

Mastère Avancé

Ingénierie et Qualité du Software



Mastère Avancé Ingénierie et Qualité du Software

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Euromed University
- » Accréditation: 120 ECTS
- » Horaire: à votre rythme
- » Examens: en ligne

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

Sommaire

01

Présentation du programme

page 4

02

Pourquoi étudier à TECH?

page 8

03

Programme d'études

page 12

04

Objectifs pédagogiques

page 34

05

Opportunités de carrière

page 40

06

Méthodologie d'étude

page 44

07

Corps Enseignant

page 54

08

Diplôme

page 60

01

Présentation du programme

L'Ingénierie Logicielle est devenue la pierre angulaire de la transformation numérique. Aujourd'hui, tous les secteurs s'appuient sur des solutions technologiques pour optimiser les processus, améliorer l'expérience des clients et rester compétitifs. La qualité du Software, quant à elle, garantit que ces solutions sont fiables, évolutives et sûres. Cette discipline est une branche de l'ingénierie qui combine l'expertise technique et de gestion pour garantir que les produits et les systèmes développés sont fonctionnels et durables. C'est pourquoi ce programme va au-delà de la simple programmation et se concentre sur l'ensemble du cycle de vie des Logiciels, de la conception initiale à la maintenance et à l'évolution du système. L'objectif principal est d'offrir aux étudiants une opportunité académique unique qui leur fournit des informations à la pointe de la technologie. TECH Euromed University a développé ce diplôme multidisciplinaire et 100% en ligne qui couvre tout, des fondamentaux du génie Logiciel aux dernières tendances en matière de méthodologies agiles.

```
...se if (settings[0].getStr  
if (name.compare  
name += "  
}  
name += Date  
...se if (set  
...me.
```

“

Rejoignez-nous dès maintenant et commencez à transformer des idées complexes en solutions technologiques tangibles qui auront un impact sur le monde”

La qualité du Logiciel garantit que les systèmes ne répondent pas seulement aux exigences fonctionnelles, mais qu'ils sont également intuitifs, sûrs et durables à long terme. Ceci est particulièrement important dans les secteurs critiques tels que la finance, la santé ou les transports, où les défaillances peuvent avoir de graves conséquences. En outre, en donnant la priorité à la qualité, elle permet aux entreprises de s'adapter avec souplesse aux progrès technologiques constants et de répondre efficacement aux demandes croissantes du marché.

En utilisant des méthodologies telles que le développement agile, *DevOps* et la mise en œuvre de normes de qualité internationales, l'ingénierie Logicielle garantit la livraison de produits dans des délais plus courts. En outre, la maîtrise des coûts et un niveau de qualité qui minimise les erreurs critiques ont été amplifiés par l'intégration de technologies émergentes, telles que l'intelligence artificielle, le *cloud computing* et la cybersécurité. Dans ce contexte, le programme conçu par TECH Euromed University vise à former des professionnels hautement qualifiés dans la conception, le développement, la gestion et l'assurance qualité des Logiciels. Pour acquérir les compétences nécessaires, le programme du Mastère Avancé inclut les concepts les plus récents en matière de gestion de projets technologiques et de gestion stratégique. Cette approche représente une valeur ajoutée tant pour les ingénieurs qui occupent déjà des postes à responsabilité et souhaitent mettre à jour leurs connaissances, que pour ceux qui aspirent à diriger des équipes et des projets pour la première fois dans ce domaine.

L'un des principaux avantages de ce programme est qu'il sera entièrement en ligne, ce qui élimine les déplacements et la nécessité de s'adapter à des horaires spécifiques. En outre, les étudiants bénéficieront de la méthode d'apprentissage *Relearning*, qui s'adapte à leur rythme d'étude. Cette approche flexible est très utile, car elle permet aux étudiants d'organiser efficacement leurs obligations quotidiennes, qu'elles soient professionnelles ou familiales, et de s'épanouir pleinement.

Ce **Mastère Avancé en Ingénierie et Qualité du Software** contient le programme le plus complet et le plus actualisé du marché. Ses caractéristiques sont les suivantes:

- ♦ Le développement d'études de cas présentées par des experts en informatique
- ♦ Le contenu graphique, schématique et éminemment pratique du programme fournit des informations scientifiques et pratiques sur les disciplines essentielles à la pratique professionnelle
- ♦ Les exercices pratiques où effectuer le processus d'auto-évaluation pour améliorer l'apprentissage
- ♦ Il met l'accent sur les méthodologies innovantes en matière d'Ingénierie et Qualité du Software
- ♦ 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



Avec TECH Euromed University, vous n'apprendrez pas seulement à développer du Software, mais aussi à créer des systèmes qui font une différence dans la vie des gens et des entreprises"

“

Maîtriser les compétences et les outils d'ingénierie les plus avancés avec la méthodologie d'enseignement la plus innovante sur la scène académique actuelle”

