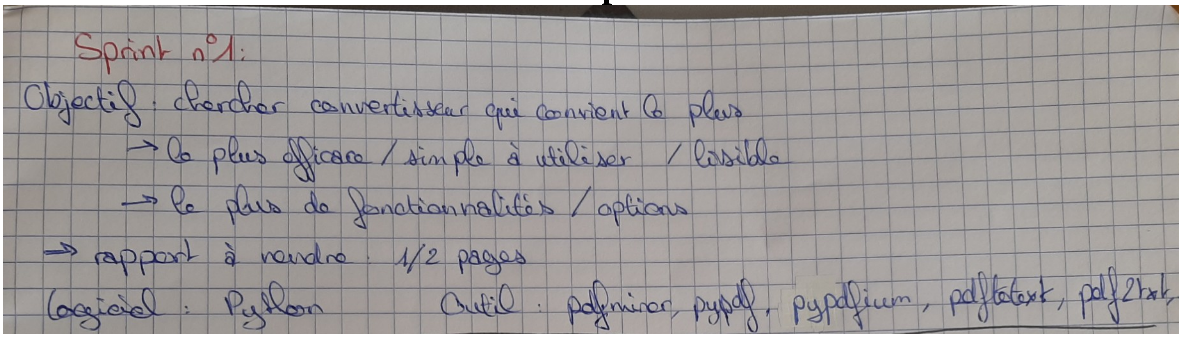
LouisonRABREAU – AlexisGOHIER – SimonCHEVE L3 INF/CMI

Sprint 1


Sprint n°1 :
Objectif : chercher convertisseur qui convient le plus
→ le plus efficace / simple à utiliser / lisible
→ le plus de fonctionnalités / options
⇒ rapport à rendre 1/2 pages
Logiciel : Python Outil : pdfminer, pypdf, pypdfium, pdftotext, pdf2txt,

Résumé des consignes et objectif du sprint1 : description de la compréhension des personnes de l’équipe après lecture du sujet, et choix du logiciel qui sera utilisé (Python), étant connu et déjà utilisé par l’équipe.

→ permet de savoir exactement quoi faire pour ne pas se perdre

```
            Poppler + pip install pdftotext -> Windows
import pdftotext
with open("input.pdf", "rb") as f:                # Load your PDF
            pdf = pdftotext.PDF(f)                #option "secret" if it's password-protected
with open('output.txt', 'w') as f:                # Save all text to a txt file
            f.write("\n\n".join(pdf))
- len(pdf) = nombre de pages

Debian, Ubuntu, and friends : "sudo apt install build-essential libpoppler-cpp-dev pkg-config python3-dev"
Fedira, Red Hat, and friends :"sudo yum install gcc-c++ pkgconfig poppler-cpp-devel python3-devel" ????
Windows : "conda install -c conda-forge poppler"
            https://github.com/jalan/pdftotext

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install PyPDF2
import PyPDF2
pdffileobj=open('input.pdf','rb')   #create file object variable and open it in "read binary" mode
pdfreader=PyPDF2.PdfFileReader(pdffileobj) #create reader variable that will read the pdffileobj
x=pdfreader.numPages                              #store the number of pages of this pdf file
pageobj=pdfreader.getPage(x+1)                    #create a variable that will select the selected number of pages
text=pageobj.extractText()                        #create text variable which will store all text datafrom pdf file
file1=open(r"output.txt","a")                     #save the extracted data from pdf to a txt file ('r' before the file path)
file1.writelines(text)

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install aspose-words
import aspose.words as aw
doc = aw.Document("document.pdf")                 # Load your PDF
doc.save("pdf-to-text.txt")                                   # Save all text to a txt file

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pdf2txt
https://github.com/euske/pdfminer/blob/master/tools/pdf2txt.py

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install pypdf
from pypdf import PdfReader
reader = PdfReader("input.pdf")
text = ""
for page in reader.pages:
    text += page.extract_text() + "\n"

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

            pip install pypdfium2

import pypdfium2 as pdfium
text = ""
pdf = pdfium.PdfDocument(data)
for i in range(len(pdf)):
    page = pdf.get_page(i)
    textpage = page.get_textpage()
    text += textpage.get_text()
    text += "\n"
    [g.close() for g in (textpage, page)]
pdf.close()
```

Recherches sur les differents convertisseurs de pdf → txt sur Python (Recherche_de_parseurs.txt) : comment les utiliser et quelques options.

