

Intégration et travail en équipe

Durée : 1 jour

Objectifs

Cette journée a pour objectif :

- L'intégration des participants au groupe en mettant en valeur l'importance de réussir cette phase dans toute prise de fonction
- Une prise de conscience de l'importance de la dimension relationnelle et comportementale dans l'atteinte des objectifs pour les amener à accorder de l'attention dans la relation à des aspects comme l'écoute, la disponibilité, la confiance, l'esprit de coopération, etc.
- Une valorisation de l'impact de la méthodologie choisie sur l'atteinte des objectifs pour amener les participants à développer une approche structurée de leur travail.
- Un accompagnement à l'ancrage d'une projection individuelle et collective positive à l'issue du parcours.

Contenu

Accueil et Présentation du Parcours

- Présentation du Groupe Global Knowledge
- Présentation du Parcours de formation et de ses moments clés
Présentation des règles générales d'organisation

Activité d'intégration et de construction d'une équipe collaborative

Dans un cadre décontracté et une ambiance de jeu, les participants sont accompagnés tout au long de la journée à travers des jeux pédagogiques et ludiques pour les amener à s'intégrer et à prendre conscience de l'importance des dimensions non techniques et de leur impact sur l'atteinte de leurs objectifs et sur leur réussite dans leur rôle.

- Tirage au sort et organisation en équipes
- Jeu 1 : Plus tu connais plus tu gagnes
 - ✚ Principe : présentation croisée des participants
 - ✚ Objectifs pédagogiques : Se connaître les uns les autres en mettant en relief la richesse du groupe
- Jeu 2 : Les 5 objets
 - ✚ Principe : Transcription d'un schéma d'objets par commande distante
 - ✚ Objectifs :
 - Mettre en évidence l'importance de la phase de stratégie et planification dans tout projet
 - Analyser les causes possibles des écarts entre le message transmis et le message reçu
 - Valorisation de sens du détail et de la gestion optimisée du temps
- Jeu 3 : Le cercle des objets
 - ✚ Principe : Transmission de multiples objets au sein du groupe avec diverses contraintes
 - ✚ Objectifs pédagogiques :
 - Développement de la capacité à gérer la pression et le stress lié à la multiplicité des objets et à leur spécificité
 - Respect du récepteur dans l'acte de transmission et intégration de ses contraintes
 - Lien avec l'équilibre vie professionnelle/vie professionnelle d'une part et avec le renvoi des balles dans l'entreprise
- Jeu 4 : Le mur de la confiance
 - ✚ Principe : Aller vers un « mur de collègues »
 - ✚ Objectifs pédagogiques :
 - Prise de conscience des participants de leur distance spontanée de la notion de confiance quand le vis-à-vis en est incontestablement digne
 - Valoriser la dimension stratégie de l'équipe et son impact sur la performance individuelle et collective
- Jeu 5 : La corde au carré
 - ✚ Principe : Construction d'un carré parfait
 - ✚ Objectifs pédagogiques :
 - Renforcement de l'importance de l'organisation, de la planification et de la stratégie de groupe dans l'atteinte des objectifs
 - Mise en évidence des aptitudes du groupe à l'application minutieuse d'un plan de déploiement préétabli
 - Valorisation de la capacité du groupe à rebondir en cas d'imprévu
- Jeu 6 : La tour de papier
 - ✚ Principe : Construction de la plus haute tour
 - ✚ Objectifs pédagogiques :
 - Développement de la stratégie et de la différenciation
 - Accompagnement des participants à la valorisation du sens du détail et de la rigueur ainsi que leur impact sur

-
- la performance
 - Renforcement du sens d'équipe et de l'importance de la planification et de l'organisation et la répartition des rôles
 - Jeu 7 : Hymne du Groupe
 - ✚ Principe : Construire une chorégraphie d'équipe avec une chanson autour du métier
 - ✚ Objectifs pédagogiques :
 - Construction d'équipe
 - Activation des compétences de créativité des participants
 - Accompagnement des participants à la mise en avant des dimensions de leur futur métier
 - Valorisation du métier et projection positive des participants
 - Ancrage de l'émotion positive à travers la musique et la chorégraphie

Visualisation et débriefing

- La visualisation des différents exercices permet aux participants de prendre du recul face à leurs attitudes et comportement.
- Accompagnement vers une synthèse des enseignements de la journée autant à l'échelle individuelle que collective.
- Mise en avant de l'importance des notions suivantes :
 - ✚ Intégration au sein d'un nouvel environnement
 - ✚ Cohésion d'équipe
 - ✚ Travail en équipe
 - ✚ Stratégie et organisation du travail
 - ✚ Sens du détail et rigueur dans la mise en œuvre d'un plan d'action
 - ✚ Le récepteur est central dans l'acte de communication
 - ✚ Plaisir et travail

Le plus de cette formation

Pédagogie ludique et participative qui accélère les rencontres et la construction de l'équipe des stagiaires.

Introduction à la gestion de projets

Durée : 1 jour

Résumé

Cette formation théorique permettra de sensibiliser les stagiaires à la gestion de projets.

Public visé

Toute personne amenée à travailler en mode projet.

Prérequis

Pas de prérequis particulier.

Contenu :

Le logiciel

- Qu'est-ce qu'un logiciel
- Classification des logiciels
- Impact de la qualité d'un logiciel
- Conception
- Codage
- Tests
- Maintenance

Le développement de logiciels

- Qu'est-ce que c'est
- Qu'est-ce qu'un bon logiciel du point de vue du client
- Que faut-il pour le développement
- Définitions : MOA/MOE

Le Génie logiciel

- Echs des projets de développement
- Les difficultés
- Qu'est-ce que le génie logiciel

Activités de développement

- Les étapes d'un développement
- Les principales activités
 - Analyse des besoins

Cycle de vie de logiciels

- Procédé logiciel
 - Modèles de procédés : Méthodes classiques/Méthodes agiles
 - Modèle en cascade
 - Modèle en fontaine
 - Les limites des cycles de vie linéaires
 - Le modèle en V
 - Le modèle en W
 - Les limites des cycles de vie en V
 - Le modèle incrémental
 - Le modèle en spirale
 - Principes des méthodes Agile
-

Informations complémentaires

Support de cours remis aux participants

Les fondamentaux de la méthode Agile SCRUM

Durée: 2 Jours

Résumé:

L'objet de la formation "Les fondamentaux" est de permettre à toute personne, quel que soit son poste dans l'entreprise et son rôle dans le projet Agile, de comprendre la démarche Agile et la rupture culturelle qu'elle induit par rapport aux méthodes classiques de gestion de projets.

Public visé:

Tous les candidats désirant s'initier aux méthodes agiles (les profils pouvant être très hétérogènes). Ce module est un prérequis pour de nombreuses autres formations - certifiantes ou non.

- Développeurs
 - Architectes
 - Chefs de projets
 - Directeurs de projets
 - Futurs Scrum Masters/ Futurs Managers Agiles
 - Responsables Méthodes/Qualité
 - Décideurs
 - MOA/Client/Product Owner
 - Commerciaux
-

Objectifs:

- Comprendre cette rupture
 - Faire un choix de méthode en gestion de projet
 - Parler des méthodes agiles et de les faire partager
-

Pré-requis:

Aucun pré-requis si ce n'est l'envie de découvrir l'agilité.

Contenu:

Serious Game « Icebreaker Agile » 1

Le "Icebreaker Agile" a pour but de démarrer la journée, d'amener les participants à exprimer par leur choix ce que représentent les méthodes Agiles pour eux et partager ces représentations avec le groupe. Ils expérimentent l'auto-organisation, la limitation du temps (le timeboxing) ainsi que la possibilité d'échouer à atteindre l'objectif prévu pour une itération.

