Plan de management projet

Théo Delmas

Lauric Teysseyre

Pierre-Louis Renon

Julien Wattier

Université Paul Sabatier

Master Informatique 1

Table des matières

1 Objectif du document

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

1.1 Contexte du projet

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.

1.2 Objectifs du projet

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.

2 Plans de gestion

2.1 Approche de développement

2.1.1 Workflow git

Le développement suivra le workflow git. Il y aura donc une branche maîtresse intégrant les releases (artefact d’un sprint), une branche de développement intégrant les features en cours de développement et des branches de features correspondant au traitement d’un besoin.

2.2 Plan de gestion du cadre

2.2.1 Définition du cadre de base

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

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

2.2.2 Validation du cadre de base

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

2.3 Cadre de base

Le projet se tient sur la période fixe du 05/01/2023 au 28/04/2023.

Il a pour objectif de produire les produits listés dans le product breakdown structure.

2.3.1 Work breakdown structure

La nature agile du projet impose que le work breakdown structure corresponde à la Liste des livrables.

2.3.2 Product breakdown structure

Le product breakdown structure défini l'ensemble des produits que doit produire ce projet. Il est exposé dans le document dédié.

Chaque produit y est décrit ainsi que les sous-produits le composant.

2.3.3 Liste des livrables

La nature agile du projet impose que les livrables soient les artefact issues d'un sprint. Relativement à la durée d'un sprint définie dans le Plan de gestion du calendrier, la liste des livrables est la suivante :

- Artefact du sprint 0 (05/01/2023 - 03/02/2023)
- Artefact du sprint 1 (09/02/2023 - 24/02/2023)
- Artefact du sprint 2 (02/03/2023 - 10/03/2023)
- Artefact du sprint 3 (09/03/2023 - 17/03/2023)
- Artefact du sprint 4 (23/03/2023 – 31/03/2023)
- Artefact du sprint 5 (06/04/2023 – 14/04/2023)
- Artefact du sprint 6 (20/04/2023 - 28/04/2023)

2.3.4 Critère d'acceptation

Chaque livrable est présenté au client lors d'une revue de sprint en fin de sprint. Durant cette réunion le client accepte généralement l'artefact produit et émet éventuellement des demandes de modification.

Il peut néanmoins refuser l'artefact si ce dernier ne correspond pas à son besoin. Auquel cas l'équipe repartira du dernier artefact accepté par le client pour le sprint suivant.

2.4 Évolution du cadre

La période couverte par le projet est immuable. De fait la liste des livrable ainsi que le WBS ne peuvent changer. En revanche le product breakdown structure peut changer.

Lors de la Capture des besoins, si il apparaît qu'ils sont attachés à un produit non identifié, alors le product breakdown structure sera mis à jour.

2.5 Plan de gestion des besoins

2.5.1 Référencement des besoins

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 prenante de comprendre sur quel produit l'équipe travaille actuellement. Les produits ayant été défini lors de la kick-off meeting.

La seconde est un backlog lié au produit en cours. L'ensemble des besoins y sont retranscrits et décrits 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 du 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.
- A 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 du client.

2.5.2 Capture des besoins

Une première partie des besoins ont été capturé 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.

2.5.3 Gestion des changements dans les besoins

Les changements dans la liste des tâches à effectuer pour chaque besoin ne sont pas tracés.

En revanche l'ensemble des besoins annulés sont récupérables dans l'archive du kanban.

Chaque développeur peut manipuler le kanban sans autorisation particulière.

2.6 Plan de gestion du calendrier

Le manque d’expérience de l’équipe l’empêche d’estimer raisonnablement la durée des activités. De fait les activités n’auront pas d’estimation de durée.

La durée des sprint est fixée à deux semaines.

2.7 Plan de gestion de la qualité

2.7.1 Outil de qualité de code pour Javascript

Le code produit est soumis à l’outil Prettier.

2.7.2 Documentation

Le code produit est commenté, idéalement par des outils dédié lorsque c’est possible (JSDoc).