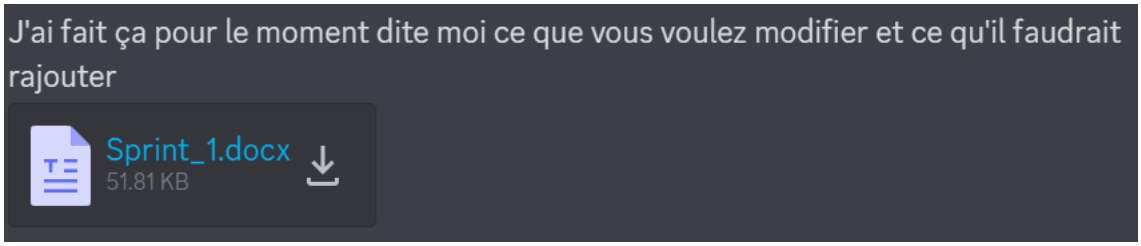
→ permet d’avoir des idées sur comment faire

Module	Temps d'exécution	Qualité	Commentaire	Options
PDF Miner	7 secondes	6/10	Très bien mais ne fonctionne vraiment pas bien sur certain pdf, mieux vaut ne pas prendre le risque.	Aucune option intéressante.
Aspose Word	30 secondes	8/10	Propre dans la globalité mais pas trop lors des données non textuelles (graphiques, ...)	Aucune option intéressante.
PyPDF	2 secondes	5/10	Paraît compliquer à exploiter par la suite.	Aucune option intéressante.
PyPDF2	3 secondes	5/10	Paraît compliquer à exploiter par la suite.	Aucune option intéressante.
PDFIUM	Moins d'1 seconde	9/10	Propre et assez homogène, semble le plus simple a exploiter.	Modification du PDF avant analyse.

