



Configuration Interface for MESSage ROUTing

# **Plan de développement logiciel**

---

---

Date : 22/08/07  
Version : 1.2  
Statut : diffusable

**BAGNARD Natacha**  
Auteurs : **FOROT Julien**

## Table des révisions

<i>Version</i>	<i>Date</i>	<i>Modifications</i>
0.1	30/01/07	Création du document
0.2	07/02/07	Modification du document suite à des changements dans le cahier des charges
0.3	07/03/07	Modification du document suite à une relecture de Jérôme Camilleri (fiche PDL-Camilleri_07032007.odt)
1.0	20/03/07	Révision suite à une réunion informelle avec Jérôme Camilleri et validation du document
1.1	05/04/07	Modification suite au premier audit avec Martine Tasset (fiche audit1_040407.odt)
1.2	22/08/07	Ajout du planning ajusté

# Table des matières

<b>1.Introduction.....</b>	<b>4</b>
1.1.Objectifs du document.....	4
1.2.Portée du document.....	4
1.3.Documentation de référence.....	4
1.4.Glossaire.....	4
<b>2.Projet concerné.....</b>	<b>4</b>
2.1.Contexte et but du projet.....	4
2.2.Communication au sein du projet.....	5
<b>3.Cycle de vie.....</b>	<b>5</b>
3.1.Choix du cycle de vie.....	5
3.2.Incréments identifiés.....	6
3.3.Phases du cycle de vie.....	8
<b>4.Planification du projet.....</b>	<b>8</b>
4.1.Dates et durée imposées.....	8
4.1.1.Soutenances.....	8
4.1.2.Durée du projet.....	8
4.1.3.Durée réelle de travail.....	8
4.2.Estimation.....	9
4.2.1.Mode de développement.....	10
4.2.2.Taille du logiciel.....	10
4.2.3.Répartition de l'effort.....	11
4.2.4.Répartition de la durée.....	11
4.2.5.Analyse du projet précédent.....	12
4.3.Planning prévisionnel.....	12
<b>5.Gestion des risques.....</b>	<b>13</b>
<b>6.Annexes.....</b>	<b>14</b>
6.1.Planning ajusté.....	14

## **1.Introduction**

### **1.1.Objectifs du document**

Ce document présente le Plan de Développement Logiciel (PDL) adopté pour le projet CIMERO.

Le plan de développement logiciel entre dans le cadre de l'organisation et de la gestion du projet. Il a pour objectif de fixer de façon détaillée les principes et les moyens pour mener à bien le développement du logiciel.

Il précise le rôle de chaque personne intervenant dans le projet et détaille les tâches à accomplir ainsi que leur enchaînement. Le Plan de Développement Logiciel présente l'estimation de l'effort et de la durée nécessaires pour l'accomplissement de chaque tâche du cycle de vie ainsi que le suivi et la planification de toutes les tâches du projet.

### **1.2.Portée du document**

Ce document est destiné :

- au responsable du stage, Jérôme Camilleri
- à la consultante : Martine Tasset
- au jury du Master 2 Pro GI pour l'évaluation du stage
- à l'équipe projet : Natacha Bagnard et Julien Forot

### **1.3.Documentation de référence**

Le présent document fait référence au Cahier des Charges du projet (« CahierDesCharges.odt ») et est rédigé en fonction des clauses qualités définies dans le Plan d'Assurance Qualité Logicielle (« PlanAssuranceQualiteLogicielle.odt »).

### **1.4.Glossaire**

--> voir la section 1.3 (« Définitions, acronymes et abréviations ») du Cahier des Charges (CahierDesCharges.odt)