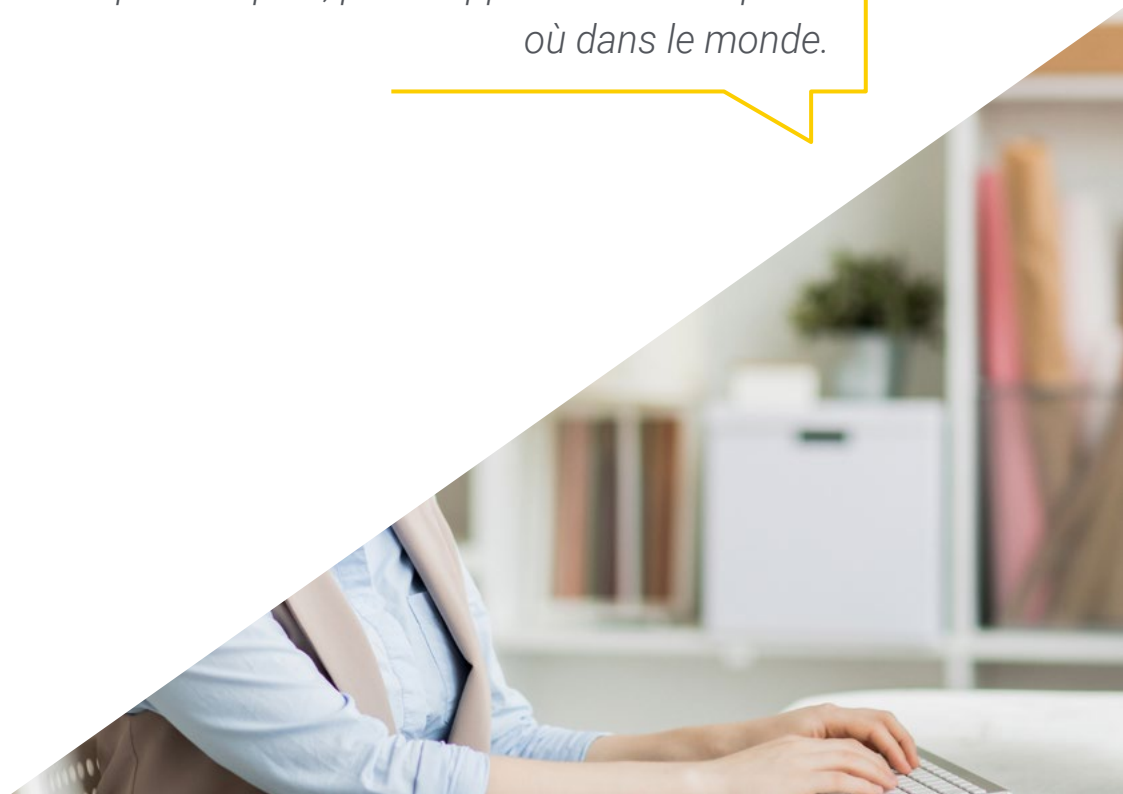
Son corps enseignant comprend des professionnels du domaine de l'Informatique, qui apportent leur expérience professionnelle à ce programme, ainsi que des spécialistes reconnus de grandes sociétés et d'universités prestigieuses.

Son contenu multimédia, développé avec les dernières technologies éducatives, permettra au professionnel un apprentissage situé et contextuel, c'est-à-dire un environnement simulé qui fournira un étude immersif programmé pour s'entraîner dans des situations réelles.

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

Améliorez vos attentes professionnelles en apprenant 100% en ligne, sans interférer avec vos responsabilités personnelles et familiales.

Devenez un professionnel de l'ingénierie de premier plan, prêt à apprendre de n'importe où dans le monde.



02

Pourquoi étudier à TECH?

TECH Euromed University est la plus grande Université numérique du monde. Avec un catalogue impressionnant de plus de 14 000 programmes universitaires, disponibles en 11 langues, elle se positionne comme un leader en matière d'employabilité, avec un taux de placement de 99%. En outre, elle dispose d'un vaste corps professoral composé de plus de 6 000 professeurs de renommée internationale.



“

Étudiez dans la plus grande université numérique du monde et assurez votre réussite professionnelle. L'avenir commence à TECH Euromed University”

La meilleure université en ligne du monde, selon FORBES

Le prestigieux magazine Forbes, spécialisé dans les affaires et la finance, a désigné TECH Euromed University comme « la meilleure université en ligne du monde ». C'est ce qu'ils ont récemment déclaré dans un article de leur édition numérique dans lequel ils se font l'écho de la réussite de cette institution, « grâce à l'offre académique qu'elle propose, à la sélection de son corps enseignant et à une méthode d'apprentissage innovante visant à former les professionnels du futur ».

Le meilleur personnel enseignant top international

Le corps enseignant de TECH Euromed University se compose de plus de 6 000 professeurs jouissant du plus grand prestige international. Des professeurs, des chercheurs et des hauts responsables de multinationales, parmi lesquels figurent Isaiah Covington, entraîneur des Boston Celtics, Magda Romanska, chercheuse principale au Harvard MetaLAB, Ignacio Wistumba, président du département de pathologie moléculaire translationnelle au MD Anderson Cancer Center, et D.W. Pine, directeur de la création du magazine TIME, entre autres.

La plus grande université numérique du monde

TECH Euromed University est la plus grande université numérique du monde. Nous sommes la plus grande institution éducative, avec le meilleur et le plus vaste catalogue éducatif numérique, cent pour cent en ligne et couvrant la grande majorité des domaines de la connaissance. Nous proposons le plus grand nombre de diplômes propres, de diplômes officiels de troisième cycle et de premier cycle au monde. Au total, plus de 14 000 diplômes universitaires, dans dix langues différentes, font de nous la plus grande institution éducative au monde.



