Post Mortem Report

Equipe Projet

Axel BERNARD
Moussa BERTHE
Yao KONAN

SOMMAIRE

I. Causes d’échec du projet

A la fin de notre travail, l’étude du backlog de notre projet, nous montre quelques causes potentielles qui pourraient expliquer le non aboutissement du projet notamment :

1. Une Mauvaise Estimation des Durées :

En effet, dans plusieurs cas lors de la répartition des durées sur les différentes tâches des sprints, nous avons observé à la fin de celles-ci qu’il y a un déphasage entre la durée estimée et la durée réelle mise pour la réaliser. Par exemple, dans le :

- Sprint 1: La tâche "Establish a mechanism to initiate consumption tracking" a été estimée à 4h30 mais a pris 14h. Cette différence significative s’explique par une mauvaise estimation initiale et des complications imprévues, cette tâche particulièrement a d’ailleurs constitué notre plus grand problème à la réalisation de ce projet.

2. Défis techniques et imprévus:

Durant tout le long de la réalisation du projet nous avons été confrontée à beaucoups de difficultés techniques et certains gros imprévus comme:

- Sprint 3 : La tâche “Obtain the different time range of the different stages” nous avons eu beaucoups de mal à récupérer les données de consommation des processus au sein d’un build avec PowerAPI.
- Sprint 1 : Les difficultés techniques rencontrées pour établir le mécanisme de suivi de la consommation montrent que nous n'étions pas préparé à certains aspects techniques du projet.
- Sprint 3 : La tache "Set up the graphical display" a connu un gros bug avec Javascript qu’on ne percevait pas. Les valeurs ne posaient pas problème, le problème venait en fait des labels dû à une spécificité de Javascript.

3. Mauvaise visualisation des différentes choses à faire pour atteindre l’objectif :

Nous avons rencontré des problèmes et en plus de cela, nous avons réalisé certaines tâches qui n’ont pas été intégrées par la suite dans le projet. Cela est dû au fait que du fait d’avoir dû changer de manière par rapport à ce qui était prévu nous avons eu du mal à clairement identifier les différentes étapes pour la réalisation du projet. En voici des exemples :

- Sprint 3 : Les tâches “Install and configure InfluxDB” et “Develop a database integration for data insertion and retrieval” n’ont pas été intégrés dans dans le projet à suite. Cela s’explique par le fait que nous avons trouvé une autre manière pour conserver l’historique via une fonctionnalité de Jenkins que nous n’imaginions pas au début.
- Sprint 1 : Plusieurs tâches du sprint 1 n’ont pas pu être mobilisé par la suite tel que notamment l’utilisation d’un script bash que nous avons remplacé par du code java.

4. Problèmes de coordination et de répartition des tâches :

- Plusieurs fois dans les sprint 1 et 2 les personnes qui ont réalisé des tâches sont différentes de la personne à qui la tâche avait été attribuée. Pour le sprint 2 cela s’explique en partie par le fait que certaines choses du sprint 1 n’avaient pas été réalisées comme on voulait et donc cela a dû mobiliser quelqu’un lors du sprint 2 qui ne pouvaient donc pas faire les tâches qui lui avaient été initialement attribuées pour ce sprint. Aussi de manière générale (sprint 1 comme sprint 2) ce constat vient du fait qu’il arrivait que lorsque quelqu’un éprouvait des difficultés pour sa tâche d’autres pouvaient venir l’aider ou le remplacer ce qui cause des problèmes pour l’organisation et pour le respect du temps consacré à la tâche en question.

5. Surcharge de Travail pour Certains Membres de l'Équipe :

- Cette réatribution des tâches après qu’elles aient été assignées (souvent par dépit) a causé plusieurs problèmes :
    - Cela crée une irrégularité dans le travail fourni et donc une mauvaise utilisation du temps total imparti (temps imparti x nombre de membres) ce qui nuit à l’avancement.
    - De plus cela cause le dépassement de la durée définie pour la tâche car on se retrouve à avoir à compter à la fois le temps de la personne qui a commencé le tâche + le temps de la personne qui l’a reprise.
    - Ce phénomène est aussi causé par le fait d’avoir une personne qui restait sur l’étude de l’intégration de PowerAPI.

Ceci réduit donc l'efficacité globale.

6. Environnement Jenkins

- Nous avons constaté lors du développement qu’il y avait plus de choses à maîtriser dans le développement Jenkins que ce que nous avions vu lors du sprint 0. Ce qui faisait qu’en même temps que développer les fonctionnalités nous devions continuer

à apprendre à manipuler les particularités de cet outil (Listeners, Actions, agents d'exécutions des builds, etc…)

II. Leçons apprises et comment prévenir ces problèmes

De ces causes nous tirons différentes leçons ainsi que certaines perspectives qui auraient pu améliorer le déroulement du projet :

1. Choses qui auraient pu être faites autrement

Notre plus grand problème a résidé dans le lien à faire entre PowerAPI avec Jenkins. Une piste pour éviter l'impact que cela a eu dans la réalisation de notre projet aurait pu être de se contenter de la solution alternatives que nous avions trouvé au sprint 1, cela aurait permis de consacrer plus de temps au développement de notre plugin mais aussi d'y avoir plus clair dans l'avancement de notre projet et dans les choses à réaliser (car ce point sur PowerAPI créait une grosse incertitude et on courrait le risque de devoir changer certaines choses de Jenkins en fonction de ce qui était trouvé pour l'utilisation de PowerAPI sans pour autant savoir à quels moments des solutions émergeraient). D'autant que la solution que nous avons finis par trouvé pour PowerAPI implique de créer des nouveaux Docker à chaque build Jenkins ce qui a plusieurs défaut :

- ce n'est pas pratique
- cela ralentit grandement le build
- cela consomme de l'énergie alors que justement notre projet vise à l'inverse

Cependant il est à noter que ceci n'était pas prévisible, il était tout à fait possible qu'une solution plus satisfaisante aurait émerger - d'où l'incertitude qui était à prendre en compte dans la décision de continuer à creuser PowerAPI au détriment de plus de temps accordé à Jenkins.

2. Leçons apprises pour la suite

Premièrement nous constatons qu'après le sprint 1 les temps prévu et temps consacrés réellement aux tâches étaient beaucoup plus précis, ce qui constitue déjà un point d'amélioration que nous avons eu lors de ce projet.
Une grande leçon que nous tirons de cette expérience est qu'il est essentiel d'avoir une organisation rigoureuse dès le début afin d'optimiser au maximum la totalité du temps consacré au projet.

Nous constatons aussi qu'il important d'adopter dès le début une stratégie efficace dans la manière d'aborder le projet et de judicieusement choisir les solutions à sélectionner quant aux difficultés rencontrées.
Il faudrait aussi pouvait respecter de manière plus assidue l'attribution des tâches afin de ne pas perturber l'organisation établie.