

Mastère Avancé

Ingénierie du Logiciel



Mastère Avancé Ingénierie du Logiciel

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web: www.techtitute.com/fr/informatique/mastere-avance/mastere-avance-ingenierie-logiciel

Sommaire

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 14

05

Direction de la formation

page 18

04

Structure et contenu

page 40

06

Méthodologie

page 40

07

Diplôme

page 48

01

Présentation

Ces dernières années, la demande de logiciels est montée en flèche. Avec l'émergence de nouvelles plateformes numériques, d'un matériel plus sophistiqué et d'une virtualisation croissante des processus quotidiens, les ingénieurs *logiciel* sont sans cesse confrontés à de nouveaux défis. Alors que le public s'habitue de plus en plus aux nouvelles technologies et que ses exigences deviennent plus élevées, les professionnels du développement *logiciel* doivent s'adapter à ces demandes et créer des produits à la hauteur des attentes du marché. Cela exige un niveau technique élevé dans de multiples domaines de connaissances informatiques.



“

Vous jouerez un rôle clé dans l'avenir technologique de nombreuses entreprises. Spécialisez-vous dans le génie logiciel et commencez à développer les systèmes qui feront la différence”

L'industrie technologique est l'une des industries les plus pertinentes aujourd'hui, car presque tout le monde interagit quotidiennement avec une sorte d'appareil numérique. Dans ce contexte, les ingénieurs de software sont en première ligne de tout le processus de développement technologique, car ce sont eux qui doivent constamment mettre à jour les systèmes, en développer de nouveaux et proposer des solutions intelligentes aux problèmes qui se posent. Vu sous cet angle, les professionnels de l'ingénierie logicielle doivent être des personnes très décisives, dotées de grandes connaissances techniques et d'une capacité exceptionnelle à s'adapter à tous les types de développement et d'environnements.

C'est dans cet objectif que TECH a conçu ce Mastère Avancé en Ingénierie du Logiciel, offrant une formation complète et de haut niveau à tous les développeurs qui souhaitent spécialiser leur carrière et l'orienter vers la création de systèmes. D'une part, le programme aborde les différentes méthodologies pour créer et gérer un projet de développement logiciel, ainsi que tous les aspects à prendre en compte en matière de calcul, d'exigences et de plateformes. D'autre part, elle traite également de la sécurité tant du logiciel lui-même que des systèmes d'information et de l'environnement de travail utilisés au cours du processus. Une fois diplômé, l'étudiant aura toutes les connaissances nécessaires pour devenir un expert efficace et hautement compétent en Ingénierie du Logiciel.

En outre, l'un des principaux avantages de ce programme est son caractère 100% en ligne. Cela signifie que l'étudiant ne doit pas s'adapter à des horaires fixes et n'est pas obligé de fréquenter un centre physique spécifique. De cette façon, l'étudiant a la liberté de gérer l'étude de la matière de son choix, à son propre rythme et en tenant compte de ses obligations, en planifiant l'horaire qui lui convient le mieux.

Ce **Grand Mastère Avancé en Ingénierie du Logiciel** contient le programme le plus complet et le plus à jour du marché. Ses principales caractéristiques sont:

- ♦ Le développement d'études de cas présentées par des experts en développement de logiciels
- ♦ Le contenu graphique, schématique et éminemment pratique du programme fournit des informations scientifiques et pratiques sur les disciplines essentielles à la pratique professionnelle
- ♦ Des exercices pratiques où le processus d'auto-évaluation peut être réalisé pour améliorer l'apprentissage
- ♦ L'accent est mis sur les méthodologies innovantes dans le domaine de l'Ingénierie du Logiciel
- ♦ Les cours théoriques, questions à l'expert, forums de discussion sur des sujets controversés et travail de réflexion individuel
- ♦ Il est possible d'accéder aux contenus depuis tout appareil fixe ou portable doté d'une connexion à internet



Pouvez-vous imaginer être impliqué dans le développement de Netflix? Il est temps d'arrêter d'imaginer et de concentrer votre carrière sur les meilleurs projets logiciels"

02 Objectifs

Ce Mastère Avancé en Ingénierie du Logiciel a été développé dans le but d'offrir à tous les professionnels du domaine informatique la formation supérieure nécessaire pour orienter leur carrière vers le développement de logiciels modernes adaptés aux nouvelles réalités fluctuantes du marché. Grâce aux connaissances hautement techniques enseignées tout au long de la formation, l'étudiant augmentera considérablement ses chances d'évolution de carrière et d'accès à des emplois dans les grandes entreprises du secteur.





“

Un Mastère Avancé qui sera le plus grand coup de pouce positif que vous pourrez donner à votre carrière vers la réussite professionnelle"

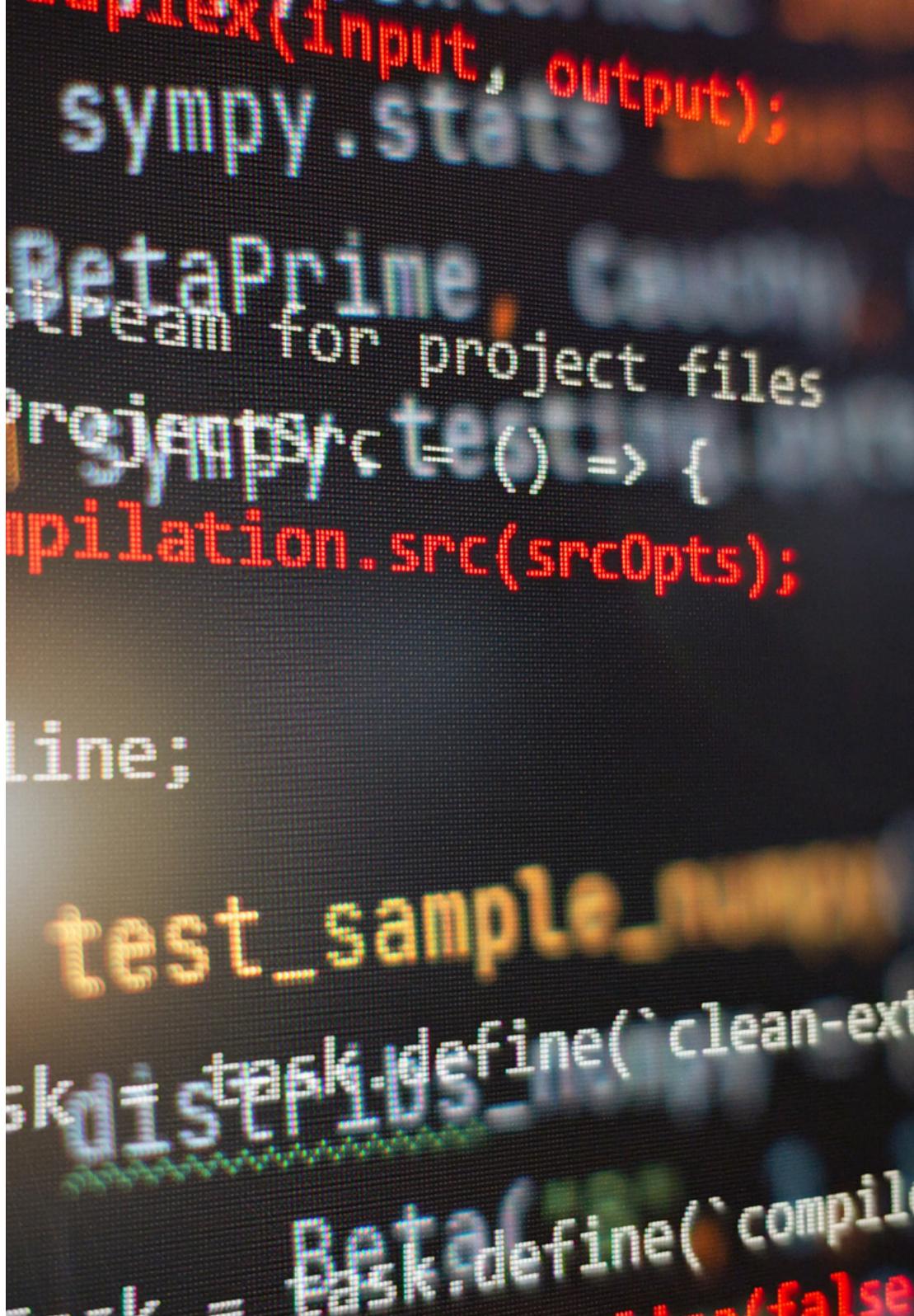


Objectifs généraux

- ◆ Acquérir les nouvelles compétences nécessaires et exigées en termes de nouvelles technologies et des derniers développements logiciels
- ◆ Compléter les connaissances acquises par des compétences dans le domaine de l'informatique et de la structure des ordinateurs, y compris les bases mathématiques, statistiques et physiques essentielles en ingénierie
- ◆ Élargir les connaissances en matière de Ingénierie du Logiciel et de systèmes Informatique avec les derniers développements et la méthodologie la plus innovante
- ◆ S'attaquer à des projets et des environnements logiciels complexes, en sachant apporter des solutions intelligentes à des problèmes divers

“

Une spécialisation qui vous aidera à maîtriser le développement de logiciels avec un ensemble unique de compétences exigées par toutes les grandes entreprises du secteur”





Objectifs spécifiques

- ◆ Connaître les bases de l'Ingénierie du Logiciel, ainsi que l'ensemble des règles ou principes éthiques et la responsabilité professionnelle pendant et après le développement
- ◆ Comprendre le processus de développement de logiciels, sous les différents modèles de programmation et le paradigme de la programmation orientée objet
- ◆ Comprendre les différents types de modélisation d'applications et les modèles de conception dans le langage unifié de modélisation (UML)
- ◆ Connaître les concepts fondamentaux de la gestion de projet et le cycle de vie de la gestion de projet
- ◆ Comprendre le fonctionnement de la gestion de la qualité dans les projets, y compris la planification, l'assurance, le contrôle, les concepts statistiques et les outils disponibles
- ◆ Acquérir les connaissances essentielles liées à la responsabilité professionnelle découlant de la gestion de projet
- ◆ Comprendre les différentes plateformes de développement de logiciels
- ◆ Acquérir les connaissances nécessaires au développement d'applications et d'interfaces graphiques dans les langages *Java* et *.NET*
- ◆ Apprendre les environnements de développement d'applications mobiles *Android* les processus de débogage et de publication
- ◆ Comprendre le développement d'applications basées sur le cloud et déterminer les procédures correctes pour leur mise en œuvre
- ◆ Comprendre les procédures et les techniques permettant d'améliorer l'apparence d'un document écrit en HTML
- ◆ Acquérir les compétences nécessaires au développement d'applications web côté client
- ◆ Développez des applications aux structures complexes, en utilisant les différentes procédures, fonctions et objets qui composent *JavaScript*
- ◆ Apprenez à utiliser l'interface de programmation DOM pour les documents HTML et XML afin de modifier leur structure, leur style et leur contenu
- ◆ Connaître le concept d'utilisabilité du web, ses avantages, ses principes, ses méthodes et ses techniques pour rendre un site web utilisable par l'utilisateur
- ◆ Comprendre l'architecture logicielle MVC (Model View Controller View) qui sépare les données, l'interface utilisateur et la logique de contrôle d'une application en trois composants distincts
- ◆ Acquérir les compétences pour l'utilisation des services web, en utilisant XML, SOA et REST
- ◆ Comprendre le processus de sécurité de l'information, ses implications sur la confidentialité, l'intégrité, la disponibilité et les coûts économiques
- ◆ Apprendre l'utilisation de bonnes pratiques de sécurité dans la gestion des services informatiques
- ◆ Acquérir les connaissances nécessaires à la certification adéquate des processus de sécurité
- ◆ Comprendre les mécanismes et les méthodes d'authentification pour le contrôle d'accès, ainsi que le processus d'audit d'accès
- ◆ Comprendre les programmes de gestion de la sécurité, la gestion des risques et la conception des politiques de sécurité
- ◆ Apprendre les plans de continuité des activités, leurs phases et le processus de maintenance
- ◆ Comprendre les procédures pour une protection adéquate de l'entreprise par le biais de réseaux DMZ, l'utilisation de systèmes de détection d'intrusion et d'autres méthodologies
- ◆ Comprendre les problèmes liés à la sécurité des logiciels, leurs vulnérabilités et la manière dont ils sont classés
- ◆ Analyser les différents serveurs web qui sont en vogue sur le marché actuel

tech 12 Objectifs

- ◆ Comprendre le processus de statistiques d'utilisation et d'équilibrage des charges sur les serveurs web
- ◆ Acquérir les connaissances nécessaires à la bonne exécution du processus de contrôle et d'audit informatique interne
- ◆ Comprendre les concepts et les processus de la conception de logiciels, en apprenant également la conception d'architecture et la conception au niveau des composants, ainsi que la conception basée sur des modèles
- ◆ Comprendre les différents modèles d'architectures de systèmes et de conception de logiciels, ainsi que l'architecture des applications en nuage
- ◆ Approfondir la compréhension de l'amélioration du processus de développement des logiciels et de la qualité des logiciels à l'aide des normes ISO/CEI
- ◆ Comprendre l'importance de l'ingénierie des exigences dans le processus de développement des logiciels
- ◆ Approfondir la compréhension des techniques de détermination des besoins et d'élicitation des besoins, car elles constituent une partie essentielle du processus
- ◆ Comprendre et appliquer le prototypage comme une partie essentielle du processus de développement
- ◆ Posez les bases de l'analyse médico-légale dans le monde des audits de logiciels et de technologies de l'information
- ◆ Connaître les concepts fondamentaux de la gestion de projet et le cycle de vie de la gestion de projet
- ◆ Apprenez à élaborer un calendrier pour la gestion du temps, le développement du budget et la réponse aux risques
- ◆ Comprendre le fonctionnement de la gestion de la qualité dans les projets, y compris la planification, l'assurance, le contrôle, les concepts statistiques et les outils disponibles