## **2.Projet concerné**

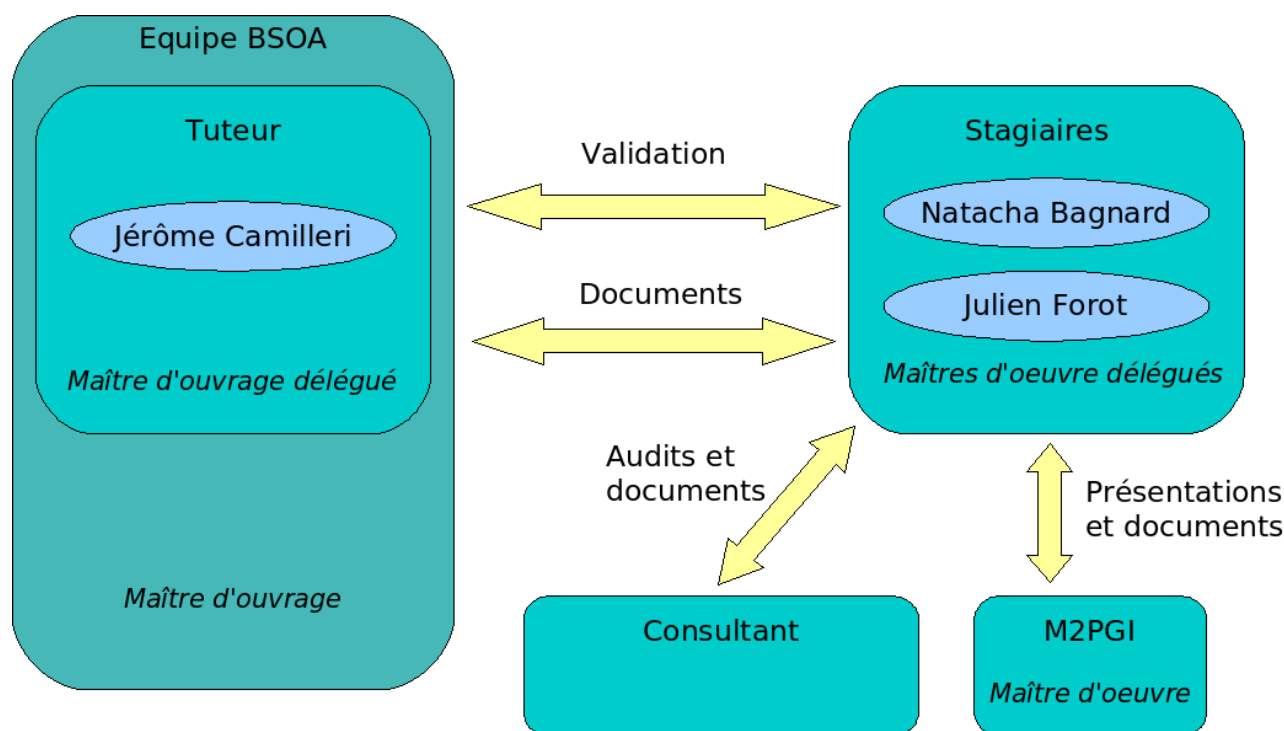
### **2.1.Contexte et but du projet**

Le logiciel visé dans ce document, nommé CIMERO (Configuration Interface for Message Routing), est une interface de configuration de flux, réalisée au sein de l'équipe BSOA. Les objectifs et les détails de ce projet sont précisés dans le cahier des charges.

## 2.2.Communication au sein du projet

Ce projet est réalisé par Natacha Bagnard et Julien Forot, étudiants en Master 2 Pro Génie Informatique (Université Joseph Fourier - Grenoble) au cours d'un stage chez Bull de janvier à fin septembre 2007.

Le schéma suivant représente les principaux intervenants du projet, leur(s) rôle(s) ainsi que les flux de communication entre ces personnes :