Le mouvement Agile

- Les origines d'Agile, la culture Agile, le manifeste Agile, les principes Agile
- Le cycle de vie Agile, et en quoi il diffère du cycle de vie classique
- Les différentes méthodes classiques (PMP, Prince2), leurs avantages et leurs inconvénients par rapport aux méthodes agiles
- Illustration de la supériorité des méthodes Agile sur les méthodes classiques au travers d'un cas d'utilisation spécifique

L'Agilité en action

- Panorama des méthodes agiles et de leurs apports respectifs (les 10 méthodes agiles et leurs spécificités propres)
- Présentation des deux méthodes agiles les plus répandues: SCRUM et XP; leurs similarités, leurs différences
- Présentation de méthodes connexes à Agile (Lean IT, Kanban)

Mise en Œuvre d'Agile

- Le changement dans l'entreprise, les résistances traditionnelles au changement
- Les freins au déploiement des méthodes agiles, les spécificités de leurs domaines d'application, leurs risques particuliers, les contraintes particulières à Agile
- Les mythes sur Agile, les mauvaises applications d'Agile et les facteurs clés de succès d'une implémentation réussie d'Agile

Les participants seront amenés à travailler en équipe, notamment sur des serious games (Icebreaker Agile) pour faire comprendre les valeurs Agile, leurs avantages et mettre en pratique les acquis du cours.

Autre(s) Information(s):

Un support de cours papier sera remis à chaque participant, reprenant les éléments, concepts, jeux et outils présentés lors du cours. Ce support est destiné à favoriser la mise en place par le participant des acquis de la formation au sein de leur entreprise.

Le rôle du Responsable produit dans un projet Agile

Durée: 2 Jours

Résumé:

Cette formation permettra aux futurs Product Owner de comprendre leur rôle tant vis-à-vis du client que de l'équipe et l'impact de leurs actions sur le produit.

Les participants apprendront à se positionner par rapport au Scrum Master et aux autres membres de l'équipe. Ils sauront comment gérer les Product Backlog, rédiger et hiérarchiser les Users Stories, valider les livraisons de chaque sprint.

Public visé:

Ce module est obligatoire pour les candidats ayant les rôles ou fonctions suivantes: Fonctionnels, MOA, Responsable de produit/ Product Owner, Scrum Master, s'il doit apporter du support au client/Product Owner

Ce module est recommandé pour les candidats ayant les rôles ou fonctions suivantes : Managers, Responsable Qualité/Méthodes

Objectifs pédagogiques:

- | | |
|--|---|
| ■ Cette formation permettra aux futurs Product Owner de | ■ Se préparer au passage de la certification PSPO Niveau 1 |
| ■ Comprendre l'investissement du Product Owner | ■ Cette formation apporte les connaissances nécessaires pour qu'un Product Owner soit capable de débiter dans son rôle. La formation comporte des exercices et des Serious Games (Environ 70% de théorie et 30% de pratique). |
| ■ Comprendre son rôle par rapports aux parties prenantes du projet (équipe/Client) | ■ La compétence du Product Owner deviendra plus forte au fur et à mesure qu'il s'impliquera dans les projets. |
| ■ Savoir hiérarchiser les exigences et savoir agir au bon moment | |
-

Pré-requis:

- Posséder les aptitudes relationnelles (ouverture d'esprit et capacités de leadership) en vue de devenir un Manager Agile.
 - Avoir suivi nécessairement le module de formation GKAG01 "[Méthodes Agiles : comprendre la démarche](#)"
-

Contenu:

Serious game choisi spécifiquement pour ce module : « FIFATIX »

Ce jeu a été conçu pour permettre de découvrir le rôle de Product Owner ou servir de support à un échange d'expériences entre Product Owners pratiquants. Il illustre également ce qu'est la gestion de projet Agile à un public qui se situe plutôt du côté métier.

Organisation Scrum

- L'importance de l'Agilité dans tous les métiers et tous les secteurs
- Les différentes raisons d'être plus Agile
- Agile= Itératif + Incrémental : le bon choix de la bonne approche
- Prédictive Versus Itérative
- Le Framework Scrum (Piliers, Valeurs, Rôles, Evénements, artefacts, Done)
- Les caractéristiques du Product Owner (video)

Identification du rôle et des activités du Responsable Produit Agile (Product Owner)

- Activités dont est responsable le Product Manager dans une entreprise
- Le Product Manager comme Mini CEO
- La fonction de Product Management
- Construire un produit Agile au travers de la planification de scénarios : Identifier la problématique, Identifier les forces en présence, Dessiner le futur, Suivre les indicateurs et s'adapter
- La Vision Produit
- Release et création de Valeur
- « Time to Market » et « First takes all »
- Agir dans l'incertitude

Relation avec le Manager Agile (Scrum Master) et les autres membres de l'équipe

- Les différents rôles et responsabilités dans Scrum
- Définir la Valeur et délivrer la Valeur
- Points clés du Rôle du Product Owner au sein de l'équipe Scrum
- Faire des choix clairs pour optimiser la Valeur
- Rôle du Product Owner au sein du Sprint planning
- Rôle du Product Owner au sein du Sprint
- Rôle du Product Owner au sein de la Sprint Review
- Transparence et amélioration du Done
- Rôle du Product Owner au sein de la Sprint Rétrospective
- Difficulté du rôle de Product Owner et étendue de son rôle
- Les écueils rencontrés par le Product Owner

Planification des activités (Product Backlog/Sprint Backlog)

- Planifier une Release
- L'horizon limité d'Agile, vu comme un avantage
- De la Roadmap au Product Backlog
- Différents types de Releases
- Capacité d'absorption du client
- Vitesse de l'équipe
- Contraindre par le temps
- Contraindre par le coût

Les prévisions en Agile

Expression du besoin, Stratégie produit, Gestion des exigences (Product Backlog)

- Caractéristiques d'un Product Backlog
- Type de composants d'un Product Backlog
- Horizon d'un Product Backlog (3 sprints « ready » maximum)
- Priorisation d'un Product Backlog par contrainte : Business, Développement Implémentation, Dépendance, Engagement, Complexité, Effort

ROI et Valeur métier

- Différence entre Prix et Valeur
- Mesurer le succès
- Index d'usage et « Minimum Viable Product »
- Key Performance Indicators (KPI)

Méthodes pédagogiques :

Un support de cours est mis à disposition de chaque participant via un accès en ligne personnalisé. Il reprend les éléments, concepts, jeux et outils présentés lors du cours. Ce support est destiné à favoriser la mise en place par le participant des acquis de la formation au sein de leur entreprise.

Nous privilégions un support rédigé en anglais afin de faciliter la préparation des participants au passage de la certification, dont l'examen n'est disponible qu'en anglais. La formation est délivrée en français par le formateur mais celui-ci présentera et utilisera les termes anglais spécifiques à Scrum.

Le Scrum Guide 2020 sera également mis à disposition pour chaque participant

Préparation à la certification Agile Scrum Product Owner

Durée: 1 Jour

Résumé:

Le haut niveau des certifications Scrum.org nécessite de se préparer sérieusement. C'est pourquoi la journée de préparation à l'examen de certification Professional Scrum Product Owner permettra aux candidats désirant maximiser leurs chances de réussite de trouver les ressources nécessaires. Bien que la journée de préparation ne soit pas un gage de réussite à 100%, elle contribue largement à la réussite des candidats.

Public visé:

Toute personne ayant suivi la formation « Devenir Responsable de Produit Agile » et dont l'objectif est de passer la certification dans les jours qui suivent ce module.

Objectifs pédagogiques:

- Comprendre la structure de l'examen PSPO de Scrum.org
- Se sentir plus en confiance pour le passage de la certification
- Se retrouver en situation de stress pour le passage de l'examen et y faire face
- Développer son aptitude à répondre rapidement aux questions

Pré-requis:

Avoir suivi la formation « Devenir Product Owner Agile »

Contenu:

La certification Scrum.org : « Trucs et astuces »

- Les stagiaires disposent d'un accès à une banque de questions via la plate-forme Web de test que Global Knowledge a développé spécifiquement.
- L'accès à la plate-forme sera disponible à partir de la journée de préparation et cela jusqu'au passage de l'examen. Le candidat pourra donc disposer des ressources de la plate-forme tout le temps nécessaire à partir de chez lui pour passer la certification lorsqu'il aura atteint le niveau adéquat.