“

Une formation complète qui vous permettra d'acquérir les connaissances nécessaires pour rivaliser avec les meilleurs”

03

Compétences

Les ingénieurs logiciels mettent constamment leurs connaissances à jour, car les outils et les réalités dans lesquels ils travaillent se modernisent chaque jour. Cela nécessite un processus d'apprentissage régulier pour lequel diverses compétences générales sont nécessaires, tant en termes de développement logiciel pur que dans d'autres disciplines telles que la gestion d'équipe. Conscient de cette situation, TECH a développé le Mastère Avancé en Ingénierie du Logiciel dans le but de fournir aux étudiants toutes les compétences possibles et nécessaires pour rendre leur propre processus d'apprentissage continu plus léger et plus automatique.



```
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
f_operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
#selection at the end -add back the dese
```

```
ror_ob.select= 1
```

```
modifier_ob.select=1
```

```
.context.scene.objects.active =
```

```
nt("Selected" + str(modifier
```

```
#mirror_ob.select = 0
```

```
e = bpy.context.selected
```

```
y.data.objects[mod
```

```
print("Selected")
```

“

Avec toutes les compétences que vous allez acquérir en étudiant ce Mastère Avancé en Ingénierie du Logiciel, vous serez le meilleur candidat pour tout poste auquel vous postulerez"



Compétences générales

- ◆ Développer un système logiciel en tenant compte de toutes les étapes du développement, des plateformes de sécurité et des problèmes de sécurité
- ◆ Traiter correctement et professionnellement toutes les données générées pendant le développement
- ◆ Appliquer la meilleure méthodologie de travail en fonction du projet ou des personnes impliquées dans le projet
- ◆ Connaître toute la réalité du génie logiciel et prévenir rapidement et efficacement les risques ou problèmes éventuels

“

Vous pouvez faire le pas vers un meilleur avenir professionnel. Faites-le et inscrivez-vous dès maintenant à ce Mastère Avancé qui vous ouvrira de nombreuses portes pour votre carrière professionnelle”





Compétences spécifiques

- ◆ Comprendre les différents types de modélisation d'applications et les modèles de conception dans le langage unifié de modélisation (UML)
- ◆ Comprendre le fonctionnement de la gestion de la qualité dans les projets, y compris la planification, l'assurance, le contrôle, les concepts statistiques et les outils disponibles
- ◆ Utiliser les connaissances nécessaires au développement d'applications et d'interfaces graphiques dans les langages *Java* et *.NET*
- ◆ Comprendre les procédures et les techniques permettant d'améliorer l'apparence d'un document écrit en HTML
- ◆ Maîtriser le processus d'interaction avec les clients, grâce à l'utilisation de: formulaires, *Cookies* et gestion des sessions
- ◆ Comprendre les mécanismes et les méthodes d'authentification pour le contrôle d'accès, ainsi que le processus d'audit d'accès
- ◆ Comprendre l'application de la sécurité dans les différentes phases du cycle de vie des logiciels
- ◆ Comprendre le concept, le fonctionnement, l'architecture, les ressources et le contenu d'un serveur web
- ◆ Comprendre les différents outils d'aide, les méthodologies et l'analyse ultérieure lors des audits de sécurité sur l'internet et les appareils mobiles
- ◆ Comprendre les politiques et les normes de sécurité à appliquer aux applications en ligne
- ◆ Être capable de rédiger, planifier, développer et signer des projets dans le domaine de l'ingénierie informatique visant le développement ou l'exploitation de systèmes, services et applications informatiques
- ◆ Gérer les activités des projets informatiques
- ◆ Être capable de définir, d'évaluer et de sélectionner des plates-formes matérielles et logicielles pour le développement et la mise en œuvre de systèmes, services et applications informatiques
- ◆ Savoir développer en utilisant *Scrum*, la programmation extrême et les techniques de développement logiciel basées sur la réutilisation
- ◆ Avoir la capacité de concevoir, développer et maintenir des systèmes, des services et des applications informatiques en utilisant des méthodes d'ingénierie logicielle comme outil d'assurance qualité
- ◆ Utiliser les principes fondamentaux de la cryptographie symétrique et de la cryptographie asymétrique, ainsi que leurs principaux algorithmes
- ◆ Appliquer les concepts essentiels liés aux systèmes d'information dans l'entreprise, ainsi qu'identifier les opportunités et les besoins des systèmes d'information
- ◆ Savoir élaborer un calendrier pour la gestion du temps, le budget et la réponse aux risques
- ◆ Comprendre le fonctionnement de la gouvernance et de la gestion des TIC, les normes ISO/IEC qui la régissent et les meilleures pratiques à mettre en œuvre
- ◆ Planifier la gestion de la sécurité et gérer les principaux mécanismes de protection des actifs informationnels

04

Direction de la formation

Ce programme académique dispose du personnel enseignant le plus spécialisé sur le marché de l'éducation actuel. Il s'agit des spécialistes sélectionnés par TECH pour développer l'ensemble de l'itinéraire. Ainsi, sur la base de leur propre expérience et des données les plus récentes, ils ont conçu le contenu le plus actuel qui offre une garantie de qualité dans un sujet aussi pertinent.

“

TECH vous offre le personnel enseignant le plus spécialisé dans le domaine d'étude. Inscrivez-vous maintenant et profitez de la qualité que vous méritez”

Directeur invité international

Darren Pulsipher est un architecte logiciel très expérimenté, un innovateur avec une expérience internationale exceptionnelle dans le développement de logiciels et de microprogrammes. Il possède en effet des compétences très développées en matière de communication, de gestion de projet et d'affaires, qui lui ont permis de diriger d'importantes initiatives mondiales.

Il a également occupé des postes à haute responsabilité tout au long de sa carrière, comme celui d'Architecte en Chef des Solutions pour le Secteur Public chez Intel Corporation, où il a promu des activités, des processus et des technologies modernes pour les clients, les partenaires et les utilisateurs du secteur public. En outre, il a fondé Yoly Inc., dont il a également été le PDG, et a travaillé au développement d'un outil d'agrégation et de diagnostic de réseaux sociaux basé sur un Logiciel en tant que service (SaaS), utilisant les technologies Big Data et Web 2.0.

Il a également travaillé dans d'autres entreprises, en tant que Directeur Senior de l'Ingénierie, chez Dell Technologies, où il a dirigé l'Unité Commerciale Big Data dans le Cloud, dirigeant des équipes aux États-Unis et en Chine pour gérer des projets de grande envergure et restructurer des divisions commerciales pour une intégration réussie. Il a également occupé le poste de Directeur de la Technologie de l'Information (Chief Information Officer) chez XanGo, où il a géré des projets tels que le support du Help Desk, le support de la production et le développement de solutions.

Parmi les nombreuses spécialités dans lesquelles il est expert figurent la technologie Edge to Cloud, la cybersécurité, l'Intelligence Artificielle Générative, le développement de logiciels, la technologie de mise en réseau, le développement cloud-native et l'écosystème des conteneurs. Il a partagé ses connaissances dans le cadre du podcast hebdomadaire et de la lettre d'information "Embracing Digital Transformation", qu'il a produits et présentés, afin d'aider les organisations à réussir leur transformation numérique en s'appuyant sur les personnes, les processus et la technologie.



M. Pulsipher, Darren

- Architecte en Chef des Solutions pour le Secteur Public chez Intel, Californie, États-Unis
- Présentateur et Producteur de "Embracing Digital Transformation", Californie
- Fondateur et PDG de Yoly Inc. dans l'Arkansas
- Directeur Senior de l'Ingénierie chez Dell Technologies, Arkansas
- Directeur de la Technologie de l'Information, XanGo, Utah
- Architecte Senior chez Cadence Design Systems, Californie
- Gestionnaire Senior de Processus de Projet chez Lucent Technologies, Californie
- Ingénieur Logiciel chez Cemax-Icon, Californie
- Ingénieur Logiciel chez ISG Technologies, Canada
- MBA en Gestion des Technologies à l'Université de Phoenix, Université de Phoenix
- Licence en Informatique et Ingénierie électrique de l'Université Brigham Young

“

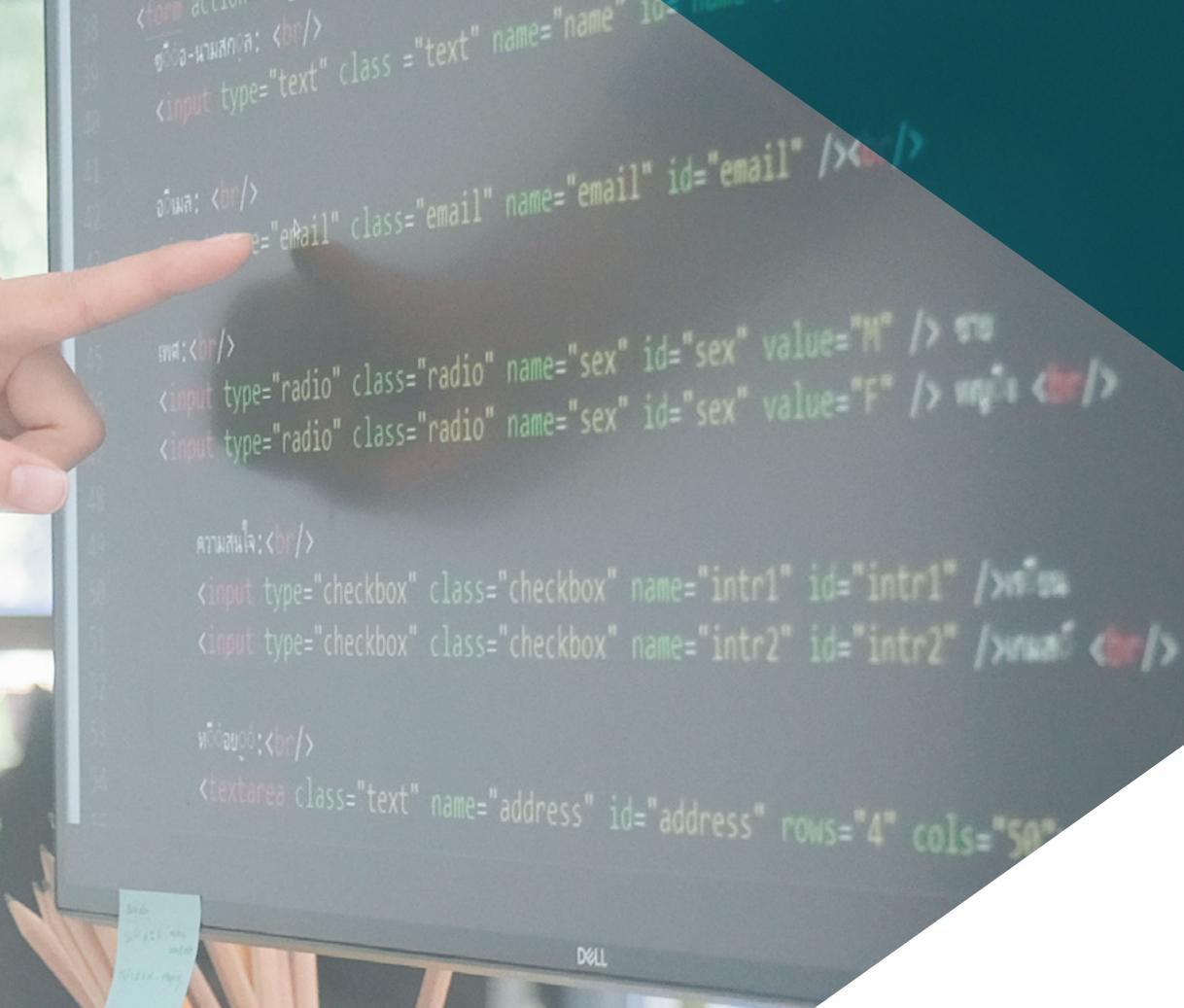
Grâce à TECH, vous pourrez apprendre avec les meilleurs professionnels du monde”

05

Structure et contenu

Le matériel didactique de ce Mastère Avancé en Ingénierie du Logiciel est conçu pour couvrir tous les enseignements nécessaires et complémentaires de la matière, avec les méthodologies, les outils et les connaissances les plus récentes disponibles sur le marché. Un programme complet et exhaustif où sont enseignés l'utilisation des langages de programmation et des environnements avancés, l'administration des serveurs web et leur intégration dans le processus de développement ou la gestion effective d'un projet. Cela garantit la meilleure opportunité possible pour l'étudiant de se spécialiser en génie logiciel et d'exceller rapidement dans ce domaine.





