

Consignes projet ingénierie logicielle

Travail demandé:

- Partager, via un espace de travail collaboratif (Par exemple Git), tous les artefacts de votre projet: dossier des spécifications, conception, comptes rendus des réunions, rapports de tests, documentation technique, historique des versions, etc. **Votre projet doit être mis dans un dépôt github privé.**
- Planification du projet en utilisant une méthode agile.
- Utilisation d'un gestionnaire de version pour organiser l'activité de développement en équipe (SVN, GIT ou Mercurial).
- Gestion efficace des dépendances en utilisant un fichier de configuration (Pom.xml ou Gradle.build).
- Build automatique de votre projet.
- Automatisation des tests avec maven ou gradle (**mvn test**).
- Expérimentation du processus d'intégration continue et observation des bénéfices pour toutes les parties prenantes du projet de développement logiciel.

Constitution des équipes de projet :

Une équipe de projet est constituée au maximum de deux personnes. Vous échangerez les rôles suivants: développeurs, testeurs, concepteurs, intégrateurs, chef de projet, DevOps, etc.

Délai : 16 Avril 2025 (délai de rigueur) présentation prévue pour 23 Avril 2025

Rapport de projet : spécifier les étapes que vous avez suivi et les résultats obtenus (captures d'écran + explication + argumentation s'il le faut).

Soumettre la version PDF sur la plateforme : [Lien](#)

Aucun travail soumis par email ne sera accepté

Livrables:

1er rendu: création d'un cahier de charges

- ☐ Création d'une page wiki sur un espace collaboratif (ex. Confluence, GitHub Wiki).
- ☐ Description détaillée du projet et des objectifs.
- ☐ Identification des acteurs et des rôles.
- ☐ Définition des exigences fonctionnelles et non fonctionnelles.
- ☐ Plan de gestion des versions et des branches Git.

2ème rendu : Planification du projet en utilisant une approche agile (Scrum) --- **Optionnel**

- ☐ Définition des rôles (Product Owner, Scrum Master, l'équipe).
- ☐ Construire le Product backlog.
- ☐ Découpage du projet en sprints (itératif + incrémental)

- ☐ Construire le sprint backlog.
- ☐ Planification des releases.
- ☐ Gestion de projet avec le Logiciel JIRA, Trello, Excel , Ice Scrum.
- ☐ Diagramme de GANTT pour tout le projet pour visualiser l'aspect itératif et incrémentale

3ème rendu : Conception (Votre Sprint) --- Optionnel

Travailler sur une demande d'utilisateur (user story) d'un sprint. Le travail demandé est:

- ☐ Conception collaborative en utilisant un AGL.
- ☐ Pour chaque cas d'utilisation décrire le nom, l'acteur, précondition, Post- condition, description du scénario nominal, description du scénario alternatif, description du scénario d'exception.
- ☐ Conception des IHM (Prototype).
- ☐ Création des diagrammes (séquences, classes au minimum).
- ☐ Partage de la conception sur l'espace collaboratif en ligne pour qu'elle soit accessible à tous les membres de l'équipe.
- ☐ Discussion de la conception et correction des erreurs.

4ème rendu: Codage (Obligatoire)

- ☐ Création et configuration d'un dépôt GitHub, GitLab ou Bitbucket.
- ☐ Utilisation d'un gestionnaire de versions (Git, SVN, Mercurial).
- ☐ Adoption d'une stratégie de branches (ex. Git Flow).
- ☐ Gestion des conflits et bonnes pratiques Git (Push, Pull, Merge, Rebase).
- ☐ Maintien de plusieurs environnements : Développement (dev), Test (QA), Production (prod).

5ème rendu: Automatisation (Obligatoire)

- ☐ Mise en place d'un système de build automatisé avec Maven ou Gradle.
- ☐ Rédaction et exécution de tests unitaires et d'intégration (JUnit, PyTest, Mocha, etc.).
- ☐ Génération d'un rapport de couverture des tests (Jacoco, SonarQube).
- ☐ Exécution automatique des tests après chaque commit avec un outil CI/CD.