*Illustration 1: Échanges au sein du projet CIMERO*

## 3.Cycle de vie

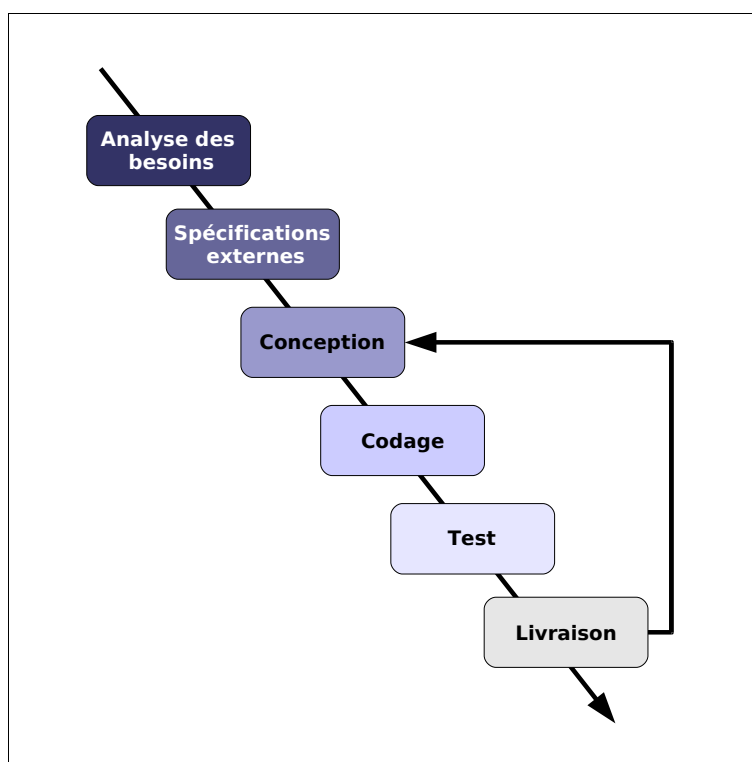
### 3.1.Choix du cycle de vie

Le but d'un cycle de vie dans un projet de génie logiciel est de représenter les étapes de développement et de maintenance pour faciliter :

- ✓ la planification et la gestion de projet
- ✓ le contrôle de la qualité
- ✓ la communication
- ✓ l'identification des métiers et des techniques propres à chaque étape

Le déroulement de ce projet suivra un cycle de vie incrémental. Ce choix s'explique par plusieurs raisons :

- Le travail à effectuer se découpe en composants distincts (les incréments). Chaque composant peut donc être vu comme un sous-projet qui pourra être développé quasi-indépendamment des autres. Les délais pourront ainsi être raccourcis grâce à un développement en parallèle de différents incréments.
- Ces différents incréments permettront d'obtenir une version provisoire mais stable de notre logiciel. Ainsi, un outil fonctionnel sera assurément livré à la date de livraison du produit.
- Si des incréments n'ont pas été implémentés avant la date de livraison, ils pourront l'être « facilement » par une autre équipe grâce à l'analyse des besoins que nous avons effectuée au préalable.



