

Rapport Sprint 4: Génie logiciel – Scrum

Temps donné total (3h00).

Objectif : Ajouter à la structure du fichier xml les parties auteur, introduction, corps, conclusion et discussion, puis ajouter un menu à afficher à l'exécution de la commande pour choisir les pdf à parser.

Product backlog :

1. Modifier la fonction `donne_xml` pour rajouter la partie auteur, qui était resté non réussi lors du sprint précédent, à la structure du fichier xml afin de rattraper le retard du Sprint 3 (*priorité haute, estimation 30 minutes*)
2. Modifier la fonction `donne_xml` pour rajouter les parties introduction, corps, conclusion et discussion à la structure du fichier xml (*priorité haute, estimation 1H*)
3. Ajouter le menu au début de l'exécution du système afin de choisir certains PDFs à parser (*priorité moyenne, estimation 1H15 minutes*)
4. Documentation complète du système dans le fichier README.md, incluant une explication générale et les instructions pour lancer le programme avec les différentes options disponible. (*priorité faible, estimation 15 minutes*)

1. Planification du Sprint:

Pour mener à bien cette implémentation, notre équipe s'est d'abord réuni pour discuter de l'objectif du sprint et des différentes exigences qui sont donnés pour y parvenir. Nous avons donc trouver les différentes tâches que nous devons réaliser et déterminer leur importances pour les lister hiérarchiquement (Product Backlog) puis nous nous sommes mis d'accord sur lesquelles nous allons pouvoir réaliser lors de ce sprint (Sprint Backlog). Ensuite, nous nous sommes pencher sur la planification du sprint, des rôles de chacun sur les différentes taches à exécuter que nous sommes sûr de terminer avant la fin sprint : Romain s'occupe d'abord de l'ajout de la partie auteur dans la structure du fichier xml que nous n'avions pas réussi au Sprint précédent (Sprint 3) afin de rattraper le retard et terminer pleinement les exigences du Sprint 3.

Enzo s'occupe de l'ajout des parties introduction, conclusion et discussion dans la structure du fichier xml.

L'ajout de la partie corps est assuré par Romain après avoir terminé sa 1^{er} tache.

Après avoir exécuté ces taches, nous allons réaliser des tests du programme sur différents textes de différents PDF et vérifier du bon fonctionnement de celui-ci, et si des problèmes surviennent, les résoudre.

Ensuite, nous allons passer à la réalisation du menu au lancement de la commande pour demander les pdf à parser et son intégration dans notre système. D'autres tests seront réalisés après cette partie.

Enfin le rapport est réalisé tout au long du sprint par Dylan.

2. Mise en œuvre du Sprint:

Scrum 1: réunion de 30 minutes pour définir les objectifs, les tâches et répartir les rôles de chacun. Nous avons discuter oralement à côté devant nos ordinateurs. Pour les rôles du début du Sprint, Romain s'occupe de modifier la fonction `donne_xml` pour ajouter la partie auteur à parser dans le pdf pour l'ajouter dans la structure du fichier xml et doit également tester sa fonction pour vérifier si des erreurs sont présentes et les résoudre s'il y en a. Enzo quant à lui s'occupe des parties introduction, conclusion et discussion à parser des pdf pour les ajouter dans la structure du fichier xml en modifiant la fonction `donne_xml`. Et enfin Dylan s'occupe de rédiger le rapport.

1^{er} étape : Romain réalise l'ajout de l'auteur parsé du pdf à la structure du fichier xml.

2^e étape : Enzo commence par ajouter l'introduction parsé du pdf à la structure du fichier xml.

Scrum 2: Au bout de 45H , soit 15 minutes après, nous nous réunissons rapidement pour discuter de l'avancement des taches. Du côté de Romain, l'ajout de la partie auteur est terminé et Romain a comme nouvelle tâche l'ajout du corps dans la structure du fichier xml. Enzo continue ses tâches comme convenu.

3^e étape : Romain est passé à l'ajout du corps parsé dans la structure du fichier xml.

Scrum 3: Au bout de 1H30, soit 45 minutes après, nous nous réunissons pour discuter des avancées. Romain a fini d'ajouter le corps parsé dans la structure du fichier xml. Enzo a fini la partie introduction et continue la partie conclusion et discussion. Dylan a essayé de prendre de l'avance en cherchant comment réaliser le menu. Romain et Dylan vont s'y occuper.

4^e étape : Enzo ajoute la conclusion et la discussion.

5^e étape : Romain et Dylan commence le menu.

Scrum 4: Au bout de 2H20 du Sprint, on se réuni pour discuter de l'avancement des taches avant les 30 minutes de la fin du Sprint: le menu est presque terminé il manque plus qu'à pouvoir écrire dans le terminal les pdf qu'on souhaite convertir et que le parseur lise notre choix et convertisse uniquement les pdf choisis. Enzo à terminé l'ajout de la conclusion et de la discussion pour la structure du fichier xml.

6^e étape : Romain et Dylan termine le menu.

7^e étape : Documentation du README et finalisation du rapport.

8^e étape : Rendu du Sprint.

3. Revue de Sprint:

Évaluation des Résultats:

- L'ajout et les tests des parties auteur, introduction, corps, conclusion et discussion sont terminés
- L'implémentation du menu est passé par l'ajout d'une liste créée et qui va contenir tout les fichiers du dossier passé en paramètre. Grace à cette liste nous pouvons convertir uniquement les fichiers désirés en entrant leur numero dans le terminal.
- L'affichage du menu est la sélection des pdf via le terminal est réalisé.

Démonstration: Présentation des fichiers sorties après extraction. (voir git : <https://github.com/RomainAllegre/Genie/tree/TP4>)

Rétrospective:

- Nous avons bien évalué la difficulté de rajouter les différentes parties à la structure du fichier xml qui nous ont pris beaucoup de temps lors de ce Sprint, et également de la réalisation du menu, et nous avons essayé d'estimer des temps de réalisation en fonction du temps disponibles pour réaliser le Sprint (3H).

4. Conclusion du Sprint:

Résultats:

- La fonction donne_xml permet de créer un fichier xml et organise le contenu sous la structure : article contenant preamble, titre, auteur, abstract, introduction, corps, conclusion, discussion et biblio.
- Le menu permet de convertir les fichiers pdf donné via le terminal sous forme de chiffre.
- Nous avons tout réalisé.

Les connaissances acquises :

- convertir uniquement certains pdf via un terminal.