



INSA RENNES

QUATRIÈME ANNÉE - INFORMATIQUE

Projet 4ème année INFO

Rapport de planification

Réalisé par :

NOUR ROMDHANE
LIANTSOA RASATA
MATHILDE LEPARQUIER
IBRAHIM BENALI
OTHMANE KABIR

Sous la direction de :

JEAN-LOUIS PAZAT

Table des matières

1	Introduction	2
2	Rappel du contexte	3
2.1	Problématique	3
2.2	Acteurs	3
2.3	Périmètre fonctionnel	3
2.4	Eléments en entrée	3
2.5	Périmètre de qualification	3
2.6	Calendrier	4
2.7	Analyse des risques	4
3	Organisation	6
3.1	Description du cycle de production	6
3.2	Description du cycle de qualification	6
3.3	Description de l'organisation	7
3.4	Description du mode de pilotage	7
3.5	Approche d'estimation	7
4	Planification à partir de Project	8
4.1	Hierarchie des tâches	8
4.2	Affectation des ressources par tâches	9
4.3	Planning	11
5	Conclusion	13

1 Introduction

Dans le cadre de notre deuxième année d'étude au département Informatique de l'INSA de Rennes, nous allons réaliser un projet technique en équipe. Le sujet de notre projet porte sur les mécanismes de synchronisation avancée en Java.

Ce rapport présente une planification initiale de notre projet. Dans un premier temps, nous rappelons le contexte de notre projet. Dans un deuxième temps, nous présentons les méthodes utilisées pour la gestion de projet suite à laquelle nous proposerons une estimation des tâches ainsi que l'affectation des ressources. Dans un dernier temps, nous présentons les résultats de notre planification à partir de Microsoft Project.

2 Rappel du contexte

2.1 Problématique

L'apprentissage des outils de synchronisation est souvent fastidieuse pour les programmeurs. Comment pouvons nous faciliter l'utilisation des outils de parallélisme ?

2.2 Acteurs

Nous pouvons distinguer deux acteurs pour notre projet.

Nous avons d'un côté le client : la problématique nous a été posé par Jean-Louis Pazat, enseignant-chercheur au département informatique, qui est donc le demandeur "client" mais aussi notre encadrant de projet.

Et de l'autre côté, nous avons l'équipe projet : notre équipe est composé de cinq étudiants en quatrième année en informatique. Pour la réalisation du projet, nous nous mettons tous dans le rôle de développeur.

2.3 Périmètre fonctionnel

Le but de notre projet est de traduire automatiquement un langage source Java avec des conditions et des compteurs vers un langage Java utilisant la synchronisation. Ainsi, pour la réalisation de ce projet, il est nécessaire de maîtriser les notions de parallélisme, le langage de programmation Java et l'environnement de développement Eclipse. De plus lors de la phase d'analyse, nous avons identifié la compilation comme l'une des solutions techniques pour résoudre la problématique. De ce fait, une connaissance des techniques de compilation est souhaitable.

2.4 Eléments en entrée

L'utilisation des compteurs de synchronisation pour faire du parallélisme n'est pas très répandue. Pendant la phase d'analyse du projet, nous n'avons pas trouvé beaucoup d'articles sur Internet pouvant nous orienter sur leur fonctionnement. Donc nous n'avons pas pu nous appuyer sur beaucoup de documents de recherches pour analyser le problème. Ainsi, nous avons du définir le mode de fonctionnement et de traduction de nos compteurs par l'établissement d'exemples de traduction à générer afin d'en déduire des schémas types.

2.5 Périmètre de qualification

Pour valider le développement, des tests unitaires devront être effectués et l'intégration de chaque différent module au reste du projet devra être tester.

Le résultat du projet sera utiliser dans les TP de programmation parallèle pour les étudiants de quatrième année au département informatique et sera distribué plus largement.

2.6 Calendrier

Au début du projet, un calendrier nous a été fourni. Ce dernier définissait les différentes phases du projet ainsi que toutes les dates d'échéance pour chaque livrable.