*Illustration 2: Cycle de vie incrémental*

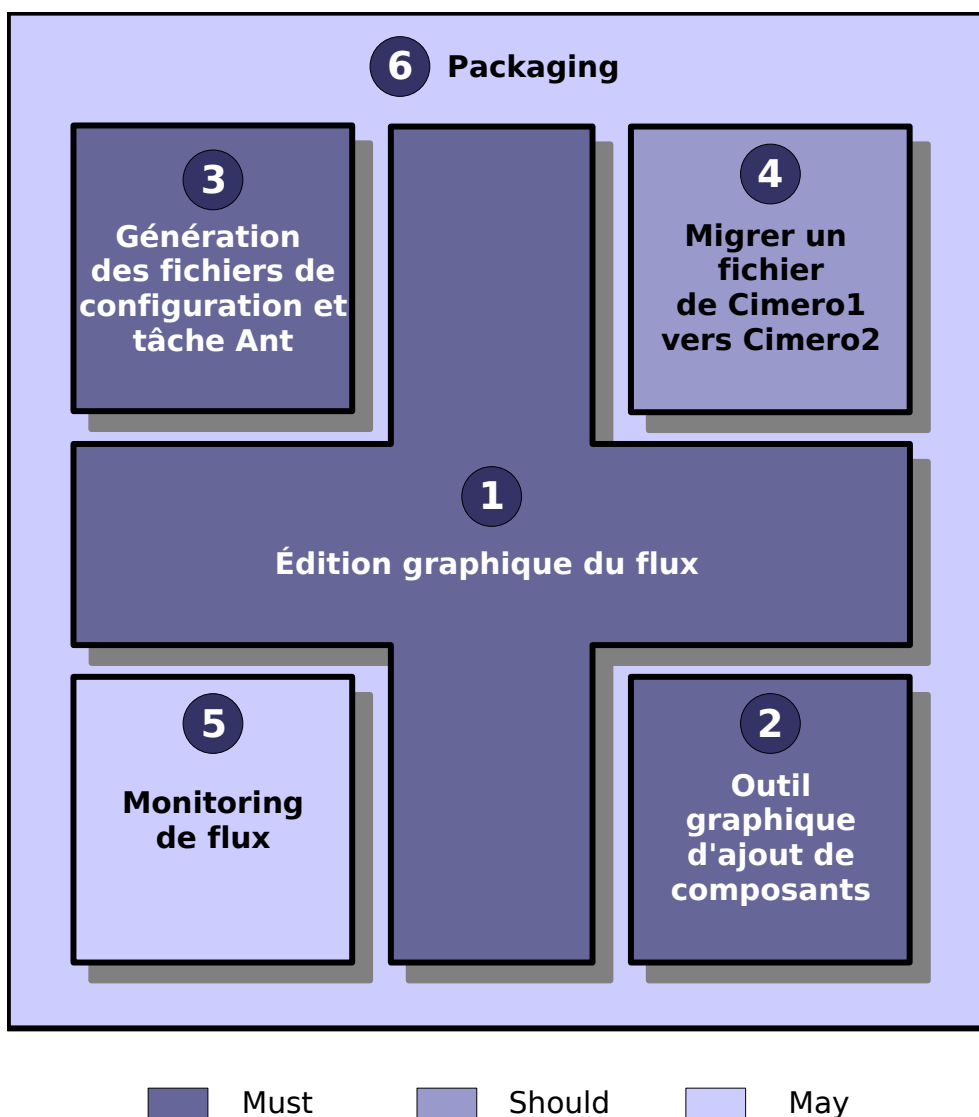
### 3.2. Incréments identifiés

Les incréments suivants ont été identifiés suite à l'analyse des besoins détaillée dans le cahier des charges<sup>1</sup>.

- *1<sup>er</sup> incrément* : Redéveloppement de la partie graphique de CIMERO grâce au framework GMF : Pas de possibilité de générer et déployer. Ajout manuel des composants dans la palette.

<sup>1</sup> Voir CahierDesCharges.odt section 3

- 2<sup>ème</sup> incrément : Outil graphique d'ajout de composant.
- 3<sup>ème</sup> incrément : Redéveloppement de la partie génération et déploiement (Package JBI correspondant au graphe, arborescence de répertoires contenant les SAs et SUs , tâche Ant permettant de créer un ou plusieurs SAs).
- 4<sup>ème</sup> incrément : Migrer un fichier Cimero1 vers Cimero2.
- 5<sup>ème</sup> incrément : Développement d'un outil de monitoring du flux.
- 6<sup>ème</sup> incrément : Mise en place des différents packaging : plugin classique et standalone.



*Illustration 3: Incréments identifiés*

### **3.3. Phases du cycle de vie**

- « Analyse des besoins » : Établissement du cahier des charges et des contraintes du système.
- « Spécifications externes » : Description des fonctionnalités et description des interfaces du logiciel avec le monde extérieur (utilisateurs et autres systèmes).
- « Conception » : Définition de l'architecture du logiciel, des principes de réalisation et des structures de données. Description de la structure générale du logiciel. Découpage du logiciel en modules.
- « Codage » : Réalisation du système.
- « Tests » : Comparaison du comportement du code produit avec celui attendu lors de la conception. Test(s) d'intégration du(des) module(s) : Vérification de l'architecture décrite lors de la conception. Tests systèmes et d'intégration du logiciel. Vérification de l'adéquation entre les propriétés du système et la description des besoins.
- « Livraison » : livraison au client

## **4. Planification du projet**

### **4.1. Dates et durée imposées**

#### **4.1.1. Soutenances**

Le déroulement de ce projet se faisant dans le cadre du Master 2 Pro GI, les dates de soutenance suivantes nous sont imposées :

- soutenance de mi-parcours : 15 mai 2007
- soutenance finale : septembre 2007

#### **4.1.2. Durée du projet**

Le temps imparti pour le projet est de :

- ✓ 3 mois à mi-temps (3 jours par semaine) : du 09/01/07 au 31/03/07
- ✓ 6 mois à plein temps : du 01/04/07 au 30/09/07

La durée du projet est donc de 9 mois.

#### **4.1.3. Durée réelle de travail**

La durée réelle du projet est de 7.4 mois :



- 3 mois à 3 jours par semaine – 3 jours de vacances (20,21,22 février) :  $3 \times 60\% - (1/4 \times 60\%) = 1.8 - 0.15 = 1.65$  mois
- 6 mois – 1 semaine de vacance (fin juillet) = 5.75 mois

Jours fériés durant la période de stage :

- avril : 1 jour férié + 1 jour de congé
- mai : 4 jours fériés + 1 jour de congé
- août : 1 jour férié

Le tableau suivant représente le nombre de jours de travail effectifs en considérant 7 heures de travail quotidien.