Les questions de test se concentrent particulièrement sur les connaissances et compétences suivantes :

- La Gestion de produit agile
- Le développement axé sur la valeur
- Les Principes Scrum ; Empirisme
- Le cadre Scrum
- La Gestion du backlog de produit
- La Gestion des versions

Analyses des réponses

JIRA et gestion de projet Agile

Durée: 2 jours

Résumé:

Jira est une gamme de produits de la société Atlassian conçus pour aider tous les types d'équipes à gérer leur travail. Jira procure plusieurs produits et options de déploiement qui ont été spécifiquement pensés pour les équipes de développement, informatiques, métier, opérationnelles et bien d'autres.

Cette formation porte sur deux modules de JIRA :

- Jira Service Desk : gestion des demandes de service et des anomalies,
 - Jira Software : gestion de projet de développement Agile
-

Public visé

Ce module s'adresse aux professionnels des tests MOA / MOE : testeurs, analystes, chefs de projet, toute personne déjà formée à ISTQB Foundation et souhaitant préparer la certification ISTQB® Niveau Foundation V2018 (CTFL- Testeur de Logiciel Certifié) Cette formation s'adresse aux professionnels de l'activité de tests de logiciels (Maîtrise d'Ouvrage, Maîtrise d'œuvre, Services et Exploitation, Utilisateurs finaux) souhaitant mettre en place un référentiel de test outillé.

Objectifs :

Les objectifs de la formation sont de :

- Apprendre à utiliser le module de gestion des demandes et anomalies.
 - Apprendre à gérer les différentes entités d'un projet Agile : release, sprints, équipes, backlog product, user stories, tâches, anomalies, rapports
-

Pré-requis :

Connaître les fondamentaux d'un projet Agile Scrum.

Contenu :

PARTIE 1 : gestion des demandes et anomalies

- Présentation de Jira
- Fonctionnalités
- Création d'un projet
- Création des demandes
- Création d'une anomalie
- Rapports

- Création d'un projet
- Gestion du backlog product
- Création d'une release
- Création et planification d'un sprint
- Affectation et traitement des demandes
- Gestion des anomalies
- Clôturer les tâches
- Reporting

PARTIE 2 : gestion de projet de développement agile

Travaux pratiques

Partie 1 - Gestion des demandes de service et des anomalies :

- Après la théorie, mise en pratique sous forme de jeu en équipes.
- Sous la forme de questions / réponses sur le thème de ISTQB Foundation (ou tout autre thème selon les formations déjà suivies).
- Les équipes gagnent des points selon leur rapidité à saisir leurs questions dans Jira sous forme de demandes, puis à répondre rapidement et correctement aux questions d'une autre équipe en complétant les demandes sous Jira.

Partie 2 - Gestion de projet de développement agile :

- Alternance d'exercices entre chapitres théoriques sur un thème transverse (application web d'achat en ligne de musiques) en guise de fil conducteur, permettant une progression dans l'application des activités d'un projet agile.
 - Espace Pilotage
 - Administration
-

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

Algorithmique

Durée : 2 jours

Résumé

Cette formation permet de s'approprier les structures logiques de base ainsi qu'une démarche de construction de solutions à partir d'un problème

Public visé

Développeurs

Prérequis

Aucune connaissance informatique n'est nécessaire pour ce stage

Contenu

L'algorithme dans le contexte projet

- De la demande utilisateur à la réalisation du logiciel.
- Qu'est-ce qu'un algorithme ?
- En quoi est-il indispensable en programmation ?
- Qu'entend-on par programmation procédurale ?
- Qu'en est-il de la programmation événementielle, de la programmation objet ?

L'algorithmique de base

- Intérêt de l'utilisation d'un langage symbolique.
- Les séquences alternatives simples et itérations simples.
- Les opérateurs logiques ET, OU et NON : Algèbre de Boole et loi de Morgan.
- Analyse : la démarche descendante par étapes.
- Les tableaux à une dimension.
- Les alternatives et débranchements complexes.
- Les différents types de données.
- **Mise en pratique : 5 Workshop de création d'algorithme + Exercices « L'enquête comparative et « Le palindrome »**

L'analyse d'un programme

- Intérêt de la structuration d'un programme.
- La modularisation d'un programme.
- La structuration d'un programme.
- Intervention en maintenance dans un programme structuré.
- Modules externes, procédures et fonctions.
- Différence entre procédure et fonction ?
- Transmission de paramètre.
- **Mise en pratique : 3 workshops +° Exercice « La compétition internationale »**

Les tests

- Intérêt et mise en œuvre.

Les fichiers et les bases de données

- Les fichiers : notions d'enregistrements, de rubriques.
- Les différentes organisations de fichiers.
- Etude détaillée des fichiers séquentiels et séquentiels indexés.
- Les bases de données relationnelles : organisation et manipulation des données.

La conception orientée objet

Durée : 2 jours

Résumé

Cette formation permet d'acquérir les connaissances de base solides sur la conception objet et celle basée sur les composants, en manipulant des outils de conception et de développement. Elle présente et démystifie le vocabulaire de cette technologie, souvent considéré, par les développeurs de l'approche procédurale, comme complexe.

Les participants seront capables de comprendre et de raisonner en classes et objets, d'appréhender la notation UML et la nécessité de la modélisation. Il s'agit d'assimiler les architectures et technologies objets avant d'entreprendre l'étude d'un langage objet ou des développements "orientés objets".

Objectifs

A l'issue de la formation, les stagiaires seront en mesure de :

- Connaître les concepts essentiels de l'Objet pour faciliter l'approche de tout langage de programmation Orienté Objet.
- Connaître les manipulations de base des outils de conception et développement (Eclipse, Visual Studio...).
- Appréhender la modélisation UML pour l'analyse fonctionnelle et technique.

Public visé

Développeurs, Concepteurs, Chefs de projets, Architectes techniques, Testeurs

Prérequis

Les participants doivent avoir des connaissances générales en informatique, notamment en matière de conception d'applications.

Pédagogie

Cette formation permet aux participants une mise en pratique progressive de tous les concepts objets, via un langage objet dans un outil de développement. Une application simple sera mise en œuvre afin d'aborder la conception et la modélisation objet et d'en comprendre les tenants et les aboutissants. Le participant sera ainsi opérationnel pour débiter dans le développement d'application et ce quel que soit le langage objet choisi, l'outil de développement et le type d'application à mettre en œuvre (Standalone, Web, mobile,...).

40% Théorie 60% de pratique

Contenu :

Objectifs de la conception Objet

- Historique de l'objet.
- L'objet par rapport à l'approche procédurale.
- Les principes fondamentaux de la conception orientée "objet".
- La nécessité de modéliser, la notation UML, les outils de modélisation.

Conception d'applications objets, l'industrie de l'objet

- Le vocabulaire : Objet – Composant – Framework
- Les Framework Java EE et .Net.
- Les langages Objet (Java, C++, C#, VB net, PHP...).
- L'objet et les bases de données relationnelles.
- Le modèle en couches dans les architectures n-tiers. Le design pattern MVC (Modèle – Vue – Contrôleur), les frameworks implémentant le MVC.
- Les objets métier, les composants.

- Le principe de la réutilisation des objets dans différentes applications (client lourd, client léger, client riche).
- Les outils de développement, les IDE (Integrated Development Environment) : Eclipse, NetBeans, Visual studio...

Les tests dans la conception d'applications objets

- Les tests avec les cas d'utilisation d'UML.
- Les outils de tests, l'intégration avec les outils de conception et de développement.
- Les outils de debug.
- Intérêt dans la conception de séparer le conceptuel (le modèle) du visuel (la présentation).
- **Travaux Pratiques** : Démonstration des outils, via la mise en œuvre dans l'application de gestion commerciale de la classe Client.

Classes – Objets - Instanciation

- Les packages et les espaces de noms, les bibliothèques.
- Les classes, les attributs, les méthodes (constructeurs, destructeurs et accesseurs).
- L'instanciation, les objets.
- **Travaux Pratiques** : Manipuler un outil de développement, écrire du code, mettant en œuvre la classe *Produit* de la gestion commerciale en respectant le pattern MVC.

