

Mastère Spécialisé

Ingénierie Software Avancée

```
back the deselected mirror modifier object
```

```
jects.active = modifier_ob  
str(modifier_ob)) # modifier ob is the active ob
```



Mastère Spécialisé Ingénierie Software Avancée

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

Accès au site web: www.techtitute.com/fr/informatique/master/master-ingenierie-software-avancee

Sommaire

01

Présentation

page 4

02

Objectifs

page 8

03

Compétences

page 14

04

Structure et contenu

page 18

05

Méthodologie

page 32

06

Diplôme

page 40

01

Présentation

Le développement de logiciels ne serait pas possible sans l'existence de l'Ingénierie Software. Grâce aux progrès technologiques, il est aujourd'hui possible de trouver des systèmes et des structures de plus en plus complexes et spécifiques, conçus sur la base de spécifications de programmation dans le but de répondre aux besoins et aux exigences du marché. Il s'agit d'un processus méthodique et ordonné qui nécessite des connaissances spécialisées en informatique et dans ses outils, ce que le diplômé pourra acquérir au cours de ce programme très complet. Grâce à une expérience pluridisciplinaire, vous serez immergé dans les clés des processus informatiques et de l'intégration des systèmes et vous acquerrez les compétences nécessaires pour mener à bien des projets au plus haut niveau en fonction des exigences des *stakeholders*. Tout cela en seulement 12 mois de formation 100% en ligne.





“

Tout cela en seulement 12 mois de formation 100% en ligne. Si vous recherchez un programme qui vous guidera pour lancer votre propre projet de logiciel en partant de zéro, ce Mastère Spécialisé est parfait pour vous. Qu'attendez-vous pour vous inscrire?"

Depuis plus de six décennies, l'Ingénierie Software est à la pointe de la révolution technologique grâce au développement de programmes et d'applications de plus en plus complexes et spécialisés. C'est un domaine qui a servi de support à de nombreux autres pour avancer vers le progrès et dont l'application est extrapolée à pratiquement toutes les spécialités existantes: médecine, agriculture, enseignement, administration, industrie, etc. Quelle que soit la façon dont on l'envisage, même le processus informatique le plus simple, comme l'envoi d'un courrier électronique ou l'utilisation d'une messagerie instantanée, un événement franchement quotidien de nos jours, a nécessité une conception et une programmation exhaustives pour atteindre son objectif: satisfaire les besoins des êtres humains.

Le large éventail de possibilités qui découlent de cette science et de ses multiples applications en fait l'une des plus demandées sur le marché du travail, non seulement pour créer de nouveaux projets, mais aussi pour superviser, maintenir et mettre à jour les projets existants. Pour cette raison, et suivant la maxime de TECH d'offrir à tous ses diplômés la possibilité de se spécialiser dans ce domaine, l'université a décidé de lancer ce programme très complet en Ingénierie Software Avancée.

Il s'agit d'une expérience académique qui comprend 1.500 heures du meilleur contenu théorique-pratique et complémentaire, couvrant l'ensemble du domaine, de l'origine à la conception, la création et la gestion de systèmes d'information innovants et modernes. Tout au long des 12 mois de développement du programme, l'informaticien pourra s'immerger dans les tenants et aboutissants de cette spécialité: ses exigences techniques et structurelles, les clés de la création d'architectures sécurisées, l'intégration de services basés sur les TIC, la gestion des *stakeholders* et de leur périmètre, le développement d'un projet de la base à son lancement et bien plus encore !

Tout cela de manière 100% en ligne, grâce à quoi le diplômé pourra accéder au cours de ce Mastère Spécialisé quand et où il le souhaite, sans cours en classe ni horaires restreints. En outre, vous pouvez accéder au Campus Virtuel depuis n'importe quel appareil doté d'une connexion internet, qu'il s'agisse d'un PC, d'une tablette ou d'un téléphone portable. Il s'agit donc d'une occasion unique de se spécialiser en Génie Logiciel grâce à un diplôme adapté à vos besoins académiques et aux exigences les plus élevées du secteur informatique actuel.

Ce **Mastère Spécialisé en Ingénierie Software Avancée** contient le programme académique le plus complet et le plus actuel du marché. Les principales caractéristiques sont les suivantes:

- ◆ Le développement d'études de cas présentées par des experts en Ingénierie informatique
- ◆ Les contenus graphiques, schématiques et éminemment pratiques avec lesquels ils sont conçus fournissent des informations sanitaires essentielles à la pratique professionnelle
- ◆ Les exercices pratiques où le processus d'auto-évaluation peut être réalisé pour améliorer l'apprentissage
- ◆ Il met l'accent sur les méthodologies innovantes
- ◆ Cours théoriques, questions à l'expert, forums de discussion sur des sujets controversés et travail de réflexion individuel
- ◆ La possibilité d'accéder aux contenus depuis n'importe quel appareil fixe ou portable doté d'une connexion internet



Un diplôme qui approfondit les bases de l'Ingénierie Software: depuis les origines, jusqu'aux processus informatiques qui se déroulent aujourd'hui"

“

Vous aurez accès à un guide pratique qui reprend les principes de l'Ingénierie Software, du processus initial à la construction et au déploiement”

Le programme comprend, dans son corps enseignant, des professionnels du secteur qui apportent à cette formation l'expérience de leur travail, ainsi que des spécialistes reconnus de grandes sociétés et d'universités prestigieuses.

Grâce à son contenu multimédia développé avec les dernières technologies éducatives, les spécialistes bénéficieront d'un apprentissage situé et contextuel, ainsi, ils se formeront dans un environnement simulé qui leur permettra d'apprendre en immersion et de s'entraîner dans des situations réelles.

La conception de ce programme est axée sur l'Apprentissage par les Problèmes, grâce auquel le professionnel doit essayer de résoudre les différentes situations de la pratique professionnelle qui se présentent tout au long du Mastère Spécialisé. Pour ce faire, l'étudiant sera assisté d'un innovant système de vidéos interactives, créé par des experts reconnus.

Un programme conçu pour vous permettre d'appliquer les stratégies les plus innovantes en matière de modélisation des besoins dès le départ.

Souhaitez-vous acquérir les compétences nécessaires pour concevoir des architectures complexes et alternatives à travers le flux de données? Inscrivez-vous à ce Mastère Spécialisé vous l'obtiendrez en 12 mois seulement.



02 Objectifs

L'Ingénierie Software est devenue l'un des principaux outils du développement technologique actuel. Sans le travail de millions de professionnels de l'informatique, il n'aurait pas été possible d'atteindre le très haut niveau d'innovation numérique que l'on connaît aujourd'hui. Pour cette raison, l'objectif de ce Mastère Spécialisé est d'apporter aux diplômés intéressés par ce domaine toutes les informations dont ils ont besoin pour les connaître en détail et pouvoir développer leurs propres programmes et applications de manière réussie, efficace et sur la base de la plus haute qualité et rigoureuse.