Voici une frise illustrant ce calendrier :

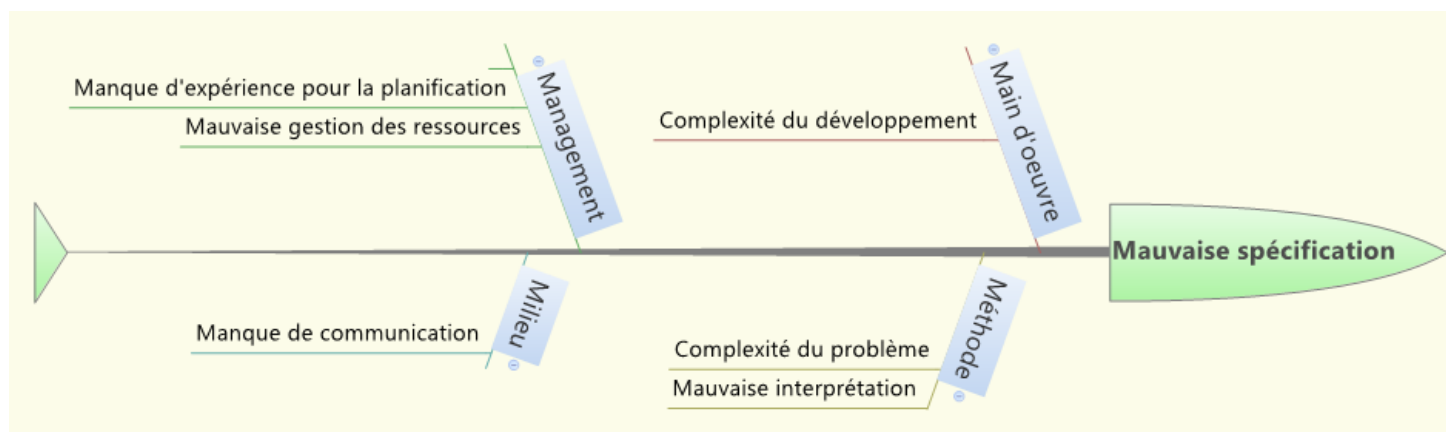


2.7 Analyse des risques

Dès le départ, la problématique était pour nous difficile à cerner. En effet, le projet requiert la maîtrise du parallélisme, notion que nous n'avions pas au début du projet. Il nous a donc fallu beaucoup de temps pour acquérir les connaissances nécessaires pour ensuite pouvoir comprendre la problématique et proposer une solution au problème.

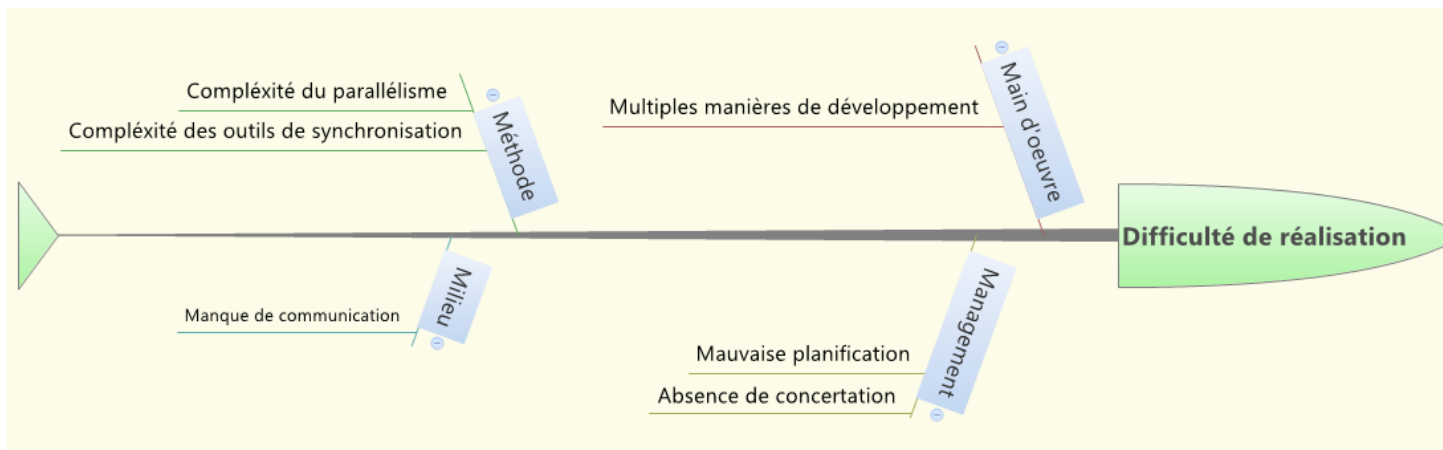
En conséquence, un risque que nous pouvons identifier ici est une mauvaise spécification.

