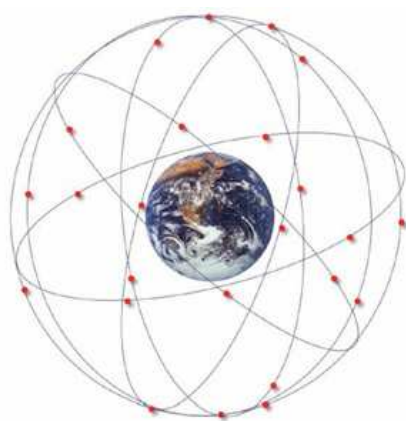


Florian Bastien
Fabien Cornic
Antoine Després
François Droumaguet
Bastien Przybylski



Rapport de planification initiale du projet NAVinc



Jean-Louis Pazat
Nikolaos Parlavantzas

SOMMAIRE

Sommaire	2
Chapitre 1 MÉTHODE DE PLANIFICATION	3
1.1) Rappel du besoin	3
1.2) Présentation de la méthode de planification	3
Chapitre 2 DOCUMENTS DE PLANIFICATION	5
2.1) Calendrier et temps de travail	5
2.2) Ordonnancement des tâches	6
2.3) Diagramme de Gantt.....	8
2.4) Utilisation des ressources	10
2.5) Analyse des risques	10

CHAPITRE 1 MÉTHODE DE PLANIFICATION

1.1) Rappel du besoin

Nous devons développer un logiciel de navigation de type « GPS » en utilisant une architecture orientée services. Nous devons pour cela développer les services permettant la réalisation des différents cas d'utilisation.

Plus particulièrement, le logiciel doit être capable de s'adapter en fonction du contexte. Pour cela, il nous est demandé de mettre en place une gestion des services basée sur l'utilisation du Framework OSGi.

Les technologies retenues pour la réalisation de ce logiciel sont le Framework OSGi, Apache iPOJO et Eclipse.

1.2) Présentation de la méthode de planification

Pour planifier le développement du logiciel de navigation nous avons suivi une méthode de planification classique, décomposée de la manière suivante :

Définir le calendrier de travail

Il s'agit d'abord de définir les limites dans le temps du projet avec la date de début et la date de fin. Il faut aussi définir les périodes chômées. Nous obtenons ainsi les jours de travail disponibles.

Déterminer les ressources

Il s'agit de déterminer les personnes disponibles qui peuvent travailler sur le projet. Dans notre cas, cette étape était évidente.

Déterminer les tâches à réaliser et les ordonner

Il s'agit de décomposer le développement du logiciel en tâches. Nous obtenons ainsi l'ensemble des tâches à réaliser. Il faut ensuite les ordonner en fonction de leur priorité et des dépendances qu'il y a entre elles.

Déduire le diagramme de Gantt

Il s'agit de construire le diagramme de Gantt correspondant à l'ordonnancement réalisé précédemment.

Déterminer les risques

Il s'agit d'effectuer une analyse des risques pouvant compromettre le déroulement du développement du logiciel. Cette analyse permet de mettre en évidence les options de secours en cas de problème.

CHAPITRE 2 DOCUMENTS DE PLANIFICATION

2.1) Calendrier et temps de travail

Le projet est délimité par les dates ci-dessous :

Date de début : Lundi 04 janvier 2010

Date de fin : Vendredi 28 mai 2010

Pour tenir compte des périodes chômées nous avons redéfini le calendrier standard.