Encapsulation

- Définition et intérêt.
- Visibilité des variables
- Mise en œuvre de l'encapsulation.
- **Travaux Pratiques** : Modifier la classe *Produit* en respectant l'encapsulation. Mettre en œuvre plusieurs constructeurs. Effectuer des tests. Faire évoluer le modèle de la gestion commerciale sans impact sur l'application cliente.

Liens entre les classes : Composition, agrégation

- Définition et utilité des liens de composition et d'agrégation.
- La représentation UML des liens dans le diagramme de classe.
- **Travaux Pratiques** : Conception de classes de composition et d'agrégation pour la gestion commerciale. Intégration des liens dans la classe *Produit*. Impacts sur la maintenance de l'application cliente.

Liens entre les classes : Héritage simple et multiple

- Définition et utilité des liens de généralisation/spécialisation.
- Le polymorphisme, la surcharge.
- La représentation UML des liens d'héritage.
- Les classes abstraites.
- Les références à l'objet.
- L'héritage multiple et les limites dans les langages Objet.
- **Travaux Pratiques** : Evolution de notre modèle objet, conception de deux classes filles, de la classe *Produit*. Manipulation des méthodes héritées, redéfinition d'une méthode de la classe mère dans une classe fille, utilité de méthode abstraite, maîtriser l'intérêt de l'héritage.

Les interfaces

- Définition et intérêts des interfaces.
- L'héritage multiple avec les interfaces.
- Les interfaces dans les architectures distribuées.
- **Travaux Pratiques** : Mettre en œuvre un héritage multiple sur les classes filles de notre *Produit*, en concevant et manipulant une interface, afin de comprendre le polymorphisme.

Travaux pratiques

Définition du produit

- Faire la représentation uml
- Définir le code java pour la classe produit
- Définir le code java pour tester la classe produit

Agrégation ou composition

- Faire la représentation uml
- Définir la classe dénomination
- Modifier la classe produit
- Modifier la classe testproduit

Héritage simple

- Faire le diagramme de classes
- Définir les classes *Produitperissable* et

Produitsaisonnier

- Créer la classe testproduits

Polymorphisme

- Modifier la classe produitsaisonnier
- Classe testproduits

Héritage multiple

- Création de l'interface saison
- Création de la classe produitfugace
- Modification de la classe produitsaisonnier
- Modification de la classe testproduits
- Création de la classe testsaisons

Autre(s) Information(s):

Un support de cours électronique en français sera remis aux stagiaires.

Introduction à Java

Durée : 2 jours

Résumé

Cette formation permet de prendre en main l'environnement JAVA.

Public visé

Administrateurs systèmes et réseaux.

Prérequis

Il est nécessaire d'avoir suivi la formation GKCOBJ « Les concepts : de la modélisation au développement objet » ou posséder les connaissances équivalentes.

Contenu :

Introduction Java, Java SE

- Les origines de Java, son historique
- Le JDK (Java Development Kit), ses versions
- L'outil de développement : Eclipse
- La plate-forme Java SE 8, Standard Edition
- Le fonctionnement de Java, la machine virtuelle (JVM), les exécutable, la gestion de la mémoire (garbage collector)
- La documentation, Javadoc

Le langage de programmation Objet de Java

- Les packages, les classes, les méthodes
- Les objets, les variables, les composants
- La création du projet Java sous Eclipse

Java EE : Les architectures logicielles et les techniques

- Architecture logicielle et technique en couches
- La plate-forme Java EE Enterprise Edition
- Les principaux composants Java EE (EJB, Servlet, JSP)
- Les containers et leurs utilités
- Java embarqué, Android

Les archives Java et le déploiement d'applications

- Les archives .jar, .war, .ear
- Les serveurs d'applications

L'architecture Web

- Mise en œuvre d'un projet Web
 - Servlet – JSP – Java Bean
 - Déploiement sous Apache - Tomcat
-

Autre(s) Information(s)

Un support de cours sera remis aux stagiaires

HTML5, CSS3, JavaScript : développement client

Durée: 3 Jours

Résumé:

Initialement destinés au partage d'information, le Web et les standards d'Internet ont démontré leur capacité de s'imposer pour la mise en œuvre d'applications d'entreprise. Cette formation vous permet de maîtriser le développement de sites Web, dans le respect des standards d'interopérabilité et d'accessibilité du W3C. Le langage HTML devenu universel, comme support de communication entre Clients et Serveurs, il est nécessaire de le maîtriser. Cette formation est le point d'entrée pour tous les types de développement Web (Java, PHP, C++, C# ...) respectant le responsive design, le modèle MVC. Elle vous apprend à créer des IHM riches. Elle aborde tous les concepts importants des langages et de leurs évolutions HTML, CSS, JavaScript et introduit les frameworks Bootstrap, Ajax pour une meilleure dynamique du site et une conception « mobile first ».

Cette formation s'appuie sur de nombreux exemples concrets que les participants mettront en œuvre et pourront réutiliser.

Public visé:

Toutes les personnes désireuses d'appliquer les dernières technologies du W3C pour mettre en œuvre leurs projets Web.

Objectifs:

- Maîtriser HTML, CSS et JavaScript pour la mise en œuvre de page Web en responsive design.
 - Manipuler les outils de développement et de debug.
 - Comprendre l'évolution des langages pour la maintenance d'applications et manipuler leurs nouveautés.
 - Comprendre Javascript et l'interaction avec le document pour dynamiser le site Web.
-

Pré-requis:

La connaissance de l'algorithmique est nécessaire. La connaissance d'un langage de programmation objet est un plus.

Contenu:

Rappels des architectures Web

- Internet, ses organismes, ses protocoles, ses normes et standards.
- Les composants architecturaux d'une application Web: navigateur, serveur, ressource.
- Caractéristiques de http (hyper text transfer protocol) et fonctionnement.
- Du web statique au web dynamique.

Le langage HTML (Hyper Text Markup Language)

- Les différentes évolutions de HTML (Hyper text markup language).
- Le positionnement du W3C (World Wide Web Consortium), le DOM (Document Object Model).
- Structure d'un document HTML : Doctype., entête, corps.
- Contenu d'un document : les titres, le texte, les listes, les liens, les inclusions multimédias
- Les apports du HTML5

CSS3 (Cascading Style Sheets)

- Les concepts de base, les propriétés de style
- Le positionnement des éléments
- Comment gérer la compatibilité des navigateurs.
- Les apports des nouveautés CSS3.
- Couleurs et effets : transparence, dégradés, coins arrondis, ombres portées.
- Caractères : web fonts, ombrage.
- Transformations et animations : rotation, redimensionnement, déformation, transitions, animations.
- Mise en page : multicolonnage, mise en page flexible.

HTML et CSS

- Les formulaires, les nouveaux champs de saisie HTML5.
- La mise en page et les balises d'organisation (header, footer, nav, section, article, aside)
- Les tableaux, les Iframes
- Techniques de validation des formulaires
- Le responsive design et les Média Queries

Le langage Javascript

- L'objectif de JavaScript.
- Eléments syntaxiques
- La structure générale d'un programme de scripts.
- Les données, les opérateurs, les instructions de contrôle, les tableaux.
- Les fonctions et l'appel de scripts.

La programmation avec Javascript

- Rappels sur les concepts objets, les objets.
- Les différentes catégories d'objets du navigateur.
- Les objets du DOM (Window, Document...) et leurs manipulations.
- Les objets spécifiques à JavaScript.
- Le contrôle des formulaires et la gestion d'évènements
- Le Dynamic HTML : présentation du concept DHTML.
- Modification des styles CSS en JavaScript.

Autre(s) Information(s):

Support de cours en français remis aux participants

Linux / Unix les bases et programmation Shell

Durée : 5 jours

Résumé:

Cette formation présente les concepts fondamentaux d'Unix-Linux et constitue un tronc commun technique indispensable pour les informaticiens ayant à travailler sur un système Unix ou Linux quelle qu'en soit la version.

