

Plan de management projet

Théo Delmas
Lauric Teyseyre
Pierre-Louis Renon
Julien Wattier

Université Paul Sabatier
Master Informatique 1

Table des matières

1	Objectif du document	3
2	Contexte	3
2.1	Contexte	3
2.2	Objectifs	3
3	Plans	3
3.1	Scope management plan	3
3.1.1	Définition de la scope baseline	3
3.1.2	Validation de la scope baseline	3
3.2	project baseline	3
3.3	Requirement management plan	4
3.3.1	Backlog produit	4
3.3.2	Capture des besoins	4
3.3.3	Gestion des changements dans les besoins	4
3.4	Schedule management plan	5
3.5	Quality management plan	5
3.6	Ressource management plan	5
3.6.1	Entrainement de l'équipe	5
3.7	Communication management plan	5
3.8	Risk management plan	5
3.8.1	Référencement des risques	5
3.9	Stakeholder engagement plan	5
3.9.1	Procédé d'identification des parties prenantes	5
3.9.2	Référencement des parties prenantes	5
3.10	Change management plan	6
3.10.1	Référencement des requêtes de changement	6
3.10.2	Traitement des requêtes de changement	6
3.10.3	Émission des requêtes de changement	6
3.11	Configuration management plan	6
3.12	Scope baseline	7
3.13	Schedule baseline	7

1 Objectif du document

Ce document a pour objectif d'exposer les méthodes de management appliqué à ce projet.

2 Contexte

2.1 Contexte

FlopEDT! est une application permettant de créer un emploi du temps satisfaisant une série de contraintes basées sur la programmation linéaire. Cependant la gestion actuelle de ces contraintes est relativement technique et peu documentée.

De plus les principaux collaborateurs du projet ont initié un découplage entre le back-end et le front-end, les deux parties étant actuellement gérées par un unique serveur.

2.2 Objectifs

L'objectif est de créer une interface intuitive permettant de rechercher des contraintes à ajouter à la génération d'un emploi du temps, tout en y intégrant la documentation des contraintes, le tout en respectant le contexte de découplage front/back de l'application.

3 Plans

3.1 Scope management plan

3.1.1 Définition de la scope baseline

Les produits du projet ont été définis lors de la kick-off meeting et sa préparation.

Ils sont référencé dans la scope baseline.

Au fur et à mesure du projet, si de nouveau produits apparaissent, ils seront ajoutés à la WBS via une demande de changement interne.

3.1.2 Validation de la scope baseline

Lorsque le document sera rédigé, une demande de validation sera envoyé aux parties prenantes.

3.2 project baseline

Description du cadre

Liste des livrables == WBS en agile == sprint

Critère d'acceptation

Exclusion

3.3 Requirement management plan

Les besoins sont suivis grâce à un backlog produit. Celui-ci se décompose en deux parties. La première définit les principaux produits du projet et permet aux parties prenantes de comprendre sur quelle produit l'équipe travaille actuellement. Les produits ayant été défini lors de la kick-off meeting.

3.3.1 Backlog produit

La seconde est un backlog lié au produit en cours. L'ensemble des besoins y sont retranscrit et décrit grâce à des user stories sur lesquelles l'équipe fournisseur adjoint une liste de tâches à effectuer afin de satisfaire la user story.

Le backlog se divise en 5 colonnes :

- Idée floue : La user story nécessite d'être raffinée. Cette colonne permet de noter informellement les besoins des clients
- À faire : La user story a été définie mais pas choisie pour le sprint en cours.
- En cours : La user story a été choisie pour le sprint en cours.
- À tester : La user story est satisfaite mais nécessite une validation.
- Ok : La user story est terminée et testée.

Chaque user story est priorisée via sa position dans son couloir, et relativement aux besoins des clients.

3.3.2 Capture des besoins

Une première partie des besoins ont été capturée lors de la kick-off meeting et sa préparation.

Par la suite les besoins seront capturés et définis précisément au travers des revues de sprint.