Périodes chômées

Période de DS semestre 7 : du 11/01/2010 au 15/01/2010

Salon du CEBIT : du 01/03/2010 au 05/03/2010

Période de DS semestre 8 : du 05/05/2010 au 14/05/2010

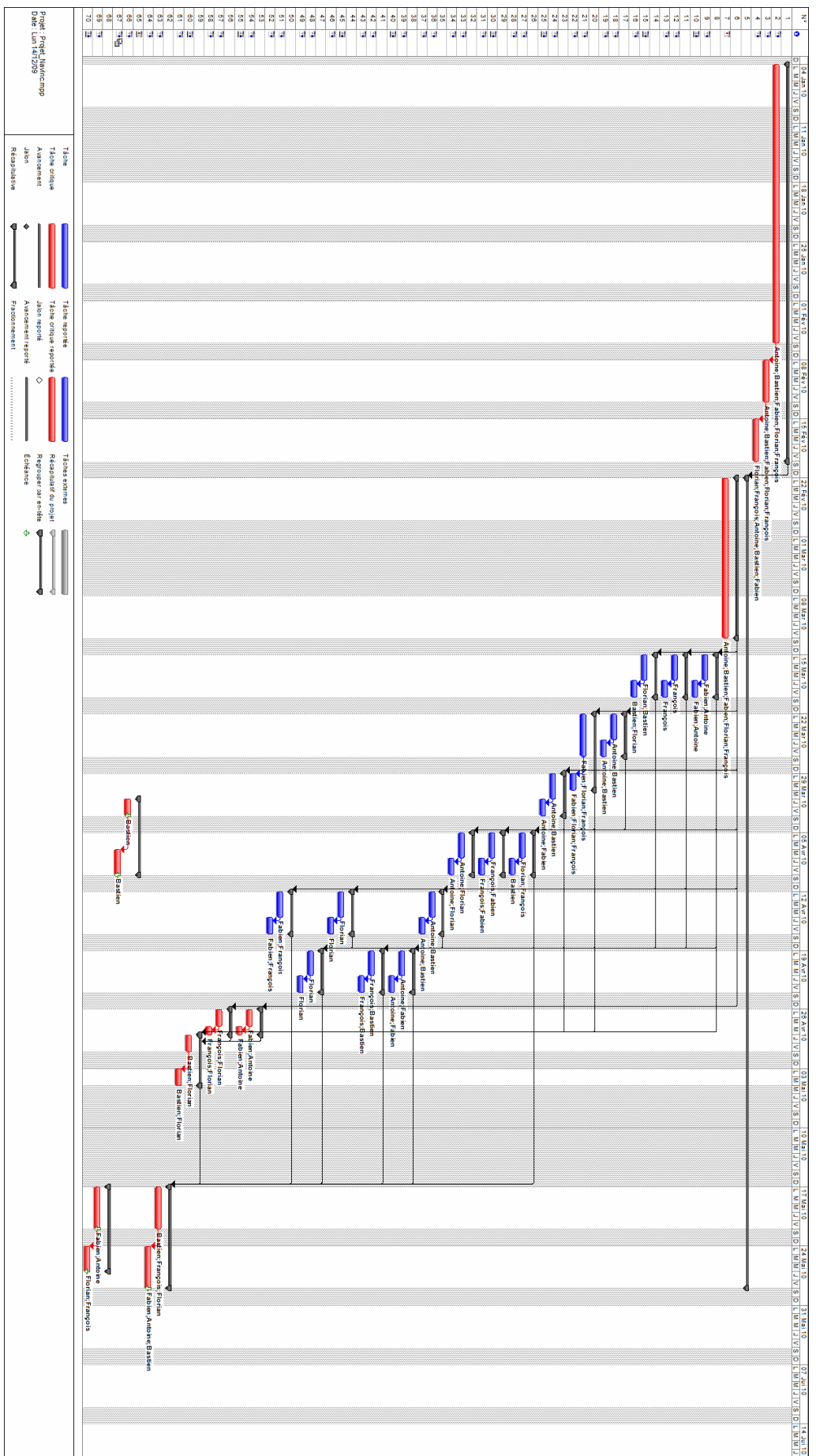
La semaine de travail commence le lundi et se termine le vendredi. Le temps de travail a été prévu tous les jours de 19h à 21h soit une durée de travail de 10 heures par semaine et par ressource. Nous avons compté 20 jours par mois, ce qui fait un total de 40 heures de travail par mois et par ressource.