“

Montrez que vos connaissances sont à la hauteur de votre attitude pour être le meilleur professionnel et ajoutez une grande valeur à votre CV avec ce Mastère Avancé en Ingénierie du Logiciel”

Module 1. Méthodologies, développement et qualité en Ingénierie du Logiciel

- 1.1. Introduction à la Ingénierie du Logiciel
 - 1.1.1. Introduction
 - 1.1.2. La crise du logiciel
 - 1.1.3. Différences entre l'Ingénierie du Logiciel et l'Informatique
 - 1.1.4. Éthique et responsabilité professionnelle dans l'Ingénierie du Logiciel
 - 1.1.5. Usines de logiciels
- 1.2. Le processus de développement de logiciels
 - 1.2.1. Définition
 - 1.2.2. Modèle de processus logiciel
 - 1.2.3. Le processus unifié de développement de logiciels
- 1.3. Développement de logiciels orientés objet
 - 1.3.1. Introduction
 - 1.3.2. Principes de l'orientation objet
 - 1.3.3. Définition de l'objet
 - 1.3.4. Définition de la classe
 - 1.3.5. Analyse orientée objet vs. Conception orientée objet
- 1.4. Développement logiciel guidé par le modèle
 - 1.4.1. La nécessité de la modélisation
 - 1.4.2. Modélisation des systèmes logiciels
 - 1.4.3. Modélisation d'objets
 - 1.4.4. UML
 - 1.4.5. Outils de CASE
- 1.5. Modélisation d'applications et patrons de conception
 - 1.5.1. Modélisation avancée des exigences
 - 1.5.2. Modélisation statique avancée
 - 1.5.3. Modélisation dynamique avancée
 - 1.5.4. Modélisation des composants
 - 1.5.5. Introduction aux modèles de conception avec UML
 - 1.5.6. *Adapter*
 - 1.5.7. *Factory*
 - 1.5.8. *Singleton*
 - 1.5.9. *Strategy*
 - 1.5.10. *Composite*
 - 1.5.11. *Facade*
 - 1.5.12. *Observer*
- 1.6. Ingénierie guidée par le modèle
 - 1.6.1. Introduction
 - 1.6.2. Métamodélisation des systèmes
 - 1.6.3. MDA
 - 1.6.4. DSL
 - 1.6.5. Raffinements de modèles avec OCL
 - 1.6.6. Transformations de modèles
- 1.7. Ontologie dans l'Ingénierie du Logiciel
 - 1.7.1. Introduction
 - 1.7.2. Ontologies dans l'Ingénierie du Logiciel
 - 1.7.3. Application des Ontologies dans l'Ingénierie du Logiciel
- 1.8. Méthodologies agiles pour le développement du logiciel, *Scrum*
 - 1.8.1. Qu'est-ce que la agilité dans le logiciel?
 - 1.8.2. Le manifeste agile
 - 1.8.3. La charte d'un projet agile
 - 1.8.4. Le *Product Owner*
 - 1.8.5. Les histoires de l'utilisateur
 - 1.8.6. Planification et estimation agile
 - 1.8.7. Mesures dans le développement agile
 - 1.8.8. Introduction à Scrum
 - 1.8.9. Les rôles
 - 1.8.10. Le *Product Backlog*
 - 1.8.11. Le *Sprint*
 - 1.8.12. Réunions

- 1.9. La méthodologie de développement de logiciels *Lean*
 - 1.9.1. Introduction
 - 1.9.2. *KANBAN*
- 1.10. Qualité et amélioration des processus logiciels
 - 1.10.1. Introduction
 - 1.10.2. Mesure du logiciel
 - 1.10.3. Tests du logiciel
 - 1.10.4. Modèle de qualité de processus logiciel: CMMI

Module 2. Gestion de projets logiciels

- 2.1. Concepts fondamentaux de la gestion de projet et le cycle de vie de la gestion de projet
 - 2.1.1. Qu'est-ce qu'un projet?
 - 2.1.2. Méthodologie commune
 - 2.1.3. Qu'est-ce que la gestion/leadership de projet?
 - 2.1.4. Qu'est-ce qu'un plan de projet?
 - 2.1.5. Bénéfices
 - 2.1.6. Cycle de vie d'un projet
 - 2.1.7. Groupes de processus ou cycle de vie de la gestion de projet
 - 2.1.8. La relation entre les groupes de processus et les domaines de connaissances
 - 2.1.9. Relations entre le cycle de vie du produit et du projet
- 2.2. Initiation et planification
 - 2.2.1. De l'idée au projet
 - 2.2.2. Élaboration de la charte du projet
 - 2.2.3. Réunion de lancement du projet
 - 2.2.4. Tâches, connaissances et compétences dans le processus de démarrage
 - 2.2.5. Le plan de projet
 - 2.2.6. Développement du plan de base Étapes
 - 2.2.7. Tâches, connaissances et compétences dans le processus de planification
- 2.3. La gestion des *Stakeholders* et de la sensibilisation
 - 2.3.1. Identifier les parties prenantes
 - 2.3.2. Développer un plan de gestion des parties prenantes
 - 2.3.3. Gérer l'engagement des parties prenantes
 - 2.3.4. Contrôler l'engagement des parties prenantes
 - 2.3.5. L'objectif du projet
 - 2.3.6. La gestion de la portée et son plan
 - 2.3.7. Recueillir les besoins
 - 2.3.8. Définir l'énoncé de la portée
 - 2.3.9. Créer le WBS
 - 2.3.10. Vérifier et contrôler la portée
- 2.4. L'élaboration du calendrier
 - 2.4.1. La gestion du temps et son plan
 - 2.4.2. Définir les activités
 - 2.4.3. Établissement de la séquence des activités
 - 2.4.4. Estimation des ressources des activités
 - 2.4.5. Estimation des ressources pour les activités
 - 2.4.6. Développement du calendrier et calcul du chemin critique
 - 2.4.7. Contrôle des horaires
- 2.5. Élaboration du budget et réponse aux risques
 - 2.5.1. Estimation des coûts
 - 2.5.2. Élaboration du budget et de la courbe en S
 - 2.5.3. Contrôle des coûts et méthode de la valeur acquise
 - 2.5.4. Concepts de risque
 - 2.5.5. Comment faire une analyse de risque?
 - 2.5.6. L'élaboration du plan de réponse
- 2.6. Gestion de la qualité
 - 2.6.1. Planification de la qualité
 - 2.6.2. Assurance de la qualité
 - 2.6.3. Contrôle de la qualité
 - 2.6.4. Concepts statistiques de base
 - 2.6.5. Outils de la gestion de la qualité

- 2.7. La communication et les ressources humaines
 - 2.7.1. Planification de la gestion des communications
 - 2.7.2. Analyse des besoins de communication
 - 2.7.3. Technologie des communications
 - 2.7.4. Modèle de communication
 - 2.7.5. Méthodes de communication
 - 2.7.6. Plan gestion des communications
 - 2.7.7. Gestion des communications
 - 2.7.8. La gestion des ressources humaines
 - 2.7.9. Principaux acteurs et leurs rôles dans les projets
 - 2.7.10. Types d'organisations
 - 2.7.11. Organisation du projet
 - 2.7.12. L'équipe de travail
- 2.8. L'approvisionnement
 - 2.8.1. Le processus d'acquisitions
 - 2.8.2. Planification
 - 2.8.3. Recherche de fournisseurs et appels d'offres
 - 2.8.4. Attribution du contrat
 - 2.8.5. Administration des contrats
 - 2.8.6. Contrats
 - 2.8.7. Types de contrats
 - 2.8.8. Négociation de contrats
- 2.9. Exécution, suivi et contrôle et clôture
 - 2.9.1. Groupes de processus
 - 2.9.2. L'exécution des projets
 - 2.9.3. Suivi et contrôle des projets
 - 2.9.4. Clôture du projet
- 2.10. Responsabilité professionnelle
 - 2.10.1. Responsabilité professionnelle
 - 2.10.2. Caractéristiques de la responsabilité sociale et professionnelle
 - 2.10.3. Code d'éthique du chef de projet
 - 2.10.4. Responsabilité vs PMP®
 - 2.10.5. Exemples de responsabilité
 - 2.10.6. Avantages de la professionnalisation



Module 3. Plateformes de développement de logiciels

- 3.1. Introduction au développement d'application
 - 3.1.1. Applications de bureau
 - 3.1.2. Langage de programmation
 - 3.1.3. Environnements de développement intégrés
 - 3.1.4. Applications web
 - 3.1.5. Applications mobiles
 - 3.1.6. Applications en nuage
- 3.2. Développement d'applications et d'interfaces graphiques en *Java*
 - 3.2.1. Environnements de développement intégrés pour *Java*
 - 3.2.2. Principaux IDE de *Java*
 - 3.2.3. Introduction à la plateforme de développement *Eclipse*
 - 3.2.4. Introduction à la plateforme de développement *Eclipse*
 - 3.2.5. Modèle contrôleur-vue pour les interfaces utilisateur graphiques
 - 3.2.6. Réaliser une interface utilisateur graphique dans *Eclipse*
 - 3.2.7. Réaliser une interface utilisateur graphique dans *NetBeans*
- 3.3. Débogage et test dans *Java*
 - 3.3.1. Test et débogage des programmes *Java*
 - 3.3.2. Débogage dans *Eclipse*
 - 3.3.3. Débogage dans *NetBeans*
- 3.4. Développement d'applications et d'interfaces graphiques en .NET
 - 3.4.1. *Net Framework*
 - 3.4.2. Composants de la plate-forme de développement .NET
 - 3.4.3. Visual Studio .NET
 - 3.4.4. Outils GUI .NET
 - 3.4.5. L'interface graphique avec *Windows Presentation Foundation*
 - 3.4.6. Déboguer et compiler une application WPF
- 3.5. Programmation pour les réseaux .NET
 - 3.5.1. Introduction à la programmation réseau .NET
 - 3.5.2. Demandes et réponses dans .NET
 - 3.5.3. Utilisation des protocoles d'application dans .NET
 - 3.5.4. Sécurité dans programmation réseau .NET
- 3.6. Environnements de développement d'applications mobiles
 - 3.6.1. Applications mobiles
 - 3.6.2. Applications mobiles *Android*
 - 3.6.3. Étapes du développement d' *Android*
 - 3.6.4. L'IDE *Android Studio*
- 3.7. Développer des applications dans l'environnement *Android Studio*
 - 3.7.1. Installer et démarrer *Android Studio*
 - 3.7.2. Exécution d'une application *Android*
 - 3.7.3. Développer l'interface graphique dans *Android Studio*
 - 3.7.4. Lancement d'activités dans *Android Studio*
- 3.8. Débogage et publication des applications *Android*
 - 3.8.1. Déboguer une application dans *Android Studio*
 - 3.8.2. Stockage des applications dans *Android Studio*
 - 3.8.3. Publier une application sur *Google Play*
- 3.9. Développer des applications pour le cloud
 - 3.9.1. *Cloud computing*
 - 3.9.2. Niveaux du *Cloud*: SaaS, PaaS, IaaS
 - 3.9.3. Principales plateformes de développement en nuage
 - 3.9.4. Références bibliographiques
- 3.10. Introduction à Google *Cloud Platform*
 - 3.10.1. Notions de base de Google *Cloud Platform*
 - 3.10.2. service Google *Cloud Platform*
 - 3.10.3. Outils de Google *Cloud Platform*

Module 4. Informatique client Web

- 4.1. Informatique du client Web
 - 4.1.1. Structure d'un document
 - 4.1.2. Couleur
 - 4.1.3. Texte
 - 4.1.4. Liens hypertextes
 - 4.1.5. Images
 - 4.1.6. Listes
 - 4.1.7. Tables
 - 4.1.8. Cadres (*Frames*)
 - 4.1.9. Formulaire
 - 4.1.10. Éléments spécifiques aux technologies mobiles
 - 4.1.11. Éléments désaffectés
- 4.2. Fiches de style Web (CSS)
 - 4.2.1. Éléments et structure d'une fiche de style
 - 4.2.1.1. Création de fiches de style
 - 4.2.1.2. Application des modèles. Sélecteurs
 - 4.2.1.3. Héritage de style et cascade
 - 4.2.1.4. Mise en forme des pages à l'aide de modèles
 - 4.2.1.5. Mise en forme des pages à l'aide de modèles. Le modèle de boîte
 - 4.2.2. Concevoir des styles pour différents appareils
 - 4.2.3. Types de feuilles de style: statiques et dynamiques Pseudo-classes
 - 4.2.4. Bonnes pratiques dans l'utilisation des fiches de style
- 4.3. Introduction et histoire de *JavaScript*
 - 4.3.1. Introduction
 - 4.3.2. Histoire de *JavaScript*
 - 4.3.3. Environnement de développement que nous allons utiliser
- 4.4. Notions de base de la programmation web
 - 4.4.1. Syntaxe de base de *JavaScript*
 - 4.4.2. Types de données et opérateurs primitifs
 - 4.4.3. Variables et champs d'application
 - 4.4.4. Chaînes de texte et *Template Literals*
 - 4.4.5. Nombres et booléens
 - 4.4.6. Comparaisons
- 4.5. Structures complexes en *JavaScript*
 - 4.5.1. Vecteurs ou *Arrays* et objets
 - 4.5.2. Conjonctifs
 - 4.5.3. Cartes
 - 4.5.4. Disjonctions
 - 4.5.5. Boucles
- 4.6. Fonctions et objets
 - 4.6.1. Définition et invocation des fonctions
 - 4.6.2. Arguments
 - 4.6.3. Fonctions des flèches
 - 4.6.4. Fonctions de rappel ou *Callback*
 - 4.6.5. Fonctions d'ordre supérieur
 - 4.6.6. Objets littéraux
 - 4.6.7. L'objet *This*
 - 4.6.8. Objets en tant qu'espaces de noms: l'objet *Math* et l'objet *Date*
- 4.7. Le modèle d'objet de document (DOM)
 - 4.7.1. Qu'est-ce que le DOM?
 - 4.7.2. Un peu d'histoire
 - 4.7.3. Navigation et récupération d'éléments
 - 4.7.4. Un DOM virtuel avec JSDOM
 - 4.7.5. Sélecteurs de requêtes ou *Query Selectors*
 - 4.7.6. Navigation dans les propriétés
 - 4.7.7. Attribution d'attributs aux éléments
 - 4.7.8. Création et modification de nœuds
 - 4.7.9. Mise à jour du style des éléments du DOM
- 4.8. Développement web moderne
 - 4.8.1. Flux piloté par les événements et *Listeners*
 - 4.8.2. *Toolkits* web modernes et systèmes d'alignement
 - 4.8.3. Mode écrit de *JavaScript*
 - 4.8.4. Un peu plus sur les fonctions
 - 4.8.5. Fonctions asynchrones et promesses
 - 4.8.6. *Fermetures*
 - 4.8.7. Programmation fonctionnelle
 - 4.8.8. OOP dans *JavaScript*