Si il apparaît que des besoins sont attachés à un produit non identifié, alors la WBS sera mise à jour.

3.3.3 Gestion des changements dans les besoins

Les changements dans la liste des tâches à effectuer pour chaque besoin n'est pas tracée. En revanche l'ensemble des besoins annulés sont récupérables dans l'archive du kanban.

Chaque développeur peut manipuler le kanban sans autorisation particulière.

3.4 Schedule management plan

3.5 Quality management plan

3.6 Ressource management plan

Ce projet est sans ressources matériels.

3.6.1 Entraînement de l'équipe

L'entraînement de l'équipe se fait au fur et à mesure du développement, sans activité dédiée.

3.7 Communication management plan

3.8 Risk management plan

3.8.1 Référencement des risques

La liste des risques et opportunité est listée dans le registre des risques et opportunité. Chaque risque y est décrit par les champs suivants :

3.9 Stakeholder engagement plan

3.9.1 Procédé d'identification des parties prenantes

Une première phase d'identification a été menée lors de l'élaboration de la charte de projet.

Par la suite, les parties prenantes seront potentiellement identifiées au fur et à mesure que des besoins apparaissent. Auquel cas l'équipe fournisseur émettra une demande de changement.

3.9.2 Référencement des parties prenantes

La liste des parties prenante est listée dans le registre des parties prenantes. Chaque partie prenante y est décrite par son rôle, ses besoins principaux, son niveau d'engagement, ainsi que son niveau d'influence sur le projet.

Le niveau d'engagement représente à quel point la partie prenante est impliqué dans le projet. Il influence notamment le niveau de communication qui doit être maintenu avec la partie prenante. Les niveau d'engagement possible sont les suivants :

- Ignorant : La partie prenante n'est pas au courant du projet.
- Résistant : La partie prenante est au courant du projet et s'y oppose.
- Neutre : La partie prenante est au courant du projet mais n'y accorde pas d'importance.
- Favorable : La partie prenante est au courant du projet, et le supporte.
- Impliqué : La partie prenante est au courant du projet, et s'implique dans sa réalisation

Les niveau d'influence possibles sont les suivants et représente à quel point la partie prenante peut imposer son pouvoir de décision sur le projet.

- Faible : La partie prenante n'a pas ou peu de pouvoir de décision sur le projet.
- Moyen : La partie prenante peut influencer dans une certaine mesure le projet.
- Fort : La partie prenante a un fort pouvoir de décision.

3.10 Change management plan

3.10.1 Référencement des requêtes de changement

L'ensemble des requêtes de changement sont listés dans le document requêtes de changement. Elles y sont décrite par les champs suivants :

- Titre de la requête : Description succincte de la requête.
- Date d'émission.
- Émetteur de la requête.
- Domaine de connaissance impacté.
- Description : quelle modification est demandé.
- Date de traitement.
- Responsables de la requête : Qui a appliqué la réponse à cette requête.
- Réponse : ensemble des modifications ayant été effectué après traitement.

3.10.2 Traitement des requêtes de changement

Les requêtes sont traitées en priorité par l'équipe de fournisseur, idéalement juste après leur émission. Le responsable de la requête veillera a completé les champs de la requête traité.

3.10.3 Émission des requêtes de changement

Tout événement donnant lieu à une modification des informations relatives à ce projet produira une demande de changement. De fait certaines partie prenante de part leur niveau d'influence vont naturellement produire ce genre d'événement. L'équipe fournisseur veillera donc à produire les requête associée.

Dans le cas d'événement impactant plusieurs domaines de connaissances, une requête par domaine devra être produite.

3.11 Configuration management plan

Les produits de ce projet seront tracés à l'aide de git, sur le dépôt officiel de FlopEDT. D'autre part le plan de projet est également tracé à l'aide de git et consultable sur le dépôt distant.

Dans les deux cas, chaque développeur dispose des droits et responsabilités sur les actions qu'il entreprend relativement à ces dépôts.

3.12 Scope baseline

3.13 Schedule baseline