**Suivi de planification et motivation des choix**

**Groupe05**

**Itération 1 :**

Lors de cette première itération, nous avons convenu avec le client de réaliser l’histoire 4 et le début de l’histoire 11.

**Implémentations des fonctionnalités :**

Toutes les fonctionnalités ont été implémentés. Par soucis technique et de temps, nous avons limité le nombre de site source à 3 en attendant la réunion avec le client qui décidera des sources dont il souhaite importer les articles.

**Difficultés rencontrées :**

**-**Temps d’adaptation avec les outils intelliji et git.

-Quelques difficultés rencontrés pour certains membres du groupe lors de la prise en main de JavaFX et maven.

-Nous avons décidé de récupérer un article ainsi que ses informations (auteur, titre…) sans utiliser de librairie. Nous avons implémenté un parser qui récupère les informations de l’article dans le code html du site donné 🡺 perte de temps à l’implémenter + parser non conventionnel à tous les sites. Ce mauvais choix a été commis par manque de documentation.

**Design pattern + librairies :**

Nous avons utilidé le pattern DAO pour faciliter l’accès à la base de données

Nous utilisons Sqlite pour gérer notre base de données, nous avons choisi Sqlite car elle est entièrement manipulable car l'intégralité de la base de données est **stockée dans un fichier.**

**Itération 2 :**

Lors de cette deuxième itération et après discussion avec le client concernant les sources des articles, nous avons convenu d’intégrer cinq sources et 10 catégories prédéfini au logiciel. Le client a souhaité que l’histoire 11 soit complètement fini ainsi que l’entièreté de l’histoire 1 et le début de l’histoire 12.

**Implémentations des fonctionnalités + difficultés rencontrées :**

Dans certains cas, certaines informations concernant l’articles sont absentes (auteur, mots clé). Dans ce cas, nous avons décidé de n’afficher que les informations disponible (description, source…).

La localisation d’un article n’est jamais donnée, nous discuterons avec le client concernant ce point afin de trouver une solution qui lui convient.

Nous avons rencontré des difficultés pour récupérer les vidéos des articles, nous n’avons pas trouvé de solution technique à ce problème. Nous avons pris l’initiative d’implémenter une nouvelle catégorie appelée vidéo qui permettra de visionner des vidéos facilement accessibles sur internet.

**Design pattern + librairies :**

Jsoup : librairie qui nous permet de parser plus facilement un code html.

Flux rss : nous permet de récupérer les informations d’un article.

observer observable : pour un rafraîchissement automatique de la vue lors du changement de la mise à jour des données

Singleton : usage dans le routeur, permet de changer de vue et se charge de la logique des transitions entre les vues.