Public visé

Cette formation s'adresse aux administrateurs et ingénieurs amenés à travailler en environnement Unix-Linux.

Objectifs

- Acquérir les compétences indispensables pour une utilisation maîtrisée d'un système Unix ou Linux.
 - Comprendre les fondamentaux d'Unix et sélectionner les commandes et utilitaires essentiels dans la pratique quotidienne
 - Appréhender les aspects essentiels de la programmation Shell (écriture de scripts).
-

Prérequis :

Aucune connaissance préalable sur Unix ou Linux n'est nécessaire. Avoir de bonnes connaissances générales en informatique.

Contenu :

Caractéristiques générales et composantes du système.

Panorama des versions.

Linux par rapport à Unix, logiciels «open source».

Connexion et déconnexion, premières commandes simples.

Accès à la documentation.

Approche utilisateur de l'organisation des disques.

Types et noms de fichiers, arborescence, commandes essentielles.

Maîtrise des droits d'accès.

Prise en main de l'éditeur de texte «vi».

Processus (définitions, attributs, commande «ps», interruptions).

Mécanismes disponibles pour l'utilisateur :

- redirections
- «pipeline»
- arrière-plan
- processus stoppés

Utilisation du shell, variables d'environnement, alias,

historique des commandes, fichiers de connexion.

Manipulation de commandes essentielles :

- find
- commandes d'impression
- sauvegardes
- filtres

Commandes réseau

Comprendre les interfaces graphiques

- principe fonctionnel
- CDE, KDE, GNOME

Programmation Shell

- Les différents Shells (sh, ksh, bash)
 - Portabilité, contextes d'utilisation
 - Procédures et paramètres
 - Codes retour
 - Tests
 - Boucles
 - Lectures clavier
 - Expressions arithmétiques
 - Aiguillages
 - Fonctions
-

Qualité logicielle et Test

Durée: 4 jours

Résumé:

Les objectifs de ce module sont les suivants :

- Présenter les enjeux du test, les principes de base du test.
 - Comprendre comment s'inscrit le test dans un projet informatique.
 - Décrire le processus de test dans son ensemble en se basant sur la méthodologie ISTQB.
 - Connaître les livrables du test.
 - Découvrir les principales notions fondamentales du test : exigences, cas de test, scénario de test.
 - Connaître les différents rôles des acteurs d'une équipe de test et des parties prenantes interagissant avec l'équipe de test
-

Public visé

Ce module s'adresse aux professionnels des tests MOA / MOE : testeurs, analystes, chefs de projet, toute personne souhaitant acquérir des méthodes, des savoir-faire dans le test logiciel

Pré-requis :

Pour suivre ce module, il est préférable d'avoir quelques connaissances de base du cycle de vie logiciel, de la gestion de projet et de la méthode Agile

Contenu :

Connaître et comprendre les enjeux du test logiciel

- Les tests dans notre quotidien
- Que sont les tests ?
- Pourquoi les tests sont-ils nécessaires ?
- Les 7 principes sur les tests
- Processus de test
- La psychologie des tests

Les méthodes projet & Les tests

- Les modèles de développement logiciel
- Les niveaux de test
- Les types de test
- Les tests de maintenance

Principes de la conception des tests

- Comprendre la notion d'exigence et de condition de test

- Cas de test et scénario de test
- Principe de conception de test

Tests statiques et processus de revue

- Bases des tests statiques
- Processus de revue
- Techniques de revue

La gestion des tests

- Organisation des tests
 - Planification et estimation des tests
 - Pilotage et contrôle des tests
 - Gestion de configuration
 - Risques et tests
 - Gestion des défauts
-

Travaux pratiques (40% du module)

Un TP fil rouge se déroule sur l'ensemble du module :

- Objectif : permettre aux participants de synthétiser les connaissances acquises sur le processus de test.
- Déroulement : les participants complètent chacun une page au format A3 en plusieurs étapes au fil des acquisitions sur les activités du test et les acteurs liés au test. Le formateur distribue et commente une correction en fin de module.

Exercices :

- Première approche sur les exigences : à partir d'une expression de besoin, en déterminer des exigences et leur niveau de criticité.
- Première approche sur la conception de test : à partir d'une spécification sur le thème des habilitations, déterminer les cas de test et les scénarios de test à prévoir.

QCM : à la fin de chaque chapitre, sont projetées quelques questions de QCM portant sur les connaissances fondamentales abordées, afin de mieux mémoriser ces connaissances et de commencer à s'habituer aux types de questions présentes dans un examen ISTQB

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

Définition des exigences à partir des spécifications

Durée: 2 jours

Résumé:

L'une des premières activités du testeur logiciel est la phase d'analyse qui consiste à lire et analyser des documents de spécifications d'exigences (expression de besoin, cahier des charges ou spécifications fonctionnelles) pour d'une part en vérifier la qualité, d'autre part en extraire une liste d'exigences sur laquelle le testeur va se baser pour concevoir ses tests.

Public visé

Cette formation s'adresse aux professionnels des tests MOA / MOE, essentiellement les testeurs, analystes de test et manager de test.

Objectifs :

Ce module a pour objectifs de :

- Apprendre ce qu'est une exigence et en connaître les différentes catégories
 - Apprendre à décrire des exigences à partir d'une expression de besoins ou cahier des charges
 - Apprendre à identifier des exigences de test à partir de spécifications fonctionnelles
-

Pré-requis :

Connaître les principes de base du test et son processus.

Contenu :

JOUR 1

Fondamentaux sur les exigences

- Définition d'une exigence, d'une partie prenante
- Les activités principales de l'ingénierie des exigences
- Les catégories d'exigences

Définir les exigences à partir d'une expression de besoins

- Définir le périmètre et le contexte du système
- Elucider les exigences
- Documenter les exigences en langage naturel
- Critères qualité des exigences

Définir les exigences à partir d'une spécification

- Identifier les exigences selon leur catégorie
- Les modèles et leurs caractéristiques
- Modèle de cas d'utilisation
- Diagramme de classes UML
- Diagramme d'activité UML

JOUR 2

Travaux pratiques

- Définir les exigences à partir d'une spécification fonctionnelle
-

Travaux pratiques (60% du module)

Alternance d'exercices entre sous-chapitres théoriques sur un thème transverse (application mobile de réservation d'une place de parking pour automobiliste) en guise de fil conducteur, permettant une progression dans l'application de la méthode :

- Identifier les parties prenantes
- Etablir la liste des exigences utilisateurs
- Définir le périmètre et le contexte de l'application
- Choisir les bonnes techniques d'élucidation
- Valider la qualité des exigences
- Représenter les exigences par des modèles

Mise en œuvre de la méthodologie sous forme de travaux pratiques le 2^{ème} jour :

- Identifier et documenter les exigences de test à partir d'une spécification fonctionnelle d'une application client/serveur de réservation de vols d'avion
-

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

Conception des test et jeux de données

Durée: 2 jours

Résumé:

Le testeur peut intervenir sur un ou plusieurs niveaux de tests et pour un ou plusieurs types de tests. Dans tous les cas, il doit concevoir des tests avec efficacité. Les techniques de conception de tests lui permettent de concevoir les tests avec efficacité, avec le juste effort de test. Les cas de tests doivent être agrémentés des jeux de données les plus pertinents pour pouvoir être exécutés.

