Table des matières

[1. Objet du document : 2](#_Toc340735125)

[2. Présentation du Projet : 2](#_Toc340735126)

[2.1 Objectifs du projet : 2](#_Toc340735127)

[3. Organisation du projet : 2](#_Toc340735128)

[3.1 Méthodologie du projet : 2](#_Toc340735129)

[3.2 Formalisme, Documentation 3](#_Toc340735130)

[3.3 Matrice rôle du projet 3](#_Toc340735131)

[3.3.1 Scrum Master : 3](#_Toc340735132)

[3.3.2 Membre constituant l’équipe : 4](#_Toc340735133)

[3.3.3 Product Owner : 4](#_Toc340735134)

[3.3.4 Responsable de communication : 4](#_Toc340735135)

[3.3.5 Responsables pédagogiques 4](#_Toc340735136)

[3.4 Backlog : 5](#_Toc340735137)

[3.5 Besoins en Formation : 5](#_Toc340735138)

[3.6 Ressources logiciels et environnement matériels : 5](#_Toc340735139)

[3.6.1 Progiciel de Gestion de Projet 5](#_Toc340735140)

[3.6.2 Atelier de génie logiciel 6](#_Toc340735141)

[3.6.3 Outils : 6](#_Toc340735142)

[4. Processus de développement Logiciel : 6](#_Toc340735143)

[4.1 Phase de développent : 6](#_Toc340735144)

[4.1.1 Analyse de besoins Logiciel : 6](#_Toc340735145)

[4.1.2 Conception préliminaire : 6](#_Toc340735146)

[4.1.3 Conception Détaillés : 6](#_Toc340735147)

[4.1.4 Implémentation : 6](#_Toc340735148)

[4.2 Développement mené par les tests : 6](#_Toc340735149)

[5. Pilotage du projet : 7](#_Toc340735150)

[5.1 Processus itératif et incrémental : 7](#_Toc340735151)

[5.2 Estimation des charges et durées : 7](#_Toc340735152)

[5.3 Planification : Plan de phases 7](#_Toc340735153)

[5.4 Choix de priorisation : 7](#_Toc340735154)

[5.5 Réunion de suivi 7](#_Toc340735155)

[5.6 Gestion des risques 8](#_Toc340735156)

[5.7 Tableau récapitulatifs des livrables 8](#_Toc340735157)

# Objet du document :

Ce document a pour objectif de mettre en place un plan de projet (Software Development

Plan), il s’agit de définir aussi la méthode de conduite de projet utilisée, le document traite les modèles de développement, les responsabilités organisationnelles, outils, gestion de configuration et la qualité logicielle

Le document va évoluer au fur et à mesure de l’avancement dans le cycle de vie du projet, il devra être réexaminé/modifié à la fin de chaque Release/Comité de pilotage, mais peut être aussi mis à jour à d’autres moments, si besoin est.

Le contenu de ce document sera organisé selon les axes suivants :

- Organisation du projet

- Développement du projet

- Pilotage du projet

# Présentation du Projet :

## Objectifs du projet :

Il s’agit d’implémenter une application qui exploite un réseau ferré et des équipements automatiques. Ce projet sera réalisé par une équipe de dix étudiants Miagistes spécialisés dans l’architecture des systèmes distribués, la version finale du projet devra être prête pour la troisième semaine du mois de mai 2013.

# Organisation du projet :

## Méthodologie du projet :

Cette méthode correspond au mieux pour le choix du modèle de développement du projet PDS parce que :

* Taille de l’équipe (10 personnes)
* Comités de pilotage / réunions avec le Product Owner et les responsables pédagogiques fréquentes
* La spécification du besoin dans l’énoncé du projet de synthèse demande l’utilisation des pratiques agiles

En Plus des choix de pratiques agiles, le projet va adopter certains principes des processus unifiés (UP) incluant une documentation bien détaillée au niveau des spécifications techniques

. Dans les rubriques suivantes, on va définir les pratiques définis pour le PDS, ces pratiques vont devoir s’appliquer dans le processus de développement du projet

## Formalisme, Documentation

Afin d’assurer l’accès à la documentation en tout moment, une documentation technique détaillée devra être rédigé

Chaque document doit être sous format de Template Word définissant les principaux axes et le style du document.

Les versions des documents vont être rédigés en mode révision, chaque révision ne sera acceptée qu’après validation de toute l’équipe, éventuellement des retours lors des comités de pilotage et/ou lors des réunions de méthodologie de projet de synthèse.

## Matrice rôle du projet

Le besoin en termes de formation va être assuré principalement à travers la formation ISIDIS, en plus, des formations par transfert de connaissances ou bien par de la documentation seront assurés, les charges de formation vont apparaitre dans le Backlog.

La plupart des outils et technologies utilisés ne sont pas maitrisés par les membres de l’équipe, des tutoriaux sur ces outils vont être rédigés et mis en place dans l’espace du projet, des présentations lors des réunions sont prévus aussi,

### Scrum Master :

Chaque itération doit être dirigée par un nouveau Scrum Master. Le rôle du Scrum Master est de :

* Assurer l’affectation et le suivi des taches dans l’espace de travail de Version One
* Définition de l’ordre du jour de chaque réunion, et rédaction d’un compte rendu de réunion
* Participation comme chaque membre dans la réalisation des taches du projet.