Evaluation de certain parseurs pour déterminer les plus aptes à utiliser.

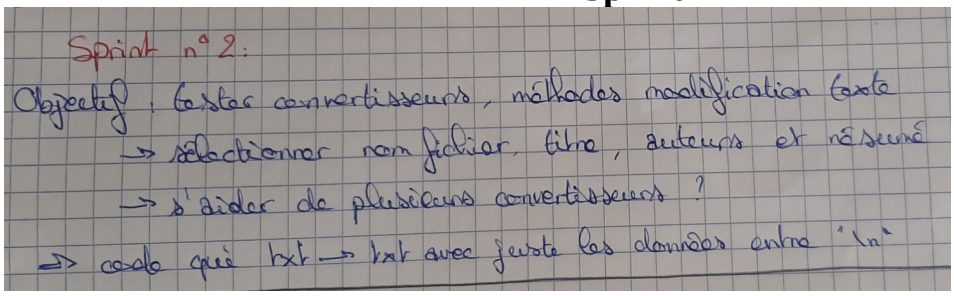
→ nombreuses discutions discord pour permettre d'atteindre ces évaluations.

→ choix «final» qui se porte sur pdfium, mais les autres modules sont gardés de côté.



Discutions discord permettant de relire le compte-rendu, le corriger et de le compléter, document repris à plus de 3reprises.

Sprint 2



Résumé des consignes et objectif du sprint2 : description de la compréhension des personnes de l'équipe après lecture du sujet.

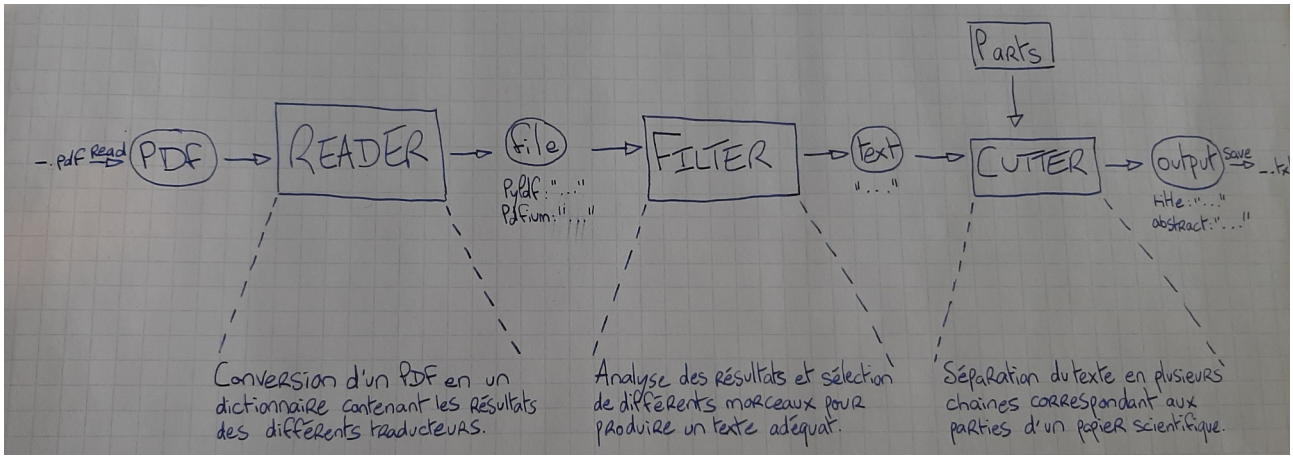
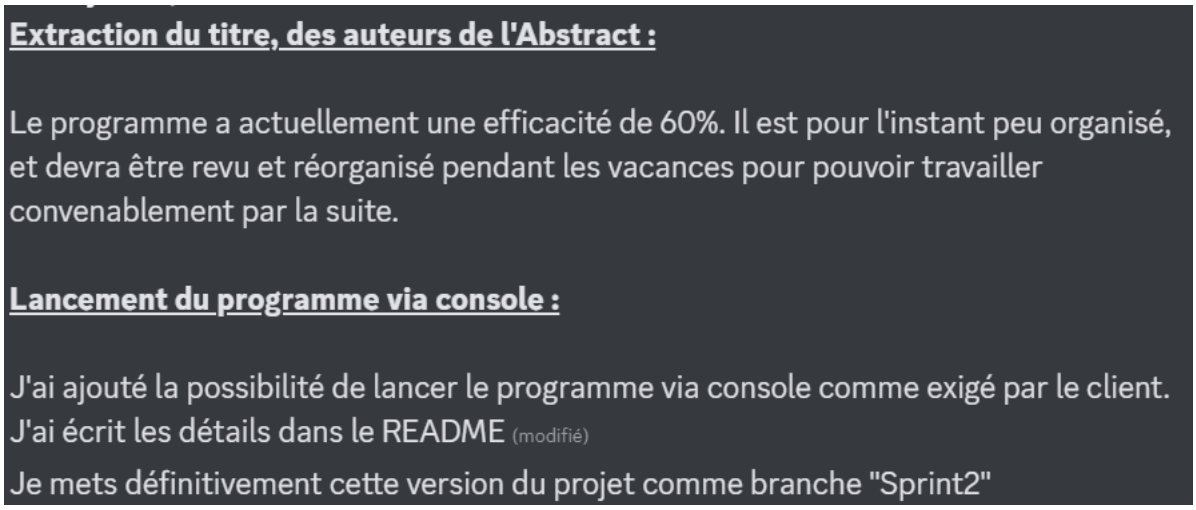