Mois	Nombre de jours de travail	Nombre d'heures de travail
Janvier	12	84
Février	12	84
Mars	12	84
Avril	20	140
Mai	16	112
Juin	21	147
Juillet	19	133
Août	19	133
Septembre	22	154
<b>Total</b>	<b>9 mois</b>	<b>1071</b>

## 4.2.Estimation

Le modèle CoCoMo (Constructive Cost Model) de base, ainsi que le planning de nos prédécesseurs seront utilisés pour estimer l'effort nécessaire pour l'accomplissement de chaque tâche ainsi que le temps nécessaire.

L'estimation se fait par étape :

- détermination du mode de développement adapté au projet
- détermination de la taille du logiciel
- détermination des tâches
- estimation de l'effort
- estimation de la durée

#### 4.2.1.Mode de développement

Ce projet s'effectue au sein d'une équipe du département R&D. Sa durée est environ de 6 mois et demi. Il peut être considéré, d'après le modèle CoComo, comme un projet dit : « moyen ». De plus, nous ne sommes pas experts dans les technologies utilisées et notre expérience est moyenne.

Le mode de développement correspondant le mieux à ce projet est donc « semidetached ».

#### 4.2.2.Taille du logiciel

Mode	Effort (hmois)	Durée (mois)
Organic	2.4 KDSI <sup>1.05</sup>	2.5 Effort <sup>0.38</sup>
Semidetached	3.0 KDSI <sup>1.12</sup>	2.5 Effort <sup>0.35</sup>
Embedded	3.6 KDSI <sup>1.20</sup>	2.5 Effort <sup>0.32</sup>

*Tableau 1: Modèle CoCoMo*

##### Effort :

Ce projet sera réalisé par deux personnes. L'effort qui sera fourni sera de deux fois la durée réelle du projet, soit :

- 2\*7.4 mois = 14.8 homme-mois
- 2\*153 jours = 306 hommes-jours

##### Taille du logiciel en LOC (Lines Of Code) :

- En fonction de l'effort (14.8hm) et du modèle CoCoMo, l'estimation du nombre de lignes écrites est de 4160 LOC. La durée devrait, toujours d'après le modèle CoCoMo, être de 6.41 mois, alors qu'elle est de 7.4 mois. L'estimation est trop pessimiste.
- En fonction de la durée réelle (7.4 mois), l'estimation du nombre de lignes écrites est de 5970 LOC. L'effort fourni serait de 22.2 hm alors qu'il est de 14.8hm. L'estimation est trop optimiste.

De plus, l'expérience montre qu'un groupe de deux personnes écrit entre 25 et 50 LOC par jour. Pour la durée totale de ce projet, ces chiffres indiquent le projet contiendra entre 3875 (assez pessimiste) et 7750 (assez optimiste) LOC.

Selon ces données, l'approximation du nombre de lignes de code à développer pour ce projet se situe en conséquence autour de 5400.

### **4.2.3. Répartition de l'effort**

En se basant sur le modèle CoCoMo, nous pouvons estimer l'effort nécessaire pour réaliser chaque phase du cycle de vie que nous avons choisi. Tout d'abord, nous devons mettre en correspondance les phases de notre cycle de vie et celles du modèle.

Phases du CV (CoCoMo)	Phases du CV (projet)
Analyse des besoins	Analyse des besoins
Spécifications	Spécifications externes
Conception	Conception
Codage et tests unitaires	Codage
Intégration et tests	Tests

L'estimation de l'effort nécessaire par phase du cycle de vie est présenté dans le tableau suivant :

Tâches	Répartition (en %)	Effort (en homme-mois)
Analyse des besoins	7	1.036
Spécifications	17	2.516
Conception	26	3.848
Codage et tests unitaires	35	5.18
Intégration et tests	22	3.256
Total	100	14.8

### **4.2.4. Répartition de la durée**

Le tableau suivant résume l'estimation du nombre de jours qui sera consacré pour chaque phase du cycle de vie.