Les programmes d'études les plus complets sur la scène universitaire

TECH Euromed University offre les programmes d'études les plus complets sur la scène universitaire, avec des programmes qui couvrent les concepts fondamentaux et, en même temps, les principales avancées scientifiques dans leurs domaines scientifiques spécifiques. En outre, ces programmes sont continuellement mis à jour afin de garantir que les étudiants sont à la pointe du monde universitaire et qu'ils possèdent les compétences professionnelles les plus recherchées. De cette manière, les diplômes de l'université offrent à ses diplômés un avantage significatif pour propulser leur carrière vers le succès.

Une méthode d'apprentissage unique

TECH Euromed University est la première université à utiliser *Relearning* dans tous ses formations. Il s'agit de la meilleure méthodologie d'apprentissage en ligne, accréditée par des certifications internationales de qualité de l'enseignement, fournies par des agences éducatives prestigieuses. En outre, ce modèle académique perturbateur est complété par la « Méthode des Cas », configurant ainsi une stratégie d'enseignement en ligne unique. Des ressources pédagogiques innovantes sont également mises en œuvre, notamment des vidéos détaillées, des infographies et des résumés interactifs.

L'université en ligne officielle de la NBA

TECH Euromed University est l'université en ligne officielle de la NBA. Grâce à un accord avec la grande ligue de basket-ball, elle offre à ses étudiants des programmes universitaires exclusifs ainsi qu'un large éventail de ressources pédagogiques axées sur les activités de la ligue et d'autres domaines de l'industrie du sport. Chaque programme est conçu de manière unique et comprend des conférenciers exceptionnels: des professionnels ayant un passé sportif distingué qui apporteront leur expertise sur les sujets les plus pertinents.

Leaders en matière d'employabilité

TECH Euromed University a réussi à devenir l'université leader en matière d'employabilité. 99% de ses étudiants obtiennent un emploi dans le domaine qu'ils ont étudié dans l'année qui suit la fin de l'un des programmes de l'université. Un nombre similaire parvient à améliorer immédiatement sa carrière. Tout cela grâce à une méthodologie d'étude qui fonde son efficacité sur l'acquisition de compétences pratiques, absolument nécessaires au développement professionnel.



Google Partner Premier

Le géant américain de la technologie a décerné à TECH Euromed University le badge Google Partner Premier. Ce prix, qui n'est décerné qu'à 3% des entreprises dans le monde, souligne l'expérience efficace, flexible et adaptée que cette université offre aux étudiants. Cette reconnaissance atteste non seulement de la rigueur, de la performance et de l'investissement maximaux dans les infrastructures numériques de TECH Euromed University, mais positionne également TECH Euromed University comme l'une des principales entreprises technologiques au monde.



L'université la mieux évaluée par ses étudiants

Le site web d'évaluation Global Score a positionné TECH Euromed University comme l'université la mieux évaluée au monde par ses étudiants. Ce portail d'évaluation, le plus fiable et le plus prestigieux car il vérifie et valide l'authenticité de chaque avis publié, a attribué à TECH Euromed University sa note la plus élevée, 4,9 sur 5, sur la base de plus de 1 000 avis reçus. Des chiffres qui placent TECH Euromed University comme la référence universitaire absolue au niveau international.



03

Programme d'études

Le programme d'études du Mastère Avancé en Ingénierie et Qualité du Software est conçu pour fournir une spécialisation complète et avancée dans tous les domaines clés de l'ingénierie du Logiciel. Les premiers modules se concentrent sur les fondamentaux, allant de la conception de Logiciels et de la gestion des exigences aux architectures technologiques et aux méthodologies agiles. Au fur et à mesure que le programme progresse, les étudiants approfondissent des domaines plus spécialisés tels que l'automatisation des tests, l'intégration continue et l'assurance qualité. En outre, des sujets sur la gestion de projets technologiques sont incorporés, où les participants apprennent à gérer des équipes multidisciplinaires.





“

*Ce Mastère Avancé vous prépare à être
l'expert qui fait la différence dans le secteur
de l'Ingénierie et la Qualité du Software”*