Le diagramme d'Ishikawa suivant illustre cette analyse :



Entre autre, nous pouvons aussi identifier la difficulté de réalisation comme un risque car le périmètre fonctionnel de notre projet est composé de notions assez complexes. De plus, pendant la phase de développement, chaque membre de l'équipe peut avoir différentes manières de réaliser le projet non compatible avec le reste du groupe. Ainsi, il est nécessaire de se concerter et de communiquer régulièrement pour ne pas diverger.

Le diagramme d'Ishikawa suivant illustre cette analyse :



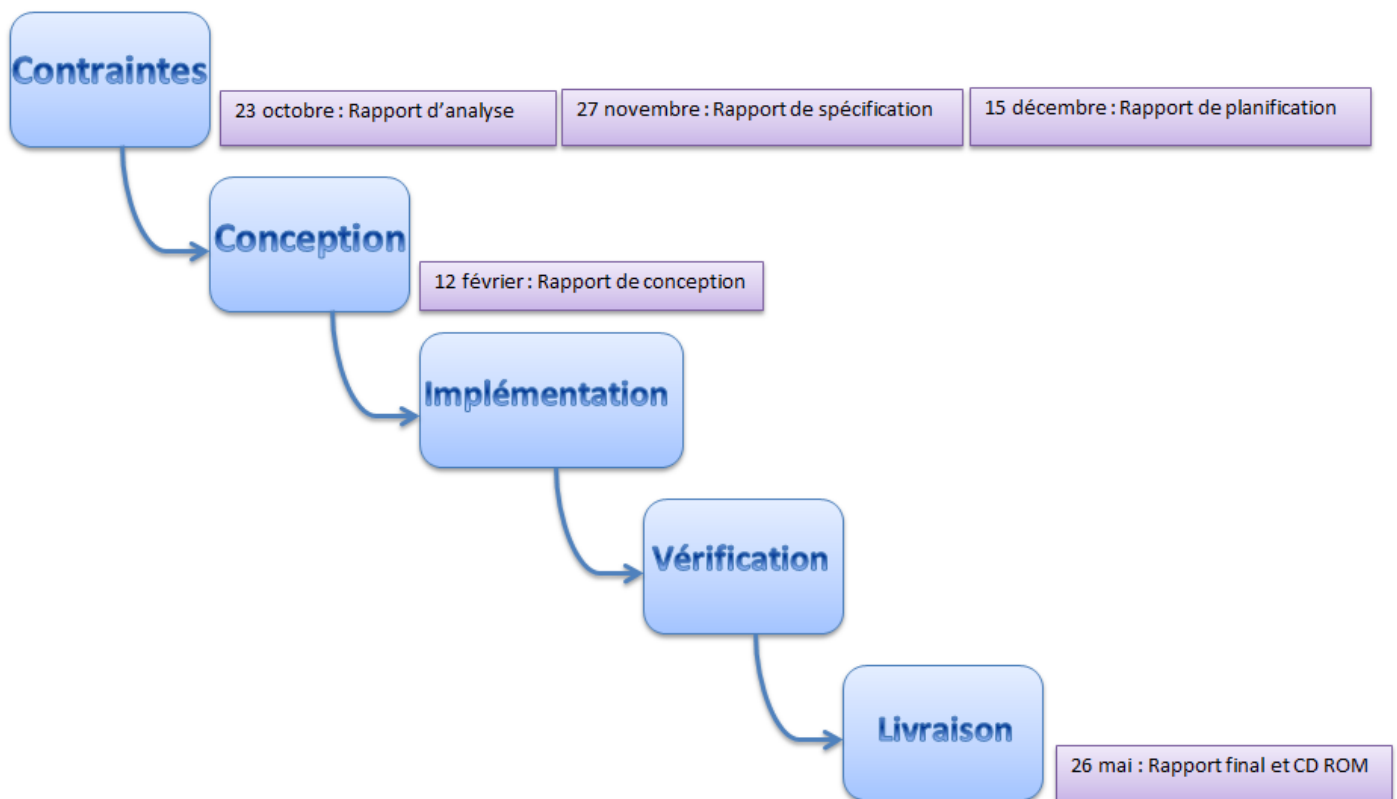
Enfin, un dernier risque serait la difficulté de déploiement. En effet, la solution que nous développerons sera déployer dans les salles de Tps pour les travaux pratiques de programmation parallèle de quatrième année et il est difficile pour nous de prévoir à ce stade les problèmes que nous pourrions rencontrer ainsi que leur source.

