Architecture tests automatiques

# Pourquoi

L’ajout de test automatique permet de préparer l’intégration continue dans le projet. En d’autres mots, cela signifie permet d’accueillir le devopps au sein du projet de manière optimale. En effet avec les pipelines github, chaque push va lancer une série de test. Afin de s’assurer de la qualité du rendu.   
Cela permet de vérifier de potentiel régression en temps réel. Cela permet aussi de vérifier la bonne intégration de chaque partie au sein du projet. En effet nous essayons ici de s’approcher du TDD (Test Driven Developement). Qui permet de fixer dans le marbre l’architecture.

# Comment

Pour ce faire il y aura 3 points clés.

## 1) Python

Python est une solution adéquate dans le contexte de test automatique en effet Python présente un faible cout en termes de développement, bien que ce projet soit étudiant et que la notion de cout n’ait pas sa place le temps de développement a quand même une importance non négligeable.

Python est par nature relativement simple à mettre en place et très polyvalent. Ce qui permet de tester de bout en bout un système.

## 2) Robot framework

Robot Framework est un outil qui permet a du personnelle non-développeur de créer des scénarios de test. En effet il s’agira d’une sorte de test à trou crée à partir d’un template.

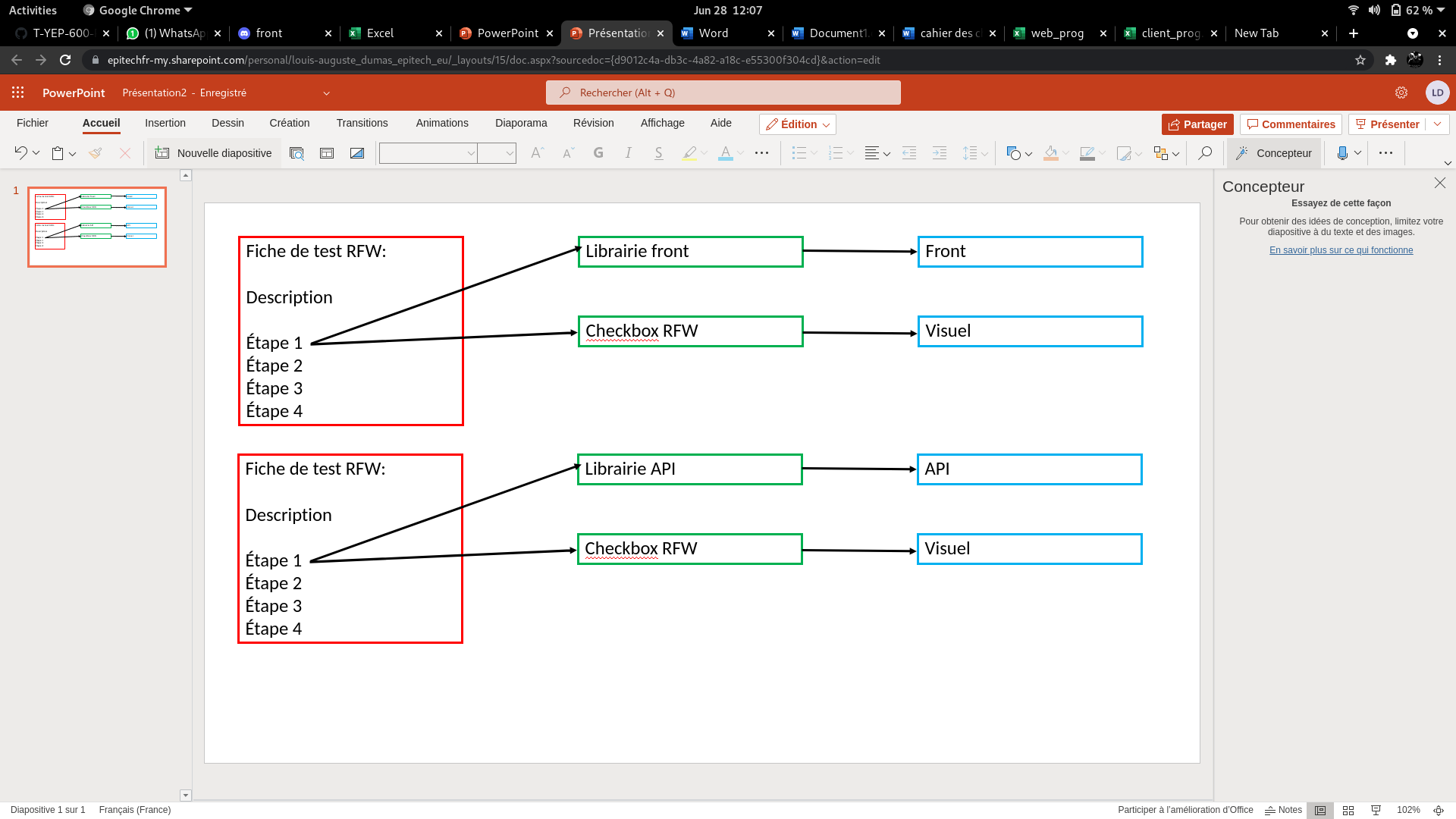
Le gros avantage de robot framework est la génération de documentation qui permet d’avoir un suivi des tests. En effet à la fin de l’exécution des tests, des fichiers seront générés avec des informations, des logs sur le test lancé. Et il nous permettra aussi de voir sous quel condition le test à fail. Et si le test fails, nous pourrons ainsi voir quelle partie a posé problèmes, comment et sous quelles conditions. Et donc nous aurons les informations nécessaires pour reproduire le bug.

## 3) Pipeline github (ou autre)

Le pipeline d’intégration github permet de lancer une série de test permettant de vérifier le bon respect de l’architecture ...

Fonctionnement

Pour ce faire nous utilisions l’architecture suivante :



La fiche de test robot Framework va appeler différentes fonctions python qui vont lancer certaines actions. Sur le drone on ne peut pas avoir d’état sur le statut du drone. Il faudra un opérateur qui va vérifier l’état du drone à l’instant T et voir son statut.

Les tests seront divisés sous forme de campagne qui est-elle même divisé sous forme de tests.   
Une campagne correspond en quelques sortes à une catégorie de test qui vont tester un type de fonctionnalité. Le cas présent nous aurons besoins de 5 campagnes de tests, la première

1. Reconnaissance vocale
2. Client web
3. Api masteur
4. Api user
5. Api son

Chaque campagne de tests va ensuite contenir différents tests. Dans le cas de la campagne “client web” toutes les actions concerneront le client web et l’opérateurs vérifiera que le drone prend bien en compte les instructions.

https://robotframework.org/robotframework/latest/libraries/Dialogs.html