Module 1. Qualité du Software. Niveaux de développement TRL

- 1.1. Éléments influençant la Qualité des Logiciels (I). La Dette Technique
 - 1.1.1. La Dette Technique. Causes et conséquences
 - 1.1.2. Qualité du Software. Principes généraux
 - 1.1.3. Qualité des logiciels avec et sans principes
 - 1.1.3.1. Conséquences
 - 1.1.3.2. La nécessité de l'application des principes de qualité dans les Logiciels
 - 1.1.4. Qualité du Software. Typologie
 - 1.1.5. Software de Qualité. Caractéristiques spécifiques
- 1.2. Éléments influençant la Qualité des Logiciels (II). Coûts associés
 - 1.2.1. Qualité du Software. Éléments d'influence
 - 1.2.2. Qualité du Software. Idées fausses
 - 1.2.3. Qualité du Software. Coûts associés
- 1.3. Modèles de Qualité des Logiciels (I). Gestion des connaissances
 - 1.3.1. Modèles de qualité générale
 - 1.3.1.1. Gestion de la qualité totale
 - 1.3.1.2. Modèle Européen d'Excellence Commerciale (EFQM)
 - 1.3.1.3. Modèle Six-sigma
 - 1.3.2. Modèles de Gestion des Connaissances
 - 1.3.2.1. Modèle Dyba
 - 1.3.2.2. Modèle Seks
 - 1.3.3. Expérience du Paradigme Factory et QIP
 - 1.3.4. Modèles de qualité d'usage (25010)
- 1.4. Modèles de Qualité des Logiciels (III). Qualité des données, des processus et des modèles SEI
 - 1.4.1. Modèle de qualité des données
 - 1.4.2. Modélisation des processus Logiciels
 - 1.4.3. Software & Systems Process Engineering Metamodel Specification (SPEM)
 - 1.4.4. Modèles de la DIE
 - 1.4.4.1. CMMI
 - 1.4.4.2. SCAMPI
 - 1.4.4.3. IDEAL
- 1.5. Normes ISO de Qualité du Software (I). Analyse des normes
 - 1.5.1. Normes ISO 9000
 - 1.5.1.1. Normes ISO 9000
 - 1.5.1.2. Famille de normes de qualité ISO (9000)
 - 1.5.2. Autres normes ISO relatives à la Qualité
 - 1.5.3. Normes de Modélisation de la Qualité (ISO 2501)
 - 1.5.4. Normes de Mesure de la Qualité (ISO 2502n)
- 1.6. Normes ISO de Qualité du Software (II). Exigences et évaluation
 - 1.6.1. Normes d'Exigences de Qualité (2503n)
 - 1.6.2. Normes sur l'Évaluation de la qualité (2504n)
 - 1.6.3. ISO/IEC 24744:2007
- 1.7. Niveaux de développement TRL (I). Niveaux 1 à 4
 - 1.7.1. Niveau TRL
 - 1.7.2. Niveau 1: principes de base
 - 1.7.3. Niveau 2: concept et/ou application
 - 1.7.4. Niveau 3: fonction analytique critique
 - 1.7.5. Niveau 4: validation des composants dans un environnement de laboratoire
- 1.8. Niveaux de développement TRL (II). Niveaux de 5 à 9
 - 1.8.1. Niveau 5: validation du composant dans un environnement pertinent
 - 1.8.2. Niveau 6: modèle de système/sous-système
 - 1.8.3. Niveau 7: démonstration en environnement réel
 - 1.8.4. Niveau 8: système complet et certifié
 - 1.8.5. Niveau 9: succès dans un environnement réel
- 1.9. Niveaux de développement TRL. Utilisations
 - 1.9.1. Exemple d'une entreprise avec un environnement de laboratoire
 - 1.9.2. Exemple d'une entreprise de R&D&I
 - 1.9.3. Exemple d'une entreprise de R&D&I industriel
 - 1.9.4. Exemple d'une entreprise commune laboratoire-ingénierie
- 1.10. Qualité du Software. Principaux détails
 - 1.10.1. Détails méthodologiques
 - 1.10.2. Détails techniques
 - 1.10.3. Détails sur la gestion des projets Logiciels
 - 1.10.3.1. Qualité Systèmes d'Information
 - 1.10.3.2. Qualité des Produits Logiciels
 - 1.10.3.3. Qualité des Processus Logiciels

Module 2. Développement de projets Software. Documentation fonctionnelle et technique

- 2.1. Gestion de projets
 - 2.1.1. Gestion de projet en matière de Qualité des Logiciels
 - 2.1.2. Gestion de projets. Avantages
 - 2.1.3. Gestion de projets. Typologie
- 2.2. Méthodologie de la gestion de projet
 - 2.2.1. Méthodologie de la gestion de projet
 - 2.2.2. Méthodologie de la gestion de projet Typologie
 - 2.2.3. Méthodologie dans la gestion de projets. Application
- 2.3. Phase d'Identification des Besoins
 - 2.3.1. Identification des besoins du projet
 - 2.3.2. Gestion des réunions de projet
 - 2.3.3. Documentation à fournir
- 2.4. Modèle
 - 2.4.1. Phase initiale
 - 2.4.2. Phase d'analyse
 - 2.4.3. Phase de construction
 - 2.4.4. Phase de test
 - 2.4.5. Livraison
- 2.5. Modèle de Données à utiliser
 - 2.5.1. Détermination du nouveau Modèle de Données
 - 2.5.2. Identification du Plan de Migration des Données
 - 2.5.3. Ensemble de données
- 2.6. Impact sur d'autres projets
 - 2.6.1. Impact d'un projet. Exemples
- 2.7. *MUST* du projet
 - 2.7.1. *MUST* du projet
 - 2.7.2. Identification du *MUST* du projet
 - 2.7.3. Identification des points de mise en œuvre pour la réalisation d'un projet
- 2.8. L'équipe de construction du projet
 - 2.8.1. Rôles à jouer en fonction du projet
 - 2.8.2. Contact avec les RH pour le recrutement
 - 2.8.3. Livrables et calendrier du projet

- 2.9. Aspects techniques d'un projet de Logiciel
 - 2.9.1. Architecte du projet. Aspects Techniques
 - 2.9.2. Responsables Techniques
 - 2.9.3. Construction du projet Logiciel
 - 2.9.4. Évaluation de la qualité du code Sonar
- 2.10. Livrables du projet
 - 2.10.1. Analyse fonctionnelle
 - 2.10.2. Modèles de données
 - 2.10.3. Diagrammes d'état
 - 2.10.4. Documentation technique