3 Organisation

3.1 Description du cycle de production

Nous avons choisi le modèle en cascade comme cycle de développement. Dans ce modèle, les phases sont effectuées simplement les unes après les autres, avec éventuellement un retour sur les précédentes étapes. Ce modèle est pour nous le mieux adapté au développement de notre projet car elle pourra nous permettre d'identifier les phases qui auront pour but de produire à une date précise des livrables définis au préalable et de ne se terminer que lorsque les livrables seront jugés satisfaisants lors d'une étape de validation-vérification.

La figure suivante montre les différentes phases de notre projet ainsi que les jalons et les productions :



3.2 Description du cycle de qualification

Afin d'assurer que le produit livré satisfait les exigences requises, nous choisissons de faire de l'intégration continue.

Pour appliquer cette technique, nous allons partager le code source en utilisant le logiciel de gestion de version Git. Chaque membre de l'équipe devra intégrer quotidiennement ses modifications et des unitaires notamment avec JUnit et des tests d'intégration devront être développés pour valider toute évolution du code.

3.3 Description de l'organisation

Avec la mise en place d'un dépôt du projet en ligne, chaque membre de l'équipe pourra développer différents modules constituant l'application et pourra voir l'évolution du projet.

Une réunion hebdomadaire est mise en place pour faire un bilan de l'avancement, pour pouvoir identifier les futurs tâches à affecter et aussi pour résoudre les difficultés éventuelles rencontrées.

3.4 Description du mode de pilotage

Le timeboxing est la technique de planification qui nous paraît le plus approprié pour le projet. Un calendrier fournit dès le départ définit un certain nombre de périodes de production avec des échéances fixées.

Ainsi, comme le délai de production est impérativement fixé, dans le cas où la limite de temps est atteinte, nous serions éventuellement amené à retirer des fonctions de l'objectif final ou de nous focaliser sur un objectif prioritaire puis d'ajouter ultérieurement des fonctions.

3.5 Approche d'estimation

Dans le cadre de notre projet, pour déduire les tâches et les charges, nous utilisons l'approche top-down qui nous permettra de délimiter et de conceptualiser rapidement le projet et de le diviser en sous-parties aisément manipulables.

Au début du projet, un calendrier ainsi qu'un cahier des charges nous a été fourni. Dans ce dernier a été défini le but final : implémenter les compteurs de synchronisation, et les moyens techniques à utiliser : Java et Eclipse.

Dans un premier temps, il nous a donc fallu définir de manière plus détaillé à partir de cas d'utilisations avec notre encadrant qui est aussi notre client le résultat final attendu.

Dans un deuxième temps, il nous a fallu identifier les solutions techniques possibles à la problématique. A l'issue duquel nous avons identifier la technique de la compilation et l'utilisation de l'annotation comme solutions adéquates. Et c'est à partir de ces techniques que nous pouvons concevoir l'architecture de notre projet et d'identifier les tâches.

4 Planification à partir de Project

Dans ce chapitre, nous allons présenter les résultats de notre planification à partir de Microsoft Project.

La hiérarchie des tâches, l'affectation des ressources, les jalons et la répartition des ressources ont été défini de manière à respecter au mieux les différents dates d'échéance imposés par le cahier des charges.

4.1 Hiérarchie des tâches

Pour les phases principales du projet, nous commençons par une analyse de la problématique pour pouvoir d'abord cerner le périmètre fonctionnel et ensuite pour pouvoir proposer une solution technique adéquate qui définira les tâches ainsi que les tests nécessaires pour vérifier et valider le développement.

