|  |  |
| --- | --- |
| **Bloc de compétences 2 SLAM – Conception et développement d’applications Semestre 2** | |
| **Finalité métier :** Vous exercez les activités suivantes sous la responsabilité d’un chef de projet pour répondre aux besoins de digitalisation de vos clients :   * Recueil et analyse des besoins issus des utilisateurs et/ou des métiers * Conception de la solution applicative * Gestion des données * Implémentation de la solution applicative avec les technologies retenues * Conception et réalisation de plans de tests * Rédaction de la documentation technique * Recette de l’application et accompagnement des utilisateurs * Gestion de la maintenance corrective et/ou évolutive de l’application | **Contexte professionnel** : Vous travaillez pour le compte d’une entreprise de services du numérique qui intervient auprès de différentes entreprises clientes pour assurer des missions de conception, de développement d’applications informatiques. Vous assurez également la maintenance corrective ou évolutive de leurs applications existantes ainsi que la gestion des données de leurs applications. Le déploiement des applications ainsi que le stockage des données peuvent être réalisés à travers une solution en nuage informatique (*cloud computing*). Les projets sont réalisés dans le respect du cahier des charges et en tenant compte des contraintes techniques, environnementales et réglementaires. Vous assurez une veille technologique régulière et appliquez les bonnes pratiques de qualité et de développement pour apporter des solutions performantes et des technologies innovantes à vos clients. Les projets de développement de ces applications digitales sont menés à l’aide de méthodes classiques de gestion de projets (cycle en V) ou en mode agile. |

|  |  |
| --- | --- |
| **2.2 Assurer la maintenance corrective ou évolutive d’une solution applicative Semestre 2** | |
| **Votre mission :** Rattachée ou rattaché au responsable d’activité, vous réalisez des prestations d’analyse, de maintenances préventives, curatives, adaptatives et évolutives des applications existantes avec le souci de respecter les contraintes d’organisation, de contexte, de réglementation et d’architecture cible. | **Contexte professionnel** : Vous travaillez pour le compte d’une entreprise de services du numérique qui assure auprès de ses entreprises clientes des prestations de tierce maintenance applicative (TMA). |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.2. A2D** | **Participer à la maintenance corrective des applications logicielles d’une entreprise cliente** | | | | | |
| Durée totale en heures du scénario pour la séquence  6 h | Vous travaillez pour le compte d’une entreprise cliente et vous assurez, à distance, des prestations de maintenance corrective de ses applications informatiques dans le respect du niveau de service du contrat de TMA. L’équipe informatique sur site est composée de 4 techniciens, dont deux développeurs, dirigés par une responsable informatique spécialisée dans l’administration des systèmes et réseaux. | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Recueillir, analyser et mettre à jour les informations sur une version d’une solution applicative. * Analyser et corriger un dysfonctionnement. * Mettre à jour des documentations technique et d’utilisation d’une solution applicative. | Savoirs technologiques   * Méthodes, normes et standards associés au processus de conception et de développement d’une solution applicative. * Concepts de la programmation objet : classe, objet, abstraction, interface, héritage, polymorphisme, annotations, patrons de conception, interface de programmation d’applications. * Techniques de gestion de versions.   Savoirs économiques, juridiques et managériaux   * Contrat de développement (formation, exécution, inexécution) et ses clauses spécifiques. * Typologie des licences logicielles et droits des utilisateurs. | | * Un outil collaboratif de gestion des versions est utilisé. * Le dysfonctionnement de la solution existante est corrigé selon les procédures en vigueur et dans les délais. * L’application améliorée et/ou corrigée est opérationnelle et stable dans l’environnement de production. * Les composants logiciels adaptés et/ou corrigés sont validés par les procédures de tests unitaires et fonctionnels. | | Prérequis :  B1.2. Répondre aux incidents et aux demandes d’assistance et d’évolution.  Compétence : Traiter des demandes concernant les applications.  Transversalité :  B2.1 Concevoir et développer une solution applicative. |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Vous travaillez en étroite collaboration avec les développeurs internes de l’entreprise cliente. Afin d’harmoniser les plateformes de travail et de test, les développeurs vous ont fourni une liste des outils qu’ils exploitent en interne.  Votre tâche consiste à préparer votre environnement de développement, de débogage et de collaboration en conformité avec les termes des licences des logiciels cibles installés.   1. Installer et/ou mettre à jour l’environnement de développement intégré utilisé dans l’entreprise cliente. 