2.2) Ordonnancement des tâches

N°	Nom de la tâche	Durée	Début	Fin	Prédécesseurs	Noms ressources
1	Conception logicielle	30 jours	Lun 04/01/10	Ven 19/02/10		
2	Conception détaillée (UML)	20 jours	Lun 04/01/10	Ven 05/02/10		Antoine,Bastien,Fabien,Florian,François
3	Rapport de conception (version préliminaire)	5 jours	Lun 08/02/10	Ven 12/02/10 2		Antoine,Bastien,Fabien,Florian,François
4	Rapport de conception (version finale)	5 jours	Lun 15/02/10	Ven 19/02/10 3		Florian,François,Antoine,Bastien,Fabien
5	Développement logiciel	67 jours	Lun 22/02/10	Ven 28/05/10 1		
6	Préparation	10 jours	Lun 22/02/10	Ven 12/03/10		
7	Préparation du développement (formation)	10 jours	Lun 22/02/10	Ven 12/03/10		Antoine,Bastien,Fabien,Florian,François
8	Développement de la base du service Navhic (IHM)	5 jours	Lun 15/03/10	Ven 19/03/10 6		
9	Développement du bundle	3 jours	Lun 15/03/10	Mer 17/03/10		Fabien,Antoine
10	Tests unitaires	2 jours	Jeu 18/03/10	Ven 19/03/10 9		Fabien,Antoine
11	Développement du service de gestion des cartes	5 jours	Lun 15/03/10	Ven 19/03/10 6		
12	Développement du bundle	3 jours	Lun 15/03/10	Mer 17/03/10		François
13	Tests unitaires	2 jours	Jeu 18/03/10	Ven 19/03/10 12		François
14	Développement du service de cartographie	5 jours	Lun 15/03/10	Ven 19/03/10 6		
15	Développement du bundle	3 jours	Lun 15/03/10	Mer 17/03/10		Florian,Bastien
16	Tests unitaires	2 jours	Jeu 18/03/10	Ven 19/03/10 15		Bastien,Florian
17	Développement de la base du service de géolocalisation (hors scénario)	5 jours	Lun 22/03/10	Ven 26/03/10 6		
18	Développement du bundle	3 jours	Lun 22/03/10	Mer 24/03/10		Antoine,Bastien
19	Tests unitaires	2 jours	Jeu 25/03/10	Ven 26/03/10 18		Antoine,Bastien
20	Développement du gestionnaire de services	7 jours	Lun 22/03/10	Mar 30/03/10 6		
21	Développement du bundle	5 jours	Lun 22/03/10	Ven 26/03/10		Fabien,Florian,François
22	Tests unitaires	2 jours	Lun 29/03/10	Mar 30/03/10 21		Fabien,Florian,François
23	Développement de l'interface de suivi des scénarii	5 jours	Lun 29/03/10	Ven 02/04/10 6		
24	Développement du bundle	3 jours	Lun 29/03/10	Mer 31/03/10		Antoine,Bastien
25	Tests unitaires	2 jours	Jeu 01/04/10	Ven 02/04/10 24		Antoine,Fabien
26	Intégration des services pour les scenarii de géolocalisation	5 jours	Lun 05/04/10	Ven 09/04/10 17:20:23:8		
27	Intégration des services	3 jours	Lun 05/04/10	Mer 07/04/10		Florian,François
28	Tests d'intégration	2 jours	Jeu 08/04/10	Ven 09/04/10 27		Bastien
29	Développement du service de localisation	5 jours	Lun 05/04/10	Ven 09/04/10 6		
30	Développement du bundle	3 jours	Lun 05/04/10	Mer 07/04/10		François,Fabien
31	Tests unitaires	2 jours	Jeu 08/04/10	Ven 09/04/10 30		François,Fabien
32	Développement du service de routage	5 jours	Lun 05/04/10	Ven 09/04/10 6		
33	Développement du bundle	3 jours	Lun 05/04/10	Mer 07/04/10		Antoine,Florian
34	Tests unitaires	2 jours	Jeu 08/04/10	Ven 09/04/10 33		Antoine,Florian
35	Développement du service de navigation	5 jours	Lun 12/04/10	Ven 16/04/10 6		