Module 3. *Testing* de Software. Automatisation des Tests

- 3.1. Modèles de qualité des Logiciels
 - 3.1.1. Qualité du produit
 - 3.1.2. Qualité du processus
 - 3.1.3. Qualité de l'utilisation
- 3.2. Qualité du processus
 - 3.2.1. Qualité du processus
 - 3.2.2. Modèles de maturité
 - 3.2.3. Norme ISO 15504
 - 3.2.3.1. Objectifs
 - 3.2.3.2. Contexte
 - 3.2.3.3. Étapes
- 3.3. Norme ISO/IEC 15504
 - 3.3.1. Catégories de processus
 - 3.3.2. Processus de Développement. Exemple
 - 3.3.3. Fragment de profil
 - 3.3.4. Étapes
- 3.4. CMMI (*Capability Maturity Model Integration*)
 - 3.4.1. CMMI. Intégration du Modèle de Maturité de la Capacité
 - 3.4.2. Modèles et zones. Typologie
 - 3.4.3. Domaines de processus
 - 3.4.4. Niveaux de Capacité
 - 3.4.5. Gestion des processus
 - 3.4.6. Gestion de projet

- 3.5. Gestion des changements et des référentiels
 - 3.5.1. Gestion des changements Logiciels
 - 3.5.1.1. Élément de configuration. Intégration continue
 - 3.5.1.2. Lignes
 - 3.5.1.3. Organigrammes
 - 3.5.1.4. *Branches*
 - 3.5.2. Référentiel
 - 3.5.2.1. Contrôle de la version
 - 3.5.2.2. Équipe de travail et utilisation du référentiel
 - 3.5.2.3. Intégration continue dans le référentiel
- 3.6. *Team Foundation Server (TFS)*
 - 3.6.1. Installation et configuration
 - 3.6.2. Création d'un projet d'équipement
 - 3.6.3. Ajouter du contenu au contrôle de la source
 - 3.6.4. *TFS on Cloud*
- 3.7. *Testing*
 - 3.7.1. Motivation pour les tests
 - 3.7.2. Test de vérification
 - 3.7.3. Tests bêta
 - 3.7.4. Mise en œuvre et maintenance
- 3.8. Essais de charge
 - 3.8.1. *Load testing*
 - 3.8.2. Tests avec *LoadView*
 - 3.8.3. Tests avec *K6 Cloud*
 - 3.8.4. Tests avec *Loader*
- 3.9. Tests unitaires de stress et d'endurance
 - 3.9.1. Raison d'être des tests unitaires
 - 3.9.2. Outils pour *l'unit testing*
 - 3.9.3. Motivation des tests de stress
 - 3.9.4. Test en utilisant le *StressTesting*
 - 3.9.5. Motivation pour les tests de résistance
 - 3.9.6. Test à l'aide de *LoadRunner*

- 3.10. Échelles Conception de Logiciels évolutifs
 - 3.10.1. Scalabilité et architecture Logicielle
 - 3.10.2. Indépendance entre les couches
 - 3.10.3. Couplage entre les couches. Modèles architecturaux

Module 4. Méthodologies de gestion de projets de Software. Méthodologies Waterfall par rapport aux Méthodologies Agiles

- 4.1. Méthodologie Waterfall
 - 4.1.1. Méthodologie Waterfall
 - 4.1.2. Méthodologie Waterfall. Influence sur la Qualité des Logiciels
 - 4.1.3. Méthodologie Waterfall. Exemples
- 4.2. Méthodologie Agile
 - 4.2.1. Méthodologie Agile
 - 4.2.2. Méthodologie Agile. Influence sur la Qualité des Logiciels
 - 4.2.3. Méthodologie Agile Exemples
- 4.3. Méthodologie SCRUM
 - 4.3.1. Méthodologie SCRUM
 - 4.3.2. Manifeste de SCRUM
 - 4.3.3. Applications du SCRUM
- 4.4. Panel Kanban
 - 4.4.1. Méthode Kanban
 - 4.4.2. Panel Kanban
 - 4.4.3. Panel Kanban. Exemples d'application
- 4.5. Gestion de Projet en Waterfall
 - 4.5.1. Phases d'un projet
 - 4.5.2. Vision dans un projet Waterfall
 - 4.5.3. Livrables à considérer
- 4.6. Gestion de projet en SCRUM
 - 4.6.1. Phases d'un projet SCRUM
 - 4.6.2. Vision dans un projet SCRUM
 - 4.6.3. Produits livrables à considérer
- 4.7. Waterfall vs SCRUM. Comparaison
 - 4.7.1. Approche par projet pilote
 - 4.7.2. Projet utilisant Waterfall. Exemple
 - 4.7.3. Projet utilisant SCRUM. Exemple

- 4.8. Aperçu des Clients
 - 4.8.1. Documents dans un Waterfall
 - 4.8.2. Documents dans un SCRUM
 - 4.8.3. Comparaison
- 4.9. Structure Kanban
 - 4.9.1. Histoires d'Utilisateurs
 - 4.9.2. Backlog
 - 4.9.3. Analyse Kanban
- 4.10. Projets Hybrides
 - 4.10.1. Construction du projet
 - 4.10.2. Gestion de projet
 - 4.10.3. Produits livrables à considérer