2. Installer l’outil d’analyse statique de code exploité par l’équipe dans l’IDE. 3. Installer l’outil de génération de documentation de code dans l’IDE. 4. Tester l’installation et l’exploitation des outils à travers une application de test. | | * Environnement(s) de développement intégré (*IDE exploité dans le bloc 2*). * Outil(s) d’analyse statique de code (*Lint*, *SonarLint, checkstyle, ou équivalent*). * Procédures d’installation et de configuration des outils cités (*lien vers la ressource de type tutoriel ou vidéo*). * Feuille de route pour une prise en main des outils installés. * Application de test présentant des manquements aux bonnes pratiques de développement. | | * L’environnement de développement intégré est installé ou mis à jour. * Les outils fournis sont installés dans l’IDE (*extensions*) pour assurer la qualité logicielle et la maintenabilité du code. * L’application de test est analysée par l’outil d’analyse statique. * L’application est revue et corrigée. * La documentation technique est mise à jour. * Les bonnes pratiques de développement sont respectées. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 3 h | L’équipe des développeurs utilise un outil de gestion de versions collaboratif.   1. Créer un compte d’accès au serveur de gestion des versions d’applications utilisée par l’équipe. 2. Installer le plugin qui permet d’avoir l’interface graphique du service dans votre IDE. 3. Configurer l’espace de travail dans l’IDE et activer la vue des dépôts. 4. Installer la solution de gestion des versions en local pour gérer les dépôts en local (commit, historique). 5. Prendre en main la solution de gestion des versions en local et sur un serveur distant. 6. Rédiger un rapport d’activités. | | * Fiche des savoirs technologiques portant sur les solutions de gestion de versions des solutions applicatives. * Synthèse des commandes les plus utilisées de la solution de gestion de versions. * Feuille de route pour une prise en main des fonctionnalités de base de la solution de gestion de versions (*liens vers des tutoriels ou des vidéos*). * Application de test. | | * La solution de gestion de versions des applications est intégrée à l’IDE. * La solution de gestion de versions des applications est configurée pour un usage en local et pour des dépôts distants. * L’application de test est poussée sur le dépôt distant. * Les modifications de code (*commits*) sont commentées avec des messages pertinents. * Le rapport d’activité est rédigé et les opérations de suivi et de gestion de versions sont illustrées pour servir d’aide-mémoire. | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | Une application de l’entreprise cliente présente une anomalie. L’application a été développée par un prestataire externe. Un des développeurs a déposé sur le serveur de dépôt le code source de l’application et a renseigné dans l’onglet « *issues* » l’anomalie constatée qu’il vous a soumise.   1. Récupérer l’application du dépôt distant dans l’IDE. 2. Tester l’application et reproduire l’anomalie. 3. Prendre en charge l’anomalie et la corriger. 4. Envoyer les modifications sur le dépôt distant. | | * Fiche des savoirs technologiques portant sur la qualité logicielle (norme ISO 9126), les meilleures pratiques de codage (KISS, DRY, SOLID…) et les techniques de gestion d’incidents. * Fiche des savoirs relatifs aux types de maintenance au sens de l’AFNOR, au contrat de TMA et aux niveaux de services. * Application présentant une anomalie liée à plusieurs dysfonctionnements mettant l’accent sur les impacts engendrés par la non application des meilleures pratiques de codage. * Description de l’anomalie : contexte, cas d’utilisation concerné, comportement attendu et constaté, priorité de l’anomalie. | | * L’anomalie est traitée en tenant compte de sa priorité et des termes du SLA (*Service Level Agreement*). * L’origine de l’anomalie est détectée et corrigée. * Les éléments adaptés et corrigés sont testés. * L’application est opérationnelle dans l’environnement de test et poussée sur le dépôt distant. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.2. B2D** | **Participer à la qualité logicielle du patrimoine applicatif d’une entreprise cliente** | | | | | |
| Durée totale en heures du scénario pour la séquence  4 h | Vous assurez des prestations de maintenance curative auprès d’une entreprise cliente sur son site. Le patrimoine applicatif compte plusieurs applications comportant du code hérité (*legacy code*). Le responsable de la qualité logicielle a réalisé une cartographie de ces applications suite à un audit de qualité des solutions applicatives développées en interne ou par des prestataires.  