N°	Nom de la tâche	Durée	Début	Fin	Prédécesseurs	Noms ressources
36	Développement du bundle	3 jours	Lun 12/04/10	Mer 14/04/10		Antoine,Bastien
37	Tests unitaires	2 jours	Jeu 15/04/10	Ven 16/04/10	36	Antoine,Bastien
38	Intégration des services pour les scenarii de navigation (cas du tunnel)	5 jours	Lun 19/04/10	Ven 23/04/10	35;20;26;23;8	
39	Intégration des services	3 jours	Lun 19/04/10	Mer 21/04/10		Antoine,Fabien
40	Tests d'intégration	2 jours	Jeu 22/04/10	Ven 23/04/10	39	Antoine,Fabien
41	Intégration des services pour les scenarii de gestion des cartes	5 jours	Lun 19/04/10	Ven 23/04/10	11;14;20;23;8	
42	Intégration des services	3 jours	Lun 19/04/10	Mer 21/04/10		François,Bastien
43	Tests d'intégration	2 jours	Jeu 22/04/10	Ven 23/04/10	42	François,Bastien
44	Développement de la base du service d'info-traffic (hors scénario)	5 jours	Lun 12/04/10	Ven 16/04/10	6	
45	Développement du bundle	3 jours	Lun 12/04/10	Mer 14/04/10		Florian
46	Tests unitaires	2 jours	Jeu 15/04/10	Ven 16/04/10	45	Florian
47	Intégration des services pour les scenarii d'info-traffic	5 jours	Lun 19/04/10	Ven 23/04/10	44;20;23;8	
48	Intégration des services	3 jours	Lun 19/04/10	Mer 21/04/10		Florian
49	Tests d'intégration	2 jours	Jeu 22/04/10	Ven 23/04/10	48	Florian
50	Développement du service de lieux d'intérêt	5 jours	Lun 12/04/10	Ven 16/04/10	14;11	
51	Développement du bundle	3 jours	Lun 12/04/10	Mer 14/04/10		Fabien,François
52	Tests unitaires	2 jours	Jeu 15/04/10	Ven 16/04/10	51	Fabien,François
53	Développement du service de monitoring voiture (hors scénario)	3 jours	Lun 26/04/10	Mer 28/04/10	6	
54	Développement du bundle	2 jours	Lun 26/04/10	Mar 27/04/10		Fabien,Antoine
55	Tests unitaires	1 jour	Mer 28/04/10	Mer 28/04/10	54	Fabien,Antoine
56	Développement du service de monitoring système (hors scénario)	3 jours	Lun 26/04/10	Mer 28/04/10	6	
57	Développement du bundle	2 jours	Lun 26/04/10	Mar 27/04/10		François,Florian
58	Tests unitaires	1 jour	Mer 28/04/10	Mer 28/04/10	57	François,Florian
59	Intégration des services pour les scenarii de monitoring	4 jours	Jeu 29/04/10	Mar 04/05/10	53;56;20;23;8	
60	Intégration des services	2 jours	Jeu 29/04/10	Ven 30/04/10		Bastien,Florian
61	Tests d'intégration	2 jours	Lun 03/05/10	Mar 04/05/10	60	Bastien,Florian
62	Tests et validation	10 jours	Lun 17/05/10	Ven 28/05/10	59;47;41;38;26;50	
63	Test du logiciel fini	5 jours	Lun 17/05/10	Ven 21/05/10		Bastien,François,Florian
64	Validation du logiciel fini	5 jours	Lun 24/05/10	Ven 28/05/10	63	Fabien,Antoine,Bastien
65	Page HTML	7 jours	Jeu 01/04/10	Ven 09/04/10		
66	Page HTML (version préliminaire)	2 jours	Jeu 01/04/10	Ven 02/04/10		Bastien
67	Page HTML (version finale)	3 jours	Mer 07/04/10	Ven 09/04/10	66	Bastien
68	Rapport final	8 jours	Lun 17/05/10	Mer 26/05/10		
69	version préliminaire	5 jours	Lun 17/05/10	Ven 21/05/10		Fabien,Antoine
70	version finale	3 jours	Lun 24/05/10	Mer 28/05/10	69	Florian,François