“

Si parmi vos objectifs avec le cours de ce Mastère Spécialisé est de maîtriser Scrum et d'autres techniques de méthodologies agiles, TECH mettra à votre disposition tout ce dont vous avez besoin pour les surmonter de manière garantie"



Objectifs généraux

- ◆ Former scientifiquement et technologiquement, ainsi que préparer à la pratique professionnelle du génie Logiciel, le tout avec une expérience académique transversale et polyvalente adaptée aux nouvelles technologies et aux innovations dans ce domaine
- ◆ Obtenir des connaissances approfondies dans le domaine du génie Logiciel, mais aussi dans le domaine de l'informatique et de la structure des ordinateurs, y compris les bases mathématiques, statistiques et physiques essentielles en ingénierie

“

Vous travaillerez intensément sur les tests de logiciels via TDD, ATDD et BDD, pour que vous puissiez construire des structures informatiques de la plus haute qualité”





Objectifs spécifiques

Module 1. Ingénierie Software

- ◆ Poser les bases du génie Logiciel et de la modélisation, en apprenant les principaux processus et concepts
- ◆ Comprendre le processus Logiciel et les différents modèles pour son développement, y compris les technologies agiles
- ◆ Connaître les principales normes relatives à la qualité des Logiciels et à la gestion de projet

Module 2. Ingénierie Software Avancée

- ◆ Découvrez en profondeur les différentes méthodologies agiles utilisées dans l'ingénierie logicielle
- ◆ Apprendre à développer en utilisant les techniques de Scrum, de programmation extrême et de développement logiciel basé sur la réutilisation
- ◆ Comprendre les concepts et les processus de conception logicielle, en apprenant également sur la conception de l'architecture et sur la conception au niveau des composants et basée sur des modèles
- ◆ Introduire le concept de DevOps et quelles sont ses principales pratiques
- ◆ Apprendre à tester le logiciel, avec des méthodologies telles que Test Driven Development, Acceptance Test Driven Development, Behavior Driven Development, BDD et Cucumber
- ◆ Comprendre les différents modèles d'architecture de système et de conception logicielle, ainsi que l'architecture d'applications cloud

Module 3. Ingénierie des exigences

- ◆ Comprendre l'ingénierie des exigences, leur développement, leur élaboration, leur négociation et leur validation
- ◆ Apprenez la modélisation des exigences et les différents éléments tels que les scénarios, les informations, les classes d'analyse, le flux, le comportement et les modèles
- ◆ Comprendre l'importance de l'ingénierie des besoins dans le processus de développement de logiciels
- ◆ Apprendre à réaliser des analyses et à documenter correctement les exigences
- ◆ Approfondir les sources d'exigences et les techniques d'appel d'offres, car elles sont un élément essentiel du processus
- ◆ Comprendre les processus de validation et de négociation des exigences, ainsi que la modélisation et la gestion des exigences
- ◆ Acquérir les connaissances nécessaires à la gestion des systèmes critiques et à la spécification formelle des exigences

Module 4. Processus d'Ingénierie Software

- ◆ Approfondir l'amélioration du processus de développement de logiciels et de la qualité des logiciels en utilisant les normes ISO/IEC
- ◆ Comprendre et appliquer le prototypage comme partie essentielle du processus de développement
- ◆ Connaître le cadre de référence de l'ingénierie logicielle et la norme ISO/IEC 12207
- ◆ Apprendre les caractéristiques du processus unifié de développement de logiciels et de planification dans le contexte du développement de logiciels agiles
- ◆ Connaître les différents styles de conception de logiciels distribués et les architectures logicielles orientées services
- ◆ Apprendre les concepts essentiels dans la conception d'interfaces graphiques utilisateur
- ◆ Comprendre les bases du développement d'applications web

Module 5. Qualité et audit des systèmes d'information

- ◆ Approfondir les stratégies et techniques de test des logiciels, les facteurs de qualité des logiciels et les différentes mesures utilisées
- ◆ Acquérir les connaissances essentielles des systèmes de gestion de la sécurité informatique
- ◆ Introduire les concepts de propriété intellectuelle dans les systèmes de gestion de l'information
- ◆ Préparer les étudiants à créer des plans de continuité d'activité et de reprise après sinistre
- ◆ Apprendre à planifier la gestion de la sécurité et à gérer les principaux mécanismes de protection des ressources et des informations
- ◆ Connaître les différents types d'audits et connaître le processus d'audit informatique

Module 6. Intégration des systèmes

- ◆ Acquérir les concepts essentiels liés aux systèmes d'information dans l'entreprise et identifier les opportunités et les besoins des systèmes d'information dans l'entreprise
- ◆ Connaître les bases du *Business Intelligence*, ses stratégies et sa mise en oeuvre, ainsi que le présent et l'avenir du BI
- ◆ Comprendre le fonctionnement des systèmes de gestion intégrée des ressources de l'entreprise
- ◆ Comprendre la transformation numérique du point de vue de l'innovation commerciale, de la gestion financière et de la production, du marketing et de la gestion des ressources humaines

Module 7. Réutilisation de software

- ◆ Présentation de la stratégie de réutilisation software
- ◆ Apprendre les différents schémas liés à la réutilisation des logiciels, qu'ils soient de conception, de création, de structure ou de comportement
- ◆ Introduire le concept de *framework*, et en connaître les principaux types, tels que ceux destinés à la conception d'interfaces graphiques utilisateur, au développement d'applications Web et à la gestion de la persistance d'objets dans les bases de données
- ◆ Comprendre le fonctionnement du modèle largement utilisé actuellement de Model Vista Controller (MVC)

Module 8. Services informatiques

- ◆ Formation à la prise de décision d'investissement dans les TIC et à la planification des systèmes d'information
- ◆ Connaître les objectifs de contrôle pour les technologies et les informations connexes (COBIT)
- ◆ Apprendre le fonctionnement de la bibliothèque d'infrastructure des technologies de l'information (ITIL), les stratégies, la conception des services, les transitions et les opérations
- ◆ Approfondir le système de gestion des services, en connaissant les principes de base de UNE-ISO/IEC 20000-1, la structure de la série de normes ISO/IEC 20000 et les exigences du Système de Gestion des Services (SGS)
- ◆ Comprendre le fonctionnement des systèmes et des technologies de l'information, leurs composants, leurs classifications, leurs architectures et leurs modalités d'intégration
- ◆ Apprendre la norme ISO/IEC 12207, l'analyse, la conception, la mise en oeuvre et l'acceptation des systèmes d'information

Module 9. Sécurité des systèmes d'information

- ◆ Apprendre à élaborer un calendrier de gestion du temps, de développement budgétaire et de gestion des risques
- ◆ Analyser la nature des attaques réseau et les différents types d'architectures de sécurité
- ◆ Comprendre les différentes techniques de protection des systèmes et de développement de code sécurisé
- ◆ Connaître les composants essentiels des *botnets* et du spam, ainsi que des logiciels malveillants et du code malveillant
- ◆ Jeter les bases de l'analyse médico-légale dans le monde du logiciel et de l'audit informatique
- ◆ Obtenir une vue d'ensemble de la sécurité, de la cryptographie et de la cryptanalyse classique
- ◆ Comprendre les bases de la cryptographie symétrique et de la cryptographie asymétrique, ainsi que leurs principaux algorithmes

Module 10. Gestion de projets

- ◆ 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
- ◆ Comprendre le fonctionnement des processus d'approvisionnement, d'exécution, de surveillance, de contrôle et de clôture d'un projet
- ◆ Acquérir les connaissances essentielles relatives à la responsabilité professionnelle découlant de la gestion de projets
- ◆ Connaître les concepts fondamentaux de la gestion de projet et le cycle de vie de la gestion de projet
- ◆ Comprendre les différentes étapes de la gestion de projet telles que le démarrage, la planification, la gestion des *stakeholders* et la portée

03

Compétences

Parmi les caractéristiques les plus significatives de tous les diplômes offerts par TECH, il y a la possibilité pour le diplômé d'affiner ses compétences à travers le cours. Cela est possible grâce à l'acquisition d'une connaissance spécialisée en la matière, dans ce cas sur l'Ingénierie Software. Cependant, la clé pour y parvenir de manière garantie réside dans la résolution de cas d'utilisation basés sur des situations du contexte actuel du paysage informatique, grâce auxquels l'élève peut mettre en pratique ses compétences, appliquer les stratégies indiquées dans le programme et élargir ses compétences dans la conception, l'analyse, la gestion et le lancement d'applications et de programmes.



