



MASTER 2 – ISIDIS

Projet de synthèse

V.A.L.

Plan du Projet

Table des matières

1. Objet du document :	3
2. Présentation du Projet :	3
3. Organisation du projet :	3
3.1 Méthodologie du projet :	3
3.2 Formalisme, Documentation.....	4
3.3 Matrice rôle du projet.....	5
3.3.1 Scrum Master :	5
3.3.2 Membre constituant l'équipe :	6
3.3.3 Product Owner :	6
3.3.4 Responsable de communication :	6
3.3.5 Responsables pédagogiques	6
3.4 Backlog :	7
3.5 Besoins en Formation :	7
3.6 Ressources logiciels et environnement matériels :	7
3.6.1 Progiciel de Gestion de Projet	8
3.6.2 Atelier de génie logiciel	8
4. Processus de développement Logiciel :	8
4.1 Phase de développement :	8
4.1.1 Analyse de besoins Logiciel :	8
4.1.2 Conception préliminaire :	8
4.1.3 Conception Détaillés :	8
4.1.4 Implémentation :	9
4.2 Développement mené par les tests :	9
4.3 Intégration Continue :	9
5. Pilotage du projet :	10
5.1 Processus itératif et incrémental :	10
5.2 Planification : Plan de phases	10
5.3 Choix de priorisation :	10
5.4 Réunion de suivi.....	10
5.5 Gestion des risques.....	11
5.6 Tableau récapitulatifs des livrables.....	11

1. Objet du document :

Ce document a pour objectif d'indiquer d'une manière précise comment sera piloté le projet VAL, il s'agit de définir aussi la méthode de conduite de projet utilisée, de traiter les modèles de développement.

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 et après chaque réunion de méthodologie au cas où il y a des remarques.

Le plan de développement contient l'ensemble des informations destinées à la direction du projet :

- Organisation de l'équipe
- Organisation du projet
- Développement du projet
- Pilotage du projet

2. Présentation 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 finalisation de ce projet doit avoir lieu à la fin du mois de mai 2013.

3. Organisation du projet :

3.1 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

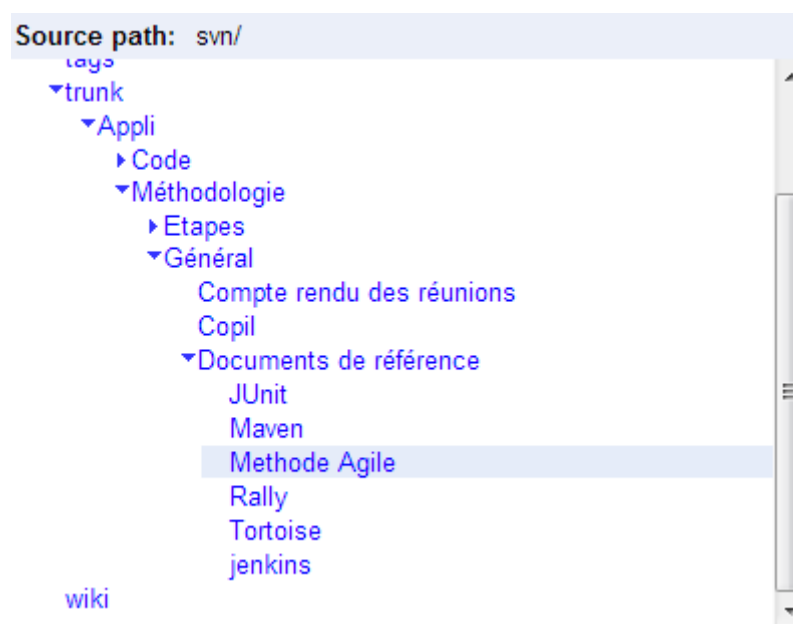
En Plus des choix de pratiques agiles, le projet va adopter certains principes des processus unifiés (UP) incluant une documentation bien détaillée au niveau des spécifications techniques

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

3.2 Formalisme, Documentation

Afin d'assurer la maîtrise de la méthode choisie et d'assurer une bonne gestion du projet, toute l'équipe s'est documentée sur cette méthode.