Schéma provisoire du fonctionnement/des différentes parties du projet : le input.pdf sera converti en plusieurs temporaire.txt en fonction des différents parseurs utilisés, les résultats seront analysés pour produire un input.txt, le plus correct possible. Un dernier programme découpera juste les élèments demandés en un output.txt



Discutions discord permettant de se mettre au courant des différentes avancés du produit.

Bilan Sprint 2

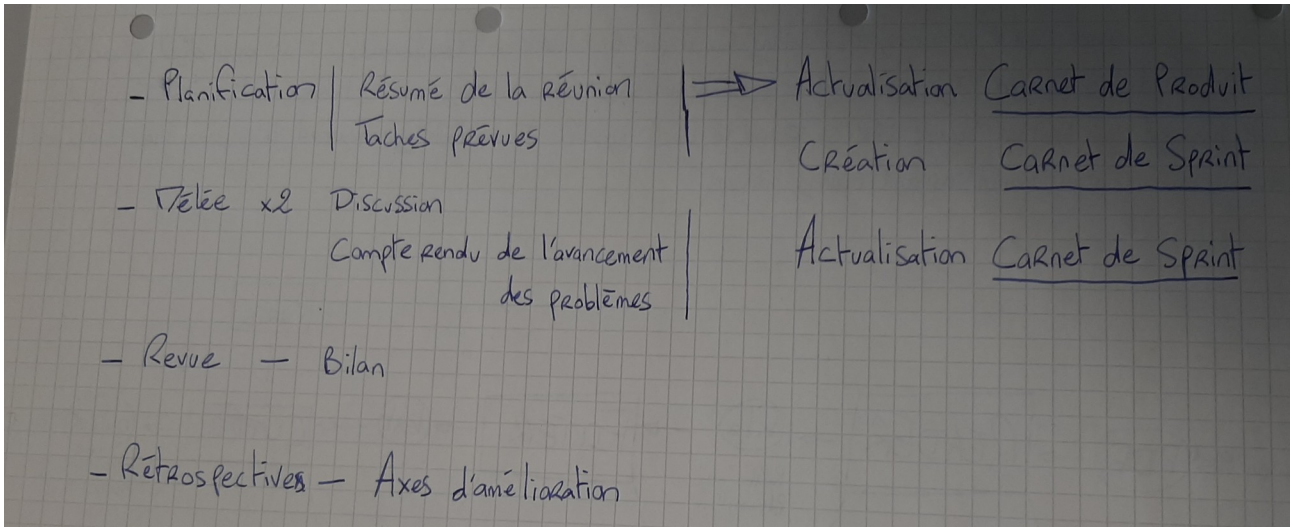
Pour la suite

→ se retouver en dehors des cours pour des réunions d'avancement

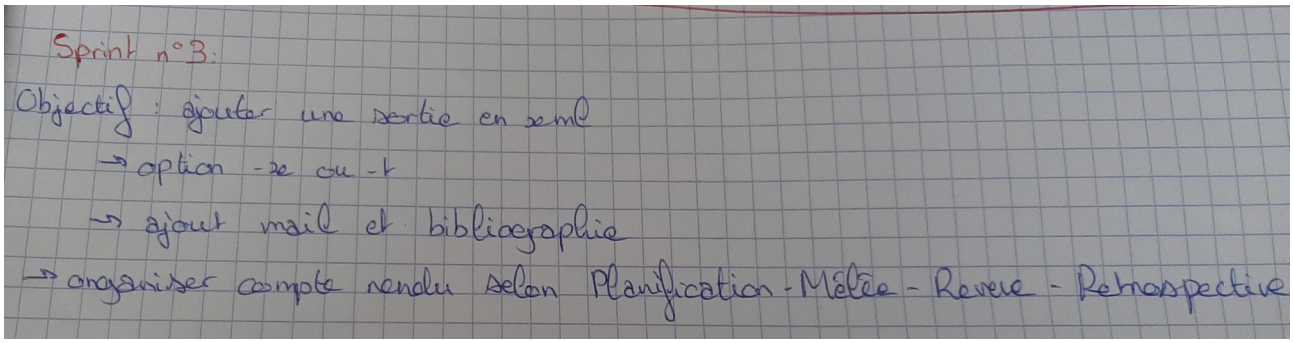
→ améliorement du code et création du «Filter»

Sprint 3

Planification :



Description des étapes d'un sprint.



Résumé des consignes et objectif du sprint3 : description de la compréhension des personnes de l'équipe après lecture du sujet.

Carnet de produit : résumé des points clés à fournir pour le projet, séparé en trois types (obligatoires, importants et optionnels) ainsi qu'en trois parties (a faire, en continu et fait). Le carnet de sprint ne comporte que les deux premières parties.

Revue :

→ Continuer à améliorer l'efficacité et l'organisation logistique
→ Commenter le code et compléter le README
→ Faire en sorte que le projet s'installe/s'exécute en une ligne de commande

Rétrospective :

-La récupération des mails fonctionne à 50%.

-La principale difficulté est de séparer les noms des auteurs puis de les lier à un mail de façon fiable. Il faudra dévelloper ou trouver des outils pour ça.

-La bibliographie est difficile à obtenir dans le parseur actuel, il faudra s'intéresser aux autres parseurs.

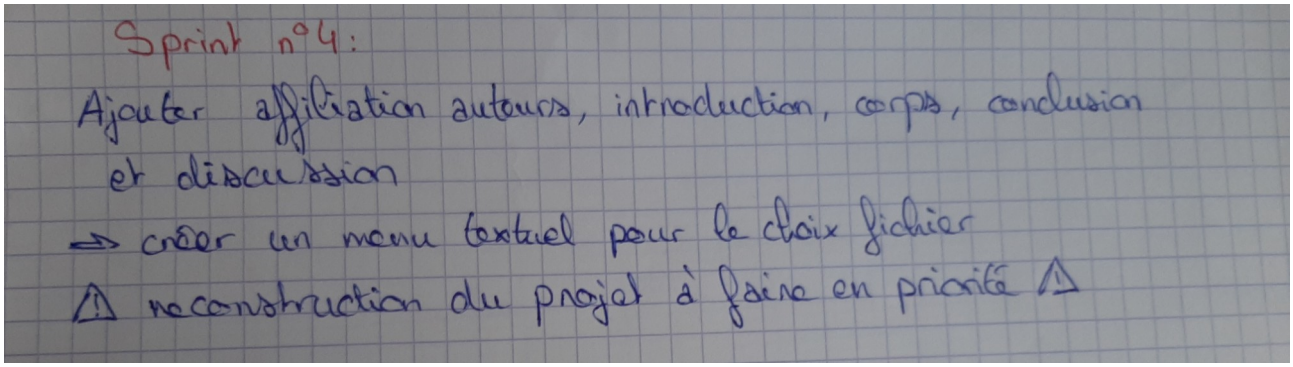
-La gestion des options fonctionne mais est assez primitive, c'est dur de faire quelque chose de propre quand on ne sait pas ce que voudra le client !!!

-La sortie XML est opérationelle mais je ne comprends pas comment on est sensé la lire, tous les logiciels de lecture de xml que je trouve la lisent comme un fichier texte.