“

Le meilleur programme du marché académique actuel pour implémenter dans votre pratique informatique les techniques de prototypage les plus innovantes et efficaces de l'environnement logiciel"

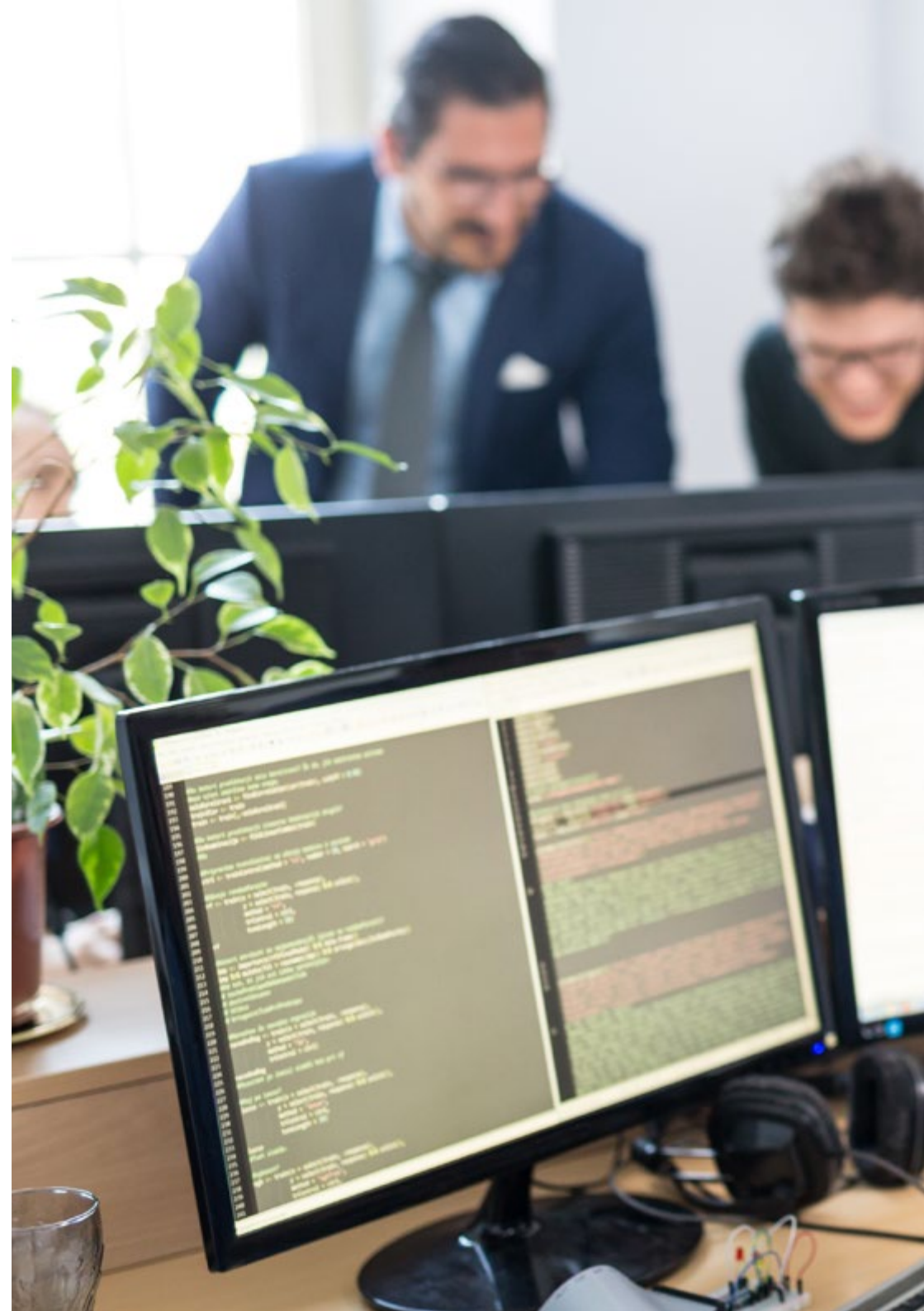


Compétences générales

- ◆ Répondre aux besoins actuels dans le domaine de l'Ingénierie des Logiciels Avancée
- ◆ Avoir la maîtrise des différents systèmes de travail dans l'Ingénierie des Logiciels Avancée
- ◆ Décrire et exploiter les logiciels libres et les connaissances ouvertes existant sur le réseau

“

Grâce à la connaissance des stratégies et des techniques de test des logiciels, vous pourrez travailler sur l'amélioration des projets d'autres personnes pour les rendre plus intuitifs et spécialisés”





Compétences spécifiques

- ◆ Découvrir en profondeur toutes les facettes de l'interaction homme-ordinateur et la façon dont les développements informatiques sont impliqués
- ◆ Être soluble dans l'utilisation des bases de données
- ◆ Développer différents types d'applications en réseau
- ◆ Travailler comme ingénieur logiciel
- ◆ Contrôler l'utilisation des bases de données avancées
- ◆ Planification avancée
- ◆ Savoir comment le logiciel est réutilisé
- ◆ Création d'applications et d'interfaces réseau

04

Structure et contenu

Le programme a été conçu sur la base de l'efficacité éducative, en sélectionnant soigneusement les contenus pour offrir un cours complet, qui inclut tous les domaines d'étude essentiels pour atteindre une réelle connaissance du sujet. Avec les dernières mises à jour et aspects du secteur.



“

Tous les sujets et domaines de connaissances ont été rassemblés dans un programme complet et d'une actualité absolue, pour amener l'élève au plus haut niveau tant au niveau théorique que pratique"

Module 1. Ingénierie Software

- 1.1. Introduction au Génie Logiciel et à la modélisation
 - 1.1.1. La nature des logiciels
 - 1.1.2. La nature unique des webApps
 - 1.1.3. Ingénierie Software
 - 1.1.4. Le processus logiciel
 - 1.1.5. La pratique du Génie Logiciel
 - 1.1.6. Mythes sur les logiciels
 - 1.1.7. Comment tout commence
 - 1.1.8. Concepts orientés objet
 - 1.1.9. Introduction à UML
- 1.2. Le processus logiciel
 - 1.2.1. Un modèle général de processus
 - 1.2.2. Modèles de processus prescriptifs
 - 1.2.3. Modèles de processus spécialisés
 - 1.2.4. Le processus unifié
 - 1.2.5. Modèles de processus personnels et d'équipe
 - 1.2.6. Qu'est-ce que l'agilité?
 - 1.2.7. Qu'est-ce qu'un processus agile?
 - 1.2.8. Scrum
 - 1.2.9. Boîte à outils du processus Agile
- 1.3. Principes guidant la pratique du Génie Logiciel
 - 1.3.1. Principes guidant le processus
 - 1.3.2. Principes guidant la pratique
 - 1.3.3. Principes de la communication
 - 1.3.4. Principes de planification
 - 1.3.5. Principes de modélisation
 - 1.3.6. Principes de construction
 - 1.3.7. Principes de déploiement
- 1.4. Comprendre les besoins
 - 1.4.1. Ingénierie des exigences
 - 1.4.2. Établir la base
 - 1.4.3. Détermination des besoins
 - 1.4.4. Développer des cas d'utilisation
 - 1.4.5. Élaboration du modèle d'exigences
 - 1.4.6. Négociation des exigences
 - 1.4.7. Validation des exigences
- 1.5. Modélisation des exigences I: scénarios, classes d'information et d'analyse
 - 1.5.1. Analyse des besoins
 - 1.5.2. Modélisation basée sur des scénarios
 - 1.5.3. Modèles UML fournissant le cas d'utilisation
 - 1.5.4. Concepts de modélisation des données
 - 1.5.5. Modélisation basée sur les classes
 - 1.5.6. Diagrammes de classes
- 1.6. Modélisation des exigences II: flux, comportement et modèles
 - 1.6.1. Stratégies de modélisation des exigences
 - 1.6.2. Modélisation orientée flux
 - 1.6.3. Diagrammes d'état
 - 1.6.4. Création d'un modèle comportemental
 - 1.6.5. Diagrammes de séquence
 - 1.6.6. Diagrammes de communication
 - 1.6.7. Modèles pour la modélisation des exigences
- 1.7. Concepts de design
 - 1.7.1. Conception dans le contexte de l'Ingénierie Logicielle
 - 1.7.2. Le processus de conception
 - 1.7.3. Concepts de design
 - 1.7.4. Concepts de conception orientée objet
 - 1.7.5. Le modèle de conception

