

**Module : M202 – Approche Agile**

**Filière : Développement digital – option web full stack**

**Mini\_Projet**

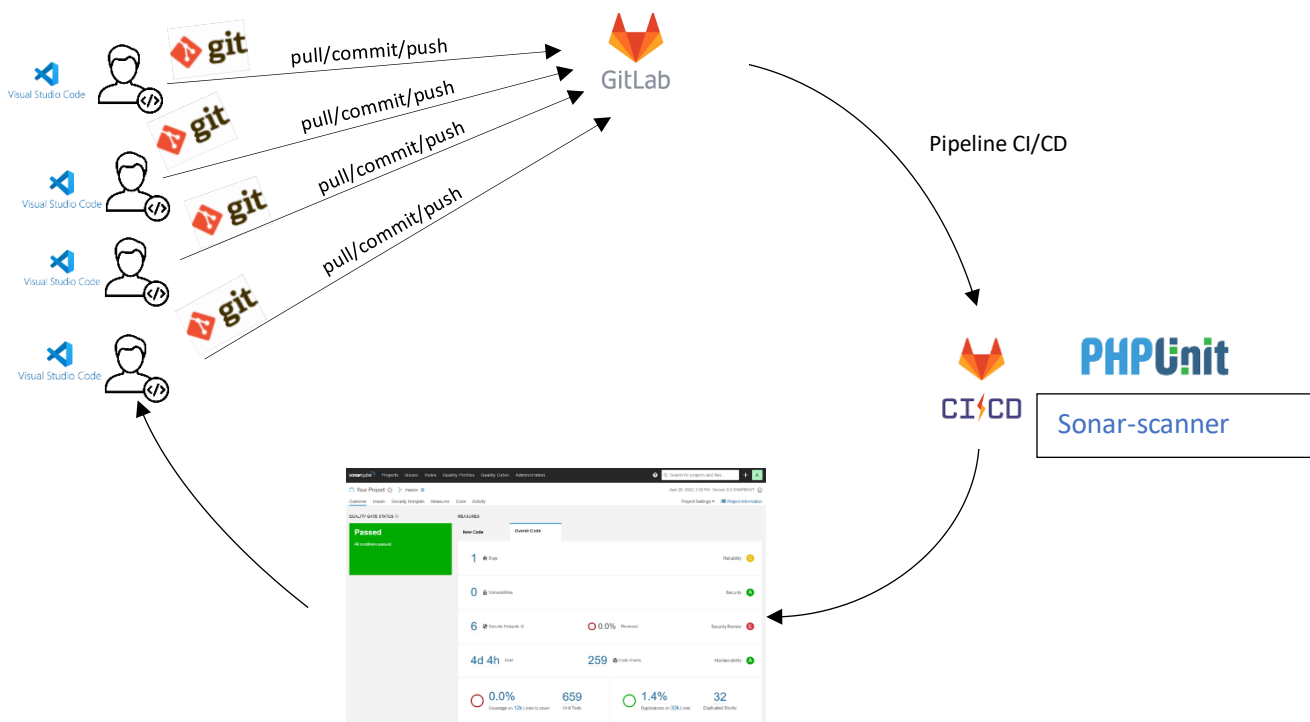
***A rendre avant le 09/01/2023***

**Objectifs :**

- Mettre en pratique tous les acquis du module Approche Agile
  - Gérer le projet avec la méthode agile SCRUM
  - Gérer les versions du projet à l'aide de git et Gitlab
  - Analyser la qualité du code avec Sonarqube
  - Configurer les tests unitaires avec PHPUnit
  - Configurer et afficher le taux de couverture de code
  - Configurer un pipeline CI/CD et instaurer la culture DevOps

L'objectif de ce mini-projet est de mettre en pratique toutes les compétences acquises lors du module « Approche agile ».

La réalisation de ce mini-projet doit s'effectuer en mettant en place une chaîne DevOps ; cette chaîne permettra ainsi d'automatiser au maximum le processus du développement de ce mini-projet, voici l'architecture visée :



Formez des groupes d'environ quatre stagiaires et choisissez un titre à votre projet :

Le projet doit être une application web gérant un sujet de votre choix ;

Réalisez le projet en suivant les indications suivantes :

**1. Configuration du projet sous Jira :**

- a. Le responsable du groupe doit créer un projet sous Jira et inviter les autres membres de son équipe.