Tâches	Répartition (en %)	Durée (en jours)
Analyse des besoins	18	27.9
Spécifications	25	38.75
Programmation ( <i>conception, codage et tests unitaires</i> )	52	80.6
Intégration et tests	23	35.65

Tâches	Répartition (en %)	Durée (en jours)
Total	118	182.9

#### 4.2.5. Analyse du projet précédent

Voici le planning final du projet CIMERO 2006 :

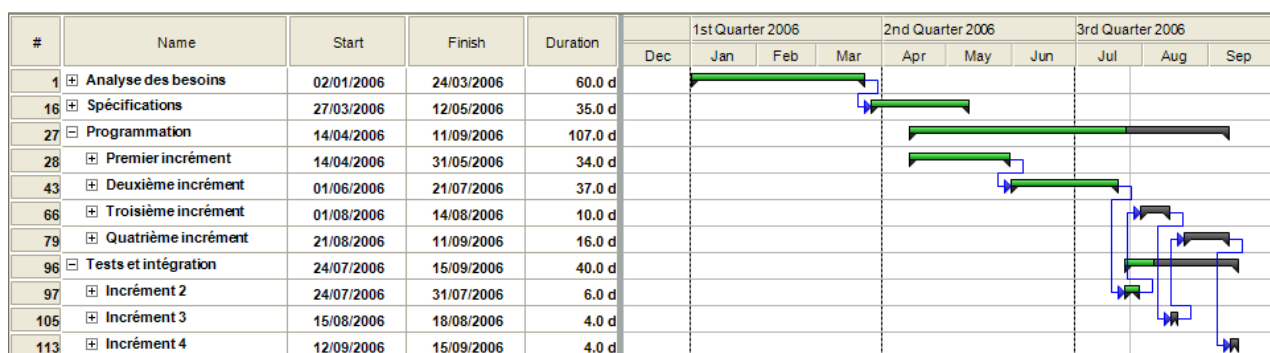


Illustration 4: Planning CIMERO 2006

Le cycle de vie choisi est un cycle incrémental (6 incréments). La phase de spécification a été menée parallèlement à la phase de programmation et les phases de test ont été menée parallèlement aux phases de programmation. Il ne semble donc pas possible de mener toutes les étapes séquentiellement.

Le nombre de lignes produites réel est de 25000 lignes. Le taux de commentaire est estimé à 40%. Le nombre de lignes de code produites est donc de  $25000 \times 60\% = 15000$ . Leur estimation basé sur le modèle CoCoMo était de 4500 lignes. Leur estimation a été dépassée de plus de 200%. Nous pouvons donc penser que notre estimation CoCoMo est à revoir à la hausse.

#### 4.3. Planning prévisionnel

Voici le planning prévisionnel construit à partir du modèle CoCoMo et des observations du projet précédent.

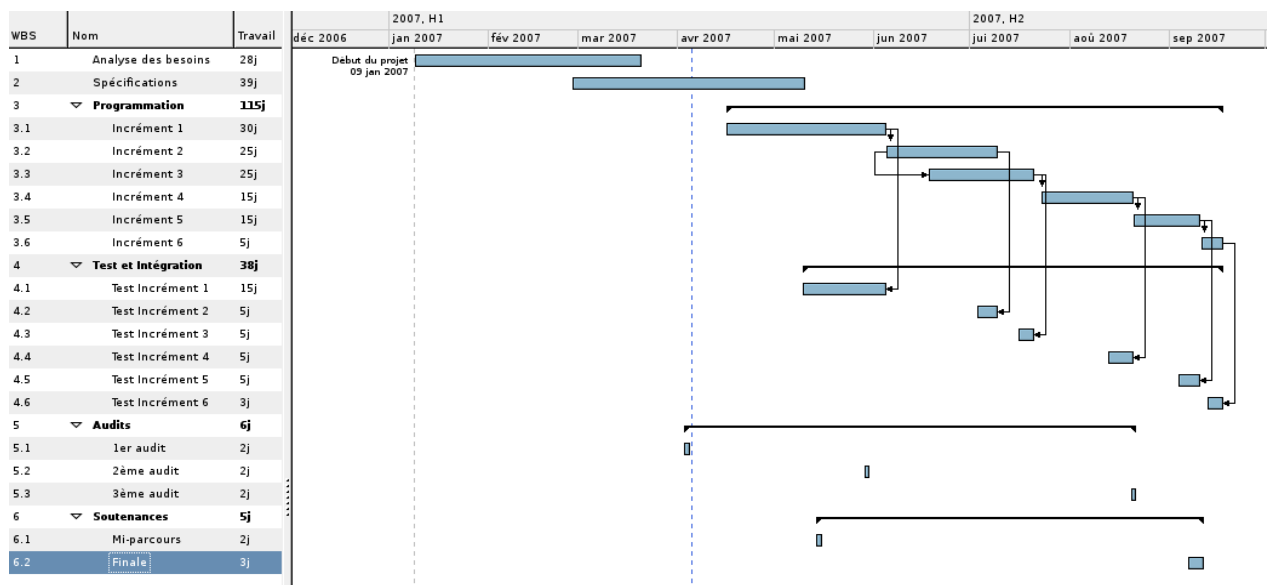


Illustration 5: Planning prévisionnel

## 5. Gestion des risques

Cette partie présente les risques qui peuvent survenir lors de ce projet. Pour chacun d'entre eux, l'impact sur le projet, les solutions préventives ainsi que les solutions curatives seront explicitées.