- 1.8. Architecture de conception
 - 1.8.1. Architecture logicielle
 - 1.8.2. Genres architecturaux
 - 1.8.3. Styles architecturaux
 - 1.8.4. Conception architecturale
 - 1.8.5. Évolution des conceptions alternatives de l'architecture
 - 1.8.6. Cartographie de l'architecture à l'aide du flux de données
- 1.9. Conception au niveau des composants et des modèles
 - 1.9.1. Qu'est-ce qu'un composant?
 - 1.9.2. Conception de composants basée sur les classes
 - 1.9.3. Réalisation de la conception au niveau des composants
 - 1.9.4. Conception traditionnelle des composants
 - 1.9.5. Développement basé sur les composants
 - 1.9.6. Modèles de conception
 - 1.9.7. Conception de logiciels basée sur des modèles
 - 1.9.8. Modèles architecturaux
 - 1.9.9. Patrons de conception au niveau des composants
 - 1.9.10. Modèles de conception d'interface utilisateur
- 1.10. Qualité des logiciels et gestion de projet
 - 1.10.1. Qualité
 - 1.10.2. Qualité des logiciels
 - 1.10.3. Le dilemme de la qualité des logiciels
 - 1.10.4. Atteindre la qualité des logiciels
 - 1.10.5. Assurance qualité des logiciels
 - 1.10.6. Le spectre de la gestion
 - 1.10.7. Personnel
 - 1.10.8. Le produit
 - 1.10.9. Le processus
 - 1.10.10. Le projet
 - 1.10.11. Principes et pratiques

Module 2. Ingénierie Software Avancée

- 2.1. Introduction aux méthodologies Agiles
 - 2.1.1. Modèles de processus et méthodologies
 - 2.1.2. Agilité et processus agiles
 - 2.1.3. Manifeste agile
 - 2.1.4. Quelques méthodologies agiles
 - 2.1.5. Agile vs. Traditionnel
- 2.2. Scrum
 - 2.2.1. Origines et philosophie de Scrum
 - 2.2.2. Valeurs Scrum
 - 2.2.3. Flux de processus Scrum
 - 2.2.4. Les rôles Scrum
 - 2.2.5. Les artefacts de Scrum
 - 2.2.6. Les événements Scrum
 - 2.2.7. Histoires d'utilisateur
 - 2.2.8. Extensions Scrum
 - 2.2.9. Estimations rapides
 - 2.2.10. Escalade de Scrum
- 2.3. Programmation extrême
 - 2.3.1. Justification et présentation de XP
 - 2.3.2. Le cycle de vie au XP
 - 2.3.3. Les cinq valeurs de base
 - 2.3.4. Les douze pratiques de base de XP
 - 2.3.5. Rôles des participants
 - 2.3.6. XP Industriel
 - 2.3.7. Évaluation critique de XP
- 2.4. Développement de logiciels basés sur la réutilisation
 - 2.4.1. Réutilisation de software
 - 2.4.2. Niveaux de réutilisation du code
 - 2.4.3. Techniques spécifiques de réutilisation
 - 2.4.4. Développement basé sur les composants
 - 2.4.5. Avantages et problèmes de réutilisation
 - 2.4.6. Planification de la réutilisation

- 2.5. Modèles d'architecture système et de conception logicielle
 - 2.5.1. Conception architecturale
 - 2.5.2. Modèles architecturaux généraux
 - 2.5.3. Architectures tolérantes aux pannes
 - 2.5.4. Architectures de systèmes distribués
 - 2.5.5. Modèles de conception
 - 2.5.6. Modèles Gamma
 - 2.5.7. Modèles de conception d'interaction
- 2.6. Architecture d'application cloud
 - 2.6.1. Principes fondamentaux de Cloud Computing
 - 2.6.2. Qualité d'application cloud
 - 2.6.3. Styles d'architecture
 - 2.6.4. Modèles de conception
- 2.7. Tests logiciels: TDD, ATDD et BDD
 - 2.7.1. Vérification et validation du logiciel
 - 2.7.2. Tests logiciels
 - 2.7.3. Test Driven Development (TDD)
 - 2.7.4. Acceptance Test Driven Development (ATDD)
 - 2.7.5. Behavior Driven Development (BDD)
 - 2.7.6. BDD et Cucumber
- 2.8. L'amélioration du processus logiciel
 - 2.8.1. L'amélioration du processus logiciel
 - 2.8.2. Le processus d'amélioration de processus
 - 2.8.3. Modèles de maturité
 - 2.8.4. Le modèle CMMI
 - 2.8.5. CMMI V2.0
 - 2.8.6. CMMI et Agile

- 2.9. Qualité des systèmes informatiques: SQuaRE
 - 2.9.1. Qualité des logiciels
 - 2.9.2. Modèles de qualité des produit logiciels
 - 2.9.3. Famille ISO/IEC 25000
 - 2.9.4. ISO/IEC 25010: modèle et caractéristiques de qualité
 - 2.9.5. ISO/IEC 25012: qualité des données
 - 2.9.6. ISO/IEC 25020: mesure de la qualité du logiciel
 - 2.9.7. ISO/IEC 25022, 25023 et 25024: mesures de qualité des logiciels et des données
 - 2.9.8. ISO/IEC 25040: évaluation du logiciel
 - 2.9.9. Processus de certification
- 2.10. Présentation de DevOps
 - 2.10.1. Concept DevOps
 - 2.10.2. Principales pratiques

Module 3. Ingénierie des exigences

- 3.1. Introduction à l'Ingénierie des Exigences
 - 3.1.1. L'importance des exigences
 - 3.1.2. Concept d'exigence
 - 3.1.3. Dimensions des exigences
 - 3.1.4. Niveaux et types d'exigences
 - 3.1.5. Caractéristiques des exigences
 - 3.1.6. Ingénierie des exigences
 - 3.1.7. Le processus d'Ingénierie des Exigences
 - 3.1.8. *Frameworks* pour l'ingénierie des exigences
 - 3.1.9. Meilleures pratiques en matière d'ingénierie des exigences
 - 3.1.10. L'analyste d'affaires
- 3.2. Les sources des besoins
 - 3.2.1. Le réseau des exigences
 - 3.2.2. Les *stakeholders*
 - 3.2.3. Les exigences commerciales
 - 3.2.4. Document sur la vision et le champ d'application

- 3.3. Techniques d'élicitation des besoins
 - 3.3.1. Détermination des besoins
 - 3.3.2. Problèmes liés à la détermination des besoins
 - 3.3.3. Contextes de découverte
 - 3.3.4. Interviews
 - 3.3.5. Observation et "apprentissage"
 - 3.3.6. Ethnographie
 - 3.3.7. Workshops
 - 3.3.8. *Focus groups*
 - 3.3.9. Questionnaires
 - 3.3.10. *Brainstorming* et techniques de création
 - 3.3.11. Médias du groupe
 - 3.3.12. Analyse des interfaces du système
 - 3.3.13. Analyse de documents et "archéologie"
 - 3.3.14. Cas d'utilisation et scénario
 - 3.3.15. Prototypes
 - 3.3.16. Ingénierie inverse
 - 3.3.17. Réutilisation des exigences
 - 3.3.18. Bonnes pratiques en matière d'appels d'offres
- 3.4. Besoins des utilisateurs
 - 3.4.1. Personnes
 - 3.4.2. Cas d'utilisation et histoires d'utilisateurs
 - 3.4.3. Scénario
 - 3.4.5. Types de scénarios
 - 3.4.6. Comment découvrir des scénarios
- 3.5. Techniques de prototypage
 - 3.5.1. Prototypage
 - 3.5.2. Prototypes en fonction de leur portée
 - 3.5.3. Prototypes en fonction de leur temporalité
 - 3.5.4. La fidélité d'un prototype
 - 3.5.5. Prototypes d'interface utilisateur
 - 3.5.6. Évaluation des prototypes
- 3.6. Analyse des besoins
 - 3.6.1. Analyse des besoins
 - 3.6.2. Bonnes pratiques en matière d'analyse des exigences
 - 3.6.3. Dictionnaire de données
 - 3.6.4. Hiérarchisation des besoins
- 3.7. Documentation des exigences
 - 3.7.1. Le document de spécification des exigences
 - 3.7.2. Structure et contenu d'un BSS
 - 3.7.3. Documentation en langage naturel
 - 3.7.4. EARS: Easy Approach to Requirements Syntax
 - 3.7.5. Exigences non fonctionnelles
 - 3.7.6. Attributs et modèles de table
 - 3.7.7. Bonne pratique de spécification
- 3.8. Validation et négociation des exigences
 - 3.8.1. Validation des exigences
 - 3.8.2. Techniques de validation des exigences
 - 3.8.3. Négociation des exigences
- 3.9. Modélisation et gestion des exigences
 - 3.9.1. Modélisation des besoins
 - 3.9.2. Le point de vue de l'utilisateur
 - 3.9.3. Perspective des données
 - 3.9.4. La perspective fonctionnelle ou axée sur le flux
 - 3.9.5. La perspective comportementale
 - 3.9.6. Volatilité des exigences
 - 3.9.7. Processus de gestion des exigences
 - 3.9.8. Outils de gestion des exigences
 - 3.9.9. Bonnes pratiques en matière de gestion des exigences
- 3.10. Systèmes critiques et spécification formelle
 - 3.10.1. Systèmes critiques
 - 3.10.2. Spécification axée sur le risque
 - 3.10.3. Spécification formelle