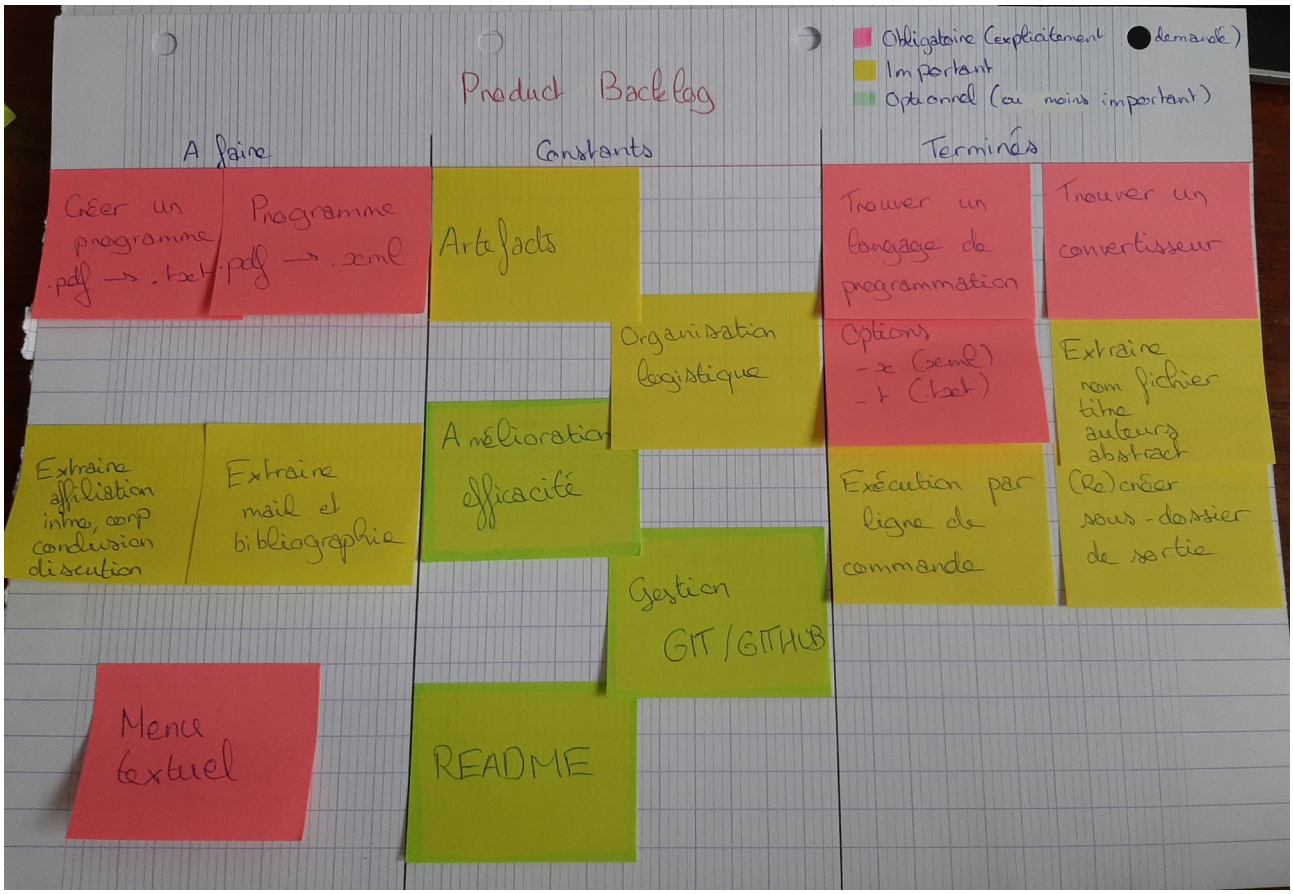
Discutions et mise au points des avancements.

Sprint 4

Planification :



Résumé des consignes et objectif du sprint4 : description de la compréhension des personnes de l'équipe après lecture du sujet.



Carnet de produit : résumé des points clés à fournir pour le projet, séparé en trois types (obligatoires, importants et optionnels) ainsi qu'en trois parties (a faire, en continu et fait). Le carnet de sprint ne comporte que les deux premières parties.
→ «Extraire mail» reste dans la partie «a faire» car non 100% optionnel