Risque	Impact	Probabilité	Solutions préventives	Solutions curatives
Modification des besoins du client	Important  De quelques jours (solution 1) à quelques mois (solution 2)	faible	Réunions fréquentes avec le maître d'ouvrage	1. Tentative d'adaptation aux nouveaux besoins  2. Retour à la phase d'analyse des besoins

<i>Risque</i>	<i>Impact</i>	<i>Probabilité</i>	<i>Solutions préventives</i>	<i>Solutions curatives</i>
Technologies utilisées mal connues et immatures	Important  De quelques jours (solution 1) à quelques mois (solution 2)	important	- Étude préalable approfondie des technologies utilisées  - Maquettage	1. Utilisation si possible de technologies alternatives  2. Retour à la phase d'analyse des besoins
Indisponibilité d'un membre	Faible à important  De quelques jours à quelques semaines	faible	- Réunion quotidienne des membres de l'équipe  - Tenue d'un carnet de bord	1. Correction du planning prévisionnel  2. Redistribution du travail à effectuer

## 6. Annexes

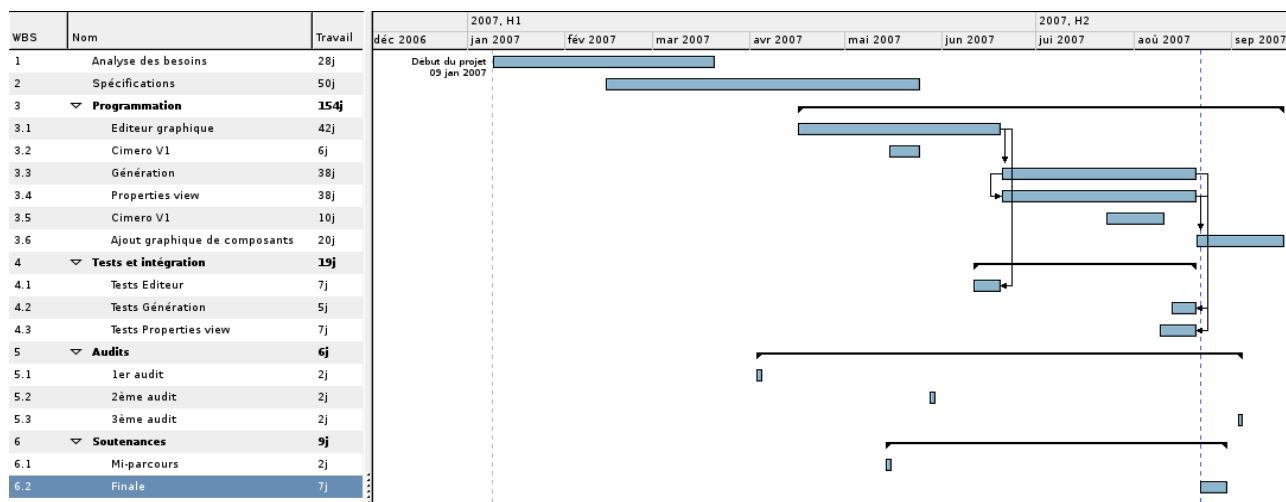
### 6.1. Planning ajusté

Les incréments 2 (Outil graphique d'ajout de composant) et 3 (Génération des fichiers de configuration et tâche Ant) ont été inversés pour avoir plus rapidement une version fonctionnelle.

Un nouvel incrément a été ajouté lors du développement. En effet, le développement d'une Properties View plus adaptée que celle de la précédente version était nécessaire.

De plus, un travail sur la version précédente de Cimero a été effectué.

De ce fait, le planning prévisionnel a subi des modifications. Le planning suivant est la planification prévisionnelle adaptée à ces modifications.



*Illustration 6: Planning prévisionnel ajusté*