Module 4. Processus d'Ingénierie Software

- 4.1. Cadre de l'Ingénierie logicielle
 - 4.1.1. Caractéristiques des logiciels
 - 4.1.2. Les principaux processus de l'Ingénierie Software
 - 4.1.3. Modèles de processus de développement de logiciels
 - 4.1.4. Cadre de référence normalisé pour le processus de développement logiciel: la norme ISO/CEI 12207
- 4.2. Processus unifié de développement de logiciels
 - 4.2.1. Processus Unifié
 - 4.2.2. Les dimensions du Processus Unifié
 - 4.2.3. Processus de développement basé sur les cas d'utilisation
 - 4.2.4. Flux de travail fondamentaux du Processus Unifié
- 4.3. La planification dans le contexte du développement logiciel agile
 - 4.3.1. Caractéristiques du développement logiciel agile
 - 4.3.2. Différents horizons de planification dans le développement agile
 - 4.3.3. Cadre de développement agile Scrum et planification des horizons temporels
 - 4.3.4. User stories comme unité de planification et d'estimation
 - 4.3.5. Techniques courantes de calcul d'une estimation
 - 4.3.6. Échelles d'interprétation des estimations
 - 4.3.7. *Planning poker*
 - 4.3.8. Types courants de planification: planification des livraisons et planification des itérations
- 4.4. Styles de conception de logiciels distribués et architectures logicielles orientées services
 - 4.4.1. Modèles de communication dans les systèmes logiciels distribués
 - 4.4.2. Couche intermédiaire
 - 4.4.3. Modèles architecturaux pour les systèmes distribués
 - 4.4.4. Processus général de conception de services logiciels
 - 4.4.5. Aspects de la conception de services logiciels
 - 4.4.6. Composition des services
 - 4.4.7. Architecture des services web
 - 4.4.8. Composants d'infrastructure et SOA
- 4.5. Introduction au développement logiciel guidé par le modèle
 - 4.5.1. Le concept de modèle
 - 4.5.2. Développement logiciel guidé par le modèle
 - 4.5.3. MDA cadre de développement guidé par le modèle
 - 4.5.4. Éléments d'un modèle de transformation
- 4.6. Conception d'interfaces utilisateur graphiques
 - 4.6.1. Principes de conception d'interfaces utilisateur
 - 4.6.2. Modèles de conception architecturale pour les systèmes interactifs: Modèle Vue Contrôleur Vue (MVC)
 - 4.6.3. Expérience utilisateur (UX User Experience)
 - 4.6.4. Conception centrée sur l'utilisateur
 - 4.6.5. Processus d'analyse et de conception d'une interface utilisateur graphique
 - 4.6.6. Utilisabilité d'interfaces utilisateurs
 - 4.6.7. Accessibilité des interfaces utilisateurs
- 4.7. Conception d'applications web
 - 4.7.1. Caractéristiques des applications web
 - 4.7.2. Interface utilisateur d'une application web
 - 4.7.3. Conception de la navigation
 - 4.7.4. Protocole d'interaction de base pour les applications web
 - 4.7.5. Styles d'architecture pour les applications web
- 4.8. Stratégies et techniques de test des logiciels et facteurs de qualité des logiciels
 - 4.8.1. Stratégies d'essai
 - 4.8.2. Designs de cas d'essai
 - 4.8.3. Rapport qualité-prix
 - 4.8.4. Modèles de qualité
 - 4.8.5. Famille de normes ISO/IEC 25000 (SQuaRE)
 - 4.8.6. Modèle de qualité du produit (ISO 2501n)
 - 4.8.7. Modèles de qualité des données (ISO 2501n)
 - 4.8.8. Gestion de la qualité des logiciels

- 4.9. Introduction aux métriques de l'ingénierie logicielle
 - 4.9.1. Concepts de base: mesures, métriques et indicateurs
 - 4.9.2. Types de métriques en génie logiciel
 - 4.9.3. Le processus de mesure
 - 4.9.4. ISO 25024. Mesures externes et de qualité utilisées
 - 4.9.5. Métriques orientées objet
- 4.10. Maintenance et réingénierie des logiciels
 - 4.10.1. Processus de maintenance
 - 4.10.2. Cadre du processus de maintenance standard. ISO/EIEC 14764
 - 4.10.3. Le modèle de processus d' réingénierie des de software
 - 4.10.4. Ingénierie inverse

Module 5. Qualité et audit des systèmes d'information

- 5.1. Introduction aux Systèmes de Gestion de la Sécurité de l'Information
 - 5.1.1. Principes fondamentaux du SMSI
 - 5.1.2. Règles d'or du SMSI
 - 5.1.3. Rôle de l'audit informatique dans les SGSI
- 5.2. La planification dans la gestion de la sécurité
 - 5.2.1. Concepts liés à la gestion de la sécurité
 - 5.2.2. Classification de l'information: objectifs, concepts et rôles
 - 5.2.3. Mise en œuvre des politiques de sécurité: politiques, normes et procédures de sécurité
 - 5.2.4. Gestion des risques: principes et analyse des risques liés aux actifs informationnels
- 5.3. Principaux mécanismes de protection des actifs informationnels I
 - 5.3.1. Aperçu des principaux outils cryptographiques pour la protection de la triade CID
 - 5.3.2. Prise en compte de la vie privée, de l'anonymat et des exigences de gestion de la traçabilité des utilisateurs
- 5.4. Principaux mécanismes de protection des actifs informationnels II
 - 5.4.1. Sécurité des communications: protocoles, dispositifs et architectures de sécurité
 - 5.4.2. Sécurité des systèmes d'exploitation
- 5.5. Contrôle interne des SGSI
 - 5.5.1. Taxonomie des contrôles du SGSI: contrôles administratifs, logiques et physiques
 - 5.5.2. Classification des contrôles en fonction de la manière dont ils traitent la menace: contrôles pour la prévention, la détection et la remédiation des menaces
 - 5.5.3. Mise en œuvre de systèmes de contrôle interne dans les SGSI
- 5.6. Types d'audits
 - 5.6.1. Différence entre l'audit et le contrôle interne
 - 5.6.2. Audit interne et externe
 - 5.6.3. Classification de l'audit en fonction de l'objectif et du type d'analyse
- 5.7. Scénariste et scénario: sujet et objet protégés par la Propriété Intellectuelle
 - 5.7.1. Introduction aux tests de pénétration et à l'analyse médico-légale
 - 5.7.2. Définition et pertinence des concepts de *fingerprinting* et *footprinting*
- 5.8. Analyse de vulnérabilité et surveillance du trafic réseau
 - 5.8.1. Outils pour l'analyse de la vulnérabilité des systèmes
 - 5.8.2. Principales vulnérabilités dans le contexte des applications web
 - 5.8.3. Analyse des protocoles de communication
- 5.9. Le processus d'audit informatique
 - 5.9.1. Concept de cycle de vie du développement des systèmes
 - 5.9.2. Suivi des activités et des processus: collecte et traitement des preuves
 - 5.9.3. Méthodologies d'audit informatique
 - 5.9.4. Le processus d'audit informatique
 - 5.9.5. Identification des principales infractions et délits dans le contexte informatique
 - 5.9.6. Enquête sur la criminalité informatique: introduction à l'analyse judiciaire et à sa relation avec l'audit informatique
- 5.10. Plans de continuité des activités et de reprise après sinistre
 - 5.10.1. Définition du plan de continuité d'activité et du concept d'interruption d'activité
 - 5.10.2. Recommandation du NIST sur les plans de continuité des activités
 - 5.10.3. Plan de reprise après sinistre
 - 5.10.4. Processus de plan de reprise après sinistre