6ème rendu : Rapport d'intégration continue (Obligatoire)

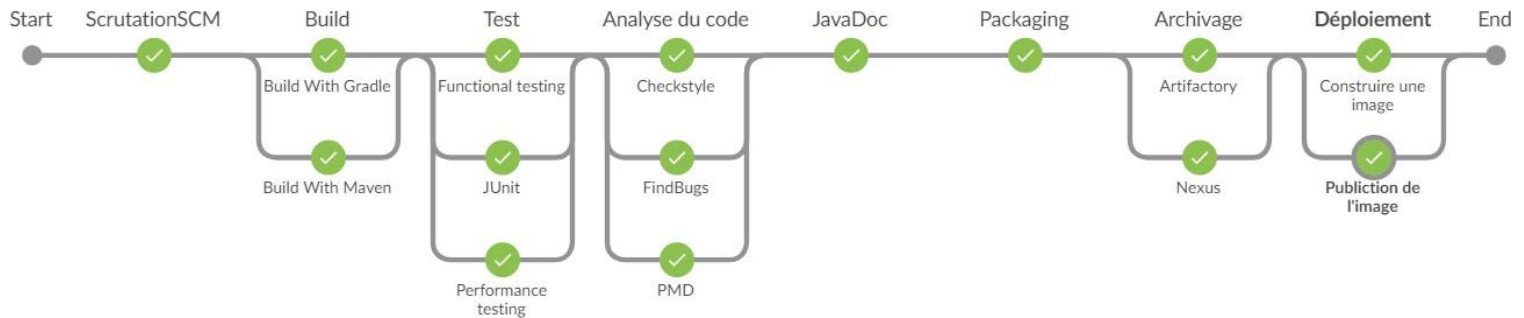
- ☐ Installation et configuration de Jenkins, GitHub Actions ou GitLab CI/CD.
- ☐ Création d'un Job de compilation et d'exécution des tests.
- ☐ Mise en place de la scrutation automatique du dépôt et déclenchement des builds.
- ☐ Envoi de notifications en cas d'échec des tests ou du build.

8ème rendu: Rapport d'intégration continue avancé (Obligatoire)

- ☐ Rédaction d'un script Groovy (Jenkinsfile) décrivant le pipeline CI/CD.
- ☐ Versionnement du script dans le dépôt GitHub.
- ☐ Configuration du job CI/CD pour exécuter le script et automatiser le build (Vous pouvez utiliser le plugin Blue Ocean de Jenkins).
- ☐ Automatisation complète du pipeline : build, test, package, déploiement.
- ☐ Utilisation de Nexus, Artifactory pour la gestion des artefacts.
- ☐ Mise en place d'un déploiement continu (CD) avec Docker et Kubernetes (Optionnel).
- ☐ Simulation d'un environnement de production avec conteneurisation (Optionnel).
- ☐ S'il y'a une erreur dans une des étapes de build, envoyer une notification par mail à

l'administrateur de Jenkins.

Exemple de workflow :

**9^{ème} Rendu: mise en œuvre full stack d'un projet logiciel (Très obligatoire)**

Répondre aux questions suivantes à la fin de votre rapport :

1. Que veut dire la mise en œuvre full stack d'un projet logiciel ?
2. Décrire Le cycle de vie d'un build avec maven ou gradle ?.
3. Que représentent les coordonnées maven d'un projet ?
4. Quel est l'objectif de la pratique «Convention Over Configuration » dans maven
5. Quel est le rôle des cibles maven ? quel est leur équivalent dans Gradle ?
6. Quel est l'avantage des systèmes de gestion de version distribués par rapport aux systèmes de gestion de version centralisés ?
7. Pourquoi suivre en version votre projet logiciel est une bonne pratique ?
8. Quelle est la différence entre la commande Rebase et Merge dans Git ?
9. Qu'est-ce qu'on entend par l'expression « se détacher du HEAD » ?
10. Selon vous, quelle est la meilleure approche d'intégration continue parmi celles que vous avez expérimentées dans le TP.
11. A quoi sert un Webhook et quel est l'avantage de son utilisation ?
12. Quel est l'avantage des workflows dans un serveur IC ?
13. Quel est l'avantage de la stratégie Maître/Esclave dans le serveur d'Intégration Continue ?
14. Quelle est la différence entre un build échoué et un build instable ?