*En plus, le responsable est amené à inviter l'adresse suivante :*

[asmae.youala2@gmail.com](mailto:asmae.youala2@gmail.com) ;

- b. Ajouter à votre backlog product au moins 10 user stories ;
- c. Créer le sprint 0 où figureront les tâches suivantes :
  - Installation des environnements (php, mysql ...) ;
  - Analyse et conception : Création des diagrammes de merise (MCD, MLD ...) ou UML (diagramme de cas d'utilisation, diagramme de classe ...) ;
    - ➔ *Ajouter les documents produits comme pièces jointes de cette tâche ;*
  - Elaboration du script de création de la base de données ;
    - ➔ *Ajouter les documents produits comme pièces jointes de cette tâche ;*
  - Création et configuration du projet sous Gitlab (voir 2.) ;
    - ➔ *(Optionnel) Vous pouvez lier votre projet sous Jira avec un compte Gitlab (voir le menu « code »)*
  - Configuration de Sonarqube (voir 3.)
  - Configuration d'un pipeline CI/CD (voir 4.)

**2. Création et configuration du projet sous Gitlab :**

- a. Créer un groupe sous Gitlab en lui attribuant un nom de votre choix. Ajouter les membres de votre équipe à ce nouveau groupe ;
- b. Créer un projet sous le namespace de votre groupe ;
- c. Chacun des membres de l'équipe doit cloner le projet sur sa machine locale ;

**3. Configuration de Sonarqube :**

- a. Créer un nouveau projet sous Sonarqube ;
- b. Configurer votre projet et récupérer la commande nécessaire au scan
  - ➔ *(Optionnel) Vous pouvez effectuer des recherches pour voir comment configurer un projet Gitlab sous Sonarqube*

**4. Ajout d'un pipeline CI/CD :**

- d. Ajouter un Runner à votre projet Gitlab (*On peut configurer plusieurs Runners*) ;
- e. Ajouter le pipeline par défaut proposé par Gitlab (Tester son exécution) ;
- f. Configurer le fichier `.gitlab-ci.yml` pour qu'il se compose de trois « *stages* » :
  - **initiation** : affichant un message d'initialisation du pipeline ;
  - **tests-unitaires** : lançant les tests unitaires ;
  - **qualite-du-code** : lançant l'analyse avec sonar scanner

Vous pouvez utiliser le modèle suivant :

```
stages:
  - xxxxx
  - xxxxx
  - xxxxx

xxxxx-job:
  stage: xxxxxx
  script:
    - xxxxxxxxx

xxxxx-job:
  stage: xxxxxxxx
  variables:
    XDEBUG_MODE: "coverage"
  artifacts:
    when: always
    reports:
      junit: phpunit-execution-result.xml
  paths:
    - phpunit-execution-result.xml
    - phpunit-coverage-result.xml
  script:
    - vendor/bin/phpunit --coverage-clover=phpunit-coverage-
      result.xml --log-junit=phpunit-execution-result.xml

xxxxx-job:
  stage: xxxxxxxx
  variables:
    GIT_CLEAN_FLAGS: none
  script:
    x:\...\sonar-scanner\bin\sonar-scanner.bat -
      D"sonar.qualitygate.wait=true" -D"sonar.projectKey=xxxxxxxxxx" -
      D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -
      D"sonar.login=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```

➔ *A chaque commit/push, le pipeline démarrera pour exécuter les différentes tâches (job) indiquées ;*

**5. Planification d'un sprint :**

- a. Ajouter trois à quatre user stories à votre sprint ;
- b. Décomposer les user stories en des tâches en estimant la durée de réalisation de chacune des tâches ;
  - ➔ La tâche d'écriture des tests unitaires est à prévoir parmi les tâches ;
- c. Affecter les tâches au différent membre de l'équipe ;
- d. Démarrer le sprint avec une durée d'une semaine ;
- e. Organiser des réunions quotidiennes pour discuter les avancements effectués ;
- f. A la fin du sprint, organisez une démonstration : chaque membre de l'équipe présentera son travail.

La démo doit être planifiée sous teams, chaque équipe est amené à inviter le compte « asmae.youala@ofppt-edu.ma » à la réunion. La démo doit être enregistrée.

➔ Chaque projet doit comporter au minimum quatre sprints.