- 4.9. Utilisabilité du Web
 - 4.9.1. Introduction à l'utilisabilité
 - 4.9.2. Définition de l'utilisabilité
 - 4.9.3. Importance d'une conception web centrée sur l'utilisateur
 - 4.9.4. Différences entre l'accessibilité et la facilité d'utilisation
 - 4.9.5. Avantages et problèmes liés à la combinaison de l'accessibilité et de la facilité d'utilisation
 - 4.9.6. Avantages et difficultés de la mise en œuvre de sites web utilisables
 - 4.9.7. Méthodes d'utilisabilité
 - 4.9.8. Analyse des besoins des utilisateurs
 - 4.9.9. Principes de conception Prototypage orienté vers l'utilisateur
 - 4.9.10. Directives pour la création de sites web utilisables
 - 4.9.10.1. Directives d'utilisabilité selon *Jakob Nielsen*
 - 4.9.10.2. Directives d'utilisabilité selon Bruce Tognazzini
 - 4.9.11. Évaluation de la convivialité
- 4.10. Accessibilité du Web
 - 4.10.1. Introduction
 - 4.10.2. Définition de l'accessibilité du Web
 - 4.10.3. Types de handicaps
 - 4.10.3.1. Handicaps temporaires ou permanents
 - 4.10.3.2. Déficiences visuelles
 - 4.10.3.3. Déficiences auditives
 - 4.10.3.4. Handicaps moteurs
 - 4.10.3.5. Handicaps neurologiques ou cognitifs
 - 4.10.3.6. Difficultés liées au vieillissement
 - 4.10.3.7. Limitations environnementales
 - 4.10.3.8. Obstacles à l'accès au web
 - 4.10.4. Aides techniques et produits d'assistance pour surmonter les obstacles
 - 4.10.4.1. Aides pour les aveugles
 - 4.10.4.2. Aides pour les personnes atteintes de basse vision
 - 4.10.4.3. Aides pour les personnes atteintes de daltonisme
 - 4.10.4.4. Aides pour les malentendants
 - 4.10.4.5. Aides pour les personnes souffrant d'un handicap moteur
 - 4.10.4.6. Aides pour les personnes souffrant de handicaps cognitifs et neurologiques
 - 4.10.5. Avantages et difficultés de la mise en œuvre de l'accessibilité du Web
 - 4.10.6. Réglementations et normes en matière d'accessibilité du Web
 - 4.10.7. Organismes de réglementation de l'accessibilité du Web
 - 4.10.8. Comparaison des normes et des standards
 - 4.10.9. Lignes directrices pour le respect des réglementations et des normes
 - 4.10.9.1. Description des principales orientations (images, liens vidéo, etc.)
 - 4.10.9.2. Directives pour une navigation accessible
 - 4.10.9.2.1. Perceptibilité
 - 4.10.9.2.2. Exploitabilité
 - 4.10.9.2.3. Compréhensibilité
 - 4.10.9.2.4. Résistance
 - 4.10.10. Description du processus de conformité de l'accessibilité du Web
 - 4.10.11. Niveaux de conformité
 - 4.10.12. Critères de conformité
 - 4.10.13. Exigences de conformité
 - 4.10.14. Méthodologie d'évaluation de l'accessibilité des sites web

Module 5. Informatique du serveur web

- 5.1. Introduction à la programmation côté serveur: PHP
 - 5.1.1. Concepts de base de la programmation côté serveur
 - 5.1.2. Syntaxe PHP de base
 - 5.1.3. Générer du contenu HTML avec PHP
 - 5.1.4. Environnements de développement et de test: XAMPP
- 5.2. PHP avancé
 - 5.2.1. Structures de contrôle avec PHP
 - 5.2.2. Fonctions en PHP
 - 5.2.3. Gestion de *Arrays* en PHP
 - 5.2.4. Manipulation des chaînes en PHP
 - 5.2.5. Orientation objet en PHP
- 5.3. Modèles de données
 - 5.3.1. Concept de données. Cycle de vie des données
 - 5.3.2. Types de données
 - 5.3.2.1. Principes de base
 - 5.3.2.2. Enregistrements
 - 5.3.2.3. Dynamiques

- 5.4. Modèle relationnel
 - 5.4.1. Description
 - 5.4.2. Entités et types d'entités
 - 5.4.3. Éléments de données. Attributs
 - 5.4.4. Relations: types, sous-types, cardinalité
 - 5.4.5. Clés. Types de clés
 - 5.4.6. Normalisation Formes normales
- 5.5. Construction du modèle logique de données
 - 5.5.1. Spécification des tables
 - 5.5.2. Définition des colonnes
 - 5.5.3. Spécification clés
 - 5.5.4. Conversion en formes normales Dépendances
- 5.6. Le modèle physique de données. Fichiers de données
 - 5.6.1. Description des fichiers de données
 - 5.6.2. Types de fichiers
 - 5.6.3. Modes d'accès
 - 5.6.4. Organisation des fichiers
- 5.7. Accès aux bases de données depuis PHP
 - 5.7.1. Introduction à *MariaDB*
 - 5.7.2. Travailler avec une base de données *MariaDB*: le langage SQL
 - 5.7.3. Accéder à la base de données *MariaDB* desde PHP
 - 5.7.4. Introduction à *MySQL*
 - 5.7.5. Travailler avec une base de données *MySQL*: le langage SQL
 - 5.7.6. Accéder à la base de données *MySQL* desde PHP
- 5.8. Interaction avec le client à partir de PHP
 - 5.8.1. Formulaires PHP
 - 5.8.2. *Cookies*
 - 5.8.3. Traitement des sessions
- 5.9. Architecture d'applications Web
 - 5.9.1. Le modèle Modèle Vue Contrôleur Modèle
 - 5.9.2. Contrôle
 - 5.9.3. Modèle
 - 5.9.4. Voir

- 5.10. Introduction aux services Web
 - 5.10.1. Introduction à XML
 - 5.10.2. Architectures orientées services (SOA): Services Web
 - 5.10.3. Création de services web SOAP et REST
 - 5.10.4. Le protocole SOAP
 - 5.10.5. Le protocole REST

Module 6. Gestion de la sécurité

- 6.1. La sécurité de l'information
 - 6.1.1. Introduction
 - 6.1.2. La sécurité des informations implique la confidentialité, l'intégrité et la disponibilité
 - 6.1.3. La sécurité est une question économique
 - 6.1.4. La sécurité est un processus
 - 6.1.5. La classification des informations
 - 6.1.6. La sécurité de l'information implique la gestion des risques
 - 6.1.7. La sécurité s'articule avec les contrôles de sécurité
 - 6.1.8. La sécurité est à la fois physique et logique
 - 6.1.9. La sécurité implique des personnes
- 6.2. Le professionnel de la sécurité de l'information
 - 6.2.1. Introduction
 - 6.2.2. La sécurité de l'information en tant que profession
 - 6.2.3. Les certifications (ISC)²
 - 6.2.4. La norme ISO 27001
 - 6.2.5. Bonnes pratiques de sécurité dans la gestion des services informatiques
 - 6.2.6. Modèles de maturité de la sécurité de l'information
 - 6.2.7. Autres certifications, normes et ressources professionnelles
- 6.3. Contrôle d'accès
 - 6.3.1. Introduction
 - 6.3.2. Exigences en matière de contrôle d'accès
 - 6.3.3. Mécanismes d'authentification
 - 6.3.4. Méthodes d'autorisation
 - 6.3.5. Comptabilité et audit des accès
 - 6.3.6. Technologies "Triple A"

- 6.4. Programmes, processus et politiques de sécurité de l'information
 - 6.4.1. Introduction
 - 6.4.2. Programmes de gestion de la sécurité
 - 6.4.3. La gestion des risques
 - 6.4.4. Conception de la politique de sécurité
- 6.5. Plans de continuité des activités
 - 6.5.1. Introduction aux PCA
 - 6.5.2. Phase I et II
 - 6.5.3. Phase III et IV
 - 6.5.4. Maintenance du PCA
- 6.6. Procédures pour la protection adéquate de l'entreprise
 - 6.6.1. Réseaux DMZ
 - 6.6.2. Systèmes de détection d'intrusion
 - 6.6.3. Listes de contrôle d'accès
 - 6.6.4. Apprendre de l'agresseur *Honeypot*
- 6.7. Architectures de sécurité. Prévention
 - 6.7.1. Aperçu général. Activités et modèle de superposition
 - 6.7.2. Défense du périmètre (*Firewalls, WAFs, IPS, etc.*)
 - 6.7.3. Défense des points d'extrémité (équipements, serveurs et services)
- 6.8. Architectures de sécurité Détection
 - 6.8.1. Aperçu de la détection et de la surveillance
 - 6.8.2. Journaux, rupture de trafic crypté, enregistrement et Siems
 - 6.8.3. Alertes et renseignements
- 6.9. Architectures de sécurité Réaction
 - 6.9.1. Réaction Produits, services et ressources
 - 6.9.2. Gestion des incidents
 - 6.9.3. CERTS y CSIRTs
- 6.10. Architectures de sécurité Récupération
 - 6.10.1. Résilience, concepts, exigences commerciales et normes
 - 6.10.2. Solutions informatiques de résilience
 - 6.10.3. Gestion de crise et gouvernance

Module 7. Sécurité des systèmes d'information

- 7.1. Un aperçu de la sécurité, de la cryptographie et des cryptanalyses classiques.
 - 7.1.1. Sécurité informatique: perspective historique
 - 7.1.2. Mais qu'est-ce que la sécurité exactement?
 - 7.1.3. Histoire de la cryptographie
 - 7.1.4. Chiffres de substitution
 - 7.1.5. Étude de cas: la machine Enigma
- 7.2. Cryptographie symétrique
 - 7.2.1. Introduction et terminologie de base
 - 7.2.2. Cryptage symétrique
 - 7.2.3. Modes de fonctionnement
 - 7.2.4. DES
 - 7.2.5. La nouvelle norme AES
 - 7.2.6. Cryptage de flux
 - 7.2.7. Cryptanalyse
- 7.3. Cryptographie asymétrique
 - 7.3.1. Origines de la cryptographie à clé publique
 - 7.3.2. Concepts de base et fonctionnement
 - 7.3.3. L'algorithme RSA
 - 7.3.4. Certificats numériques
 - 7.3.5. Stockage et gestion des clés
- 7.4. Attaques de réseau
 - 7.4.1. Menaces et attaques contre les réseaux
 - 7.4.2. Énumération
 - 7.4.3. Interception du trafic *sniffers*
 - 7.4.4. Attaques par déni de service
 - 7.4.5. Attaques par empoisonnement ARP
- 7.5. Architectures de sécurité
 - 7.5.1. Architectures de sécurité traditionnelles
 - 7.5.2. *Secure Socket Layer*: SSL
 - 7.5.3. Protocole SSH
 - 7.5.4. Réseaux privés virtuels (VPN)
 - 7.5.5. Mécanismes de protection des unités de stockage externes
 - 7.5.6. Mécanismes de protection du matériel

- 7.6. Techniques de protection des systèmes et développement de code sécurisé
 - 7.6.1. Sécurité des opérations
 - 7.6.2. Ressources et contrôle
 - 7.6.3. Suivi
 - 7.6.4. Systèmes de détection d'intrusion
 - 7.6.5. IDS de l'hôte
 - 7.6.6. IDS réseau
 - 7.6.7. IDS basé sur les signatures
 - 7.6.8. Systèmes de leurres
 - 7.6.9. Principes de base de la sécurité dans le développement du code
 - 7.6.10. Gestion des pannes
 - 7.6.11. Ennemi public numéro 1: le dépassement de tampon
 - 7.6.12. Botches cryptographiques
- 7.7. Botnets et spam
 - 7.7.1. Origine du problème
 - 7.7.2. Processus de spam
 - 7.7.3. Envoi de spam
 - 7.7.4. Affinement des listes de diffusion
 - 7.7.5. Techniques de protection
 - 7.7.6. Service anti-spam offert par des tiers
 - 7.7.7. Étude de cas
 - 7.7.8. Spam exotique
- 7.8. Audit et attaques du Web
 - 7.8.1. Collecte d'informations
 - 7.8.2. Techniques d'attaque
 - 7.8.3. Outils
- 7.9. Malware et code malveillant
 - 7.9.1. Qu'est-ce qu'un malware?
 - 7.9.2. Types de malware
 - 7.9.3. Virus
 - 7.9.4. Cryptovirus
 - 7.9.5. Vers
 - 7.9.6. Adware
 - 7.9.7. Spyware