Module 5. TDD (*Test Driven Development*). Conception de Logiciels pilotés par les tests

- 5.1. TDD. *Test Driven Development*
 - 5.1.1. TDD. *Test Driven Development*
 - 5.1.2. TDD. Influence du TDD sur la qualité
 - 5.1.3. Conception et développement pilotés par les Tests. Exemples
- 5.2. Cycle TDD
 - 5.2.1. Choix d'une exigence
 - 5.2.2. Test. Typologie
 - 5.2.2.1. Tests unitaires
 - 5.2.2.2. Tests d'Intégration
 - 5.2.2.3. Preuves *End To End*
 - 5.2.3. Vérification des tests. Défaillances
 - 5.2.4. Création de la Mise en Oeuvre
 - 5.2.5. Exécution de tests automatisés
 - 5.2.6. Élimination des doubles emplois
 - 5.2.7. Mise à jour de la liste des exigences
 - 5.2.8. Répétition du cycle TDD
 - 5.2.9. Cycle TDD. Exemple théoriques et pratiques
- 5.3. Stratégies de Mise en œuvre du TDD
 - 5.3.1. Mise en œuvre fictive
 - 5.3.2. Mise en œuvre triangulaire
 - 5.3.3. Mise en œuvre évidente
- 5.4. TDD. Utilisation. Avantages et inconvénients
 - 5.4.1. Avantages de l'utilisation
 - 5.4.2. Limites d'utilisation
 - 5.4.3. Équilibre de la qualité dans la mise en œuvre
- 5.5. TDD. Bonnes pratiques
 - 5.5.1. Règles TDD
 - 5.5.2. Règle 1: Faites un test préalable qui échoue avant de coder en production
 - 5.5.3. Règle 2: ne pas écrire plus d'un test unitaire
 - 5.5.4. Règle 3: ne pas écrire plus de code que nécessaire
 - 5.5.5. Erreurs et anti-modèles à éviter dans le TDD
- 5.6. Simulation d'un projet réel pour utiliser TDD (I)
 - 5.6.1. Aperçu du projet (Entreprise A)
 - 5.6.2. Application du TDD
 - 5.6.3. Exercices proposés
 - 5.6.4. Exercices *Feedback*
- 5.7. Simulation d'un projet réel pour utiliser le TDD (II)
 - 5.7.1. Aperçu du projet (Entreprise B)
 - 5.7.2. Application du TDD
 - 5.7.3. Exercices proposés
 - 5.7.4. Exercices *Feedback*
- 5.8. Simulation d'un projet réel pour utiliser le TDD (III)
 - 5.8.1. Aperçu du projet (Entreprise C)
 - 5.8.2. Application du TDD
 - 5.8.3. Exercices proposés
 - 5.8.4. Exercices *Feedback*
- 5.9. Alternatives au TDD. *Test Driven Development*
 - 5.9.1. TCR (*Test Commit Revert*)
 - 5.9.2. BDD (*Behavior Driven Development*)
 - 5.9.3. ATDD (*Acceptance Test Driven Development*)
 - 5.9.4. TDD. Comparaison théorique

- 5.10. TDD TCR, BDD et ATDD. Comparaison pratique
 - 5.10.1. Définition du problème
 - 5.10.2. Résoudre avec TCR
 - 5.10.3. Résoudre avec BDD
 - 5.10.4. Résoudre avec ATDD

Module 6. DevOps. Gestion de Qualité du Software

- 6.1. DevOps. Gestion de Qualité du Software
 - 6.1.1. DevOps
 - 6.1.2. DevOps et Qualité du Software
 - 6.1.3. DevOps. Avantages de la culture DevOps
- 6.2. DevOps. Relation avec Agile
 - 6.2.1. Livraison accélérée
 - 6.2.2. Qualité
 - 6.2.3. Réduction des coûts
- 6.3. Mise en œuvre de DevOps
 - 6.3.1. Identification des problèmes
 - 6.3.2. Mise en œuvre dans une entreprise
 - 6.3.3. Paramètres de Mise en Œuvre
- 6.4. Cycle de livraison des Logiciels
 - 6.4.1. Méthodes de Conception
 - 6.4.2. Conventions
 - 6.4.3. Feuille de route
- 6.5. Développement de Code Sans Bogues
 - 6.5.1. Un code facile à maintenir
 - 6.5.2. Modèles de développement
 - 6.5.3. *Testing* du Code
 - 6.5.4. Développement de Logiciels au Niveau du Code. Bonnes pratiques
- 6.6. Automatisation
 - 6.6.1. Automatisation. Types de tests
 - 6.6.2. Coût de l'automatisation et de la maintenance
 - 6.6.3. Automatisation. Atténuer les erreurs

- 6.7. Déploiements
 - 6.7.1. Évaluation des objectifs
 - 6.7.2. Conception d'un processus automatique et adapté
 - 6.7.3. Retour d'information et réactivité
- 6.8. Gestion des Incidents de
 - 6.8.1. Préparation aux incidents
 - 6.8.2. Analyse et résolution des incidents
 - 6.8.3. Éviter les erreurs futures
- 6.9. Automatisation des déploiements
 - 6.9.1. Préparation des déploiements automatisés
 - 6.9.2. Évaluation automatique de l'état des processus
 - 6.9.3. Métriques et capacité de retour en arrière
- 6.10. Bonnes pratiques. Évolution de DevOps
 - 6.10.1. Guide des meilleures pratiques DevOps
 - 6.10.2. DevOps. Méthodologie pour l'équipe
 - 6.10.3. Éviter les niches