### Membre constituant l’équipe :

-Khaoula Khaiter

-Hamza Lazrak

-Driss Krafess

-Nidal [Mahraz](https://www.facebook.com/nidal.mahraz)

-Soukaina Merrouche

-Sibel Ozbey

-Patrick CODO

-Hamza Saoudi

-Rafik Boutaba

-Sabri Jadoui

### Product Owner :

Afin de présenter le client et ses intérêts, le projet devra avoir besoin d’un Product Owner. Le Product Owner est quelqu’un extérieur au groupe de projet, il correspondra le plus possible aux critères suivantes :

* Disponibilité assez suffisante
* Familiarisation avec le milieu des systèmes distribués et les Framework

En l’absence du Product Owner, un autre Product Owner sera nommé à l’intérieur du projet, il devra valider les US

Le ProductOwner du projet est :

X

Mode de communication avec le ProductOwner :

* Réunion à la fin de chaque itération & communication via e-mails, …

### Responsable de communication :

Il sera responsable de communication entre l’équipe du projet et les responsables pédagogiques, ainsi, qu’avec toute personne extérieur à l’environnement du projet. Ce rôle sera tenu par Nidal Mahraz.

### Responsables pédagogiques

Les responsables pédagogiques du module sont Alexandre BRENNER (intervenant en conduite de projet en ISIDIS et ISIAD), Gilles GIRAUD (intervenant en architectures distribuées en ISIDIS).

Récapitulatif des rencontres avec les responsables pédagogiques :

|  |  |
| --- | --- |
| Lancement du projet | 19/10/2012 de 13h à 14h |
| Comité de pilotage n° 1 | 5/12/2012 |
| Comité de pilotage n° 2 | 30/01/2013 |
| Comité de pilotage n° 3 | 28/03/2013 |
| Démo finale / Soutenance | 1. au 23 mai 2013 |

## Backlog :

Issue de la rédaction du Vision Document, le Backlog est mis en place dans l’application Version One et il est accessible par tout membre de l’équipe.

Le Backlog pourra être modifié au début de chaque itération pour personnaliser la priorité et la charges des tâches, le Backlog va contenir tous les travaux à effectuer, fonctionnels et techniques

## Besoins en Formation :

Le besoin en termes de formation va être assuré principalement à travers la formation ISIDIS, en plus, des formations par transfert de connaissances ou bien par de la documentation seront assurés, les charges de formation vont apparaitre dans le Backlog.

## Ressources logiciels et environnement matériels :

Les ressources logicielles utilisées dans le cadre du projet vont permettre d’optimiser le développement. Les membres du groupe doivent impérativement installer sur leurs machines les mêmes versions des outilsafin d’assurer une uniformité de travail.

* Postes développeurs : On utilisera les postes personnels des membres ou les postes mis à disposition par l’ESIAG.
* Environnement de développement : c’est l’environnement Cible de l’intégration continue. Dans le cas du projet ça sera Jenkins.
* Environnement de test : Il sera destiné au client (Product Owner), pour, et dans le cadre de notre développement itératif, pouvoir détecter à temps les bugs et apporter des corrections.

Chaque machine des membres de l’équipe devra utiliser un client TortoiseSvn et le plugin SubVersion intégré à l’IDE Eclipse.

### Progiciel de Gestion de Projet

* **Version One** : offert en version gratuite pour 10 personnes (avec une limite de fonctionnalités), il est le mieux adapté pour le choix d’outil permettant d’organiser le travail selon des pratiques agiles, on définira sur Version One les US, les tickets, le Backlog…

### Atelier de génie logiciel

* **BOUML :** pour la réalisation des diagrammes UML, l’option de génération de code à base des diagrammes sera utilisée afin d’optimiser le temps d’implémentation et pour avoir un code source correspondant aux diagrammes réalisés

### Outils :

Les outils utilisés dans le projet vont permettre à optimiser le développement, les membres du groupe doivent obligatoirement installer sur leur machines les mêmes versions des outils afin d’assurer une uniformité de travail. Les outils utilisés sont cités et doivent être mis à jour dans le plan du projet, ils comprennent :

* Outils de gestion du projet : Version One.
* Outils d’intégration continue : Jenkins.

# Processus de développement Logiciel :

## Phase de développent :

### Analyse de besoins Logiciel :

### Conception préliminaire :

L’étape de Conception préliminaire, c’est l’étape où les paramètres de conception sont optimisés en fonction des objectifs et des besoins du client. Dans les cas complexes la conception préliminaire est divisée en étapes à précision progressive.

### Conception Détaillés :

Cette conception commence à l’issue de la conception préliminaire. Elle s’appuie sur le dossier de conception préliminaire appelé également Spécification Technique du Besoin.

### Implémentation :

Le processus d’implémentation sera la dernière étape du cycle de vie d’un US.