Module 6. Intégration des systèmes

- 6.1. Introduction aux systèmes d'information d'entreprise
 - 6.1.1. Le rôle des systèmes d'information
 - 6.1.2. Conception des systèmes d'information?
 - 6.1.3. Dimensions des systèmes d'information
 - 6.1.4. Processus d'affaires et systèmes d'information
 - 6.1.5. Le département IS/IT
- 6.2. Opportunités et besoins en matière de systèmes d'information dans l'entreprise
 - 6.2.1. Organisations et systèmes d'information
 - 6.2.2. Caractéristiques des organisations
 - 6.2.3. Impact des systèmes d'information sur l'entreprise
 - 6.2.4. Les systèmes d'information pour un avantage concurrentiel
 - 6.2.5. Utilisation de systèmes dans l'administration et la gestion des affaires
- 6.3. Concepts de base des systèmes et technologies de l'information
 - 6.3.1. Données, informations et connaissances
 - 6.3.2. Technologie et systèmes d'information
 - 6.3.3. Composants technologiques
 - 6.3.4. Classification et types de systèmes d'information
 - 6.3.5. Architectures basées sur les services et les processus d'entreprise
 - 6.3.6. Formes d'intégration des systèmes
- 6.4. Systèmes intégrés de gestion des ressources de l'entreprise
 - 6.4.1. Exigences commerciales
 - 6.4.2. Un système d'information intégré pour l'entreprise
 - 6.4.3. Acquisition vs. Développement
 - 6.4.4. Mise en œuvre de l'ERP
 - 6.4.5. Implications pour la gestion
 - 6.4.6. Principaux fournisseurs d'ERP
- 6.5. Systèmes d'information pour la gestion de la chaîne d'approvisionnement et de la relation client
 - 6.5.1. Définition de chaîne d'approvisionnement
 - 6.5.2. Gestion effets de la chaîne d'approvisionnement
 - 6.5.3. Le rôle des systèmes d'information
 - 6.5.4. Solutions pour la gestion de chaîne d'approvisionnement
 - 6.5.5. La gestion des relations avec les clients
 - 6.5.6. Le rôle des systèmes d'information
 - 6.5.7. Mise en œuvre d'un système CRM
 - 6.5.8. Facteurs critiques de succès dans la mise en œuvre de la CRM
 - 6.5.9. CRM, e-CRM et autres tendances
- 6.6. Prise de décision en matière d'investissement dans les TIC et planification des systèmes d'information
 - 6.6.1. Critères de décision en matière d'investissement dans les TIC
 - 6.6.2. Lier le projet au plan de gestion et d'affaires
 - 6.6.3. Implications en termes de gestion
 - 6.6.4. Refonte des processus d'affaires
 - 6.6.5. Décision de la direction sur les méthodologies de mise en œuvre
 - 6.6.6. Nécessité d'une planification des systèmes d'information
 - 6.6.7. Objectifs, participants et calendrier
 - 6.6.8. Structure et développement du plan des systèmes
 - 6.6.9. Suivi et mise à jour
- 6.7. Considérations de sécurité dans l'utilisation des TIC
 - 6.7.1. Analyse des risques
 - 6.7.2. Sécurité des systèmes d'information
 - 6.7.3. Conseils pratiques

- 6.8. Faisabilité de la mise en œuvre de projets TIC et aspects financiers dans les projets de systèmes d'information
 - 6.8.1. Description et objectifs
 - 6.8.2. Participants au SVE
 - 6.8.3. Techniques et pratiques
 - 6.8.4. Structure des coûts
 - 6.8.5. Projection financière
 - 6.8.6. Budgets
 - 6.9. *Business Intelligence*
 - 6.9.1. Qu'est-ce que la Business Intelligence?
 - 6.9.2. Stratégie et mise en œuvre de la BI
 - 6.9.3. Le présent et l'avenir de la BI
 - 6.10. ISO/IEC 12207
 - 6.10.1. Qu'est-ce que "ISO/IEC 12207"?
 - 6.10.2. Analyse des Systèmes d'Informations
 - 6.10.3. Conception d'un Système d'Information
 - 6.10.4. Mise en œuvre et acceptation du Système d'Information
-
- Module 7. Réutilisation de software**
- 7.1. Présentation de réutilisation software
 - 7.1.1. Qu'est-ce que la la réutilisation des logiciels
 - 7.1.2. Avantages et inconvénients de la réutilisation des logiciels
 - 7.1.3. Principales techniques de réutilisation des logiciels
 - 7.2. Introduction aux modèles de conception
 - 7.2.1. Qu'est-ce qu'un modèle de conception?
 - 7.2.2. Catalogue des principaux modèles de conception
 - 7.2.3. Comment utiliser les modèles pour résoudre les problèmes de conception
 - 7.2.4. Comment choisir le meilleur modèle de conception
 - 7.3. Modèles de création I
 - 7.3.1. Modèles de création
 - 7.3.2. Modèle de fabrique abstraite
 - 7.3.3. Exemple de mise en œuvre du modèle de fabrique abstraite
 - 7.3.4. Modèle de construction
 - 7.3.5. Exemple de mise en œuvre du constructeur
 - 7.3.6. Usine abstraite vs. Builder
 - 7.4. Modèles de création II
 - 7.4.1. Patron Factory Method
 - 7.4.2. Factory Method vs. Abstract Factory
 - 7.4.3. Modèle Singleton
 - 7.5. Modèles structurels I
 - 7.5.1. Modèles structurels
 - 7.5.2. Modèle Adapter
 - 7.5.3. Modèle Bridge
 - 7.6. Modèle structurels II
 - 7.6.1. Modèle Composite
 - 7.6.2. Modèle Decorador
 - 7.7. Modèle structurels III
 - 7.7.1. Modèle Facade
 - 7.7.2. Modèle Proxy
 - 7.8. Modèle de comportement I
 - 7.8.1. Concept de modèles comportementaux
 - 7.8.2. Modèle de comportement: chaîne de responsabilité
 - 7.8.3. Modèle de comportement Ordre
 - 7.9. Modèle de comportement II
 - 7.9.1. Modèle d'Interprète ou *Interpreter*
 - 7.9.2. Modèle Iterador
 - 7.9.3. Modèle Observador
 - 7.9.4. Modèle de Stratégie

- 7.10. Frameworks
 - 7.10.1. Concept de *framework*
 - 7.10.2. Développement à l'aide de *frameworks*
 - 7.10.3. Model View Controller
 - 7.10.4. *Framework* pour la conception d'interfaces utilisateur graphiques
 - 7.10.5. *Frameworks* pour le développement d'applications web
 - 7.10.6. *Frameworks* pour gérer la persistance des objets dans les bases de données