Votre mission consiste à participer au nettoyage de code d’une application en mettant en œuvre les bonnes pratiques de codage et les techniques de revue de code afin d’en améliorer la qualité logicielle et la maintenabilité. | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Évaluer la qualité d’une solution applicative. * Analyser et corriger un dysfonctionnement. * Recueillir, analyser et mettre à jour les informations sur une version d’une solution applicative. * Mettre à jour des documentations technique et d’utilisation d’une solution applicative. | Savoirs technologiques   * Méthodes, normes et standards associés au processus de conception et de développement d’une solution applicative. * Concepts de la programmation objet : classe, objet, abstraction, interface, héritage, polymorphisme, annotations, patrons de conception, interface de programmation d’applications. * Fonctionnalités avancées d’un environnement de développement. * Techniques de gestion de versions. * Architectures applicatives : concepts de base et typologies. | | * L’évolution de la solution applicative répond aux besoins exprimés dans le cahier des charges. * Un outil collaboratif de gestion des versions est utilisé. * Le dysfonctionnement de la solution existante est corrigé selon les procédures en vigueur et dans les délais. * L’application améliorée et/ou corrigée est opérationnelle et stable dans l’environnement de production. * Des composants logiciels sont adaptés pour améliorer la qualité de la solution applicative. * Les composants logiciels adaptés et/ou corrigés sont validés par les procédures de tests unitaires et fonctionnels. | | Prérequis :  B1.2. Répondre aux incidents et aux demandes d’assistance et d’évolution.  Compétence : Traiter des demandes concernant les applications.  B2.1 Concevoir et développer une solution applicative, séquence 2.1 C2D  B2.2 A2D |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | 1. Récupérer l’application du dépôt distant dans l’IDE. 2. Corriger toutes les erreurs non bloquantes détectées par l’outil d’analyse statique de code. 3. Appliquer les techniques de factorisation sur le code de l’application. 4. Pousser l’application sur le dépôt distant à chaque correction. | | * Fiche des savoirs technologiques portant sur la qualité logicielle (norme ISO 9126), les meilleures pratiques de codage (KISS, DRY, SOLID…) et les techniques de gestion d’incidents. * Environnement de développement de la séquence 2.2 A2D. * Application de test avec du code hérité (*legacy code*). * Description des fonctionnalités pertinentes. | | * L’application de test est analysée par l’outil d’analyse statique. * La revue de code manuelle est réalisée. * Le code de l’application est corrigé et amélioré. * Les versions de l’application sont poussées de manière incrémentale sur le dépôt distant. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | 1. Implémenter des tests unitaires. 2. Réaliser des tests fonctionnels. 3. Corriger les erreurs de l’application. 4. Pousser l’application sur le dépôt distant à chaque correction. | | * Fiche des savoirs technologiques de la POO. * Fiche des savoirs technologiques portant sur les techniques de tests logiciels (unitaires, fonctionnels). * Application améliorée avec les corrections de la séance précédente. | | * Le rapport de tests unitaires est rédigé. * Le rapport de tests fonctionnels est rédigé. * Le code de l’application est corrigé et amélioré. * Les versions de l’application sont poussées de manière incrémentale sur le dépôt distant. | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | 1. Réaliser la revue de code de l’application en s’appuyant sur les principes SOLID. 2. Générer la documentation technique de l’application. 3. Générer la documentation d’utilisation d’une fonctionnalité corrigée. | | * Fiche des savoirs technologiques portant sur la qualité logicielle, les meilleures pratiques de codage et les techniques de gestion d’incidents. * Fiche des savoirs technologiques de la POO. * Application de test améliorée avec les corrections de la séance précédente. | | * Les éléments mis à jour sont testés. * L’application est améliorée. * Le rapport de la revue de code est rédigé. * L’application corrigée est poussée sur le dépôt distant. * La documentation technique est mise à jour. * La documentation d’utilisation est mise à jour. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.2. C2D** | **Participer à la maintenance évolutive et adaptative des solutions applicatives d’une entreprise cliente** | | | | | |
| Durée totale en heures du scénario pour la séquence  6 h | Vous travaillez dans les locaux d’une jeune pousse spécialisée dans la fourniture de solutions digitales et vous assurez des prestations de maintenance évolutive et adaptative pour la cellule informatique composée de cinq développeurs.  Votre mission consiste à faire évoluer l’architecture et le déploiement d’une solution applicative client/serveur destinée à la cellule informatique. Le projet sera réalisé en mode agile avec des *sprints* de 2h sous la supervision du producteur du produit (PDP). | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Recueillir, analyser et mettre à jour les informations sur une version d’une solution applicative. * Évaluer la qualité d’une solution applicative. * Analyser et corriger un dysfonctionnement. * Élaborer et réaliser les tests des éléments mis à jour. | Savoirs technologiques   * Méthodes, normes et standards associés au processus de conception et de développement d’une solution applicative * Architectures applicatives : concepts de base et typologies * Techniques et outils d’analyse et de rétro-conception * Composition du coût d’une solution applicative * Interfaces homme-machine : principes ergonomiques, techniques de conception, d’évaluation et de validation. * Concepts de la programmation objet. * Concepts de la programmation événementielle. * Persistance et couche d’accès aux données. * Techniques et outils de documentation. * Techniques de gestion des erreurs et des exceptions. * Fonctionnalités avancées d’un environnement de développement. * Techniques de gestion de versions.   