Suivi de Planification du Projet

Toutes les fonctionnalités prévues pour cette itération ont été implémentées avec succès, notamment :

- **Lecture de morceaux** : L'application scanne un dossier pour récupérer les fichiers audio (.mp3, .flac, etc.), permettant leur lecture avec affichage de la durée totale et du temps écoulé.
- **File d'attente** : Implémentation d'une file d'attente FIFO pour gérer la lecture séquentielle des morceaux avec possibilité d'ajout, de suppression et de réorganisation.
- **Boutons de contrôle** : Ajout des boutons de lecture (lecture/pause, précédent, suivant) et d'un curseur pour naviguer dans le morceau ou ajuster le volume.
- **Contrôles divers** : Intégration des options avancées telles que la vitesse de lecture, le mode aléatoire et la balance des canaux audio.

Difficultés Techniques et Solutions

Durant cette itération, nous avons rencontré des difficultés notables liées à l'implémentation de la lecture des métadonnées des fichiers audio. Initialement, nous avions tenté une implémentation manuelle de cette fonctionnalité. Cependant, il s'est rapidement avéré que cette approche était excessivement complexe, très "low-level", et entraînait une charge de travail considérable.

Pour résoudre ce problème, et après validation par l'assistant responsable, nous avons décidé d'utiliser une librairie spécialisée existante - JAudioTagger. Ce changement a nécessité un refactoring conséquent du code existant, entraînant une perte de temps significative.

Impact sur la Planification

Ces difficultés techniques ont provoqué une perte de temps substantielle. De plus, la surestimation initiale de la charge de travail a entraîné une répartition inefficace des tâches, laissant certaines équipes avec relativement peu de travail durant une partie de l'itération.

Couverture des Tests

Actuellement, la couverture des tests unitaires n'est pas optimale. Nous sommes conscients de ce problème et avons prévu d'améliorer significativement cette couverture pour la prochaine itération.

Motivation des Choix de Conception

Le principal choix de conception adopté est le design pattern **MVC (Model-View-Controller)**, adapté à l'environnement JavaFX. Ce choix a été motivé par la familiarité de l'équipe avec ce pattern, ainsi que par sa fiabilité éprouvée et sa popularité dans le développement d'applications graphiques. Ce modèle facilite la séparation claire des responsabilités, améliorant ainsi la maintenabilité et la lisibilité du code produit.