Module 8. Services informatiques

- 8.1. La transformation digitale I
 - 8.1.1. L'innovation dans l'entreprise
 - 8.1.2. La gestion de la production
 - 8.1.3. La gestion financière
- 8.2. La transformation digitale II
 - 8.2.1. Le marketing
 - 8.2.2. La gestion de Ressources Humaines
 - 8.2.3. Un système d'information complet
- 8.3. Étude de cas
 - 8.3.1. Présentation de l'entreprise
 - 8.3.2. Méthodologie pour analyser l'acquisition de TI
 - 8.3.3. Détermination des coûts, bénéfices et risques
 - 8.3.4. Évaluation économique de l'investissement
- 8.4. La gouvernance et la gestion des TIC
 - 8.4.1. Définition de la gouvernance des technologies et des systèmes d'information
 - 8.4.2. Différence entre la gouvernance et la gestion de l'IST
 - 8.4.3. Cadres pour la gouvernance et la gestion des TSI
 - 8.4.4. Normes, gouvernance et gestion des TSI
- 8.5. Gouvernance d'entreprise dans le domaine des TIC
 - 8.5.1. Qu'est-ce qu'une bonne gouvernance d'entreprise?
 - 8.5.2. Contexte de la gouvernance des TIC
 - 8.5.3. Norme ISO/IEC 38500:2008
 - 8.5.4. Mise en œuvre d'une bonne gouvernance des TIC
 - 8.5.5. Gouvernance et meilleures pratiques en matière de TIC
 - 8.5.6. La gouvernance d'entreprise. Vue d'ensemble et tendances
- 8.6. Objectifs de contrôle pour les technologies de l'information et les technologies connexes (COBIT)
 - 8.6.1. Cadre de mise en œuvre
 - 8.6.2. Domaine: planification et organisation
 - 8.6.3. Domaine: approvisionnement et mise en œuvre
 - 8.6.4. Domaine: livraison et soutien
 - 8.6.5. Domaine: suivi et évaluation
 - 8.6.6. Application du Guide COBIT
- 8.7. La bibliothèque d'infrastructure des technologies de l'information (ITIL)
 - 8.7.1. Introduction à ITIL
 - 8.7.2. Stratégies de service
 - 8.7.3. Conception des services
 - 8.7.4. Transition des services
 - 8.7.5. Opération des services
 - 8.7.6. Amélioration des services
- 8.8. Système de gestion des services
 - 8.8.1. Principes de base de la norme UNE-ISO/IEC 20000-1
 - 8.8.2. La structure de la série de normes ISO/CEI 20000
 - 8.8.3. Exigences du système de gestion des services
 - 8.8.4. Conception et transition de services nouveaux ou modifiés
 - 8.8.5. Processus de prestation de services
 - 8.8.6. Groupes de processus
- 8.9. Système de gestion des actifs des Software
 - 8.9.1. Justification du besoin
 - 8.9.2. Historique
 - 8.9.3. Présentation de 19770
 - 8.9.4. Mise en œuvre de la gestion
- 8.10. Gestion de la continuité des activités
 - 8.10.1. Plan de continuité des activités
 - 8.10.2. Mise en place du BCM

Module 9. Sécurité des systèmes d'information

- 9.1. Un aperçu de la sécurité, de la cryptographie et des cryptanalyses classiques
 - 9.1.1. Sécurité informatique: perspective historique
 - 9.1.2. Mais qu'est-ce que la sécurité exactement?
 - 9.1.3. Histoire de la cryptographie
 - 9.1.4. Chiffres de substitution
 - 9.1.5. Étude de cas: la machine Enigma
- 9.2. Cryptographie symétrique
 - 9.2.1. Introduction et terminologie de base
 - 9.2.2. Cryptage symétrique
 - 9.2.3. Modes d'opération
 - 9.2.4. DES
 - 9.2.5. La nouvelle norme AES
 - 9.2.6. Cryptage de flux
 - 9.2.7. Cryptanalyse
- 9.3. Cryptographie asymétrique
 - 9.3.1. Origines de la cryptographie à clé publique
 - 9.3.2. Concepts de base et fonctionnement
 - 9.3.3. L'algorithme RSA
 - 9.3.4. Certificats numériques
 - 9.3.5. Stockage et gestion des clés
- 9.4. Attaques de réseau
 - 9.4.1. Menaces et attaques contre les réseaux
 - 9.4.2. Énumération
 - 9.4.3. Interception du trafic: *sniffers*
 - 9.4.4. Attaques par déni de service
 - 9.4.5. Attaques par empoisonnement ARP
- 9.5. Architectures de sécurité
 - 9.5.1. Architectures de sécurité traditionnelles
 - 9.5.2. Secure Socket Layer: SSL
 - 9.5.3. Protocole SSH
 - 9.5.4. Réseaux Privés Virtuels (VPN)
 - 9.5.5. Mécanismes de protection des unités de stockage externes
 - 9.5.6. Mécanismes de protection du matériel
- 9.6. Techniques de protection des systèmes et développement de code sécurisé
 - 9.6.1. Sécurité Opérationnelle
 - 9.6.2. Ressources et contrôles
 - 9.6.3. Suivi
 - 9.6.4. Systèmes de détection des intrusions
 - 9.6.5. IDS de l'hôte
 - 9.6.6. IDS réseau
 - 9.6.7. IDS basé sur les signatures
 - 9.6.8. Systèmes de leurres
 - 9.6.9. Principes de base de la sécurité dans le développement du code
 - 9.6.10. Gestion des pannes
 - 9.6.11. Ennemi public numéro 1: les dépassements de tampon (Buffer Overflows)
 - 9.6.12. Botches cryptographiques
- 9.7. *Botnets* et spam
 - 9.7.1. Origine du problème
 - 9.7.2. Processus de spam
 - 9.7.3. Envoi de spam
 - 9.7.4. Affinement des listes de diffusion
 - 9.7.5. Techniques de protection
 - 9.7.6. Service anti- offert par des tiers
 - 9.7.7. Étude de cas
 - 9.7.8. Spam exotique

- 9.8. Audit et attaques du Web
 - 9.8.1. Collecte d'informations
 - 9.8.2. Techniques d'attaque
 - 9.8.3. Outils
- 9.9. Malware et code malveillant
 - 9.9.1. Qu'est-ce qu'un malware?
 - 9.9.2. Types de malware
 - 9.9.3. Virus
 - 9.9.4. Cryptovirus
 - 9.9.5. Vers
 - 9.9.6. Adware
 - 9.9.7. Spyware
 - 9.9.8. Canulars
 - 9.9.9. Pishing
 - 9.9.10. Trojans
 - 9.9.11. L'économie de Malware
 - 9.9.12. Solutions possibles
- 9.10. Analyse médico-légale
 - 9.10.1. Collecte des preuves
 - 9.10.2. Analyse des preuves
 - 9.10.3. Techniques anti-forensic
 - 9.10.4. Étude de cas pratique

Module 10. Gestion de projets

- 10.1. Concepts fondamentaux de la gestion de projet et le cycle de vie de la gestion de projet
 - 10.1.1. Qu'est-ce qu'un projet?
 - 10.1.2. Méthodologie commune
 - 10.1.3. Qu'est-ce que la gestion/leadership de projet?
 - 10.1.4. Qu'est-ce qu'un plan de projet?
 - 10.1.5. Bénéfices
 - 10.1.6. Cycle de vie d'un projet
 - 10.1.7. Groupes de processus ou cycle de vie de la gestion de projet
 - 10.1.8. La relation entre les groupes de processus et les domaines de connaissances
 - 10.1.9. Relations entre le cycle de vie du produit et du projet
- 10.2. Initiation et planification
 - 10.2.1. De l'idée au projet
 - 10.2.2. Élaboration de la charte du projet
 - 10.2.3. Réunion de lancement du projet
 - 10.2.4. Tâches, connaissances et compétences dans le processus de démarrage
 - 10.2.5. Le plan de projet
 - 10.2.6. Élaboration du plan de base. Étapes
 - 10.2.7. Tâches, connaissances et compétences dans le processus de planification
- 10.3. Gestion des *stakeholders* et du champ d'application
 - 10.3.1. Identifier les parties prenantes
 - 10.3.2. Développer le plan de gestion des parties prenantes
 - 10.3.3. Gérer l'engagement des parties prenantes
 - 10.3.4. Suivre l'engagement des parties prenantes
 - 10.3.5. Objectif du projet
 - 10.3.6. La gestion de la portée et son plan
 - 10.3.7. Recueillir les besoins
 - 10.3.8. Définir l'énoncé de la portée
 - 10.3.9. Créer l'OTP
 - 10.3.10. Vérifier et contrôle de la portée