- 7.9.8. Hoaxes
- 7.9.9. Phishing
- 7.9.10. Cheval de Troie
- 7.9.11. L'économie des malware
- 7.9.12. Solutions possibles
- 7.10. Analyse médico-légale
 - 7.10.1. Collecte des preuves
 - 7.10.2. Analyse des preuves
 - 7.10.3. Techniques anti-forensic
 - 7.10.4. Étude de cas pratique

Module 8. Sécurité du logiciel

- 8.1. Questions relatives à la sécurité des logiciels
 - 8.1.1. Introduction au problème de la sécurité des logiciels
 - 8.1.2. Vulnérabilité et leur classification
 - 8.1.3. Principes de conception de la sécurité des logiciels
 - 8.1.4. Références
- 8.2. Principes de conception de la sécurité des logiciels
 - 8.2.1. Introduction
 - 8.2.2. Principes de conception de la sécurité des logiciels
 - 8.2.3. Types de S-SDLC
 - 8.2.4. Sécurité des logiciels dans les phases S-SDLC
 - 8.2.5. Méthodologies et normes
 - 8.2.6. Références
- 8.3. Sécurité dans le cycle de vie du logiciel dans les phases d'exigences et de conception
 - 8.3.1. Introduction
 - 8.3.2. Modélisation des attaques
 - 8.3.3. Cas d'abus
 - 8.3.4. Ingénierie des exigences de sécurité
 - 8.3.5. Analyse des risques Architectural
 - 8.3.6. Modèles de conception
 - 8.3.7. Références

- 8.4. Sécurité du cycle de vie des logiciels dans les phases de codage, de test et d'exploitation
 - 8.4.1. Introduction
 - 8.4.2. Tests de sécurité fondés sur le risque
 - 8.4.3. Examen du code
 - 8.4.4. Test de pénétration
 - 8.4.5. Opérations de sécurité
 - 8.4.6. Examen externe
 - 8.4.7. Références
- 8.5. Applications de codage sécurisé I
 - 8.5.1. Introduction
 - 8.5.2. Pratiques de codage sécurisées
 - 8.5.3. Traitement et validation des entrées
 - 8.5.4. Débordement de mémoire
 - 8.5.5. Références
- 8.6. Applications de codage sécurisé II
 - 8.6.1. Introduction
 - 8.6.2. *Integers Overflows*, erreurs de troncature et problèmes de conversion de type entre entiers
 - 8.6.3. Erreurs et exceptions
 - 8.6.4. Vie privée et confidentialité
 - 8.6.5. Programmes privilégiés
 - 8.6.6. Références
- 8.7. Développement et sécurité du cloud
 - 8.7.1. Sécurité du développement; méthodologie et pratique
 - 8.7.2. Modèles PaaS, PaaS et SaaS
 - 8.7.3. Sécurité dans le nuage et pour les services en nuage
- 8.8. Orchestration et automatisation de la sécurité (SOAR)
 - 8.9.1. Complexité du traitement manuel; nécessité d'automatiser les tâches
 - 8.9.2. Produits et services
 - 8.9.3. Architecture du SOAR
- 8.9. Sécurité dans le télétravail
 - 8.9.1. Besoin et scénarios
 - 8.9.2. Produits et services
 - 8.9.3. Sécurité dans le télétravail

Module 9. Qualité et contrôle des systèmes d'information

- 9.1. Introduction aux Systèmes de Gestion de la Sécurité de l'Information
 - 9.1.1. Principes fondamentaux du SMSI
 - 9.1.2. Règles d'or du SMSI
 - 9.1.3. Rôle de l'audit informatique dans les SMSI
- 9.2. Planification de la gestion de la Sécurité
 - 9.2.1. Concepts liés à la gestion de la sécurité
 - 9.2.2. Classification de l'information: objectifs, concepts et rôles
 - 9.2.3. Mise en œuvre des politiques de sécurité: politiques, normes et procédures de sécurité
 - 9.2.4. Gestion du risque: principes et analyse du risque lié aux actifs informationnels
- 9.3. Principaux mécanismes de protection des actifs informationnels (I)
 - 9.3.1. Aperçu des principaux outils cryptographiques pour la protection de la triade CID
 - 9.3.2. Prise en compte de la vie privée, de l'anonymat et de la gestion appropriée des exigences de traçabilité des utilisateurs
- 9.4. Principaux mécanismes de protection des actifs informationnels (II)
 - 9.4.1. Sécurité des communications: protocoles, dispositifs et architectures de sécurité
 - 9.4.2. Sécurité des systèmes d'exploitation
- 9.5. Contrôles internes du SMSI
 - 9.5.1. Taxonomie des contrôles du SMSI: contrôles administratifs, logiques et physiques
 - 9.5.2. Classification des contrôles en fonction de la manière dont ils traitent la menace: contrôles pour la prévention, la détection et la correction des menaces
 - 9.5.3. Mise en œuvre des systèmes de contrôle interne dans les SMSI
- 9.6. Types de contrôles
 - 9.6.1. Différence entre l'audit et le contrôle interne
 - 9.6.2. Contrôle interne et contrôle externe
 - 9.6.3. Classification du contrôle en fonction de l'objectif et du type d'analyse
- 9.7. Scénariste et scénario : sujet et objet protégés par la propriété intellectuelle
 - 9.7.1. Introduction aux tests de pénétration et à l'analyse médico-légale
 - 9.7.2. Définition et pertinence des concepts de *Fingerprinting* et *Footprinting*

- 9.8. Analyse de vulnérabilité et surveillance du trafic réseau
 - 9.8.1. Outils pour l'analyse des vulnérabilités des systèmes
 - 9.8.2. Principales vulnérabilités dans le contexte des applications Web
 - 9.8.3. Analyse des protocoles de communication
- 9.9. Le processus d'audit informatique
 - 9.9.1. Concept de cycle de vie du développement des systèmes
 - 9.9.2. Surveillance des activités et des processus: collecte et traitement des preuves
 - 9.9.3. Méthodologie de contrôle informatique
 - 9.9.4. Processus d'un contrôle informatique
 - 9.9.5. Identification des principales infractions et délits dans le contexte informatique
 - 9.9.6. Enquête sur la criminalité informatique: introduction à l'analyse judiciaire et à sa relation avec le contrôle informatique
- 9.10. Plans de continuité des activités et de reprise après sinistre
 - 9.10.1. Définition du plan de continuité des activités et du concept d'interruption des activités
 - 9.10.2. Recommandation du NIST sur les plans de continuité des activités
 - 9.10.3. Plan de reprise après sinistre
 - 9.10.4. Processus du plan de reprise après sinistre

Module 10. Administration de serveurs Web

- 10.1. Introduction aux serveurs web
 - 10.1.1. Qu'est-ce qu'un serveur web?
 - 10.1.2. Architecture et fonctionnement d'un serveur web
 - 10.1.3. Ressources et contenus dans un serveur web
 - 10.1.4. Serveurs d'application
 - 10.1.5. Serveurs Proxy
 - 10.1.6. Principaux serveurs web sur le marché
 - 10.1.7. Statistiques d'utilisation du serveur Web
 - 10.1.8. Sécurité des serveurs Web
 - 10.1.9. Équilibrage des charges dans les serveurs web
 - 10.1.10. Références

- 10.2. Gestion du protocole HTTP
 - 10.2.1. Fonctionnement et structure
 - 10.2.2. Description des demandes ou *Request Methods*
 - 10.2.3. Codes d'état
 - 10.2.4. En-têtes
 - 10.2.5. Codage du contenu Pages de code
 - 10.2.6. Effectuer des requêtes HTTP sur Internet en utilisant un proxy, *Livehttpheaders* ou une méthode similaire, en analysant le protocole utilisé
- 10.3. Description des architectures distribuées sur plusieurs serveurs
 - 10.3.1. Modèle à 3 couches
 - 10.3.2. Modèle à 3 couches
 - 10.3.3. Partage de la charge
 - 10.3.4. Magasins de l'État de la session
 - 10.3.5. Magasins de cache
- 10.4. *Internet Information Services (IIS)*
 - 10.4.1. Qu'est-ce que IIS?
 - 10.4.2. Histoire et évolution de IIS
 - 10.4.3. Principaux avantages et caractéristiques de IIS et au-delà
 - 10.4.4. Architecture IIS et supérieure
- 10.5. Installation, administration et configuration de IIS
 - 10.5.1. Préambule
 - 10.5.2. Installation d' *Internet Information Services (IIS)*
 - 10.5.3. Outils d'administration d'IIS
 - 10.5.4. Créer, configurer et administrer des sites Web
 - 10.5.5. Installation et gestion des extensions IIS
- 10.6. Sécurité avancée dans IIS
 - 10.6.1. Préambule
 - 10.6.2. Authentification, autorisation et contrôle d'accès à IIS
 - 10.6.3. Configuration d'un site Web sécurisé sur IIS avec SSL
 - 10.6.4. Politiques de sécurité mises en œuvre dans IIS 10.x
- 10.7. Introduction à Apache
 - 10.7.1. Qu'est-ce qu'Apache?
 - 10.7.2. Principaux avantages d'Apache
 - 10.7.3. Caractéristiques principales d'Apache
 - 10.7.4. Architecture

- 10.8. Installation et configuration d'Apache
 - 10.8.1. Installation initiale d'Apache
 - 10.8.2. Configuration d'Apache
- 10.9. Installer et configurer les différents modules Apache
 - 10.9.1. Installation des modules Apache
 - 10.9.2. Types de modules
 - 10.9.3. Configuration sécurisée d'Apache
- 10.10. Sécurité avancée
 - 10.10.1. Authentification, autorisation et contrôle d'accès
 - 10.10.2. Méthodes d'authentification
 - 10.10.3. Configuration sécurisée d'Apache avec SSL

Module 11. Sécurité des applications en ligne

- 11.1. Vulnérabilités et problèmes de sécurité dans les applications en ligne
 - 11.1.1. Introduction à la sécurité des applications en ligne
 - 11.1.2. Failles de sécurité dans la conception des applications web
 - 11.1.3. Vulnérabilités de sécurité dans la mise en œuvre des applications web
 - 11.1.4. Vulnérabilité de la sécurité dans le déploiement des applications web
 - 11.1.5. Listes officielles de failles de sécurité
- 11.2. Politiques et normes pour la sécurité des applications en ligne
 - 11.2.1. Piliers de la sécurité des applications en ligne
 - 11.2.2. Politique de sécurité
 - 11.2.3. Système de gestion de la sécurité de l'information
 - 11.2.4. Cycle de vie du développement logiciel sécurisé
 - 11.2.5. Normes de sécurité des applications
- 11.3. Sécurité dans la conception des applications web
 - 11.3.1. Introduction à la sécurité des applications en Web
 - 11.3.2. Sécurité dans la conception des applications web
- 11.4. Tester la sécurité en ligne et la protection des applications web
 - 11.4.1. Analyse et test de la sécurité des applications Web
 - 11.4.2. Sécurité du déploiement et de la production des applications Web
- 11.5. Sécurité des services Web
 - 11.5.1. Introduction à la sécurité des services web
 - 11.5.2. Fonctions et technologies de sécurité des services Web

- 11.6. Tester la sécurité et la protection en ligne des services web
 - 11.6.1. Évaluation de la sécurité des services web
 - 11.6.2. Protection en ligne. *Firewalls* et *Gateways* XML
- 11.7. *Hacking* éthique, malware et *Forensic*
 - 11.7.1. Piratage éthique
 - 11.7.2. Hacking éthique
 - 11.7.3. Analyse médico-légale
- 11.8. Analyse des Malwares
 - 11.8.1. Suivi
 - 11.8.2. Outils de mesure de la performance
 - 11.8.3. Mesures de confinement
 - 11.8.4. Analyse des causes premières
 - 11.8.5. Gestion proactive des problèmes
- 11.9. Meilleures pratiques pour assurer la sécurité des applications
 - 11.9.1. Manuel de bonnes pratiques pour le développement d'applications en ligne
 - 11.9.2. Manuel de bonnes pratiques pour la mise en œuvre des applications en ligne
- 11.10. Erreurs courantes qui compromettent la sécurité des applications
 - 11.10.1. Erreurs courantes de développement
 - 11.10.2. Erreurs courantes en matière d'hébergement
 - 11.10.3. Erreurs courantes dans la production

Module 12. Ingénierie du logiciel

- 12.1. Introduction à l'ingénierie du logiciel et à la modélisation
 - 12.1.1. La nature du logiciel
 - 12.1.2. La nature unique des *Webapps*
 - 12.1.3. Ingénierie du logiciel
 - 12.1.4. Le processus du logiciel
 - 12.1.5. La pratique de l'ingénierie du logiciel
 - 12.1.6. Mythes sur les logiciels
 - 12.1.7. Comment tout cela commence-t-il?
 - 12.1.8. Concepts orientés objet
 - 12.1.9. Introduction à UML

