

Rapport Sprint 3: Génie logiciel – Scrum

Temps donné total (3h00).

Objectif : Ajouter au parseur la possibilité de générer soit un fichier TXT soit un fichier XML structuré sous forme d'un article qui contient dans l'ordre : preambule, titre, auteur, abstract et biblio.

Product backlog :

1. Modification du main du fichier `convert_txt.py` pour pouvoir prendre en compte les deux différentes options (-t ou -x) pouvant être entrée lors de la commande :
"python `convert_txt.py` <chemin_dossier> [-t | -x]" qui permettra de choisir si l'on veut des fichiers de type texte ou XML. (*priorité haute, estimation 30 minutes*)
2. Création d'une nouvelle fonction "donne_xml" permettant de prendre un dossier de fichier et de les convertir en fichier XML en respectant la structure donné si l'option -x est entrée dans la commande, sous forme d'article avec le préambule, titre, auteur, abstract et biblio. (*priorité haute, estimation 45 minutes*)
3. Test des différents ajouts lors du sprint, tel que la possibilité de mettre une option pour choisir le type de sortie et vérifier si l'option est pris en compte par le programme, essayer les différentes options -t ou -x, vérifier les fichiers retourner par le programme et résoudre les potentiels problèmes d'exécution (*priorité moyenne, estimation 30 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 30 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 taches que nous devons réaliser et déterminer leur importances pour les lister hiérarchiquement puis nous nous sommes mis d'accord sur lesquelles nous allons réaliser lors de ce sprint.

Ensuite, nous nous sommes pencher sur la planification du sprint, des rôles de chacun sur les différentes taches à exécuter que nous sommes sur de terminer avant la fin sprint :

Romain s'occupe d'abord de la création d'une nouvelle fonction "donne_xml" permettant la création du fichier XML contenant le texte d'un fichier PDF en respectant la structure : article qui contient dans l'ordre : preambule, titre, auteur, abstract et biblio.

Enzo s'occupe de la modification du main du fichier `convert_txt.py` pour pouvoir prendre en compte les différentes options de sortie (-t) pour un fichier texte, ou (-x) pour un fichier xml grâce à la commande "`python convert_txt.py <chemin_dossier> [-t | -x]`". Après avoir exécuté ces 2 tâches, 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. Enfin le rapport est réalisé tout au long du sprint puis finalement rédigé par Dylan.

2. Mise en œuvre du Sprint:

Scrum 1: réunion d'1H pour définir les objectifs, les tâches et répartir les rôles de chacun. Nous avons travaillé sur framapad pour lister les différentes tâches afin de remplir en même temps le Product Backlog et le Sprint Backlog du rapport. Romain doit s'occuper de la fonction pour créer un fichier xml et Enzo doit modifier le fichier `convert_txt.py` pour intégrer la nouvelle option de fichier sortie. Dylan s'occupe de rédiger le rapport.

1^{er} étape : Romain s'occupe de la création de la fonction `donne_xml` et test la fonction.

2^e étape : Enzo modifie le main du fichier `convert_txt.py`.

Scrum 2: Au bout de 1H30 minutes, soit 30 minutes après, nous nous réunissons pour discuter de l'avancement des tâches : Enzo a fini de modifier le main et Romain a écrit une fonction `donne_xml` mais des problèmes surviennent lors de son exécution. Nous décidons donc de passer du temps dessus pour trouver une solution aux problèmes de la fonction.

3^e étape : Résolution commune des problèmes de la fonction `donne_xml`.

Scrum 3: Au bout de 2H, soit 30 minutes après, nous nous réunissons pour discuter de la résolution des problèmes terminés de la fonction `donne_xml` : La fonction peut bien créer un fichier xml mais il manque à rajouter dans l'article le preambule, l'auteur et la biblio. Il faut donc rajouter ces 3 éléments. Romain va s'occuper d'ajouter le preambule et l'auteur. Enzo va s'occuper d'ajouter la biblio.

4^e étape : Romain ajoute à la fonction `donne_xml` l'écriture du preambule à l'article du fichier xml et test son ajout.

5^e étape : Romain doit ajouter à la fonction `donne_xml` l'écriture de l'auteur à l'article du fichier xml et test son ajout.

6^e étape : Enzo ajoute à la fonction `donne_xml` l'écriture de la la biblio à l'article du fichier xml et test son ajout.

Scrum 4: Au bout de 2H45, on se réunit pour discuter de l'avancement : preambule et biblio sont bien ajoutés mais l'ajout d'auteur n'a pas réussi. Cependant les tests sont concluants pour afficher toute la structure sauf l'auteur. Au vu du temps qui nous reste, nous décidons de délaisser l'ajout de l'auteur et nous passons à documenter le README et finaliser le rendu du Sprint.

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

8^e étape : Rendu du Sprint.

3. Revue de Sprint:

Évaluation des Résultats:

- Le main du fichier convert_txt.py a bien été modifié pour ajouter l'option de choix d'une sortie en fichier txt ou xml.
- La fonction donne_xml a mis du temps à être fonctionnelle et les problèmes d'executions nous ont fait prendre du retard quant à notre planification de Sprint et nous avons du nous y pencher communément pour les résoudre.
- Nous n'avons pas réussi à terminé l'ajout de l'auteur pour la structure xml à respecter.
- Néanmoins le reste de la structure est présent : article qui contient preamble, titre, abstract, biblio et le choix du fichier de sortie lors de l'entrée de la commande s'exécute sans problème et le fichier de sortie (xml ou txt) est bien crée.

Démonstration: Présentation des fichiers sorties après extraction. (voir git : <https://github.com/RomainAllegre/Genie/tree/TP3>)

Rétrospective:

- Le retard de la fonction donne_xml nous a fait réfléchir sur notre planification des taches surtout l'estimation des temps de réalisation de chaque tache et nous devrions mieux calculer la difficulté des taches et d'appréhender les potentiels erreurs à résoudre pour pas avoir de mauvaise surprise la prochaine fois.

4. Conclusion du Sprint:

Résultats:

- La tache de modification du main est terminée, la création est terminée avec un élément en moins (auteur), les tests ont été exécutés, le rapport est fini, le git est complet avec des images du bon fonctionnement de la commande.

Les connaissances acquises :

- proposer un choix de fichier de sortie dans une commande : xml ou txt