Afin d'assurer l'accès à la documentation en tout moment, nous avons créé un répertoire sur Google code auquel nous accédons via l'outil TortoiseSVN, où on met ces documents sous une arborescence bien définie comme suit :



Chaque livrable doit être sous format de Template Word ([lien](#)) 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.

Les numéros de version des documents sont définis sur 3 chiffres sous la forme **X.X.X**, la première version de chaque document est R(Release).V(Version).I(Itération), le premier numéro à droite est incrémenté par 1 après chaque livraison en fin d'itération. Les livraisons qui seront faites pour le ProductOwner et/ou lors des réunions de méthodologie de conduite de projet devront incrémentés le deuxième numéro à droite par. Les livraisons de fin de phases devront incrémentés le troisième numéro à droite par 1 celui du milieu aura la valeur 0 et le numéro à droite aura la valeur 1(ex : 1.0.1).

3.3 Matrice rôle du projet

3.3.1 Scrum Master :

La responsabilité de Scrum Master sera tenue à tour de rôle par les membres de l'équipe projet. Le rôle du Scrum Master est de :

- Assurer l'affectation et le suivi des tâches 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 tâches du projet.

Releases	Itérations	Date Itération		Durée	Scrum Master
R0	R0 itération	15/11/2012 → 07/12/2012		23 jours	Khaoula khaiter
	Comité de pilotage n° 1 : 5/12/2012				
R1	itération 1	08/12/2012 → 22/12/2012		15 jours	Saoudi Hamza
	itération 2	23/12/2012 → 06/01/2013		15 jours	Sibel Ozbey
	itération 3	07/01/2013 → 21/01/2013		15 jours	Nidal Mahraz
	itération 4	22/01/2013 → 01/02/2013		11jours	Hamza Lazrak
	Comité de pilotage n° 2 : 30/01/2013				
R2	itération 5	02/02/2013 → 16/02/2013		15jours	Driss Krafess
	itération 6	17/02/2013 → 03/03/2013		15 jours	Soukaina Merrouche
	itération 7	04/03/2013 → 18/03/2013		15 jours	Rafik bouttaba
	itération 8	19/03/2013 → 30/03/2013		12 jours	Sabri Jadoui
	Comité de pilotage n° 3 : 28/03/2013				
R3	itération 9	31/03/2013 → 14/04/2013		15 jours	Patrick Codo
	itération 10	15/04/2013 → 29/04/2013		15 jours	Khaoula Khaïter
	itération 11	30/04/2013 → 14/05/2013		15 jours	Saoudi Hamza
	itération 12	15/05/2013 → 22/05/2013		8jours	Sibel Ozbey
	Démon finale / Soutenance : 22 au 23 mai 2013				

	ESIAG 2012_2013 Plan du Projet Version 0.2.0
---	---

Le projet étant pédagogique nous avons choisi la durée de 15 jours pour chaque itération afin de s'assurer que chaque membre de l'équipe soit scrum master au moins une fois au cours de la réalisation du projet dans les délais mentionner si dessus

3.3.2 Membre constituant l'équipe :

Prenom et Nom	Adress e_mail	Livrable
Khaoula Khaïter	Khaoula.khaïter@gmail.com	Plan du Projet
Hamza Lazrak	ham.lazrak@gmail.com	Support de formation
Driss Krafess	krafess.driss@gmail.com	Benchmarking
Nidal Mahraz	nidalmahraz@gmail.com	Use Case
Soukaina Merrouche	soukaina.merrouche@gmail.com	Analyse de Risque
Sibel Ozbey	sibelozbey2010@gmail.com	Benchmarking
Patrick CODO	patrick.codo@yahoo.fr	Support de formation
Hamza Saoudi	saoudi.hamza00@gmail.com	Prototype d'architecture Logiciel
Rafik Boutaba	bouttaba.rafik@gmail.com	Vision Doc
Sabri Jadoui	sabri.jadoui@gmail.com	BackLog

3.3.3 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 :

- Familiarisation avec le milieu des systèmes distribués et les Framework
- capacité de prendre des décisions.
- Disponible pour répondre aux questions de l'équipe

Dans l'attente d'une repense favorable de plusieurs piste, on s'est mis d'accord qu'un membre de l'équipe jouera le rôle du ProductOwner.

