

Validation Superviser et assurer le développement des applications logicielles

Présentation orale de 20 min suivie de 15 min d'échange

Conseils :

Vous pouvez travailler ce sujet d'évaluation de BLOC sur la durée complète du BLOC.

Compétences évaluées

Élaborer le processus d'assurance qualité logicielle	<ul style="list-style-type: none">• Evaluer la qualité du développement• Définir la politique de tests• Définir la politique de sécurité
Piloter le développement et le déploiement d'applications	<ul style="list-style-type: none">• Superviser la mise en place d'une chaîne de livraison continue• Faciliter la montée en compétences des développeurs• Orchestrer la mise en production pour garantir la disponibilité et la montée en charge
Maintenir et développer son expertise en développement d'applications	<ul style="list-style-type: none">• Expérimenter des technologies innovantes en bac à sable• Démontrer la faisabilité par la réalisation d'un POC (proof of concept)• Développer des applications complexes

Consignes sujet d'évaluation

Consignes validation Superviser et assurer le développement des applications logicielles

Enoncé

Superviser et assurer le développement des applications logicielles

Cf. contexte du sujet (cas entreprise)

Travail demandé/Livrable final

1. Travail demandé

Vous êtes le / la Lead Developer en charge du développement de l'application de l'entreprise fictive décrite dans le contexte fourni. Votre mission consiste à :

- Définir un processus de développement respectant les exigences de qualité logicielle (vous pouvez vous appuyer sur la norme ISO 25010).
- Développer la version 1 de l'application intégrant :
 - Un pipeline CI/CD avec tests automatisés
 - Une gestion du déploiement via un orchestrateur ou un fournisseur cloud
 - Une solution de sécurité minimale (HTTPS, authentification/autorisation, scan de vulnérabilités)

Dans le cadre de cet exercice, il ne vous est pas demandé d'implémenter l'ensemble des fonctionnalités. Vous devez cependant à minima implémenter une fonctionnalité métier.

- Analyser la sécurité de cette version 1 et proposer un plan de remédiation.

Votre mission se déroule donc en 3 phases :

1. Structuration du processus de développement
2. Développement et déploiement : Réalisation de l'application
3. Plan de remédiation : Analyse et recommandations

La mission sera clôturée par une présentation à l'entreprise de vos résultats lors d'une soutenance orale de 20 minutes suivie d'un échange de 15 minutes.

2. Consignes détaillées de chaque phase

1.1. Structuration du processus de développement

Il s'agit de s'assurer que l'application livrée sera conforme aux exigences de qualité logicielle (performance, fiabilité, disponibilité, flexibilité, capacité fonctionnelle, etc.). Vous pouvez vous appuyer sur la classification de l'ISO 25010.

1.1.1. Indicateurs

Vous devez identifier quatre indicateurs ou métriques permettant d'évaluer la conformité aux exigences de qualité. Chaque indicateur / métrique peut couvrir plusieurs exigences de qualité ou être associée à une seule exigence. Il n'est pas demandé de couvrir l'ensemble des exigences de qualité.

Vous prendrez soin de justifier comment le suivi de ces métriques permettra d'éviter l'accumulation de dette technique dans le temps.

1.1.2. Cycle de vie et sécurité du développement

Consignes sujet d'évaluation

Consignes validation Superviser et assurer le développement des applications logicielles

Vous devez formaliser le cycle de vie du développement en intégrant la démarche DevSecOps. Pour chaque étape clé, il faut préciser les mesures de sécurité à mettre en place.

Vous devrez schématiser de façon détaillée le processus de livraison/déploiement continu (CI/CD).

Votre pipeline doit intégrer les tests automatisés : types de tests (unitaires, intégration, etc.), outils utilisés (frameworks de tests, analyse de code, etc.)

Vous devez mettre en avant comment ces outils permettent de suivre vos indicateurs / métrique qualité.

1.1.3. Compétences et formation

Cartographiez les compétences nécessaires au projet.

Proposez une action de formation pour renforcer les compétences de l'équipe.

Dans le cadre de cette évaluation il n'y pas de contrainte budgétaire ou de limites de membres. Il faut cependant rester réaliste sur les profils et ne pas utiliser des « moutons à 5 pattes ».

1.2. 2. Développement et déploiement : Implémentation du POC

Le processus et l'équipe mis en place, vous pouvez passer à la phase de réalisation de votre application.

1.2.1. Analyse des exigences et choix techniques

Reformulez les exigences fonctionnelles que doit satisfaire l'application. La fonctionnalité implémentée dans « votre prototype » doit être clairement décrite sous forme d'un backlog (User story et critères d'acceptation)

Vous réaliserez un schéma d'architecture technique intégrant la sécurité et les solutions techniques retenues (type de base de données, protocole de communication, Framework, etc.)

1.2.2. Expérimentation et documentation

Avant de vous lancer dans le développement à proprement parlé, il est opportun de tester les technologies / solutions techniques phares pour vous assurer de leur intégration.