Module 7. DevOps et intégration continue. Solutions pratiques avancées en matière de développement de Software

- 7.1. Flux de livraison des Logiciels
 - 7.1.1. Identification des acteurs et des artefacts
 - 7.1.2. Conception du flux de livraison du Logiciel
 - 7.1.3. Flux de livraison des Logiciels. Exigences entre étapes
- 7.2. Automatisation des processus
 - 7.2.1. Intégration continue
 - 7.2.2. Intégration continue
 - 7.2.3. Configuration des environnements et gestion des secrets
- 7.3. Pipelines déclaratifs
 - 7.3.1. Différences entre les pipelines traditionnels, de type code et déclaratifs
 - 7.3.2. Pipelines déclaratifs
 - 7.3.3. Pipelines déclaratifs dans Jenkins
 - 7.3.4. Comparaison des fournisseurs d'intégration continue

- 7.4. Des portails de qualité et un retour d'information riche
 - 7.4.1. Portes de qualité
 - 7.4.2. Des normes de Qualité avec des portes de qualité. Maintenance
 - 7.4.3. Exigences commerciales sur les demandes d'intégration
- 7.5. Gestion des artefacts
 - 7.5.1. Artefacts et cycle de Vie
 - 7.5.2. Systèmes de stockage et de gestion des artefacts
 - 7.5.3. La sécurité dans la gestion des artefacts
- 7.6. Intégration continue
 - 7.6.1. Déploiement continu sous forme de conteneurs
 - 7.6.2. Déploiement continu avec PaaS
- 7.7. Amélioration de l'exécution du Pipeline: analyse statique et *Git Hooks*
 - 7.7.1. Analyse statique
 - 7.7.2. Règles de style de code
 - 7.7.3. *Git Hooks* et *Tests Unitaires*
 - 7.7.4. L'impact des infrastructures
- 7.8. Vulnérabilités dans les conteneurs
 - 7.8.1. Vulnérabilités dans les conteneurs
 - 7.8.2. Balayage d'images
 - 7.8.3. Rapports et alertes périodiques

Module 8. Conception de Bases de Données (DB). Standardisation et performance. Qualité du Software

- 8.1. Conception de bases de données
 - 8.1.1. Bases de Données. Typologie
 - 8.1.2. Bases de données utilisées actuellement
 - 8.1.2.1. Relationnel
 - 8.1.2.2. Clé-valeur
 - 8.1.2.3. Basé sur le réseau
 - 8.1.3. Qualité des données
- 8.2. Conception d'un Modèle Entité-Relation (I)
 - 8.2.1. Modèle Entité-Relation. Qualité et documentation
 - 8.2.2. Entités
 - 8.2.2.1. Entité forte
 - 8.2.2.2. Entité faible

- 8.2.3. Attributs
- 8.2.4. Ensemble de Relations
 - 8.2.4.1. 1 à 1
 - 8.2.4.2. 1 à plusieurs
 - 8.2.4.3. De plusieurs à un
 - 8.2.4.4. Beaucoup à beaucoup
- 8.2.5. Clés
 - 8.2.5.1. Clé primaire
 - 8.2.5.2. Clé étrangère
 - 8.2.5.3. Clé primaire de l'entité faible
- 8.2.6. Restrictions
- 8.2.7. Cardinalité
- 8.2.8. Héritage
- 8.2.9. Agrégation
- 8.3. Modèle Entité-Relation (II). Outils
 - 8.3.1. Modèle Entité-Relation. Outils
 - 8.3.2. Modèle Entité-Relation. Exemple pratique
 - 8.3.3. Modèle Entité-Relation réalisable
 - 8.3.3.1. Échantillon visuel
 - 8.3.3.2. Échantillon en représentation de tableau
- 8.4. Normalisation (I) des bases de données (DB). Considérations sur la Qualité des Logiciels
 - 8.4.1. Normalisation et qualité des DB
 - 8.4.2. Dépendances
 - 8.4.2.1. Dépendance fonctionnelle
 - 8.4.2.2. Propriétés de la dépendance fonctionnelle
 - 8.4.2.3. Propriétés inférées
 - 8.4.3. Clés
- 8.5. Normalisation (II) de la base de données (BD). Formes Normales et Règles de Codd
 - 8.5.1. Formes Normales
 - 8.5.1.1. Première Forme Normale (1FN)
 - 8.5.1.2. Deuxième Forme Normale (2FN)
 - 8.5.1.3. Troisième Forme Normale (3FN)
 - 8.5.1.4. Forme normale de Boyce-Codd (BCNF)
 - 8.5.1.5. Quatrième Forme Normale (4FN)
 - 8.5.1.6. Cinquième Forme Normale (5FN)