- 12.2. Le processus du logiciel
 - 12.2.1. Un modèle général de processus
 - 12.2.2. Modèles de processus prescriptifs
 - 12.2.3. Modèles de processus spécialisés
 - 12.2.4. Le processus unifié
 - 12.2.5. Modèles de processus personnels et d'équipe
 - 12.2.6. Qu'est-ce que l'agilité?
 - 12.2.7. Qu'est-ce qu'un processus agile?
 - 12.2.8. SCRUM
 - 12.2.9. Boîte à outils du processus Agile
- 12.3. Principes guidant la pratique de l'ingénierie du logiciel
 - 12.3.1. Principes guidant le processus
 - 12.3.2. Principes guidant la pratique
 - 12.3.3. Principes de communication
 - 12.3.4. Principes de planification
 - 12.3.5. Principes de modélisation
 - 12.3.6. Principes de construction
 - 12.3.7. Principes de déploiement
- 12.4. Compréhension des besoins
 - 12.4.1. Ingénierie des exigences
 - 12.4.2. Établir les bases
 - 12.4.3. Enquête sur les besoins
 - 12.4.4. Développement de cas d'utilisation
 - 12.4.5. Elaboration du modèle d'exigences
 - 12.4.6. Négociation des besoins
 - 12.4.7. Validation des exigences
- 12.5. Modélisation des exigences: scénarios, classes d'information et d'analyse
 - 12.5.1. Analyse des besoins
 - 12.5.2. Modélisation basée sur des scénarios
 - 12.5.3. Modèles UML fournissant le cas d'utilisation
 - 12.5.4. Concepts de modélisation des données
 - 12.5.5. Modélisation basée sur les classes
 - 12.5.6. Diagrammes de classes
- 12.6. Modélisation des exigences: flux, comportement et modèles
 - 12.6.1. Stratégies de modélisation des exigences
 - 12.6.2. Modélisation orientée flux
 - 12.6.3. Diagrammes d'état
 - 12.6.4. Création d'un modèle comportemental
 - 12.6.5. Diagrammes de séquence
 - 12.6.6. Diagrammes de communication
 - 12.6.7. Modèles pour la modélisation des exigences
- 12.7. Concepts de design
 - 12.7.1. Conception dans le contexte du génie logiciel
 - 12.7.2. Le processus de conception
 - 12.7.3. Concepts de design
 - 12.7.4. Concepts de conception orientée objet
 - 12.7.5. Le modèle de conception
- 12.8. Architecture de conception
 - 12.8.1. Architecture du logiciel
 - 12.8.2. Genres architecturaux
 - 12.8.3. Styles architecturaux
 - 12.8.4. Conception architecturale
 - 12.8.5. Évolution des conceptions alternatives de l'architecture
 - 12.8.6. Cartographie de l'architecture à l'aide du flux de données
- 12.9. Conception au niveau des composants et des modèles
 - 12.9.1. Qu'est-ce qu'un composant?
 - 12.9.2. Conception de composants basée sur les classes
 - 12.9.3. Réalisation de la conception au niveau des composants
 - 12.9.4. Conception traditionnelle des composants
 - 12.9.5. Développement basé sur les composants
 - 12.9.6. Modèles de conception
 - 12.9.7. Conception de logiciels basée sur des modèles
 - 12.9.8. Modèles architecturaux
 - 12.9.9. Patrons de conception au niveau des composants
 - 12.9.10. Modèles de conception d'interface utilisateur

- 12.10. Qualité des logiciels et gestion de projet
 - 12.10.1. Qualité
 - 12.10.2. Qualité des logiciels
 - 12.10.3. Le dilemme de la qualité des logiciels
 - 12.10.4. Atteindre la qualité des logiciels
 - 12.10.5. Assurance qualité des logiciels
 - 12.10.6. Le spectre de la gestion
 - 12.10.7. Le personnel
 - 12.10.8. Le produit
 - 12.10.9. Le processus
 - 12.10.10. Le projet
 - 12.10.11. Principes et pratiques

Module 13. Ingénierie du logiciel avancé

- 13.1. Introduction aux méthodologies Agiles
 - 13.1.1. Modèles de processus et méthodologies
 - 13.1.2. Agilité et processus agiles
 - 13.1.3. Manifeste agile
 - 13.1.4. Quelques méthodologies agiles
 - 13.1.5. Agile vs. Traditionnel
- 13.2. SCRUM
 - 13.2.1. Origines et philosophie de Scrum
 - 13.2.2. Valeurs de Scrum
 - 13.2.3. Processus de Scrum
 - 13.2.4. Rôles de Scrum
 - 13.2.5. Les artefacts de Scrum
 - 13.2.6. Les événements de Scrum
 - 13.2.7. Les histoires de l'utilisateur
 - 13.2.8. Extensions de Scrum
 - 13.2.9. Estimations agiles
 - 13.2.10. Scrum Scaling
- 13.3. Extreme Programming
 - 13.3.1. Justification et aperçu de l'XP
 - 13.3.2. Le cycle de vie XP
 - 13.3.3. Les cinq valeurs fondamentales
 - 13.3.4. Les douze pratiques de base de XP
 - 13.3.5. Rôles des participants
 - 13.3.6. XP industriel
 - 13.3.7. Évaluation critique de XP
- 13.4. Développement logiciel basé sur la réutilisation
 - 13.4.1. La réutilisation du logiciel
 - 13.4.2. Niveaux de réutilisation du code
 - 13.4.3. Techniques spécifiques de réutilisation
 - 13.4.4. Développement basé sur les composants
 - 13.4.5. Bénéfices et problèmes de la réutilisation
 - 13.4.6. Planification de la réutilisation
- 13.5. Architecture de systèmes et modèles de conception de logiciels
 - 13.5.1. La conception architecturale
 - 13.5.2. Modèles architecturaux généraux
 - 13.5.3. Architectures tolérantes aux pannes
 - 13.5.4. Architectures de systèmes distribués
 - 13.5.5. Modèles de conception
 - 13.5.6. Motifs de Gamma
 - 13.5.7. Modèles de conception d'interaction
- 13.6. Architecture des applications en nuage
 - 13.6.1. Principes fondamentaux de *Cloud Computing*
 - 13.6.2. Qualité des applications en nuage
 - 13.6.3. Styles d'architecture
 - 13.6.4. Modèles de conception
- 13.7. Tests du logiciel: TDD, ATDD y BDD
 - 13.7.1. Vérification et validation des logiciels
 - 13.7.2. Les tests de logiciels
 - 13.7.3. *Test Driven Development* (TDD)
 - 13.7.4. *Test Driven Development* (ATDD)
 - 13.7.5. *Behavior Driven Development* (BDD)
 - 13.7.6. BDD et Cucumber

- 13.8. L'amélioration du processus du logiciel
 - 13.8.1. L'amélioration du processus du logiciel
 - 13.8.2. Le processus d'amélioration du processus
 - 13.8.3. Modèles de maturité
 - 13.8.4. Le modèle CMMI
 - 13.8.5. CMMI V13.0
 - 13.8.6. CMMI et Agile
- 13.9. La qualité du produit logiciel: *SQuaRE*
 - 13.9.1. La qualité des logiciels
 - 13.9.2. Modèles de qualité des produits logiciels
 - 13.9.3. Famille ISO/IEC 13000
 - 13.9.4. ISO/IEC 13.010: modèle et caractéristiques de la qualité
 - 13.9.5. ISO/IEC 13.0113: la qualité des données
 - 13.9.6. ISO/IEC 13.013.: mesure de la qualité des logiciels
 - 13.9.7. ISO/IEC 13.013., 13.013. et 13.013.: Métriques de qualité des logiciels et des données
 - 13.9.8. ISO/IEC 13.040: Évaluation des logiciels
 - 13.9.9. Le processus de certification
- 13.10. Introduction à *DevOps*
 - 13.10.1. Concept de DevOps
 - 13.10.2. Pratiques principales

Module 14. Ingénierie des exigences

- 14.1. Introduction à l'ingénierie des exigences
 - 14.1.1. L'importance des exigences
 - 14.1.2. Concept d'exigences
 - 14.1.3. Dimensions des exigences
 - 14.1.4. Niveaux et types d'exigences
 - 14.1.5. Caractéristiques des besoins
 - 14.1.6. L'ingénierie des exigences
 - 14.1.7. Le processus de l'Ingénierie des Besoins
 - 14.1.8. *Frameworks* pour l'ingénierie des besoins
 - 14.1.9. Bonnes pratiques en matière d'ingénierie des besoins
 - 14.1.10. L'analyste d'affaires
- 14.2. Les sources des besoins
 - 14.2.1. Le réseau des besoins
 - 14.2.2. Les *Stakeholders*
 - 14.2.3. Les besoins de l'entreprise
 - 14.2.4. Document sur la vision et le champ d'application
- 14.3. Techniques d'élicitation des besoins
 - 14.3.1. L'élicitation des besoins
 - 14.3.2. Problèmes liés à la détermination des besoins
 - 14.3.3. Contextes de découverte
 - 14.3.4. Interviews
 - 14.3.5. Observation et "apprentissage"
 - 14.3.6. Ethnographie
 - 14.3.7. *Workshops*
 - 14.3.8. *Focus groups*
 - 14.3.9. Questionnaires
 - 14.3.10. *Brainstorming* et techniques créatives
 - 14.3.11. Médias du groupe
 - 14.3.12. Analyse des interfaces du système
 - 14.3.13. Analyse de documents et "archéologie"
 - 14.3.14. Cas d'utilisation et scénarios
 - 14.3.15. Les prototypes
 - 14.3.16. Ingénierie inverse
 - 14.3.17. Réutilisation des exigences
 - 14.3.18. Bonnes pratiques en matière d'élicitation
- 14.4. Besoins des utilisateurs
 - 14.4.1. Personnes
 - 14.4.2. Cas d'utilisation et histoires d'utilisateurs
 - 14.4.3. Scénario
 - 14.4.4. Types de scénarios
 - 14.4.5. Comment découvrir des scénarios?

- 14.5. Techniques de prototypage
 - 14.5.1. Prototypage
 - 14.5.2. Prototypes en fonction de leur portée
 - 14.5.3. Prototypes en fonction de leur temporalité
 - 14.5.4. La fidélité d'un prototype
 - 14.5.5. Prototypes d'interface utilisateur
 - 14.5.6. Évaluation des prototypes
- 14.6. Analyse des besoins
 - 14.6.1. L'Analyse des besoins
 - 14.6.2. Bonnes pratiques en matière d'analyse des besoins
 - 14.6.3. Le dictionnaire de données
 - 14.6.4. Hiérarchisation des besoins
- 14.7. Documentation des besoins
 - 14.7.1. Le document de spécification des exigences
 - 14.7.2. Structure et contenu d'un BSS
 - 14.7.3. Documentation en langage naturel
 - 14.7.4. EARS: *Easy Approach to Requirements Syntax*
 - 14.7.5. Les besoins non fonctionnels
 - 14.7.6. Attributs et modèles sous forme de tableau
 - 14.7.7. Bonnes pratiques de spécification
- 14.8. Validation et négociation des exigences
 - 14.8.1. Validation des besoins
 - 14.8.2. Techniques de validation des besoins
 - 14.8.3. Négociation des besoins
- 14.9. Modélisation et gestion des exigences
 - 14.9.1. La modélisation des besoins
 - 14.9.2. La perspective de l'utilisateur
 - 14.9.3. La perspective des données
 - 14.9.4. La perspective fonctionnelle ou axée sur le flux
 - 14.9.5. La perspective comportementale
 - 14.9.6. La volatilité des besoins
 - 14.9.7. Processus de gestion des exigences
 - 14.9.8. Outils de gestion des exigences
 - 14.9.9. Bonnes pratiques en matière de gestion des exigences

- 14.10. Systèmes critiques et spécification formelle
 - 14.10.1. Systèmes critiques
 - 14.10.2. Spécification axée sur le risque
 - 14.10.3. Spécification formelle

Module 15. Processus de l'Ingénierie du logiciel

- 15.1. Cadre de l'ingénierie du logiciel
 - 15.1.1. Caractéristiques du logiciel
 - 15.1.2. Les principaux processus de l'ingénierie du logiciel
 - 15.1.3. Modèles de processus de développement de logiciels
 - 15.1.4. Cadre de référence normalisé pour le processus de développement logiciel: la norme ISO/CEI 12207
- 15.2. Processus unifié de développement de logiciels
 - 15.2.1. Processus unifié
 - 15.2.2. Dimensions du processus unifié
 - 15.2.3. Processus de développement axé sur les cas d'utilisation
 - 15.2.4. Flux de travail fondamentaux des processus unifiés
- 15.3. Planification dans le contexte du développement logiciel agile
 - 15.3.1. Caractéristiques du développement logiciel agile
 - 15.3.2. Différents horizons de planification dans le développement agile
 - 15.3.3. Cadre de développement agile Scrum et planification des horizons temporels
 - 15.3.4. Histoires d'utilisateurs en tant qu'unité de planification et d'estimation
 - 15.3.5. Techniques courantes de calcul d'une estimation
 - 15.3.6. Échelles d'interprétation des estimations
 - 15.3.7. *Planning Poker*
 - 15.3.8. Types de planification courants: planification des livraisons et planification des itérations
- 15.4. Styles de conception de logiciels distribués et architectures logicielles orientées services
 - 15.4.1. Modèles de communication dans les systèmes logiciels distribués
 - 15.4.2. Couche intermédiaire ou *Middleware*
 - 15.4.3. Modèles d'architecture pour les systèmes distribués
 - 15.4.4. Processus général de conception de services logiciels
 - 15.4.5. Aspects de la conception des services logiciels
 - 15.4.6. Composition des services
 - 15.4.7. Architecture des services Web
 - 15.4.8. Composants d'infrastructure et SOA