2.3) Diagramme de Gantt



Quelques explications sur le Gantt

- Premier constat, il n'y a pas de chemin critique affiché par MS Project et ce pour la raison suivante. La majorité des services peut être développée dans n'importe quel ordre. Ainsi, il n'y a pas de dépendance directe entre les tâches de développement des différents services effectuées successivement. En revanche, les tâches d'intégration nécessitent qu'il y ait un ou plusieurs développements de services terminés, ce qui implique une dépendance entre l'intégration et les services nécessaires. Cependant, cela ne suffit pas à MS Project pour calculer le chemin critique. En effet, du point de vue du diagramme de Gantt, le développement et l'intégration des services peuvent toujours être effectués en parallèle puisqu'il ne prend pas en compte la disponibilité des ressources. De plus, chaque groupe de tâches parallèle dispose d'une marge libre non nulle car sa durée estimée est inférieure à la période entre la date de début au plus tôt (fin de la formation) et la date de fin au plus tard (début des tests et de la validation).
- Bien évidemment, dans le cas de notre projet, nous avons un nombre de ressources limité, c'est-à-dire les membres de notre groupe. Par conséquent, nous devons appliquer des règles de priorité afin de gérer l'utilisation concurrentielle des ressources disponibles.
Nous avons dans un premier temps utilisé l'outil d'audit des ressources de MS Project pour réorganiser les tâches en tenant compte de la disponibilité des ressources. Cependant, le résultat obtenu était totalement décousu et irréalisable. Nous n'avons donc pas jugé utile de présenter cette version.
Nous avons alors utilisé des règles de priorité heuristiques, qui nous ont permis d'obtenir le diagramme de Gantt présenté ci-avant. Pour cela, nous avons ajouté manuellement des contraintes de début sur les tâches. Nous obtenons donc un diagramme réaliste du déroulement du projet.
- Compte tenu du fait qu'il n'y ait pas de chemin critique affiché, la gestion des retards sera plus délicate mais néanmoins faisable avec l'outil de suivi Microsoft Project. De plus, l'analyse des risques qui suit permet de mettre en évidence les tâches qui risquent de retarder le projet et qui seront donc à suivre de manière plus rigoureuse.

2.4) Utilisation des ressources

Le tableau ci-dessous présente les ressources disponibles ainsi que la nombre total d'heures travaillées par chaque ressource sur le projet. Nous avons essayé de faire une répartition homogène afin qu'il n'y ait pas de gros écart entre les différentes ressources.

Nom de la ressource	Travail
Antoine	166 h
François	162 h
Fabien	164 h
Florian	170 h
Bastien	168 h

2.5) Analyse des risques

Un projet ne se déroulant jamais exactement comme il avait été planifié, il est important de prévoir dès la planification les risques encourus pendant son déroulement. Cela permet de connaître les retards les plus probables dans l'avancement des tâches. Ainsi, il est possible de les anticiper et de pouvoir réagir au plus tôt pour s'y adapter et limiter l'effet boule de neige.

Le premier risque auquel nous serons confrontés lors de la phase de développement du projet concerne notre formation sur la technologie utilisée (c'est-à-dire OSGi). En effet, malgré les recherches effectuées jusqu'à présent pour prendre en main OSGi, de nombreuses interrogations demeurent et devront être levées avant de commencer le développement. Cette formation a été prise en compte dans la planification, mais sa durée reste difficile à connaître, et pourrait donc s'étaler plus tard que prévu, ce qui aurait pour effet de retarder tout le projet puisque c'est une tâche critique. Dans ce cas, il serait préférable de commencer tout de même le développement des services, mais en gardant une ou deux personnes en formation qui pourront ensuite apporter aux autres les connaissances supplémentaires acquises.

L'autre risque majeur concerne la réutilisation de Travelling Salesman. En effet, afin de faciliter le développement du projet compte tenu de temps relativement court dont nous disposons, nous avons fait le choix de réutiliser le code des *plugins* de Travelling Salesman pour développer nos services. Cependant, nous n'avons encore aucune certitude concernant la facilité de sa réutilisation, et s'il s'avère que le code est difficilement adaptable à notre

projet, le développement d'un certain nombre de nos services serait retardé. Si cela se produit, il faudra décider d'abandonner certains services et donc un ou plusieurs *scenarii*. Il faut tout de même garder à l'esprit que l'application délivrée doit montrer l'adaptabilité des services, ce qui exclut de faire l'impasse sur des *scenarii* tel que le passage sous un tunnel. Il ne sera pas possible d'écarter le service de cartographie par exemple, sans lequel l'application ne serait pas capable d'offrir une interface graphique et réduirait grandement l'intérêt du projet.

Enfin, le dernier point sensible est le gestionnaire de services, puisque c'est le cœur du projet, qui doit démontrer l'adaptabilité des services. Même si son développement dure plus longtemps que prévu (notamment à cause de difficultés techniques ou algorithmiques), il n'est pas envisageable de le limiter ou de l'arrêter. Il faut absolument qu'il soit mis en œuvre, et pour cela, en cas de retard, il faudrait augmenter le nombre de personnes travaillant dessus et/ou prévoir de ne pas implémenter les services ou *scenarii* les moins utiles.