

Logiciel Auto Mix

Rapport bilan du kick-off

Auteurs : Maxime S., Guillaume H., Jordan E., Louis C., Pierre G.

Tuteurs : Baptiste Hemery, Estelle Cherrier

Table des matières

Introduction.....	2
1 Développement Agile.....	3
1.1 User stories et fonctionnalités	3
1.2 Evaluation continue du logiciel	3
2 Gestion des risques	4
Conclusion	4
Annexe 1	5

Introduction

Lors du kick-off meeting, les objectifs de notre projet ainsi que le principe de fonctionnement de notre logiciel ont été compris sans difficulté par notre auditoire, ce que nous considérons comme un succès. En effet, notre logiciel est destiné au grand public qui ne dispose pas forcément de connaissances en informatique ou dans le domaine musical.

Cependant, la manière dont nous allons nous organiser pour développer notre application n'a pas été perçue comme nous l'espérions. Cela est dû à notre formulation incorrecte et partiellement erronée des principes du développement agile, sur lesquels repose notre développement.

Nous avons revu nos tuteurs depuis, ce qui nous a permis de formaliser correctement les idées et ainsi s'assurer de la rigueur de notre propos et de la gestion que nous proposons.

La première partie de ce rapport consiste en l'explication de la manière dont nous allons procéder tout au long du projet. Cela passe également par la définition du produit de valeur minimale et des indicateurs qui nous permettront de nous situer dans le temps par rapport au travail restant. Dans un second temps nous reviendrons sur deux des risques que nous avons présentés.

1 Développement Agile

1.1 User stories et fonctionnalités

Nous avons défini l'ensemble (non exhaustif) des user stories de notre logiciel, c'est-à-dire tous les scénarios d'utilisation possibles. Les *user stories* sont par exemple : « L'utilisateur peut redimensionner la fenêtre du logiciel comme il le souhaite » ou encore « L'utilisateur peut choisir un dossier contenant ses musiques ».

A ces user stories nous avons associé les fonctionnalités correspondantes, ces fonctionnalités relèvent du domaine technique et ne seront abordées lors des réunions de fin¹ de *sprint* que si un problème a été rencontré lors du développement de ces dernières.

Ainsi lors de chaque réunion de fin de *sprint* nous choisirons les user stories à faire pour le prochain *sprint*. Ces réunions se dérouleront avec nos tuteurs, le chef de projet et éventuellement des membres de l'équipe.

1.2 Evaluation continue du logiciel

A partir de la liste, nous avons établi le produit de valeur minimal, cela correspond au logiciel minimal pour que nous puissions le distribuer auprès du public. Ce produit devra offrir à l'utilisateur la possibilité de générer un mix basique à partir d'une fenêtre graphique, qui s'adapte à la taille et à la résolution. Pour cela, il sera en mesure de sélectionner un dossier contenant les pistes audio et un chemin pour le fichier exporté.

Pour contrôler l'avancement du projet, nous avons opté pour la démarche suivante :

- On affecte à chaque *user story* un certain nombre de points, attribués en fonction de l'importance vis-à-vis du client.
- Ensuite, on trace 3 courbes théoriques entre le début du projet, i.e. aucune *user story* de faite, jusqu'au produit minimal, au produit intermédiaire et au très bon produit.
- A chaque réunion de fin de *sprint*, on calcule la somme des user stories effectuées puis on place un point sur le graphique.

On obtient alors le graphique présenté en **Annexe 1**.

Si durant plusieurs *sprints* la courbe réelle est au-dessus de la courbe théorique visée, alors on est en avance et on peut envisager la prise en compte de nouvelles *user stories* et placer un nouveau produit incluant ces changements sur le graphique. En revanche, si la courbe réelle est en dessous de la courbe théorique pendant plusieurs *sprints*, alors c'est qu'il y a un problème avec l'avancement du projet.

¹ Les réunions de fin de sprint feront également office de réunion de début de sprint.

2 Gestion des risques

Le kick-off a montré que nous avons mal formulé la manière dont nous allions gérer le manque de connaissance sur un sujet concernant le projet ou sur la difficulté de prise en main de la librairie de fonctions importée.

Nous avons jugé faible l'impact du manque de connaissance, à propos de l'intelligence artificielle notamment. Cela s'explique par le fait que nous avons l'intention d'être très réactif sur ce point et ne pas nous laisser dépasser par des difficultés théoriques. Il n'en demeure pas moins, qu'un tel blocage aurait un impact fort sur le projet.

Pour renforcer notre vigilance sur ces points, nous avons décidé que les membres de l'équipe feront remonter les problèmes théoriques rencontrés rapidement au chef de projet afin de prendre les dispositions appropriées dans les meilleurs délais. De plus, les réunions de fin/début de sprint, que nous avons présenté dans la partie 1, nous permettront de faire un point régulier sur les problèmes rencontrés et le *burn-up* nous donnera un indicateur sur le retard éventuel que nous aurions pris.

Conclusion

Les méthodes que nous avons mises en place devraient nous permettre de gérer correctement notre projet qui se veut ambitieux. Nous sommes une équipe de développement de cinq personnes et nous devons suivre avec rigueur les bonnes pratiques de suivi du projet mais également de génie logiciel pour permettre la réussite du projet.

Annexe 1