- 15.5. Introduction au développement logiciel guidé par le modèle
 - 15.5.1. Le concept de modèle
 - 15.5.2. Développement logiciel guidé par le modèle
 - 15.5.3. MDA cadre de développement guidé par le modèle
 - 15.5.4. Éléments d'un modèle de transformation
- 15.6. Conception d'interfaces utilisateur graphiques
 - 15.6.1. Principes de conception d'interfaces utilisateur
 - 15.6.2. Modèles de conception architecturale pour les systèmes interactifs :
Modèle Vue Contrôleur Vue (MVC)
 - 15.6.3. Expérience utilisateur (UX *User Experience*)
 - 15.6.4. design centrée sur l'utilisateur
 - 15.6.5. Processus d'analyse et de conception d'une interface utilisateur graphique
 - 15.6.6. Utilisabilité des interfaces utilisateurs
 - 15.6.7. Accessibilité des interfaces utilisateurs
- 15.7. Design d'applications Web
 - 15.7.1. Caractéristiques des applications web
 - 15.7.2. Interface utilisateur d'une application web
 - 15.7.3. Conception de la navigation
 - 15.7.4. Protocole d'interaction de base pour les applications Web
 - 15.7.5. Types d'architecture pour les applications web
- 15.8. Stratégies et techniques de test des logiciels et facteurs de qualité des logiciels
 - 15.8.1. Stratégies d'essai
 - 15.8.2. Conception des cas de test
 - 15.8.3. Compromis entre coût et qualité
 - 15.8.4. Modèles de Qualité
 - 15.8.5. Famille de normes ISO/IEC 25000 (*SQuaRE*)
 - 15.8.6. Modèle de qualité du produit (ISO 2501n)
 - 15.8.7. Modèles de qualité des données (ISO 2501n)
 - 15.8.8. Gestion de la qualité des logiciels
- 15.9. Introduction aux métriques de l'Ingénierie du logiciel
 - 15.9.1. Concepts de base: mesures, métriques et indicateurs
 - 15.9.2. Types de métriques en génie logiciel
 - 15.9.3. Le processus de mesure
 - 15.9.4. ISO 250215 Mesures externes et de qualité utilisées
 - 15.9.5. Métriques orientées objet

- 15.10. Maintenance et réingénierie des logiciels
 - 15.10.1. Processus de maintenance
 - 15.10.2. Cadre du processus de maintenance standard ISO/EIEC 115.64
 - 15.10.3. Modèle de processus de réingénierie des logiciels
 - 15.10.4. Ingénierie inverse

Module 16. Intégration des systèmes

- 16.1. Introduction aux systèmes d'information d'entreprise
 - 16.1.1. Le rôle des systèmes d'information
 - 16.1.2. Qu'est-ce qu'un système d'information?
 - 16.1.3. Dimensions des systèmes d'information
 - 16.1.4. Processus d'affaires et systèmes d'information
 - 16.1.5. Le département informatique/ TI
- 16.2. Opportunités et besoins des systèmes d'information dans l'entreprise
 - 16.2.1. Organisations et systèmes d'information
 - 16.2.2. Caractéristiques des organisations
 - 16.2.3. Impact des systèmes d'information sur l'entreprise
 - 16.2.4. Les systèmes d'information pour un avantage concurrentiel
 - 16.2.5. Utilisation de systèmes dans l'administration et la gestion des affaires
- 16.3. Concepts de base des systèmes et technologies de l'information
 - 16.3.1. Données, informations et connaissances
 - 16.3.2. Systèmes et technologies de l'information
 - 16.3.3. Composants technologiques
 - 16.3.4. Classification et types de systèmes d'information
 - 16.3.5. Architectures basées sur les services et les processus d'entreprise
 - 16.3.6. Formes d'intégration des systèmes
- 16.4. Systèmes intégrés de gestion des ressources de l'entreprise
 - 16.4.1. Exigences commerciales
 - 16.4.2. Un système d'information d'entreprise intégré
 - 16.4.3. Acquisition vs. Développement
 - 16.4.4. Mise en œuvre de l'ERP
 - 16.4.5. Implications pour la gestion
 - 16.4.6. Principaux fournisseurs d'ERP

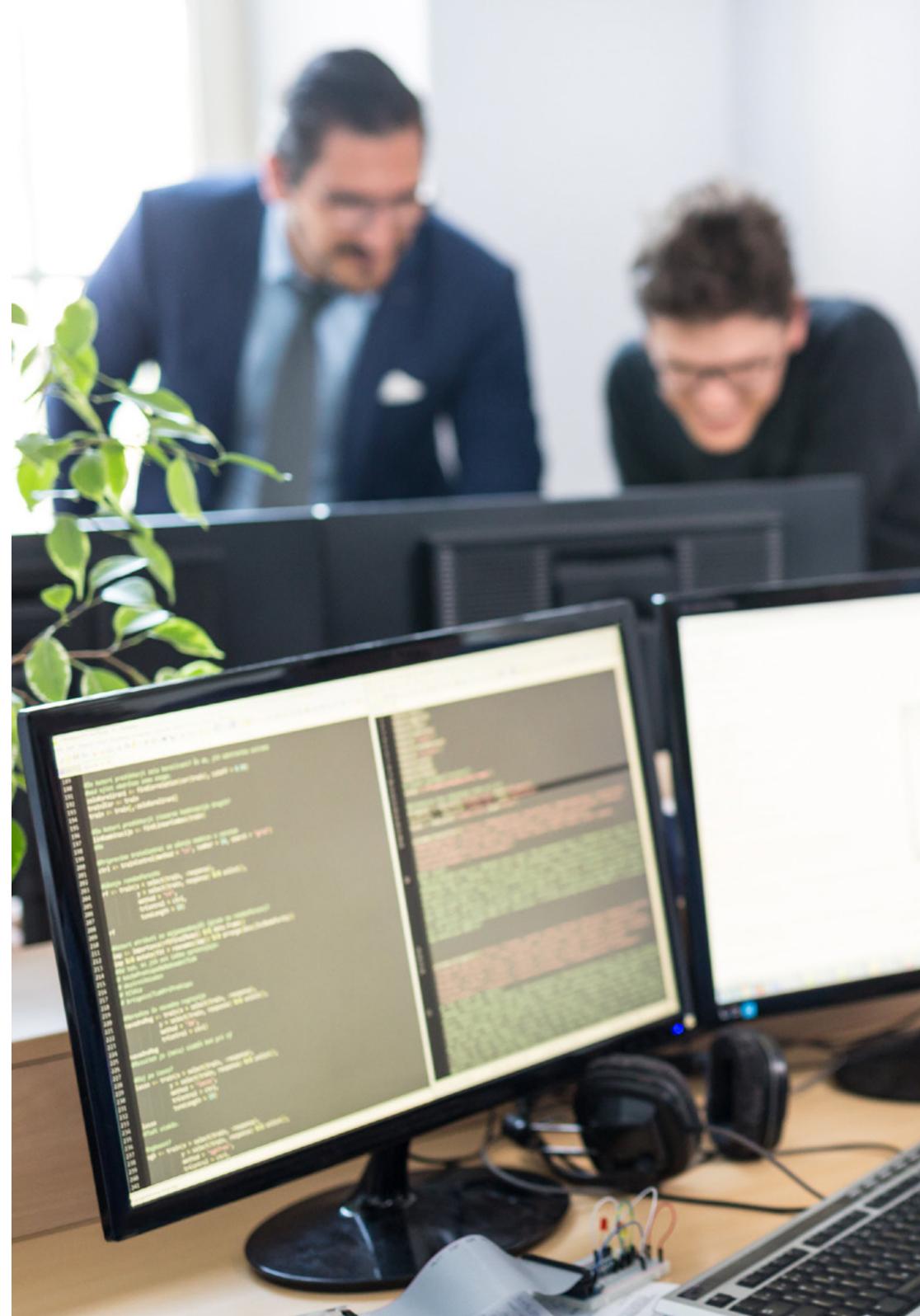
- 16.5. Systèmes d'information sur la gestion de la chaîne d'approvisionnement et de la relation client
 - 16.5.1. Définition de la chaîne d'approvisionnement
 - 16.5.2. Gestion efficace de la chaîne d'approvisionnement
 - 16.5.3. Le rôle des systèmes d'information
 - 16.5.4. Solutions de gestion de la chaîne d'approvisionnement
 - 16.5.5. Gestion de la relation client
 - 16.5.6. Le rôle des systèmes d'information
 - 16.5.7. Mise en œuvre d'un système CRM
 - 16.5.8. Facteurs critiques de succès dans la mise en œuvre de la CRM
 - 16.5.9. CRM, e-CRM et autres tendances
- 16.6. Prise de décision en matière d'investissement dans les TIC et planification des systèmes d'information
 - 16.6.1. Critères de décision en matière d'investissement dans les TIC
 - 16.6.2. Lier le projet au plan de gestion et au plan d'affaires
 - 16.6.3. Implications en termes de gestion
 - 16.6.4. Refonte des processus d'affaires
 - 16.6.5. Décision de la direction sur les méthodologies de mise en œuvre
 - 16.6.6. Nécessité d'une planification des systèmes d'information
 - 16.6.7. Objectifs, participants et calendrier
 - 16.6.8. Structure et développement du plan des systèmes
 - 16.6.9. Suivi et mise à jour
- 16.7. Considérations de sécurité dans l'utilisation des TIC
 - 16.7.1. Analyse des risques
 - 16.7.2. Sécurité des systèmes d'information
 - 16.7.3. Conseils pratiques
- 16.8. Faisabilité de la mise en œuvre des projets TIC et aspects financiers des projets de systèmes d'information
 - 16.8.1. Description et objectifs
 - 16.8.2. Participants au SVE
 - 16.8.3. Techniques et pratiques
 - 16.8.4. Structure des coûts
 - 16.8.5. Projection financière
 - 16.8.6. Budgets

- 16.9. Business Intelligence
 - 16.9.1. Qu'est-ce que la Business Intelligence?
 - 16.9.2. Stratégie et mise en œuvre de la BI
 - 16.9.3. Le présent et l'avenir de la BI
- 16.10. ISO/IEC 12207
 - 16.10.1. Qu'est-ce que "ISO/IEC 12207"?
 - 16.10.2. Analyse des systèmes d'information
 - 16.10.3. Conception de systèmes d'information
 - 16.10.4. Mise en œuvre et acceptation du système d'information

Module 17. Réutilisation du logiciel

- 17.1. Aperçu général de la réutilisation du logiciel
 - 17.1.1. Qu'est-ce que la réutilisation du logiciel?
 - 17.1.2. Avantages et inconvénients de la réutilisation du logiciel
 - 17.1.3. Principales techniques de réutilisation du logiciel
- 17.2. Introduction aux modèles de conception
 - 17.2.1. Qu'est-ce qu'un Motifs à Design?
 - 17.2.2. Catalogue des principaux modèles de conception
 - 17.2.3. Comment utiliser les modèles pour résoudre les problèmes de conception
 - 17.2.4. Comment choisir le meilleur modèle de conception
- 17.3. Création de motifs
 - 17.3.1. Modèles de création
 - 17.3.2. Modèle *Abstract Factory*
 - 17.3.3. Exemple de mise en œuvre du modèle *Abstract Factory*
 - 17.3.4. Modèle *Builder*
 - 17.3.5. Exemple de mise en œuvre du *Builder*
 - 17.3.6. Modèle *Abstract Factory* vs *Builder*
- 17.4. Modèles de création (II)
 - 17.4.1. Modèle *Factory Method*
 - 17.4.2. *Factory Method* vs. *Abstract Factory*
 - 17.4.3. Modèle *Singleton*

- 17.5. Modèles structurels
 - 17.5.1. Modèles structurels
 - 17.5.2. Modèle *Adapter*
 - 17.5.3. Modèle *Bridge*
- 17.6. Modèles structurels (II)
 - 17.6.1. Modèle *Composite*
 - 17.6.2. Modèle *Decorador*
- 17.7. Modèles structurels (III)
 - 17.7.1. Modèle *Facade*
 - 17.7.2. Modèle *Proxy*
- 17.8. Modèles de comportement
 - 17.8.1. Concept de modèles comportementaux
 - 17.8.2. Modèle de comportement: chaîne de responsabilité
 - 17.8.3. Modèle de comportement *Ordre*
- 17.9. Modèles de comportement (II)
 - 17.9.1. Modèle d'interprète ou *Interpreter*
 - 17.9.2. Modèle d'itérateur
 - 17.9.3. Modèle d'observateur
 - 17.9.4. Modèle de stratégie
- 17.10. *Cadres de travail*
 - 17.10.1. Concept de *Framework*
 - 17.10.2. Développement à l'aide de *Frameworks*
 - 17.10.3. Modèle *Model View Controller*
 - 17.10.4. *Framework* pour la conception d'interfaces utilisateur graphiques
 - 17.10.5. *Framework* pour le développement d'applications web
 - 17.10.6. *Frameworks* pour la gestion de la persistance des objets dans les bases de données