3.3.4 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.

3.3.5 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	22 au 23 mai 2013

3.4 Backlog :

Pour le découpage des Release :

Pour la Release 1 : nous avons commencé par les fonctionnalités les plus urgentes à savoir les bouchons(MOCK) qui nous serviront d'outils pour l'avancement du projet. Certains seront remplacés par des vrais d'autres des mocks tout au long du projet.

Concernant les worltems:

Nous avons procédé à un découpage très fin pour s'assurer que tout le monde puisse travailler d'une manière individuelle.

3.5 Besoins en Formation :

Suite au benchmarking des technologies jugées importantes pour le PDS, nous avons remarqué la présence de technologies dont certaines sont enseignées au sein du master ISIDIS telles que les web services et MAVEN. Néanmoins, des tutoriels expliquant ces derniers ont été mis à disposition des membres de l'équipe via l'outil TortoiseSVN. Aussi, une formation est prévue avec Mr. Redouane Qarra dans le but de bien maîtriser ActiveMQ sachant qu'il est le MOM (middleware Oriented Message) choisi pour la communication avec le composant RTDRS réalisé par nos collaborateurs ISIAD.

Sachant que presque la majorité des membres de l'équipe n'ont jamais utilisé certains outils tel que l'outil VersionOne de gestion de projet, l'outil Jenkins d'intégration continue, outil Sonar de qualité de code, des tutoriels qui décrivent l'utilisation de ces derniers ont été mis à disposition des membres de l'équipe à travers TortoiseSVN.

3.6 Ressources logicielles et environnement matériels :

Les ressources logicielles que nous utilisons dans le cadre du projet permettent d'améliorer le développement. Chaque machine des membres du groupe doit impérativement utiliser un client TortoiseSVN et aussi tous les membres de l'équipe doivent installer les mêmes versions des outils afin d'assurer une homogénéité de travail.

- Environnement de test : Il sera destiné au ProductOwner, et dans le cadre de notre développement itératif l'environnement de test va pouvoir détecter à temps les bugs et apporter des corrections.
- Environnement de développement : c'est l'environnement Cible de l'intégration continue. , il semble que l'outil le plus approprié car le plus simple à prendre en main est Jenkins qui est un outil open source.

- Postes développeurs : On va utiliser les postes personnels des membres de notre équipe et les postes qui sont mis à disposition par l'ESIAG.

3.6.1 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...

3.6.2 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

4. Processus de développement Logiciel :

4.1 Phase de développement :

4.1.1 Analyse de besoins Logiciel :

Après avoir fait le benchmarking des différentes technologies qu'on aura à utiliser tout au long de notre projet, il a résulté une multitude de technologies à utiliser.

En ce qui concerne le langage de programmation, on a comparé les langages les plus utilisés et on a trouvé que Java est le plus adéquat pour ce projet. Ensuite, on a testé les protocoles de communication et on a conclu que ce sont les web services qui nous serviraient le plus.

Par rapport aux bases de données, on a conclu d'après notre étude que MySQL est le système de gestion des bases de données à utiliser.

Windows s'est avéré être le système d'exploitation sur lequel on devra faire tourner notre projet et le Conteneur Web à utiliser est Tomcat.

4.1.2 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.

4.1.3 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.

4.1.4 Implémentation :

Le processus d'implémentation sera la dernière étape de cycle de vie d'un US

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

4.2 Développement mené par les tests :

La solution finale du projet de synthèse doit être livrée à la deuxième semaine du mois de mai 2013, donc cette solution doit être vérifiée en permanence

- Les tests unitaires :
Vont être préparés par la personne en charge de chaque US
- Les tests de montée en charges :
Ce type de teste sera utilisé uniquement sur les modules qui nécessite un équilibrage de charge.

4.3 Intégration Continue :

Le projet doit impérativement respecter la norme d'intégration continue. L'intérêt est d'avoir une intégration itérative permettant des vérifications continues évitant des régressions au fur et à mesure de l'avancement.