**Chaque classe implémentée doit avoir une classe de test assurant le bon fonctionnement de ses services.**

## Développement mené par les tests :

Le produit final de la solution du projet de synthèse doit être livré au début de mai 2013, et donc le produit dont on vérifie en permanence la qualité.

Les tests unitaires seront automatisés afin d’optimiser le temps de leurs exécution.

Les tests seront implémentés avant d’implémenter chaque US, afin de tester les fonctionnalités écrites dans les spécifications, sans avoir à mettre en place un test unitaire qui correspond aux fonctionnalités développées.

Cette méthode de développement permet de commencer avec les vérifications de bas niveau avant même que la fonctionnalité globale ne soit finalisée. Le résultat final sera donc un code plus juste et plus fiable.

# Pilotage du projet :

## Processus itératif et incrémental :

Le projet fonctionnera en cycles itératifs et incrémentaux afin d’éviter l’effet tunnel, ce processus est le plus adapté au contexte de travail du projet au fait d’avoir des comités de pilotage et des réunions assez fréquentes avec le Product Owner dans lesquelles on devra faire des démonstrations.

## Estimation des charges et durées :

L’estimation de charges collectives, mise à jour à chaque début d’itération et selon le RAF des tâches à accomplir.

## Planification : Plan de phases

Le produit doit se concrétiser et doit être terminé vers la deuxième semaine de mai 2013, les comités de pilotage vont correspondre à près de 4 jours avant les échéances des Releases pour nous laisser le temps de rectifier les erreurs commises grâce aux remarques qu’on aura lors des comités, on au aura donc 4 Releases tout au long de la durée de vie du projet, chaque Release contiendra des itérations avec une durée de deux semaines.

## Choix de priorisation :

Les « Work item » définis dans le Backlog du projet seront ordonnés de manière à ce que les fonctionnalités les plus critiques et plus risquées soient traitées au début, les fonctionnalités en haut du Backlog seront donc traités en premier, en descendant vers le bas des fonctionnalités du Backlog

## Réunion de suivi

Les réunions de suivi s’effectueront de manière régulière, deux fois par semaine, ainsi qu’en début et fin de chaque phase/itération, la durée de chaque réunion dépend du contenu à traiter et de l’état d’avancement dans le cycle phase/itération (en principe, les réunions de début/fin de Release/itération auront une durée assez longue par rapport aux deux réunions hebdomadaire). Occasionnellement, des réunions de crise pourront se tenir (selon l’état d’avancement/ changement au cours du projet)

A l’issue de chaque réunion, des comptes rendus de réunion doivent être rédigés par le ScrumMaster.

## Gestion des risques

Un document de gestion des risques du projet va être mis en place afin d’identifier les risques en début du projet et de pouvoir les surveiller en permanence, ce document va être réévalué au début de chaque phase afin de pouvoir lui ajouter, supprimer, modifier des risques selon l’état d’avancement et les changements au cours du projet.

Les risques projet vont être réévalués, au début de chaque Release.

.

## Tableau récapitulatifs des livrables

Ci-dessous un tableau récapitulant la liste des livrables et les différentes dates de livraison :

|  |  |  |
| --- | --- | --- |
| WORK ITEM / LIVRABLE | **TAILLE**  **MAXIMALE** | **RESSOURCES AFFECTEES** |
| Un Vision Document | 15 pages | 1 |
| Une analyse des risques du projet | 8 pages | 1 |
| Un « plan projet » décrivant votre méthode | 15 pages | 1 |
| Un Roadmap et un Product Backlog avec estimation des charges | - | 1 |
| Un UC Model | 5 pages | 1 |
| Un benchmarking des solutions techniques existantes + un diagramme représentant l’architecture technique préliminaire | 10 pages | 1 + 1 si l’effectif le permet |
| Un prototype d’architecture logicielle préliminaire | - | 1 |
| Une plateforme d’intégration complète + Support de formation | - | 1 + 1 si l’effectif le permet |

**Les livrables ne doivent en aucun cas dépasser la taille maximale en nombre de pages,** Le dépassement du nombre de pages imposé entraîne immédiatement le retrait d’un quart de point par page supplémentaire sur la note associée au livrable.

Les numéros de version des documents sont définis sur 3 chiffres sous la forme **X.X.X**, la première version de chaque document est 0.0.1, le premier numéro à droite est incrémenté par 1 après chaque livraison en fin d’itération. Les livraisons qui seront faites pour le Product Owner et/ou lors des réunions de méthodologie de conduite de projet devront incrémentés le deuxième numéro à droite par 1 et initialisé le premier numéro à droite à zéro (ex : après la première livraison du document pour le Product Owner, la version du document sera à 0.1.0). Les livraisons de fin de phases devront incrémenter le troisième numéro à droite par 1 et initialiser les deux autres numéros à zéro (vers la fin de la première phase, le numéro de version doit correspondre à 1.0.0)

Après chaque livraison, une étiquette de l’espace de travail en cours doit être créée et hébergée sous le serveur SVN, elle doit être nommée de la manière suivante :

**[NOM\_PROJET\_ VERSION\_AAAAMMJJ]**