



UNIVERSITÉ D'AVIGNON
ET DES PAYS DE VAUCLUSE

C E N T R E
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE



Licence Informatique

ILSEN

UE Génie Logiciel

Projet SCRUM: Séance 4-5

Groupe 2

Enzo GUENY
Adam SERGHINI
Jarod DURET

CERI - LIA
339 chemin des Meinajariès
BP 1228
84911 AVIGNON Cedex 9
France

Tél. +33 (0)4 90 84 35 00
Fax +33 (0)4 90 84 35 01
<http://ceri.univ-avignon.fr>

Encadrement

Juan Manuel TORRES MORENO

Sommaire

Titre	1
Sommaire	2
1 Introduction	3
2 Revue du Sprint	3
3 Rétrospective du sprint	3
4 Les sprints	3
4.1 Premier Sprint	3
4.2 Deuxième et dernier Sprint	4
5 Les mêlées quotidiennes	4
5.1 Première mêlée quotidienne	4
5.2 Deuxième et dernière mêlée quotidienne	4
6 Conclusion	5

1 Introduction

Ce document est un rapport sur l'avancement du projet et les étapes réalisées pour la séance du 11 jan. sur le projet **UAPV TP de Génie logiciel - Scrum**.

SCRUM Master (DURET Jarod).

Dépôt GitHub : -> https://github.com/Team-Rocket-CERI/SCRUM_Project

2 Revue du Sprint

Suite au sprint précédent, nous avons reprogrammé le parseur pour extraire les informations suivantes (sous cette forme) :

- `<article>`
 - `<preamble>` Le nom du fichier d'origine `</preamble>`
 - `<titre>` Le titre du papier `</titre>`
 - `<auteur>` La section auteurs et leur adresse `</auteur>`
 - `<abstract>` Le résumé de l'article `</abstract>`
 - `<biblio>` Les références bibliographiques du papier `</biblio>`
- `</article>`

Le résultat étant toujours autant satisfaisant (python et au rust), nous avons décidé de continuer sur cette lancée pour la dernière étape.

3 Rétrospective du sprint

Au fur et à mesure des sprints, nos méthodes de travail se précisent et s'améliorent. Les difficultés sont plus rapidement dépassées grâce à une bonne communication et cohésion de groupe.

4 Les sprints

4.1 Premier Sprint

Reprogrammation du parseur avec les informations suivantes à extraire (sous cette forme) :

- `<article>`
 - `<preamble>` Le nom du fichier d'origine `</preamble>`
 - `<titre>` Le titre du papier `</titre>`
 - `<auteur>` La section auteurs et leur adresse `</auteur>`
 - `<abstract>` Le résumé de l'article `</abstract>`
 - `<introduction>` La introduction `</introduction>`
 - `<corps>` Le développement du papier `</corps>`
 - `<conclusion>` La conclusion du papier `</conclusion>`
 - `<discussion>` La discussion du papier `</discussion>`
 - `<biblio>` Les références bibliographiques du papier `</biblio>`
- `</article>`

La majorité du code sera encore rédigée en Python, nous conservons cependant le programme `pdf_to_text` utile dans le main codé en rust. Ce script, préalablement réalisé, permet de parser une première fois le document dans un `.txt` provisoire. Le script python `"parser.py"` sera ensuite chargé d'extraire les informations utiles et les organiser dans un nouveau fichier XML.

4.2 Deuxième et dernier Sprint

Suite au brainstorming de la deuxième mêlée quotidienne vînt le moment de procéder à l'implémentation. Comme décrit précédemment d'interface terminale choisie fût implémentée.

Un lanceur sera joint au projet, il permettra de choisir le résultat de sortie ainsi que les documents à parser

5 Les mêlées quotidiennes

5.1 Première mêlée quotidienne

Lors de la mêlée du jour, nous avons décidé de continué sur notre voie et de rajouté les éléments nécessaire à la nouvelle étape; on souhaite rajouté des éléments supplémentaire sous forme de balise(<intro>, <corps>, <conclusion>, <discussion>). Afin d'exploiter les informations extraites, nous utiliserons une nouvelle fois le langage python avec les librairies etree et LXML. Nous avons rencontré peu de difficultés durant cette étape ce qui nous conforte dans l'idée que nous avons fait les bons choix précédemment.

5.2 Deuxième et dernière mêlée quotidienne

Lors de l'ultime mêlée quotidienne nos esprits ont du se mettre d'accord sur la question suivante; "comment finaliser proprement le projet?". Il a fallu trouver des solutions efficaces et intuitives pour lancer le programme de parsing. Nous avons donc réfléchi a plusieurs interfaces homme-machine éventuelles dont une interface graphique fonctionnelle dans une petite fenêtre mais cette solution, bien qu'efficace, nécessitait une durée d'implémentation supplémentaire et nous a surtout semblée peu nécessaire sur le problème en question.

Ce qui en est donc ressorti est une ligne de commande simple appelant le programme en question en passant en paramètre le format du fichier en sortie souhaité ainsi que l'emplacement des fichiers au format PDF que l'on souhaite parser.

Nous avons aussi décidé de détailler la procédure à suivre afin d'exécuter le programme dans le fichier README. Nous pensons avoir fait le nécessaire pour rendre l'utilisation du programme accessible pour quiconque connait un minimum le fonctionnement d'un terminal Bash.