La figure suivante nous montre le découpage du projet, nous y retrouvons les phases principales que nous avons décrit dans la description du cycle de production :







	 Mode Tâche	Nom de la tâche	Durée	Début	Fin	Prédécesseurs
0		▸ syncj	138 jours	Ven 12/09/14	Ven 29/05/15	
1		▸ Analyse du projet	45,25 jours	Ven 12/09/14	Ven 05/12/14	
26		▸ Conception	44,13 jours	Ven 05/12/14	Jeu 12/02/15	1
42		▸ Construction	26,25 jours	Ven 13/02/15	Jeu 02/04/15	26
51		▸ Livraison	20,38 jours	Ven 03/04/15	Mar 26/05/15	42
57		▸ Démonstrations	2,75 jours	Mer 27/05/15	Ven 29/05/15	51

FIGURE 4.1 – Principales tâches

Ensuite, chaque phase principale a été divisé en sous-tâches selon les besoins du projet. Notamment durant la phase d'analyse, il a été nécessaire d'acquérir les notions de parallélisme ainsi que la maîtrise des outils de synchronisations Java pour ensuite pourvoir comprendre le problème qui nous a été posé. Ensuite, lors de la phase de spécification, il a été nécessaire de rechercher les solutions techniques relatives dont nous avons pu en identifier trois. De ce fait, chacune de ces trois solutions devaient être analyser pour pouvoir décider laquelle nous allons mettre en oeuvre lors de la phase de construction.

La figure suivante nous montre les différents tâches à réaliser pour la phase d'analyse et de spécification du projet :






























		Mode Tâche ▾	Nom de la tâche ▾	Durée ▾	Début ▾	Fin ▾	Prédécesseurs ▾	Noms ressources ▾
1			▸ Analyse du projet	45,25 jours	Ven 12/09/14	Ven 05/12/14		
2			▸ Définition du projet	7,88 jours	Ven 12/09/14	Mer 24/09/14		
3			Définition du parallélisme	3,5 jours	Ven 12/09/14	Mer 17/09/14		Générique[200%]
4			Définition des outils de synchronisation	3,5 jours	Ven 12/09/14	Mer 17/09/14		Générique[200%]
5			Recherches sur les machines parallèles	3,5 jours	Ven 12/09/14	Mer 17/09/14		Générique
6			Problèmes de synchronisation	4,38 jours	Jeu 18/09/14	Mer 24/09/14		Générique[200%]
7			Outils de synchronisation	4,38 jours	Jeu 18/09/14	Mer 24/09/14		Générique[200%]
8			Mécanismes de synchronisation JAVA	4,38 jours	Jeu 18/09/14	Mer 24/09/14		Générique
9			▸ Spécifications générales	8,75 jours	Jeu 25/09/14	Mer 08/10/14		
10			▸ Définition du cahier des charges	4,38 jours	Jeu 25/09/14	Mer 01/10/14		
11			Définition des compteurs de synchronisation	4,38 jours	Jeu 25/09/14	Mer 01/10/14		Générique[500%]
12			▸ Recherches des solutions relatives	5,25 jours	Mer 01/10/14	Mer 08/10/14		
15			Rédaction du rapport de pré-étude	5,25 jours	Jeu 09/10/14	Jeu 16/10/14		Générique[400%]
16			Finalisation du rapport de pré-étude	4,88 jours	Ven 17/10/14	Jeu 23/10/14		Générique[400%]
17			Rapport de pré-étude	0 jour	Jeu 23/10/14	Jeu 23/10/14		
18			▸ Spécifications fonctionnelles	18,38 jours	Ven 24/10/14	Ven 05/12/14		
19			▸ Etude fonctionnelle des solutions	18,38 jours	Ven 24/10/14	Ven 05/12/14		
20			Définition du code résultat attendu	1,75 jours	Ven 24/10/14	Mer 29/10/14		Générique[200%]
21			Recherches sur les générateurs de parseur pour Java	1,75 jours	Ven 24/10/14	Mer 29/10/14		Générique
22			Recherches sur la mise en place possible de l'introspection	1,75 jours	Ven 24/10/14	Mer 29/10/14		Générique[200%]
23			Rédaction du rapport de spécification	14 jours	Jeu 30/10/14	Ven 28/11/14		Générique[400%]
24			Finalisation du rapport de spécification	4,38 jours	Sam 29/11/14	Jeu 04/12/14		Générique[400%]
25			Rapport de spécification	1,75 jours	Jeu 04/12/14	Ven 05/12/14		

FIGURE 4.2 – Zoom sur la phase d'analyse

4.2 Affectation des ressources par tâches

Nos ressources sont génériques et équivalentes à 500% durant toute la période du projet.