Un avantage majeur est un travail proactif sur la résolution des problèmes survenant à l'intégration au lieu d'être réactif en dernière minute.

Cela nous permettra également d'avoir à disposition un build opérationnel à tout moment du projet.

	Avantages notables	Inconvénients notables
Jenkins	1) La multitude de plugins existants 2) Très répandu -> beaucoup d'informations sur internet 3) Open-source 4) Intégration avec .NET 5) Intégration avec Sonar	

Pour utiliser cette norme, nous devons choisir un serveur d'intégration, nous avons opté pour Jenkins par ce que c'est l'outil le plus optimal pour notre solution.

Nous devons également planifier la réalisation des build du serveur d'intégration continue. Pour cela, nous choisissons de réaliser deux types de build :

- Le build continu : Il se déclenche à chaque « commit » d'un membre du groupe. Il exécutera la compilation des codes sources et des tests associés.

- Le build complet : Il est déclenché de façon manuelle, mais de façon fréquente (au minimum 3 fois par semaine). Il réalise les mêmes tâches que le build continu mais en y ajoutant la génération de rapports et l'exécution des tests d'acceptance.

5. Pilotage du projet :

5.1 Processus itératif et incrémental :

Pour éviter l'effet tunnel, le projet fonctionnera en cycle itératifs et incrémentaux, sachant que nous avons des comités de pilotage et des réunions fréquentes avec le ProductOwner pour lesquelles nous devons faire des démonstrations, ce processus sera le plus adapté au contexte de travail du projet

5.2 Planification : Plan de phases

La solution finale du produit doit être prête vers la deuxième semaine de mai 2013, les comités de pilotage vont correspondre à près de 2 jours avant les échéances des Releases pour nous laisser le temps de rectifier les erreurs commises et prendre en considération les aux remarques qu'on aura lors des comités, donc nous aurons 4 Releases tout au long de la durée de vie du projet, chaque Release contiendra des itérations qui ont une durée de 15 jours .

5.3 Choix de priorisation :

Les « Work Item » spécifiés dans le Backlog du projet seront triés de manière à ce que les fonctionnalités les plus risquées et plus critiques soient traitées au début, ce qui veut dire que les fonctionnalités qui sont à l'entête du Backlog seront traitées en premier.

5.4 Réunion de suivi

Les réunions s'effectuent de manière régulière en moyenne de deux fois par semaine, et en début et fin de chaque Itération ou Release. Le Scrum Master définit la date et l'heure de la réunion, et c'est lui aussi qui prépare de l'ordre du jour des prochaines réunions, en gros le Scrum Master:

- Décide de la date des réunions.
- Prépare le planning des prochaines réunions ainsi que leur durée.
- Partage l'ordre du jour soit via l'outil TortoiseSVN ou sur le groupe que nous avons créé sur « Google Groups » dont l'adresse est fatal-pds@googlegroups.com et c'est un groupe que nous avons créé au début de la Release 0, le Scrum Master c'est assuré d'y ajouter tous les

adresse des membres du groupe. On utilise Google groups comme moyen de communication entre les membres de l'équipe hors des réunions.

- Prépare les comptes rendu des réunions et met ces derniers dans un répertoire créé sur Google code auquel nous accédons via l'outil TortoiseSVN.

Sans oublier qu'on effectue quotidiennement des Stand up Meeting ces réunion en pour but de faire une mise à jour rapide de l'état d'avancement des livrables, la durée de ces réunions dure en moyenne dix minutes, le Scrum Master lui s'assure que tous les membres de l'équipe soient présent.

5.5 Gestion des risques

Un document de gestion des risques du projet est être mis en place ([Lien envoyant vers le document Analyse des risques](#)) 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 du projet vont être réévalués, au début de chaque Release.

5.6 Tableau récapitulatifs des livrables

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

5.6.1.1.1 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

	ESIAG 2012_2013 Plan du Projet Version 0.2.0
---	---

Une plateforme d'intégration complète + Support de formation	-	1 + 1 si l'effectif le permet
--	---	-------------------------------