Sélectionnez les technologies pertinentes et critiques et testez leur mise en œuvre. Vous devez rédiger une synthèse du protocole d'expérimentation. Par technologie ou solution technique, il est entendu plateforme support des composants que vous allez développer. Par exemple : middleware orienté messages, système de sécurité, framework de développement d'application d'entreprise, système de gestion de base de donnée, système d'observabilité, service mesh, orchestrateur de type Kubernetes (ex : Minikube), services d'un fournisseur cloud, etc.

Votre synthèse du protocole doit inclure l'environnement « de test », les étapes clés pour reproduire cette expérimentation, les difficultés rencontrées et les limites identifiées. La présentation des résultats obtenus doit vous permettre de justifier l'adoption des technologies phares. Vous pouvez aussi utiliser votre expérimentation en bac à sable pour rejeter certaines technologies.

Exemples d'expérimentations possibles : mise en place d'une plateforme CI/CD intégrant des outils de qualité du code et des frameworks de tests, mise en place ou exploitation d'un service mesh, intégration de services d'infrastructure (API gateway, résilience, découverte de services), déploiement d'un broker de messages dans un environnement de type Kubernetes avec tests de la publication consommation de messages , déploiement d'un cert-manager pour le provisionnement de certificats sécurisant un Ingress, clustering d'un système de gestion de bases de données, mise en

Consignes sujet d'évaluation

Consignes validation Superviser et assurer le développement des applications logicielles

plan de l'application, test d'observabilité, test d'une technologie Serveless, implémentation de traitement distribué sur plusieurs nœuds, etc.

Exemples de tests non considérés comme valides pour l'expérimentation : tests d'un langage de programmation, la communication (REST, gRPC, WebSocket, etc.) entre un composant front-end et un composant back-end, l'exécution locale de tests unitaire, l'implémentation d'un composant avec un Framework (MVC, ORM, service REST, etc.)

1.2.3. Développement de l'application

Dans le cadre de cet exercice vous restez au niveau du prototypage.

Vous devez implémenter une fonctionnalité métier en respectant le backlog que vous avez défini en amont. Le respect de la fonctionnalité devra être présenté via des tests d'appels ou des tests d'acceptation. Les tests d'acceptation peuvent faire partie des deux tests minimum qui peuvent être intégrés au pipeline CI/CD (ou plus bas).

Vous devez vous assurer que les composants / services que vous développez respectent l'architecture définie et les solutions techniques retenues.

Votre application devra intégrer une solution de sécurité (par exemple la gestion de la communication via HTTPS / TLS, la gestion de l'authentification et des autorisations via un serveur d'autorisation tel que Keycloak, la détection des vulnérabilités)

Vous devrez aussi vous assurer de la mise en place d'une des composantes de l'observabilité (tels que les logs centralisés, la collecte de métriques, la gestion des traces distribuées)

La solution doit intégrer les principales étapes du pipeline CI/CD avec à minima 2 types de tests qui permettent d'évaluer les métriques.

Note : Pour l'évaluation vous ne serez pas pénalisé si le flux de votre pipeline ne peut mettre en œuvre une étape de déploiement en (pré) production.

Lors de la présentation vous devrez présenter la montée en charge de l'application en lançant des tests de charge avec Siege ou JMeter par exemple. Bien que vous ayez le droit d'intégrer ces tests à votre pipeline CI/CD, vous n'en n'avez pas l'obligation dans le cadre de cette évaluation.

1.3. 3. Plan de remédiation : Analyse sécurité et recommandations

Votre v1 étant prête, vous devez l'auditer pour proposer le renforcement de sa sécurité.

Vous devez analyser les résultats des tests et les métriques collectées.

A partir de ces résultats et de vos connaissances de l'application, identifiez les vulnérabilités potentielles. Vous pouvez notamment vous appuyer sur l'analyse de vos tests de charge.

Plan de remédiation : vous proposez enfin des préconisations d'actions pour renforcer la sécurité adaptée au contexte de votre application. Prenez soin de justifier vos recommandations.

Déroulement et livrables intermédiaires

Contexte et consigne de l'évaluation du bloc fournis au démarrage de celui-ci.

La journée de validation de BLOC est dédiée aux soutenances individuelles.

Consignes sujet d'évaluation

Consignes validation Superviser et assurer le développement des applications logicielles

Organisation

Travail individuel

Cf. contexte du sujet

Conseils

Il s'agit d'une évaluation « technique », il n'est donc pas demandé de vulgariser votre présentation pour des non-initiés. Considérez les décideurs comme « spécialistes du développement ». Vous devez être vigilant sur la précision des termes employés (pas de « je sais faire mais pas expliqué » accepté).

Les schémas doivent être lisibles et complets mais il n'y a pas d'évaluation prenant en compte la forme d'un support de type powerpoint.

La démonstration pratique de votre application (en fait un prototype) doit être préparée pour que la présentation soit fluide et pour limiter « l'effet démo » (ça ne marche plus !). Vous pouvez d'ailleurs enregistrer une vidéo en secours en cas de problème majeur.

Pièces annexes

Contexte du sujet

Critères d'évaluation du bloc **Superviser et assurer le développement des applications logicielles**