L'affectation des ressources aux tâches a été faite selon leurs difficultés ainsi les recherches de documentation peuvent être effectués par une seule personne par contre l'identification des différents des différents modules et la modélisation de l'application sont des tâches qui devront être effectué en groupe car elles nécessitent la concertation de toute l'équipe.

La figure suivante illustre comment l'affectation des ressources lors de la phase de conception a été effectuée.

		Mode Tâche	Nom de la tâche	Durée	Début	Fin	Prédécesseurs	Noms ressources
1			► Analyse du projet	45,25 jours	Ven 12/09/14	Ven 05/12/14		
26			▄ Conception	44,13 jours	Ven 05/12/14	Jeu 12/02/15	1	
27			▄ Planification initiale	10 jours	Ven 05/12/14	Mer 17/12/14		
28			Identification des tâches pour la phase d'analyse et de conception	4,38 jours	Ven 05/12/14	Jeu 11/12/14		Générique[500%]
29			Rédaction du rapport de planification	1,75 jours	Ven 12/12/14	Sam 13/12/14		Générique[200%]
30			Préparation de la présentation du projet	5,63 jours	Ven 12/12/14	Mer 17/12/14		Générique[200%]
31			Préparation de la soutenance de planification	4,75 jours	Ven 12/12/14	Mar 16/12/14		Générique[200%]
32			Finalisation de la soutenance du projet	0,88 jour	Mer 17/12/14	Mer 17/12/14		Générique[300%]
33			Finalisation du rapport de planification	0,88 jour	Mar 16/12/14	Mar 16/12/14		Générique[200%]
34			Rapport de planification	0 jour	Mer 17/12/14	Mer 17/12/14		
35			▄ Conception logicielle	33,25 jours	Lun 22/12/14	Jeu 12/02/15		
36			▄ Approfondissement de l'architecture interne du logiciel	20,13 jours	Lun 22/12/14	Jeu 22/01/15		
37			Identification des différents modules	11,38 jours	Lun 22/12/14	Mer 07/01/15		Générique[500%]
38			Modélisation	8,75 jours	Jeu 08/01/15	Jeu 22/01/15		Générique[500%]
39			Rédaction du rapport de conception	6,13 jours	Ven 23/01/15	Lun 02/02/15		Générique[400%]
40			Finalisation du rapport de conception	6,13 jours	Mar 03/02/15	Mer 11/02/15		Générique[400%]
41			Rapport de conception	0 jour	Jeu 12/02/15	Jeu 12/02/15		
42			► Construction	26,25 jours	Ven 13/02/15	Jeu 02/04/15	26	
51			► Livraison	20,38 jours	Ven 03/04/15	Mar 26/05/15	42	
57			► Démonstrations	2,75 jours	Mer 27/05/15	Ven 29/05/15	51	

FIGURE 4.3 – Zoom sur la phase de conception

La figure suivante montre le graphe des ressources, qui montre une utilisation correcte des ressources.