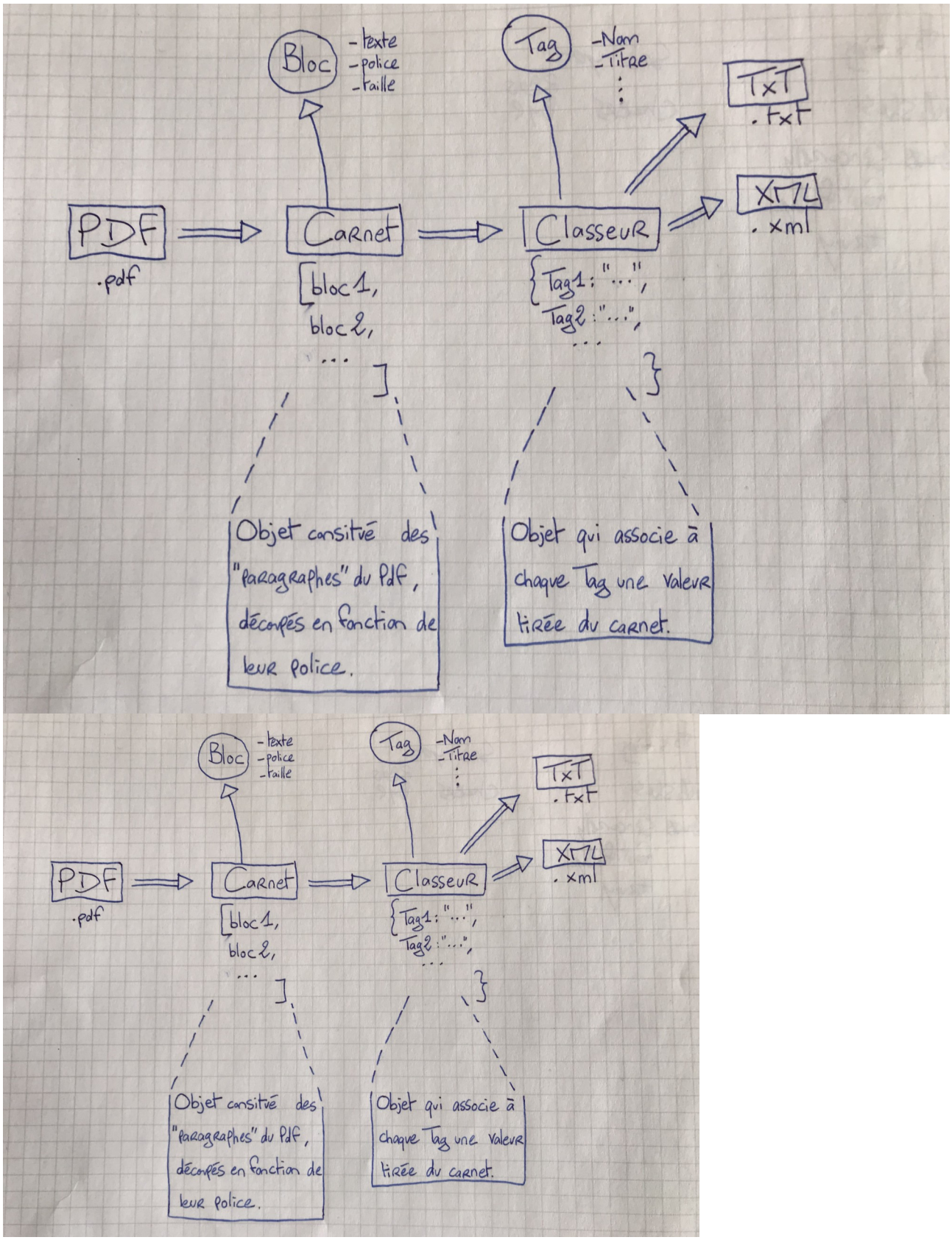
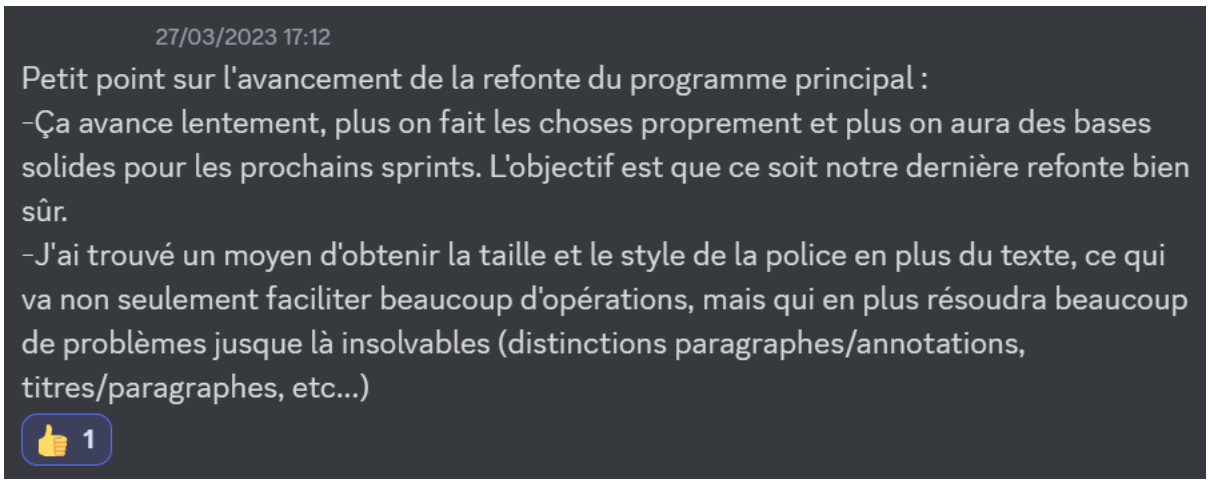