Public visé

Cette formation s'adresse aux professionnels des tests MOA / MOE, essentiellement les testeurs, analystes de test et manager de test

Objectifs :

Ce module a pour objectifs de :

- Connaître les différents types et niveaux de test
- Apprendre les techniques de conception de test
- Apprendre à définir les données de test les plus pertinentes
- Découvrir quelques techniques complémentaires spécifiques à la méthode Agile
- Découvrir des formats spécifiques de structuration des données

Pré-requis :

Connaître les principes de base du test et son processus

Contenu :

Conception des tests

- Concevoir et prioriser les cas de test
- Identifier les données de test
- Concevoir l'environnement de test
- Etablir la traçabilité

Techniques de conception des tests

- Catégories de techniques de test
- Techniques de test boîte noire
- Techniques de test boîte blanche
- Techniques de test basées sur l'expérience

Implémentation des tests

- Développer et prioriser les procédures de test
- Créer des suites de test

- Définir un calendrier d'exécution
- Construire l'environnement de test
- Préparer les données de test
- Mettre à jour la traçabilité

Implémenter des tests avec les méthodes Test First (xDD)

- Behaviour Driven Development (BDD)
- Test Driven Development (TDD)

Structures de données

- XML
 - JSON
-

Travaux pratiques (50% du module)

Alternance d'exercices entre sous-chapitres théoriques sur un thème transverse (application mobile de réservation d'une place de parking pour automobiliste) en guise de fil conducteur, permettant une progression dans l'application de la méthode :

- Concevoir des tests
- Identifier les données de test

Exercices spécifiques pour :

- Méthodes de conception de test (au moins un exercice par technique)
- Méthodes de conception de test en Agile
- Structures de données

QCM : à la fin du chapitre 2 (techniques de conception de test), sont projetées quelques questions de QCM portant sur les connaissances fondamentales abordées, afin de mieux mémoriser ces connaissances et de commencer à s'habituer aux types de questions présentes dans un examen ISTQB

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

Bases de données relationnelles - SQL

Durée: 2 jours

Résumé:

A l'issue de cette formation le stagiaire pourra

- Analyser un modèle conceptuel de données et déterminer sa cohérence générale.
 - Analyser un modèle physique de données et déterminer sa cohérence générale.
 - Ecrire des requêtes SQL de recherche et de mise à jour
-

Public visé

Cette formation s'adresse aux professionnels des tests MOA / MOE, essentiellement les testeurs, analystes de test et manager de test

Pré-requis :

Aucun

Contenu :

Jour 1

Introduction au concept des bases de données relationnelles et non relationnelles
Comprendre les modèles - conceptuel de données (MCD) et – physique de données (MPD)

- Explication des différents éléments qui le composent
- Exercices de définition de MCD et MPD associés

Jour 2

Sur la base d'un exemple vu le premier jour, le second jour est axé pour réaliser des requêtes SQL sur une base de données témoin.

L'ensemble des exercices seront réalisés sur la solution

PostgreSQL.

Présentation des principales requêtes SQL et les différents attributs
Exercices

- Faire des requêtes de sélection simples puis complexes
 - Créer, mettre à jour et supprimer des données (Construire son jeu de données)
 - Modifier une table
-

Travaux pratiques (60% du module)

La formation met en avant un exercice continu pour élaborer à partir d'un cas d'école :

- Le modèle relationnel
 - Le modèle physique de données
 - Créer un script de création d'une base de données
 - Ecrire des requêtes SQL de Création, Recherche, Mise à jour et Suppression
-

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

Stratégie de test – Plan de test

Durée: 2 jours

Résumé:

La stratégie de test est le moyen de définir un processus de test en fonction du contexte du projet, afin d'accroître l'efficacité des activités de test. Elle doit être déclinée sous forme d'approche de test par le test manager dans un plan de test, et ce dès le début du projet

Public visé

Cette formation s'adresse aux professionnels des tests MOA / MOE, essentiellement les managers de test, analystes de test et testeurs.

Objectifs :

Les objectifs de ce module sont :

- Comprendre ce que sont les politiques et stratégies de test
 - Apprendre à concevoir un plan de test
-

Pré-requis :

Connaître les principes de base du test et son processus.

Contenu :

JOUR 1

La politique de test

- Définition
- Démarche de conception
- Modèle de politique de test

La stratégie de test

- Définition
- Objectifs de la stratégie de test
- Types de stratégie de test
- L'approche de test
- Gestion des risques

Le plan de test

- Définition
- Le contenu attendu du plan de test
- Exemples de plan de test
- Exercice de conception d'un sommaire de plan de test

JOUR 2

- Travaux pratiques
 - Concevoir un plan de test à partir d'un cahier des charges
-

Travaux pratiques (70% du module)

Exercice de construction d'un sommaire de plan de test le 1er jour.

Mise en œuvre de la méthodologie sous forme de travaux pratiques le 2ème jour :

- A partir d'un cahier des charges portant sur une application client/serveur de réservation de vols d'avion, concevoir un plan de test en se basant sur les hypothèses définies par le formateur
-

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

Développer des services Web en Java : SOAP/ WSDL, REST, UDDI, AXIS

Durée: 3 Jours

Résumé:

Cette formation permet aux participants d'acquérir les compétences et connaissances nécessaires pour développer des applications distribuées d'une nouvelle génération basée sur Internet. Des applications dispersées sur le réseau peuvent alors être intégrées comme fonction d'un système plus large. Basés principalement sur XML et des protocoles d'échanges, les services web sont naturellement très adaptés au langage Java et la plate-forme Java EE.

Les participants pourront alors, grâce à de nombreux ateliers pratiques, développer, publier et utiliser leurs premiers services web.

Public visé:

Cette formation s'adresse aux développeurs et chefs de projets amenés à mettre en place des Web Services dans le domaine de Java.

Objectifs:

- | | |
|--|--|
| ■ Spécifier, développer, déployer et administrer des services Web JAX-WS et SAAJ | ■ Développer des clients synchrones et asynchrones pour les services Web dans Java |
| ■ Créer des clients pour les services Web avec JAX-WS | ■ Fournir des services Web de type RESTful pour de l'AJAX |
| ■ Créer et déployer des services Web à partir de WSDL | ■ Sécuriser de façon optimale les services Web |
-

Pré-requis:

Avoir des connaissances de base du langage XML et de l'expérience en développement Java.

Contenu:

Introduction : Architecture SOA et Web Services

- Les 5 caractéristiques de SOA
- Les 7 briques d'une architecture SOA
- Relation entre SOA et Web Services
- Technologies principales des Web Services : HTTP, XML, SOAP, WSDL, UDDI

XML dans Java

- Syntaxe et espaces de nom XML
- Description de XML avec les schémas
- Conversion et déconversion des paramètres avec JAXB
- Personnalisation des liaisons XML-Java
- Manipulation XML avec les APIs SAX, DOM et StaX

Messages SOAP

- Rôle de SOAP dans les services Web
- Structure des messages SOAP
- SOAP-ENV, Header, Body, Attachement
- Opération, messages et erreurs

Interface avec WSDL

- Rôle de WSDL dans les Web Services
- 5 sections clés d'un document WSDL
- Description du contrat / interfaces / opérations dans section portType
- Comprendre les sections "abstraites" : messages, types
- Comprendre les sections "concrètes" : binding, services
- Spécification de l'implémentation du service (service)
- Spécification du protocole d'échange de messages (Binding)

Solutions de gouvernance des web services

- Gouvernance : Respect de l'architecture SOA
- Conformité avec les normes WSI (Web Service Interoperability)
- Intégration de proxies et d'adaptateurs
- Mise en œuvre d'une terminaison de service Web avec JAX-WS
- Personnalisation de services Web JAX-WS
- Déploiement d'un fichier WAR de service Web
- Interception de trafic entre les services Web et les clients
- Optimisation de la transmission de messages

Développement Web Services SOAP en Java 6

- Créer WS à partir de WSDL (WSDL-first) ou à partir du code (code-first) ?
- Utiliser l'API de "haut niveau" JAX-WS
- Utiliser l'API de "bas niveau" SAAJ
- Valider la conformité de Web Service avec les normes WS-I
- WS-I : outils : Analyzer, Monitor
- Création d'intercepteurs (Handler) coté web service et coté client
- Optimisation de la transmission de messages avec MTOM

Développement Web Services SOAP avec frameworks / IDE

- Valeur ajoutée des frameworks Web Services
- Panorama des frameworks : CXF, AXIS, METRO, Jboss WS
- Valider la conformité de son WS avec WS-I
- Création de livrable Web Services WAR
- Création de livrable Web Services type EJB Session
- Création d'intercepteurs (Handler) coté web service et coté client
- Wizards pour créer et tester des Web Services dans Eclipse et Netbeans

Clients de services Web

- Accès à des services Web via WSDSL
- Création de fichiers sources clients à partir de WSDL
- Personnalisation des fichiers sources générés par le biais de JAX-WS
- Création de services et de clients de notification (one-way)
- Écriture de clients en multithreading
- Interception et modification de messages SOAP

Approche REST

- Développer des services Web RESTful avec JAX-WS
- Caractéristiques de POJO annotés, HTTP, XML / JSON
- Créer de l'interactivité du côté client
- Invoquer des services Web par le biais de l'API Dispatch
- Traitement des messages XML reçus
- Sécurité déléguée au transport
- Développement REST avec le Framework Jersey

Comparaison SOAP / REST

- Interopérabilité
- Spécifications
- Sécurité
- Transport

Gouvernance

Introduction à l'automatisation des tests logiciels

Durée: 1 Jour

Résumé:

Cette formation permet aux participants d'acquérir les compétences et connaissances sur les différents critères permettant d'orienter au mieux leur choix dans le cadre de l'automatisation de tests.