Savoirs économiques, juridiques et managériaux   * Réglementation en matière de collecte, de traitement et de conservation des données à caractère personnel. | | * L’évolution de la solution applicative répond aux besoins exprimés dans le cahier des charges. * Un outil collaboratif de gestion des versions est utilisé. * Des composants logiciels sont adaptés pour améliorer la qualité de la solution applicative. * Les composants logiciels adaptés et/ou corrigés sont validés par les procédures de tests unitaires et fonctionnels. * Les accès aux données persistantes à travers le langage de requête du système de gestion de base de données relationnel, le langage de requête proposé par les échanges applicatifs des technologies *Web*, le langage de requête de l’outil de correspondance objet-relationnel ou toute autre solution de persistance sont mis à jour. * La documentation technique et d’utilisateurs de la solution applicative sont mises à jour. * L’application améliorée et/ou corrigée est opérationnelle et stable dans l’environnement de production. | | Prérequis :  B1.5 Mettre à disposition des utilisateurs un service informatique.  B2.1 Concevoir et développer une solution applicative :  séquences 2.1 C2D et 2.1 D2D. |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | L’architecture de l’application ne respecte pas le modèle MVC.   1. Récupérer le code de l’application depuis le dépôt distant. 2. Tester l’application dans l’état. 3. Migrer l’application vers une architecture MVC. 4. Tester l’architecture MVC. 5. Pousser les modifications pertinentes sur le dépôt distant. | | * Fiches dse savoirs technologiques des concepts de la POO, des concepts d’IHM, des architectures applicatives. * Code de l’application (*l’application est opérationnelle et ne doit pas être complexe*). * Script de la base de données. * Serveur de la base de données (physique, virtualisé ou conteneurisé). | | * Le serveur de base de données est installé et configuré. * La base de données est installée. * L’accès à la base de données est opérationnel. * L’application est développée dans le respect de l’architecture MVC et des règles de bonnes pratiques. * L’application est opérationnelle dans l’environnement de test. * Les modifications incrémentales sont poussées sur le dépôt distant sur une branche distincte. * Les modifications sont validées par le PDP et fusionnées dans la branche principale. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | Un indicateur de supervision du serveur de bases de données remonte régulièrement des alertes alors que le seuil n’est pas atteint.  Une réflexion a été menée et, au vu des besoins actuels et futurs, le client souhaite migrer le serveur vers une solution en nuage (*cloud*).   1. Effectuer une étude comparative de cette migration du serveur et des données auprès de 4 fournisseurs de cloud. La comparaison portera sur le volume de stockage, le nombre de bases de données, la capacité maximale d’une base de données, le nombre de processeurs, le respect du RGPD, la qualité de service, la sécurité, la formation, le coût et les services annexes qui peuvent être inclus dans la proposition. 2. Tester le déploiement du serveur auprès du fournisseur qui propose le meilleur service. 3. Rédiger une proposition argumentée à votre client de la solution à retenir. | | * Fiche des savoirs technologique relatifs au *cloud*. * Fiche des savoirs juridiques relatifs au RGPD. | | * Un tableau comparatif de la migration du serveur et des données dans le *cloud* est réalisé. * Une proposition argumentée de la solution à retenir est rédigée pour le client | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | Le client a choisi son fournisseur *cloud* et souhaite désormais migrer l’application dans le *cloud*.   1. Déployer le serveur de base de données dans le *cloud*. 2. Migrer les données dans le *cloud*. 3. Déployer l’application dans le *cloud*. 4. Tester le fonctionnement de l’application dans l’environnement de production. 5. Vérifier le coût de la solution dans le *cloud* pour un usage intensif durant 1h. | | * Fiche des savoirs technologique relatifs au *cloud*. * Procédure d’installation du serveur de base de données dans le *cloud*. * Procédure de déploiement de l’application dans le *cloud*. * Procédure de migration de la base dans le *cloud*. | | * L’application est déployée dans le *cloud*. * Le serveur de base de données est virtualisé dans le *cloud*. * La migration des données dans le *cloud* est opérationnelle. * L’application est testée dans le *cloud*. * Une synthèse de la facturation de la solution est générée. | |