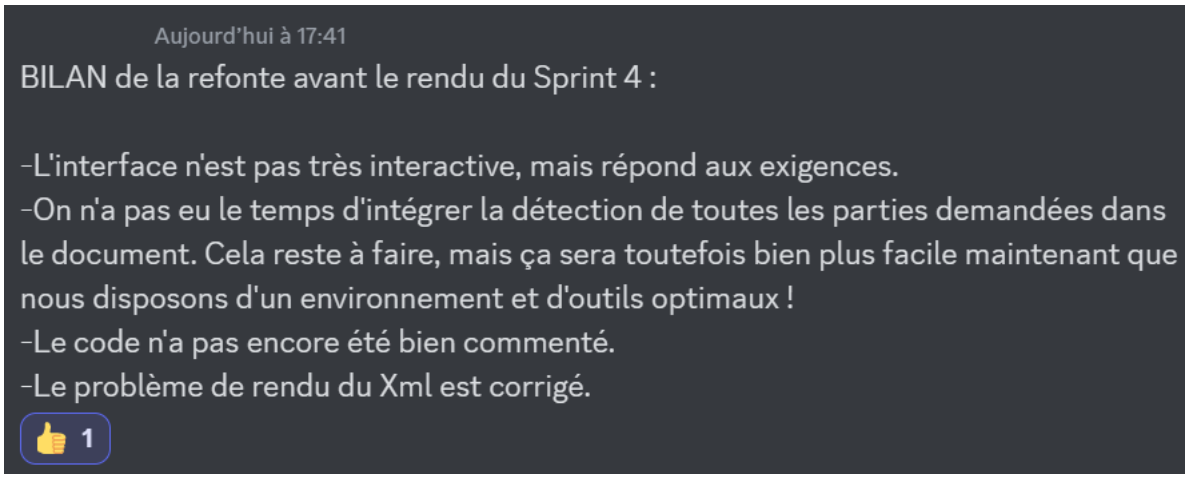
Schéma de l'organisation finale que devrait prendre le projet, après mise à jour.

```
----------------------------------
Police : 1        Taille : 9
Coling 2008: Companion volume - Posters and Demonstrations,

----------------------------------
Police : 2        Taille : 9
pages 23–26
Manchester, August 2008

----------------------------------
Police : 3        Taille : 14
A Scalable MMR Approach to Sentence Scoring
for Multi-Document Update Summarization

----------------------------------
Police : 4        Taille : 12
Florian Boudin

----------------------------------
Police : 5        Taille : 8
ɬ

----------------------------------
Police : 6        Taille : 12
and

----------------------------------
Police : 4        Taille : 12
Marc El-B`eze

----------------------------------
Police : 5        Taille : 8
ɬ
ɬ

----------------------------------
Police : 6        Taille : 12
Laboratoire Informatique d'Avignon
339 chemin des Meinajaries, BP1228,
84911 Avignon Cedex 9, France.

----------------------------------
Police : 7        Taille : 9
florian.boudin@univ-avignon.fr
marc.elbeze@univ-avignon.fr
```

Exemple d'objet de type «Carnet»



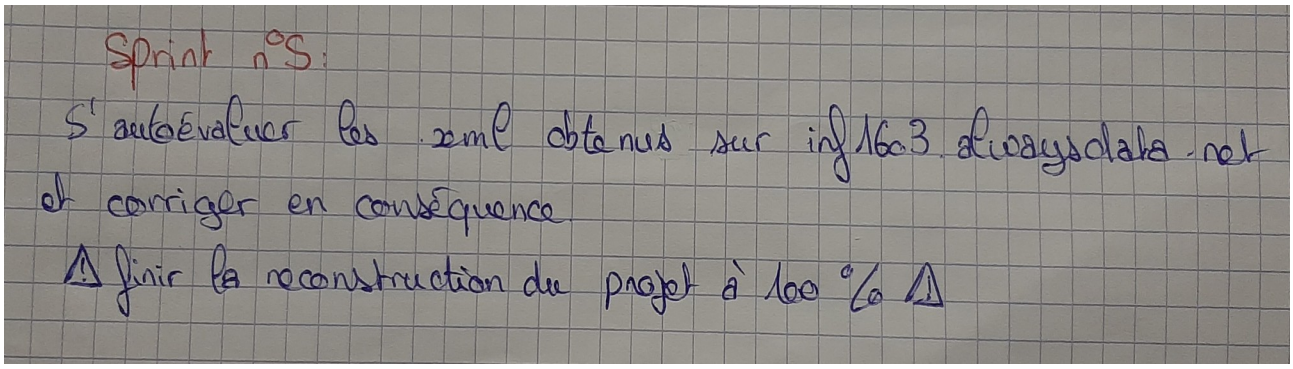
Discutions sur l'avancement du projet, tout au long du sprint. Ici un résumé de la première semaine.



Bilan de fin de sprint et résumé de la partie revue et retrospectives après discutions.

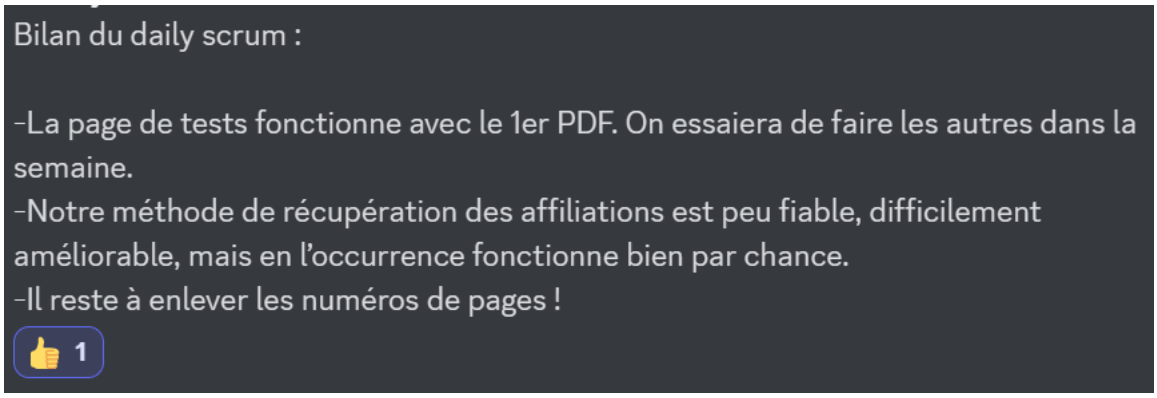
Sprint 5

Planification :