Public visé:

Cette formation s'adresse à toute personne ayant à mettre en place l'automatisation de tests logiciels.

Objectifs pédagogiques:

- Prendre les décisions concernant l'autoamtisation des tests
 - Connaître les idées reçues concernant l'automatisation et ses outils
 - Comprendre comment gagner en temps et productivité
 - Connaître les critères d'automatisation
 - Connaître les pré requis à l'automatisation
 - Comprendre les bénéfices attendus
-

Pré-requis:

Aucun

Contenu:

Contexte

- Rappels du vocabulaire
- Outillage de tests
- Principes

Qu'est-ce que l'autoamtisation des tests ?

- Pourquoi automatiser les test ?
- Quels test faut-il automatiser ?

Pilotage d'une automatisation

- Rôles et phases
- Processus d'automatisation

Modèles de maturité de l'automatisation

Tests unitaires et de performances

Bonnes pratiques, facteurs clés et choix d'outil

Méthodes pédagogiques :

Support de cours remis aux participants

Autres moyens pédagogiques et de suivi:

• Compétence du formateur : Les experts qui animent la formation sont des spécialistes des matières abordées et ont au minimum cinq ans d'expérience d'animation. Nos équipes ont validé à la fois leurs connaissances techniques (certifications le cas échéant) ainsi que leur compétence pédagogique.

Débuter avec Robot Framework

Durée: 2 jours

Résumé:

Cette formation à vocation très opérationnelle vous permettra d'écrire vos tests logiciels avec Robot Framework réelle sur un site Web et d'effectuer les tests

Public visé

Ce module s'adresse aux professionnels des tests MOA / MOE : testeurs, testeurs automaticiens, analystes, chefs de projet, toute personne souhaitant s'initier à des outils de test automatisé

Objectifs :

Les objectifs de cette formation sont de :

- Comprendre le rôle et le fonctionnement de Robot Framework
 - Ecrire des tests logiciels en utilisant l'approche de test par mot-clé de Robot Framework et la syntaxe de données de test tabulaire
 - Utiliser une syntaxe cohérente pour composer de nouveaux mots-clés à partir des mots-clés existants
 - Effectuer des tests de développement comportemental (BDD)
-

Pré-requis :

Bonne connaissance du processus de test.
Des connaissances de bases en algorithmie et en langage Python est un plus

Contenu :

Focus sur les approches de Scripting Structurés

- Par les données
- Par Mots-clés

Installation d'un projet avec Robot Framework

Présentation de la syntaxe d'écriture des tests

- Présentation d'un script

Vérification & Exécution des tests

- Comment vérifier ses scripts

- Définir sa stratégie d'exécution (Différencier les tests de confirmation / régression / et liés aux évolutions)

Maintenance des scripts et amélioration continue

- Comprendre les pièges à éviter
-

Travaux pratiques (70% du module)

Mise en œuvre de scripts sous Robot Framework

Autre(s) Information(s):

Le matériel pédagogique sera remis aux stagiaires

ISTQB Fondation

Durée: 3 Jours

Résumé:

Le tryptique Qualité/Coût/Délai est au centre de préoccupation des DSI : crash informatique, industrialisation des tests, redéveloppements coûteux d'applications existantes. Face à ces enjeux, maîtriser les activités de test devient une priorité !

Cette formation vous permet d'acquérir les connaissances nécessaires pour exercer des travaux sur les tests et, le cas échéant de préparer la certification internationale **Testeur certifié niveau fondation de l'ISTQB®**.

Public visé:

Ce cours s'adresse aux professionnels des tests MOA / MOE : testeurs, analystes, chefs de projet, toute personne souhaitant acquérir des méthodes, des savoir-faire dans le test logiciel et préparer la certification ISTQB® Niveau Fondation (CTFL- Testeur de Logiciel Certifié).

Objectifs:

- Ce cours vous aide à maîtriser les connaissances sur :
 - Le processus de tests et ses différentes étapes
 - Les types de test, leurs caractéristiques, leurs choix et leurs ISTQB® permet de mises en oeuvre dans les cycles de vie logiciel
 - Les techniques de création de cas de test, des scénarios et procédures de test
 - Comment gérer les tests, les contrôler et faire le suivi
 - Le choix des outils et leurs utilisations
 - Le cas échéant, l'obtention de la certification internationale démontrer l'acquisition d'un niveau de connaissance et de compétences en test de logiciels.
-

Pré-requis:

Pour suivre ce stage, il n'est pas demandé de prérequis particulier. Cependant, il est toutefois préférable d'avoir quelques connaissances de base du cycle de vie logiciel.

Test et certification

L'examen de certification Testeur ISTQB niveau fondation est un QCM électronique de 40 questions. Il se déroule en 1 heure sous le contrôle d'un examinateur assermenté par l'ISTQB®. Il faut 65% de bonnes réponses pour obtenir la certification. Les résultats sont connus immédiatement en fin d'examen puis confirmés avec l'envoi d'un certificat par l'organisme GASQ (Global Association for Software Quality).

Contenu:

Fondamentaux des tests

- Quels sont les enjeux des tests ? Pourquoi les tests sont-ils indispensables ?
- Le vocabulaire dans le monde du test
- Quels sont les objectifs du test ? Quelles sont les différences entre test et débogage ?
- Les 7 principes généraux des tests : quand tester ? Contexte ? Exhaustivité ?
- Processus de tests fondamentaux
- Les tests sont une activité indispensable des projets
- La psychologie des tests. Les différences d'approche entre le testeur et le développeur
- Le code éthique du testeur
- Révision de la journée sous forme d'exercices QCM servant à la préparation de l'examen

Tester pendant le cycle de vie logiciel

- Les modèles de développement logiciel : modèle en V, modèle itératif...
- Les tests dans les approches incrémentales et itératives
- Les différents tests : test de composants, test d'intégration, test système, test d'acceptation
- Les principales approches de test : Big Bang, Ad-hoc, Incrémentale, Exploratoire...
- Les types de test : les cibles de test
- Tests de maintenance. Tests de régression
- Révision du chapitre sous forme d'exercices QCM servant à la préparation de l'examen

Techniques statiques

- Revues et processus de test
- Processus de revue formelle
- Analyse statique avec des outils techniques de conception de tests
- Révision du chapitre sous forme d'exercices QCM servant à la préparation à l'examen

Techniques de conception des tests

- Identifier les conditions de test et conception des cas de tests
- Traçabilité des éléments de test
- Catégories de techniques de conception de test
- Techniques basées sur les spécifications ou techniques boîte noire
- Classes équivalence, limites, tables de décision, transition d'état...
- Techniques basées sur la structure ou boîte blanche
- Couverture de chemin exécutable, de branches, de décision, de condition
- Techniques basées sur l'expérience
- Sélectionner les techniques de test
- Révision du chapitre sous forme d'exercices QCM servant à la préparation de l'examen

Gestion des tests

- Organisation des tests, rôles, plan de test, conception, approches et stratégie
- Estimation et planification des tests
- Suivi et contrôle du déroulement des tests. Reporting
- Gestion de configuration
- Test et risques. Risques projet et produit
- Gestion des incidents
- Révision du chapitre sous forme d'exercices QCM servant à la préparation de l'examen

Outils de support aux tests

- Les outils de support à la gestion des tests
- Les outils de support aux techniques statiques de test
- Les outils de support pour la spécification des tests
- Les outils de support à l'exécution des tests
- Les outils de support pour la mesure de performance
- Bénéfices et risques de l'utilisation d'outils
- Introduire un outil dans une organisation

QCM de révision (proche des conditions de l'examen)

Autre(s) Information(s):

Conformément au programme officiel ISTQB, des travaux pratiques sous forme d'exercices ou de questions sont réalisés à la fin de chaque chapitre. En complément un examen blanc avec un questionnaire est inclus dans la formation afin de se préparer à l'examen final.