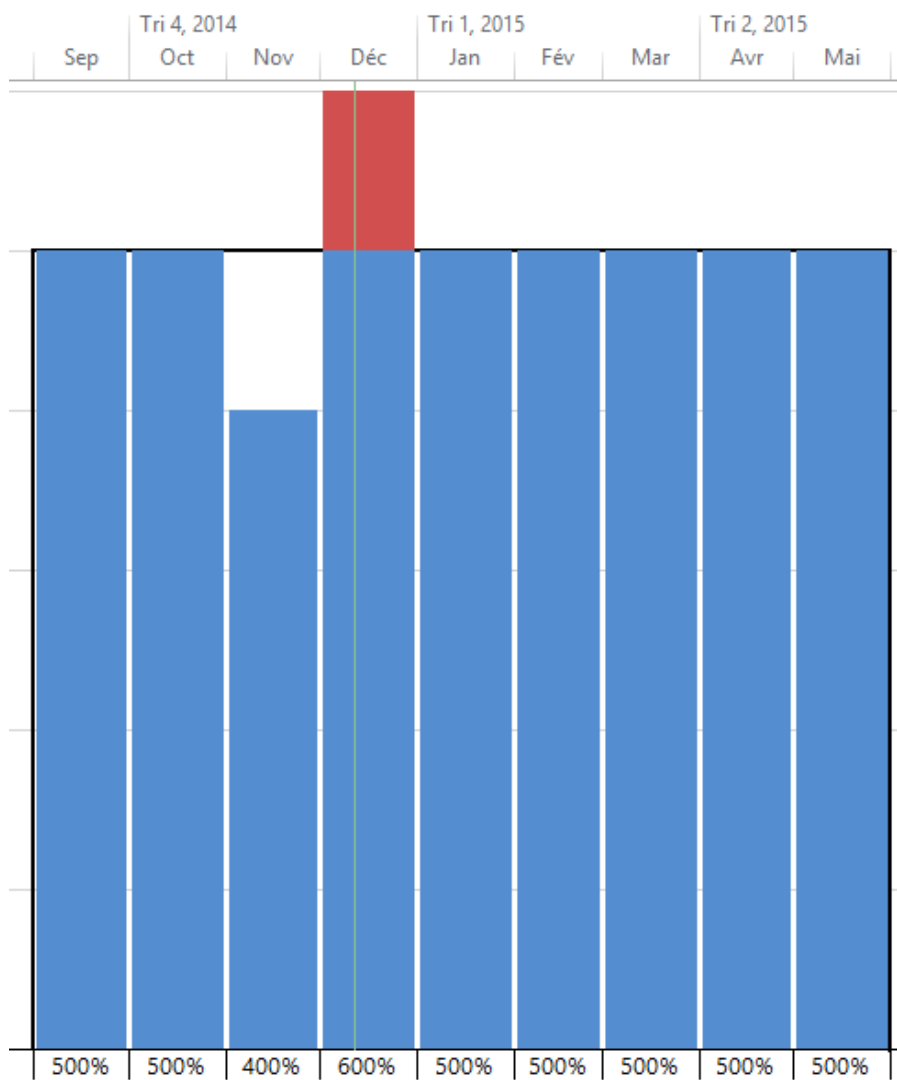


FIGURE 4.4 – Graphe des ressources

4.3 Planning

Le planning a été des grande étape du projet a été plus ou moins défini par le cahier des charges.

La figure suivante montre le diagramme de Gantt des principales tâches, chaque partie du projet étant terminée par la remise d'un rapport :

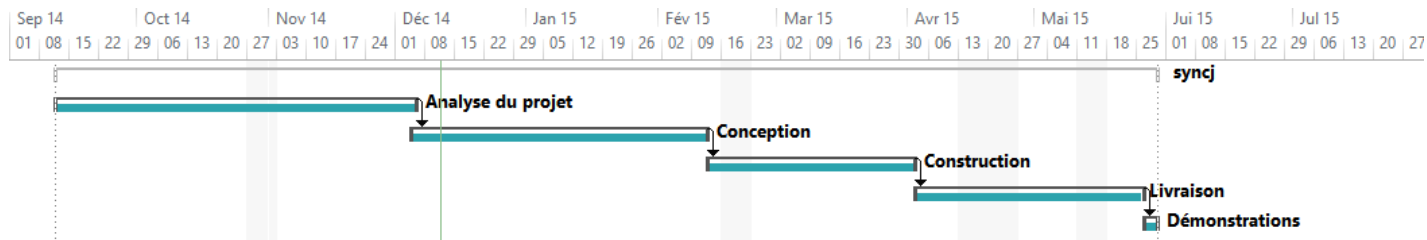


FIGURE 4.5 – Diagramme de gantt des principales tâches

Ensuite, ce planning a dû être découpé en plusieurs périodes pour nous permettre de réaliser chaque sous-tâche.

De manière générale, les sous-tâches sont réalisées en parallèle par les différents membres de l'équipe ou à la suite d'autres tâches qui y sont dépendantes et nous avons affecté une semaine à leur réalisation. Ces périodes peuvent s'allonger jusqu'à deux semaines dans le cas de la rencontre d'une difficulté éventuelle.