- 10.4. Élaboration du calendrier
 - 10.4.1. La gestion du temps et son plan
 - 10.4.2. Définir les activités
 - 10.4.3. Établir la séquence des activités
 - 10.4.4. Estimation des ressources pour les activités
 - 10.4.5. Estimation de la durée des activités
 - 10.4.6. Élaboration du calendrier et calcul du chemin critique
 - 10.4.7. Contrôle des horaires
- 10.5. Élaboration du budget et réponse aux risques
 - 10.5.1. Estimation des coûts
 - 10.5.2. Élaboration du budget et de la courbe en S
 - 10.5.3. Contrôle des coûts et méthode de la valeur acquise
 - 10.5.4. Concepts de risque
 - 10.5.5. Comment faire une analyse de risque
 - 10.5.6. L'élaboration du plan d'intervention
- 10.6. Gestion de la qualité
 - 10.6.1. Planification de la qualité
 - 10.6.2. Assurance de la qualité
 - 10.6.3. Contrôle de la qualité
 - 10.6.4. Concepts statistiques fondamentaux
 - 10.6.5. Outils de la gestion de la qualité
- 10.7. Communication et ressources humaines
 - 10.7.1. Planification de la gestion des communications
 - 10.7.2. Analyse des besoins en communication
 - 10.7.3. Technologie des communications
 - 10.7.4. Modèles de communication
 - 10.7.5. Méthodes de communication
 - 10.7.6. Plan de gestion de la communication
 - 10.7.7. Gestion des communications
 - 10.7.8. Gestion des ressources humaines
 - 10.7.9. Principaux acteurs et leurs rôles dans les projets
 - 10.7.10. Types d'organisations
 - 10.7.11. Organisation par projet
 - 10.7.12. L'équipe de travail
- 10.8. Approvisionnement
 - 10.8.1. Le processus de passation de marchés
 - 10.8.2. Planification
 - 10.8.3. Recherche de fournisseurs et appels d'offres
 - 10.8.4. Attribution du contrat
 - 10.8.5. Administration du contrat
 - 10.8.6. Contrats
 - 10.8.7. Types de contrats
 - 10.8.8. Négociation de contrats
- 10.9. Exécution, suivi et contrôle et clôture
 - 10.9.1. Les groupes de processus
 - 10.9.2. Mise en œuvre du projet
 - 10.9.3. Suivi et contrôle des projets
 - 10.9.4. Clôture du projet
- 10.10. Responsabilité professionnelle
 - 10.10.1. Responsabilité professionnelle
 - 10.10.2. Caractéristiques de la responsabilité sociale et professionnelle
 - 10.10.3. Responsabilité vs. PMP®
 - 10.10.4. Exemples de responsabilité
 - 10.10.5. Avantages de la professionnalisation



*Une expérience formation unique,
clé et décisive pour stimuler votre
développement professionnel*

05 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.



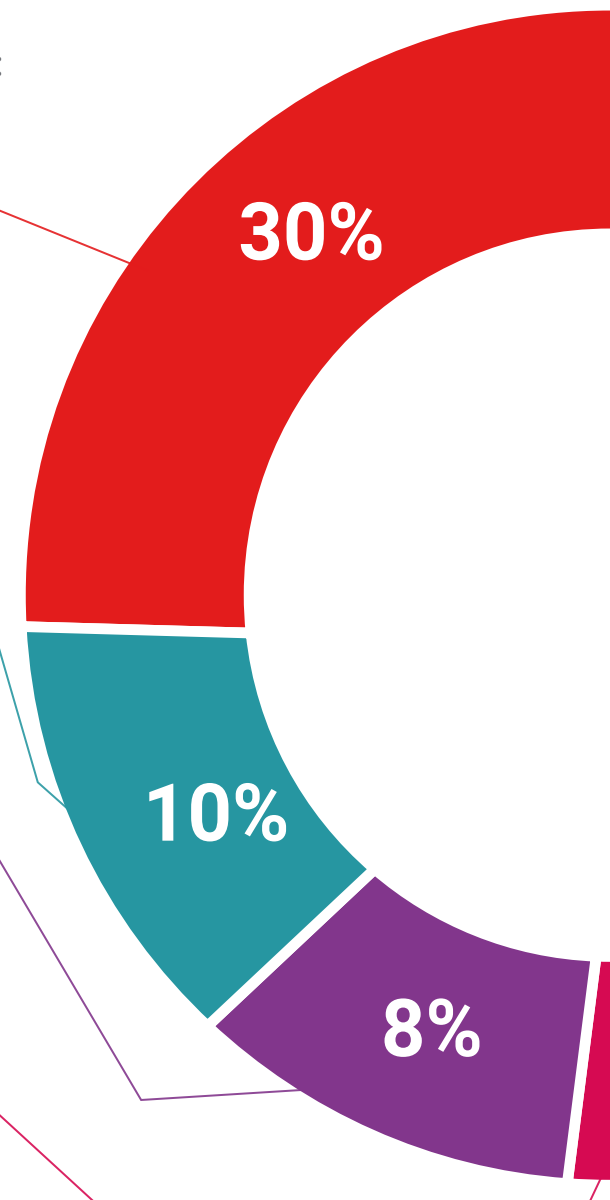
Pratiques en compétences et aptitudes

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.



06 Diplôme

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



“

Réussissez ce programme avec succès et recevez votre diplôme universitaire sans déplacements ni formalités fastidieuses”

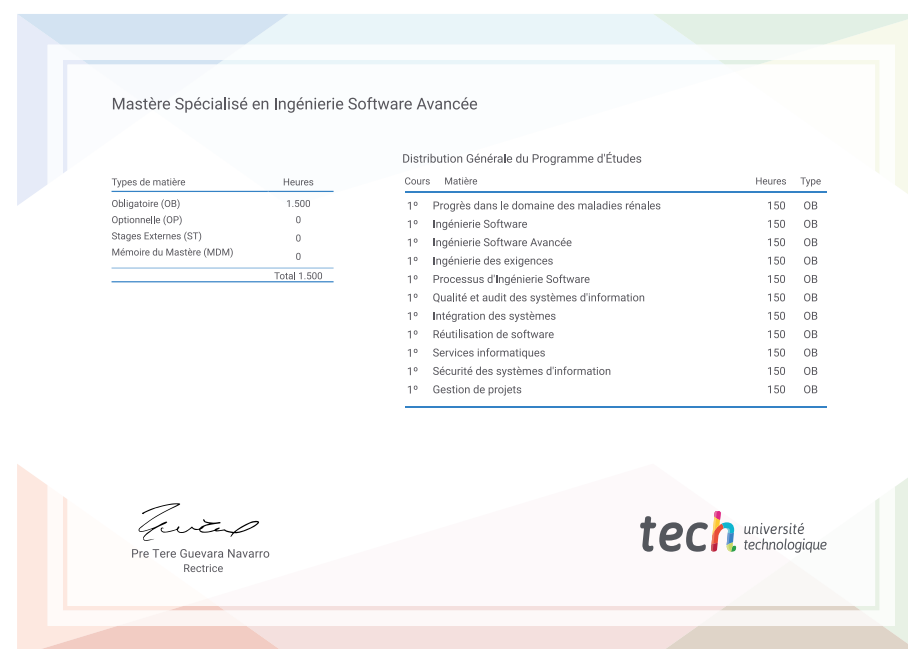
Ce **Mastère Spécialisé en Ingénierie Software Avancée** contient le programme le plus complet et le plus à jour du marché.

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

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

Diplôme: **Mastère Spécialisé en Ingénierie Software Avancée**

N.º d'Heures Officielles: **1.500 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 Spécialisé Ingénierie Software Avancée

- » Modalité: en ligne
- » Durée: 12 mois
- » Qualification: TECH Université Technologique
- » Intensité: 16h/semaine
- » Horaire: à votre rythme
- » Examens: en ligne