La gestion des containers virtuels avec Docker

Durée: 3 Jours

Résumé:

En environnement Linux, Docker fournit une solution de virtualisation qui ne nécessite pas de système d'exploitation propre puisqu'il utilise les fonctionnalités système sous-jacent. Il empaquète les applications et automatise leur déploiement en utilisant la technique de containerisation. très présent en environnement OpenStack, ce produit est aussi largement interfacé avec les outils de gestion d'infrastructure du marché, l'ensemble constituant une alternative technologique aux solutions traditionnelles de virtualisation.

Public visé:

Ce cours s'adresse aux architectes logiciels, administrateurs systèmes, DevOps et développeurs.

Objectifs pédagogiques:

- Comprendre le positionnement de Docker et principe de containerisation
 - Découvrir le principe de fonctionnement de Docker et de ses conteneurs
 - Découvrir les différentes méthodes d'installation de Docker
-

Pré-requis:

Avoir des connaissances en environnement Unix/Linux.

Contenu:

Introduction

- Présentation et mise en oeuvre des VMs Debian 9 ; CentOS 7
- Conteneurs industriels
- Salomon HYKES, de DotCloud à Docker, Inc

Grands principes de l'urbanisation logiciels

- Intérêts

La containerisation

- Les solutions existantes
- Concepts de containerisation
- Comparatifs entre containerisation et virtualisation
- Présentation de l'outil Docker, Inc
- Les versions de Docker, Inc
- Ecosystème de Docker, Inc

NameSpace ; Cgroups de Docker

- Les PID sur la machine Hôte
- Cloisonnement des PID par NameSpace
- Cloisonnement des ressources hardware par nameSpace

De LXC à LibContainer

- Imbrication des couches logiciels de Dockers, Inc

Les systèmes de fichiers en couches

- UnionFS
- Devicemapper
- AUFS
- Comparaisons entre filesystem en couches et disques virtuels

Les multiples processus d'installations

- Linux Debian
- CentOS Docker EE
- CoreOS
- DockerForMac
- DockerForWindows

La stack réseau de docker

- Le réseau bridge
- Le réseau host
- Le réseau none

Docker compose

- Principes générales
- Écriture d'un fichier docker-compose.yml
- Lancement et gestion automatique des conteneurs

Présentation du container « Hello-World »

- Vérification de la chaîne logicielle avec le container hello-world
- Les étapes clés d'une instantiation
- Les commandes Docker
- Container ID

Le registry public

- Présentation de l'interface Web <https://store.docker.com>
- Les bonnes pratiques
- Accès au registry en mode commande
- Création d'un compte en ligne
- Connexion de Docker à la plateforme en ligne

Instantiation d'images

- Les couches de l'instanciation
- Lancement en mode interactif
- Les instructions de « run »
- Le passage de variables d'environnement
- Lancement en arrière plan
- Le recyclage des processus
- Les logs
- DockerUI (Portainer)

Création d'une image pas à pas

- Création d'un serveur apache2 sur système Ubuntu

Création d'un Dockerfile

- Anatomie d'un Dockerfile
- Création d'un dockerfile serveur Ubuntu+apache2+php7.0
- Création d'une image MYSQL
- Sauvegarde des images sur le hub public

Persistence des données dans le temps

- Sur une volume simple (debian 9)
- Implémentation Docker EE sur Linux CentOS 7
- Sur système LVM thin pool (CentOS7)
- Mise en oeuvre d'un registry privée

La stack réseau de Docker

- Le réseau bridge
- Le réseau host
- Le réseau none
- Le réseau MacVlan

Debian 8 - Adaptation du noyau Linux aux VLANs avec Docker

- Vérification version de noyau et logiciel
- Téléchargement de sources
- Construction d'un noyau adapté
- Patch AUFS
- Compilation
- IpVlans sur noyau Debian

Docker compose

- Principes généraux
- Écriture d'un fichier docker-compose.yml
- Lancement et gestion automatique des conteneurs

Docker Machine

- Gestion de l'outil Docker machine
- Provisionnement local

Docker swarm

- Configuration d'un cluster swarm
- Orchestration d'un cluster de machine
- Load Balancing entre serveurs de cluster

Kubernetes avec Docker

- Concept

Annonce Docker-EE (juillet 2018)

- FAM (Fédéré Application Manager - projet Armada)
- Docker Applications Designer - projet Baguette)
- Vidéo explicative

Méthodes pédagogiques :

Un support de cours en français est remis aux participants

Autres moyens pédagogiques et de suivi:

- Compétence du formateur : Les experts qui animent la formation sont des spécialistes des matières abordées et ont au minimum cinq ans d'expérience d'animation. Nos équipes ont validé à la fois leurs connaissances techniques (certifications le cas échéant) ainsi que leur compétence pédagogique.
- Evaluation : Chaque participant, à l'issue de la formation, répond à un questionnaire d'évaluation qui est ensuite étudié par nos équipes pédagogiques en vue de maintenir et d'améliorer la qualité de nos prestations.
- Suivi d'exécution : Une feuille d'émargement par demi-journée de présence est signée par tous les participants et le formateur.

Intégration continue et industrialisation

Durée : 3 jours

Public visé

Cette formation s'adresse à tous développeur, chef de projet ou architecte impliqués dans la réalisation d'applications Java EE.

Prérequis :

Connaître le développement Java.

Contenu :

INTRODUCTION

- Les principes de l'intégration continue
- Les prérequis
- Les différents outils de l'intégration continue
- Mise en place de l'environnement d'intégration continue

LE GESTIONNAIRE DE CONTRÔLE DE VERSION

- Les fonctionnalités
- Les différents gestionnaires de sources.
- Les problématiques d'intégration des changements

L'AUTOMATISATION DES BUILDS

- Qu'est-ce que la construction d'un logiciel
- Différents types de builds
- Jenkins et Maven

GESTION DES LIVRABLES AVEC ARCHIVA

- Stratégie de mise à disposition du résultat construit
- Notion de dépôt d'artefacts
- Les principaux outils de gestion de dépôt d'artefacts: Nexus, Artifactory et Archiva
- Problématique de traçabilité du résultat et des informations de build

L'AUTOMATISATION DES TESTS

- Les différents types de tests (tests unitaires, tests d'intégration, tests d'acceptation, tests de performance)
- Les outils de test
- Automatisation des tests
- Configuration des rapports
- Les environnements de tests
- La mise en œuvre de l'automatisation des tests

LE SERVEUR D'INTÉGRATION CONTINUE JENKINS

- Le rôle du serveur d'intégration continue.
- Les grandes fonctionnalités.
- Les plugins les plus utilisés
- Gestion des autorisations et des rôles
- Gestion de l'espace disque
- Monitoring

GESTION DE LA QUALITÉ DU CODE AVEC SONARQUBE : LA MISE EN PLACE DES MÉTRIQUES

- Les outils d'analyse et de reporting (Checkstyle, Findbugs, PMD...)
- La génération de rapports d'analyse
- Configuration de SonarQube
- La publication des résultats dans SonarQube
- La publication des résultats dans Jenkins

COMMUNICATION

- Indicateurs de visibilité du résultat construit
 - Statut, Santé et tendance
 - Savoir réagir
 - Communication avec des outils externes
 - Notification par email et flux RSS
-