La figure suivante montre le diagramme de Gantt de la phase d'analyse du projet :

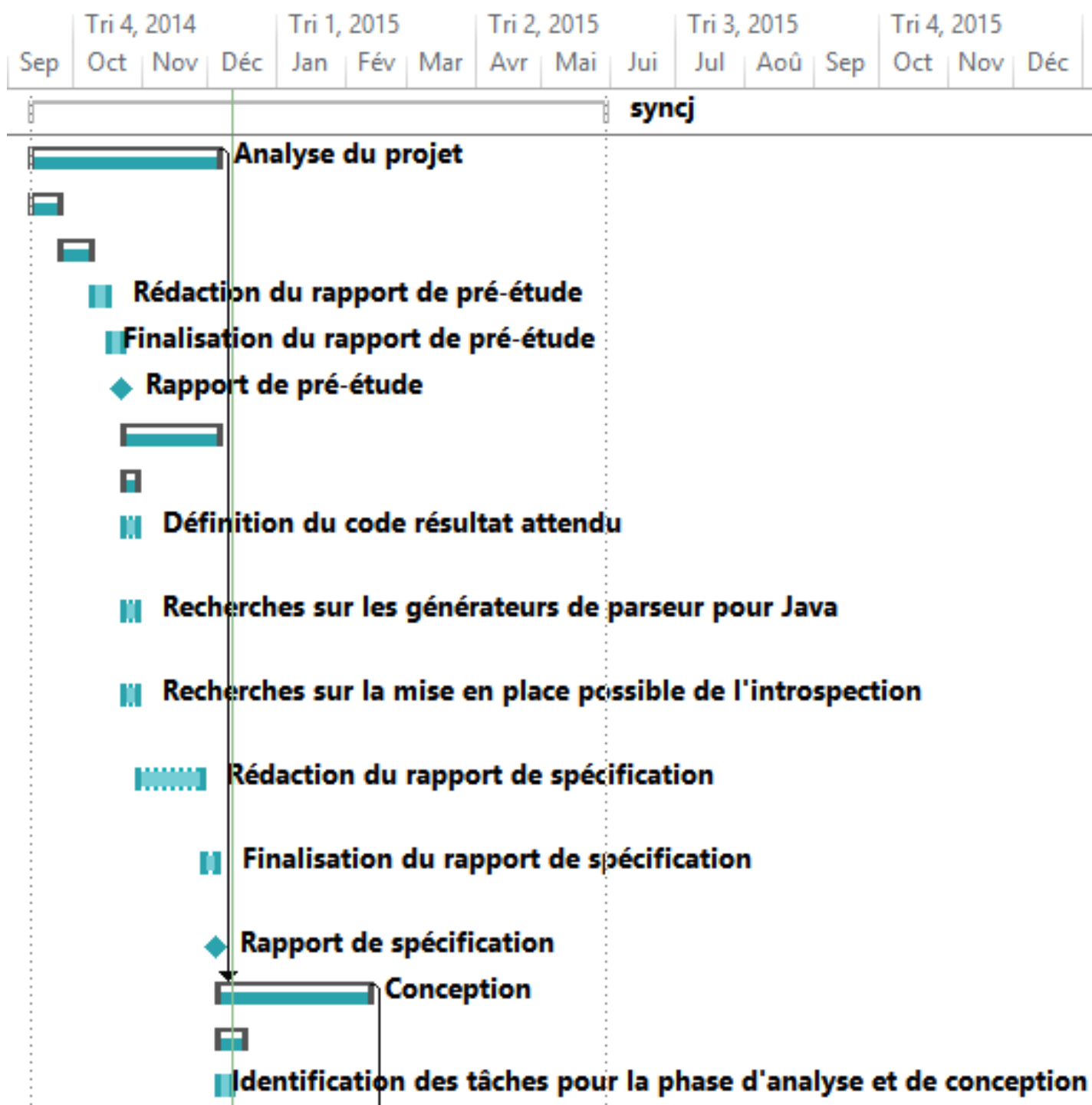


FIGURE 4.6 – Zoom sur certaines parties avec des jalons

5 Conclusion

Cette phase de planification initiale nous a permis de mieux cerner le contexte de notre projet en identifiant les acteurs, le périmètre fonctionnel et de qualification pour bien saisir ce qui est attendu par le client.

Elle nous a permis aussi d'identifier en amont les différents tâches qui composera la phase de conception logicielle et d'identifier les risques.

Elle nous a aidé à identifier le cycle de production adéquat à notre projet qui est le modèle en cascade, à réfléchir sur le cycle de qualification : l'intégration continue et à identifier le mode de pilotage que nous jugeons le plus adéquat au projet : le timeboxing.