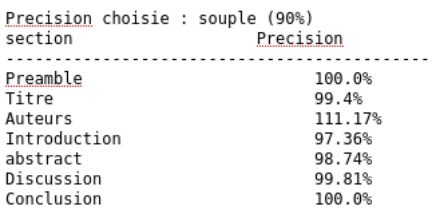


Résumé des consignes et objectif du sprint5 : description de la compréhension des personnes de l'équipe après lecture du sujet.

Carnet de produit : résumé des points clés à fournir pour le projet, séparé en trois types (obligatoires, importants et optionnels) ainsi qu'en trois parties (a faire, en continu et fait). Le carnet de sprint ne comporte que les deux premières parties.
→ Ce qui reste à faire ici reste surtout de l'amélioration en vue du rendu final

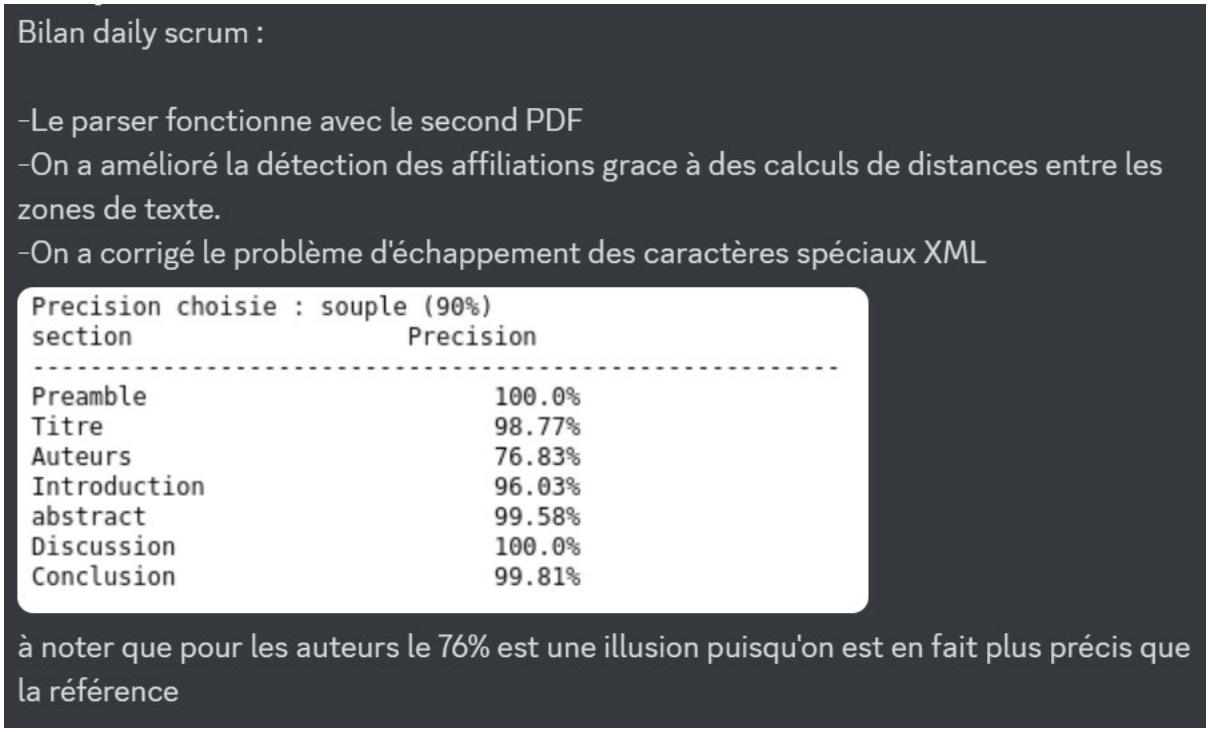


Discutions sur l'avancement du projet, tout au long du sprint. Ici un résumé de la première moitié de semaine.



```
Precision choisie : souple (90%)
section                  Precision
-----------------------------------------
Preamble                 100.0%
Titre                    99.4%
Auteurs                  111.17%
Introduction             97.36%
abstract                 98.74%
Discussion               99.81%
Conclusion               100.0%
```

Résultat correspondant au premier daily scrum.



Bilan de fin de sprint et des auto-évaluations.