de mise à jour incrémentale doit retourner un ensemble de règles dont certaines sont déjà étiquetées et les autres le seront par l'interface administrateur.

Apprentissage sur rapport

L'apprentissage sur rapport consiste à relancer les algorithmes d'apprentissage mais sur les vrais rapports et non sur l'exemple jouet. L'idée est de garder le même algorithme de mise à jour incrémentale.

Apprentissage sur exemple jouet

L'apprentissage sur l'exemple jouet consiste à développer un simulateur de notre exemple jouet pour générer des paquets d'exemples étiquetés. Le premier paquet sera utilisé pour générer l'ensemble initial de règles. Puis les suivants seront utilisés pour mettre à jour les règles grâce à l'algorithme de mise à jour incrémentale.

Génération simple de règles

La génération simple des règles est un simple compilateur qui permet de transformer les règles sorties de l'apprentissage (format dot) à notre format de règles. Cette étape de génération de règles comprendra donc deux phases : la modélisation du compilateur Dot vers notre format de règle, et la phase d'implémentation et test.

Système expert

Cette étape consistera à mettre au point le moteur d'inférence du logiciel. Il prendra en entrée la base de règles obtenue ainsi que les rapports d'état du téléphone et indiquera les actions

correctives à effectuer. Cette étape sera constituée de deux phases : la modélisation, puis le développement avec tests sur émulateur.

Base de règles

Nous allons créer manuellement une base de règles sur laquelle s'appuieront les tests du système expert.

Test système expert

Cette phase correspondra aux tests du système expert sur ordinateur à partir de la base de règle créée précédemment.

Test application téléphone

Cette étape correspondra aux tests de toute la partie embarquée sans la partie communication : intégration du système expert avec la génération de rapports et la base de règles.

Application démon sur téléphone

Cette application devra tourner en fond sur le téléphone, s'occuper de la récupération des rapports d'état, interroger le système expert, et effectuer les actions correctives en conséquence. Modélisation et implémentation constitueront cette étape.

Test génération de rapports

Cette phase consistera à vérifier que la génération des rapports d'état sur le téléphone est correctement effectuée.

Communication et tests communication

La partie communication du projet sera effectuée avec l'outil MediaContact de l'entreprise Télélogos. Nous devons donc configurer cet outil pour s'adapter à nos besoins.

Intégration et tests

Création d'un fichier .cab pour déployer le logiciel sur le téléphone mobile. Et enfin tests complets sur mobile avec toutes les nouvelles contraintes non testables sur émulateur.

Conclusion

Le présent rapport nous a permis d'établir une première planification de notre projet. Cependant, la liste des tâches n'est pas exhaustive. En effet, certaines de ces tâches pourront être amenées à être décomposées en d'autres sous-tâches. De même, l'aspect temporel reste une première évaluation du fait de notre faible expérience en termes de gestion de projet.

Ce rapport nous a donc montré l'importance d'une planification dans la démarche d'un projet : ordonnancement des tâches, évaluation de la durée de chaque tâche, et des ressources nécessaires à celles-ci. Cette planification a été réalisée avec l'outil Microsoft Project. Ce logiciel permet une gestion efficace et complète des projets et garantit leur bon déroulement futur.

Nous allons donc essayer de nous tenir à cette planification dans la réalisation du projet jusqu'à sa livraison.