Module 18. Services des technologies de l'information

- 18.1. Transformation numérique (I)
 - 18.1.1. L'innovation commerciale
 - 18.1.2. La gestion de la production
 - 18.1.3. La gestion financière
- 18.2. La transformation numérique (II)
 - 18.2.1. Marketing
 - 18.2.2. La gestion des RH
 - 18.2.3. Un système d'information intégré
- 18.3. Études de cas
 - 18.3.1. Présentation de l'entreprise
 - 18.3.2. Méthodologies d'analyse des achats de TI
 - 18.3.3. Détermination des coûts, des avantages et des risques
 - 18.3.4. Évaluation économique de l'investissement
- 18.4. Le Gouvernance et gestion des TIC
 - 18.4.1. Définition de la gouvernance des technologies et systèmes d'information
 - 18.4.2. Différence entre la gouvernance et la gestion des TSI
 - 18.4.3. Cadres Le Gouvernance et gestion des TSI
 - 18.4.4. Les normes et la gouvernance et la gestion des TSI
- 18.5. Le Gouvernance et gestion des TIC
 - 18.5.1. Qu'est-ce qu'une bonne gouvernance d'entreprise?
 - 18.5.2. Historique de la gouvernance des TIC
 - 18.5.3. La Norme ISO/IEC 318.00:2008
 - 18.5.4. Mettre en œuvre une bonne gouvernance des TIC
 - 18.5.5. Gouvernance et meilleures pratiques en matière de TIC
 - 18.5.6. Gouvernance d'entreprise. Vue d'ensemble et tendances
- 18.6. Objectifs de contrôle pour les technologies de l'information et les technologies connexes (COBIT)
 - 18.6.1. Cadre de mise en œuvre
 - 18.6.2. Domaine: Planification et organisation
 - 18.6.3. Domaine: acquisition et mise en œuvre
 - 18.6.4. Domaine: livraison et soutien
 - 18.6.5. Domaine: suivi et évaluation
 - 18.6.6. Mise en œuvre du Guide COBIT
- 18.7. La bibliothèque d'infrastructure des technologies de l'information (ITIL)
 - 18.7.1. Introduction à ITIL
 - 18.7.2. Stratégies du service
 - 18.7.3. Conception du service
 - 18.7.4. Transition du service
 - 18.7.5. Opération du service
 - 18.7.6. Amélioration du service
- 18.8. Le système de gestion des services
 - 18.8.1. Principes de base de la norme UNE-ISO/IEC 20000-1
 - 18.8.2. La structure de la série de normes ISO/IEC 20000
 - 18.8.3. Exigences du système de gestion des services (SMS)
 - 18.8.4. Conception et transition de services nouveaux ou modifiés
 - 18.8.5. Processus de prestation de services
 - 18.8.6. Groupes de processus
- 18.9. Le système de gestion actifs Du logiciel
 - 18.9.1. Justification du besoin
 - 18.9.2. Historique
 - 18.9.3. Présentation de la norme 19770
 - 18.9.4. Mise en œuvre de la gestion
- 18.10. Gestion de la continuité des activités
 - 18.10.1. Plan de continuité des activités
 - 18.10.2. Mise en œuvre du BCM



Un programme complet qui sera fondamental pour votre développement professionnel"

06

Méthodologie

Ce programme de formation offre une manière différente d'apprendre. Notre méthodologie est développée à travers un mode d'apprentissage cyclique: ***le Relearning***.

Ce système d'enseignement est utilisé, par exemple, dans les écoles de médecine les plus prestigieuses du monde et a été considéré comme l'un des plus efficaces par des publications de premier plan telles que le ***New England Journal of Medicine***.



“

Découvrez Relearning, un système qui renonce à l'apprentissage linéaire conventionnel pour vous emmener à travers des systèmes d'enseignement cycliques: une façon d'apprendre qui s'est avérée extrêmement efficace, en particulier dans les matières qui exigent la mémorisation”

Étude de Cas pour mettre en contexte tout le contenu

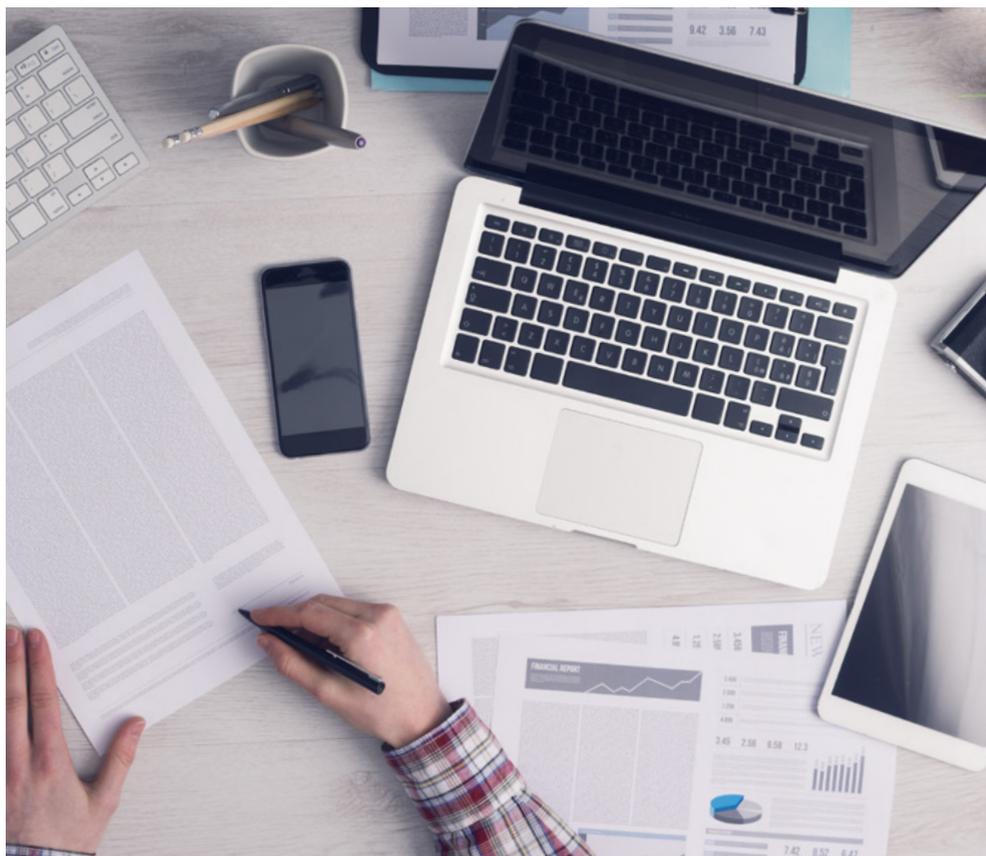
Notre programme offre une méthode révolutionnaire de développement des compétences et des connaissances. Notre objectif est de renforcer les compétences dans un contexte changeant, compétitif et hautement exigeant.

“

Avec TECH, vous pouvez expérimenter une manière d'apprendre qui ébranle les fondations des universités traditionnelles du monde entier”



Vous bénéficierez d'un système d'apprentissage basé sur la répétition, avec un enseignement naturel et progressif sur l'ensemble du cursus.



L'étudiant apprendra, par des activités collaboratives et des cas réels, à résoudre des situations complexes dans des environnements commerciaux réels.

Une méthode d'apprentissage innovante et différente

Cette formation TECH est un programme d'enseignement intensif, créé de toutes pièces, qui propose les défis et les décisions les plus exigeants dans ce domaine, tant au niveau national qu'international. Grâce à cette méthodologie, l'épanouissement personnel et professionnel est stimulé, faisant ainsi un pas décisif vers la réussite. La méthode des cas, technique qui constitue la base de ce contenu, permet de suivre la réalité économique, sociale et professionnelle la plus actuelle.

“ Notre programme vous prépare à relever de nouveaux défis dans des environnements incertains et à réussir votre carrière ”

La méthode des cas est le système d'apprentissage le plus largement utilisé dans les meilleures écoles d'informatique du monde depuis qu'elles existent. Développée en 1912 pour que les étudiants en Droit n'apprennent pas seulement le droit sur la base d'un contenu théorique, la méthode des cas consiste à leur présenter des situations réelles complexes afin qu'ils prennent des décisions éclairées et des jugements de valeur sur la manière de les résoudre. En 1924, elle a été établie comme méthode d'enseignement standard à Harvard.

Dans une situation donnée, que doit faire un professionnel? C'est la question à laquelle nous sommes confrontés dans la méthode des cas, une méthode d'apprentissage orientée vers l'action. Tout au long du programme, les étudiants seront confrontés à de multiples cas réels. Ils devront intégrer toutes leurs connaissances, faire des recherches, argumenter et défendre leurs idées et leurs décisions.

Relearning Methodology

TECH combine efficacement la méthodologie des Études de Cas avec un système d'apprentissage 100% en ligne basé sur la répétition, qui associe différents éléments didactiques dans chaque leçon.

Nous enrichissons l'Étude de Cas avec la meilleure méthode d'enseignement 100% en ligne: le Relearning.

En 2019, nous avons obtenu les meilleurs résultats d'apprentissage de toutes les universités en ligne du monde.

À TECH, vous apprendrez avec une méthodologie de pointe conçue pour former les managers du futur. Cette méthode, à la pointe de la pédagogie mondiale, est appelée Relearning.

Notre université est la seule université autorisée à utiliser cette méthode qui a fait ses preuves. En 2019, nous avons réussi à améliorer les niveaux de satisfaction globale de nos étudiants (qualité de l'enseignement, qualité des supports, structure des cours, objectifs...) par rapport aux indicateurs de la meilleure université en ligne.



Dans notre programme, l'apprentissage n'est pas un processus linéaire, mais se déroule en spirale (apprendre, désapprendre, oublier et réapprendre). Par conséquent, chacun de ces éléments est combiné de manière concentrique. Cette méthodologie a permis de former plus de 650.000 diplômés universitaires avec un succès sans précédent dans des domaines aussi divers que la biochimie, la génétique, la chirurgie, le droit international, les compétences en gestion, les sciences du sport, la philosophie, le droit, l'ingénierie, le journalisme, l'histoire, les marchés financiers et les instruments. Tout cela dans un environnement très exigeant, avec un corps étudiant universitaire au profil socio-économique élevé et dont l'âge moyen est de 43,5 ans.

Le Relearning vous permettra d'apprendre avec moins d'efforts et plus de performance, en vous impliquant davantage dans votre formation, en développant un esprit critique, en défendant des arguments et en contrastant les opinions: une équation directe vers le succès.

À partir des dernières preuves scientifiques dans le domaine des neurosciences, non seulement nous savons comment organiser les informations, les idées, les images et les souvenirs, mais nous savons aussi que le lieu et le contexte dans lesquels nous avons appris quelque chose sont fondamentaux pour notre capacité à nous en souvenir et à le stocker dans l'hippocampe, pour le conserver dans notre mémoire à long terme.

De cette manière, et dans ce que l'on appelle Neurocognitive context-dependent e-learning, les différents éléments de notre programme sont reliés au contexte dans lequel le participant développe sa pratique professionnelle.



Ce programme offre le support matériel pédagogique, soigneusement préparé pour les professionnels:



Support d'étude

Tous les contenus didactiques sont créés par les spécialistes qui enseigneront le cours, spécifiquement pour le cours, afin que le développement didactique soit vraiment spécifique et concret.

Ces contenus sont ensuite appliqués au format audiovisuel, pour créer la méthode de travail TECH en ligne. Tout cela, avec les dernières techniques qui offrent des pièces de haute qualité dans chacun des matériaux qui sont mis à la disposition de l'étudiant.



Cours magistraux

Il existe des preuves scientifiques de l'utilité de l'observation par un tiers expert.

La méthode "Learning from an Expert" renforce les connaissances et la mémoire, et donne confiance dans les futures décisions difficiles.



Pratiques en compétences et aptitudes

Les étudiants réaliseront des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Des activités pratiques et dynamiques pour acquérir et développer les compétences et aptitudes qu'un spécialiste doit développer dans le cadre de la mondialisation dans laquelle nous vivons.



Lectures complémentaires

Articles récents, documents de consensus et directives internationales, entre autres. Dans la bibliothèque virtuelle de TECH, l'étudiant aura accès à tout ce dont il a besoin pour compléter sa formation.





Case studies

Ils réaliseront une sélection des meilleures études de cas choisies spécifiquement pour ce diplôme. Des cas présentés, analysés et tutorés par les meilleurs spécialistes de la scène internationale.



Résumés interactifs

L'équipe TECH présente les contenus de manière attrayante et dynamique dans des pilules multimédia comprenant des audios, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de renforcer les connaissances. Ce système éducatif unique pour la présentation de contenu multimédia a été récompensé par Microsoft en tant que "European Success Story".



Testing & Retesting

Les connaissances de l'étudiant sont périodiquement évaluées et réévaluées tout au long du programme, par le biais d'activités et d'exercices d'évaluation et d'auto-évaluation, afin que l'étudiant puisse vérifier comment il atteint ses objectifs.



07 Diplôme

Le Mastère Avancé en Ingénierie du Logiciel vous garantit, en plus de la formation la plus rigoureuse et la plus actuelle, l'accès à un diplôme universitaire de Mastère Avancé délivré par TECH Université Technologique.



“

*Finalisez cette formation avec succès
et recevez votre Certificat sans avoir
à vous soucier des déplacements ou
des démarches administratives”*

Ce **Mastère Avancé en Ingénierie du Logiciel** contient le programme le plus complet et le plus à jour du marché.

Après avoir réussi l'évaluation, l'étudiant recevra par courrier postal* avec accusé de réception son correspondant diplôme de **Mastère Avancé** délivré par **TECH Université Technologique**.

Le diplôme délivré par **TECH Université Technologique** indiquera la note obtenue lors du Mastère Avancé, et répond aux exigences communément demandées par les bourses d'emploi, les concours et les commissions d'évaluation des carrières professionnelles.

Diplôme: **Mastère Avancé en Ingénierie du Logiciel**

N.º d'heures Officielles: **3.000 h.**



*Si l'étudiant souhaite que son diplôme version papier possède l'Apostille de La Haye, TECH EDUCATION fera les démarches nécessaires pour son obtention moyennant un coût supplémentaire.

future

santé confiance personnes

éducation information tuteurs

garantie accréditation enseignement

institutions technologie apprentissage

communauté engagement

service personnalisé innovation

connaissance présent qualité

en ligne formation

développement institutions

classe virtuelle langues

tech université
technologique

Mastère Avancé

Ingénierie du Logiciel

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Avancé

Ingénierie du Logiciel