2.7.3 Revue de code

Le code produit par les fournisseurs est soumis à des revues de code par le reste de l’équipe au moment où moment où une merge request est demandée.

2.8 Plan de gestion des ressources

Ce projet est sans ressources matérielles.

2.8.1 Gestion de l’équipe

2.8.1.1 Réunion d’équipe

Chaque journée de travail commence avec une daily meeting où chaque fournisseur expose le travail réalisé lors de la dernière journée, les difficultés rencontrées ainsi que le travail qu’il compte effectuer dans la journée.

En fin de sprint l’équipe effectue une rétrospective afin d’analyser ses méthodes de travail et les adapter au besoin.

2.8.1.2 Communication au sein de l’équipe

Les fournisseurs se retrouvent en vocal sur un serveur discord.

2.8.1.3 Entraînement de l'équipe

L’entraînement de l'équipe se fait sur le tas, sans activité dédiée.

2.9 Plan de gestion des communications

Seules les parties prenantes ayant un niveau d'engagement "impliqué" bénéficieront de communications directes, au travers des revues de sprint auxquelles elles seront invitées. Ces revues seront tenues sur le serveur discord du projet.

D’autre part des réunions informelles pourront être tenues sur le serveur discord du projet FlopEdt, ainsi que des discussions textuelles.

L’ensemble des réunions par visioconférence seront enregistrées par l’équipe de fournisseurs.

Le reste des parties prenantes pourra néanmoins bénéficier d'informations en consultant les différents dépôts distants (voir le Plan de gestion de configuration).

2.10 Plan de gestion des risques

Ce projet n'a pas pour vocation d'être piloté par les risques. De fait cet aspect ne sera que faiblement géré.

2.10.1 Référencement des risques

La liste des risques et opportunités est listée dans le registre des risques et opportunités.

Chaque risque y est décrit par les champs suivants :

- Titre du risque.
- Description du risque.
- Risque d’occurrence : sur une échelle faible - moyen - fort.
- Niveau d'impact : sur une échelle fable - moyen - fort
- Plan d'action : actions à mettre en place si le risque se déclenche ou est sur le point de se déclencher.

S'y ajoute les champs suivant si il y a déclenchement du risque :

- Date de déclenchement.
- Contexte de déclenchement.

2.10.2 Identification des risques

Les risques ont été identifiés lors de l'élaboration de la note de cadrage.

Si, lors de la capture d'un besoin, un risque apparaît comme évident aux fournisseurs, il sera ajouté dans le registre. Mais l'équipe ne dédiera pas particulièrement de temps à l'identification des risques.

2.10.3 Réponse aux risques

En cas de déclenchement d'un risque, l'équipe fournisseur suivra le plan d'action prévue dans le registre. Elle veillera également à remplir les champs associés au déclenchement dans le registre.

Dans le cas de certains risques relativement générique, l'équipe remplira plusieurs fois les champs de déclenchement.

2.11 Plan de gestion des parties prenantes

2.11.1 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 impliquée dans le projet. Il influence notamment le niveau de communication qui doit être maintenu avec la partie prenante.

Les niveau d'engagement possibles 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 représentent 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.

2.11.2 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.

2.12 Plan de gestion du changement

2.12.1 Référencement des requêtes de changement

L'ensemble des requêtes de changement sont listées dans le registre des requêtes de changement.

Elles y sont décrites 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 demandée.
- Date de traitement.
- Responsables de la requête : Qui a appliqué la réponse à cette requête.
- Réponse : ensemble des modifications ayant été effectuées après traitement.

2.12.2 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 à compléter les champs de la requête traitée.

2.12.3 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 parties prenantes de part leur niveau d'influence vont naturellement produire ce genre d’événement.

L'équipe fournisseur veillera donc à produire les requêtes associée.

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

2.13 Plan de gestion de configuration

Les produits de ce projet seront tracés à l'aide de git, sur le dépot officiel de FlopEDT, et en particulier sur la banche catalogDev.

D'autre part le plan de projet est également tracé à l'aide de git et consultable sur ce dépôt distant.

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