- 8.5.2. Les règles de Codd
 - 8.5.2.1. Règle 1: Information
 - 8.5.2.2. Règle 2: accès garanti
 - 8.5.2.3. Règle 3: Traitement systématique des valeurs nulles
 - 8.5.2.4. Règle 4: description de la base de données
 - 8.5.2.5. Règle 5: Sous-langage intégral
 - 8.5.2.6. Règle n° 6: Voir la mise à jour
 - 8.5.2.7. Règle 7: Insertion et mise à jour
 - 8.5.2.8. Règle 8: indépendance physique
 - 8.5.2.9. Règle 9: indépendance logique
 - 8.5.2.10. Règle 10: indépendance de l'intégrité
 - 8.5.2.10.1. Règles d'intégrité
 - 8.5.2.11. Règle 11: distribution
 - 8.5.2.12. Règle 12: Non-subversion
- 8.5.3. Exemple pratique
- 8.6. Entrepôt de données / Système OLAP
 - 8.6.1. Entrepôt de données
 - 8.6.2. Tableau des faits
 - 8.6.3. Tableau des dimensions
 - 8.6.4. Création du système OLAP. Outils
- 8.7. Performances des bases de données (DB)
 - 8.7.1. Optimisation de l'index
 - 8.7.2. Optimisation des requêtes
 - 8.7.3. Partitionnement des tables
- 8.8. Simulation d'un projet réel pour la conception de BD (I)
 - 8.8.1. Aperçu du projet (Entreprise A)
 - 8.8.2. Application de la conception de bases de données
 - 8.8.3. Exercices proposés
 - 8.8.4. Exercices proposés. *Feedback*
- 8.9. Simulation d'un projet réel pour la conception de BD (II)
 - 8.9.1. Aperçu du projet (Entreprise B)
 - 8.9.2. Application de la conception de bases de données
 - 8.9.3. Exercices Proposés
 - 8.9.4. Exercices Proposés. *Feedback*

- 8.10. Pertinence de l'optimisation des bases de données dans la Qualité des Logiciels
 - 8.10.1. Optimisation de la Conception
 - 8.10.2. Optimisation du Code de Requête
 - 8.10.3. Optimisation du Code des Procédures stockées
 - 8.10.4. Influence des *Triggers* sur la Qualité des Logiciels. Recommandations d'utilisation

Module 9. Conception d'architectures évolutives. L'architecture dans le cycle de vie des Logiciels

- 9.1. Conception d'architectures évolutives(I)
 - 9.1.1. Architectures évolutives
 - 9.1.2. Principes d'une architecture évolutive
 - 9.1.2.1. Fiable
 - 9.1.2.2. Évolutif
 - 9.1.2.3. Maintenable
 - 9.1.3. Types d'extensibilité
 - 9.1.3.1. Vertical
 - 9.1.3.2. Horizontal
 - 9.1.3.3. Combinaison
- 9.2. Architectures de DDD (*Domain-Driven Design*)
 - 9.2.1. Le Modèle DDD. Orientation du Domaine
 - 9.2.2. Couches, répartition des responsabilités et modèles de conception
 - 9.2.3. Le découplage comme base de la qualité
- 9.3. Conception d'architectures évolutives (II). Avantages, limites et stratégies de conception
 - 9.3.1. Architecture évolutive. Bénéfices
 - 9.3.2. Architecture évolutive. Limites
 - 9.3.3. Stratégies pour le développement d'architectures évolutives (tableau descriptif)
- 9.4. Cycle de qualité des Logiciels (I). Étapes
 - 9.4.1. Cycle de vie du Software
 - 9.4.1.1. Phase de planification
 - 9.4.1.2. Phase d'analyse
 - 9.4.1.3. Phase de conception
 - 9.4.1.4. Phase de mise en œuvre
 - 9.4.1.5. Phase de test
 - 9.4.1.6. Phase d'installation/déploiement
 - 9.4.1.7. Phase d'utilisation et de maintenance

- 9.5. Modèles de cycle de vie des Logiciels
 - 9.5.1. Modèle en cascade
 - 9.5.2. Modèle répétitif
 - 9.5.3. Modèle en spirale
 - 9.5.4. Modèle Big Bang
- 9.6. Cycle de vie des Logiciels (II). Automatisation
 - 9.6.1. Cycles de Vie du Développement Logiciel. Solutions
 - 9.6.1.1. Intégration continue et développement continu (CI/CD)
 - 9.6.1.2. Méthodologies Agiles
 - 9.6.1.3. DevOps / Opérations de Production
 - 9.6.2. Tendances futures
 - 9.6.3. Exemples pratiques
- 9.7. L'Architecture Logicielle dans le cycle de vie du Logiciel
 - 9.7.1. Bénéfices
 - 9.7.2. Limites
 - 9.7.3. Outils
- 9.8. Simulation d'un projet réel pour la conception d'une architecture Logicielle (I)
 - 9.8.1. Description générale du projet (entreprise A)
 - 9.8.2. Application de la conception de l'architecture Logicielle
 - 9.8.3. Exercices proposés
 - 9.8.4. Exercices proposés. *Feedback*
- 9.9. Simulation d'un projet réel pour la conception d'une architecture Logicielle (II)
 - 9.9.1. Aperçu du projet (Entreprise B)
 - 9.9.2. Application de la conception de l'architecture Logicielle
 - 9.9.3. Exercices proposés
 - 9.9.4. Exercices proposés *Feedback*
- 9.10. Simulation d'un projet réel pour la conception d'une architecture Logicielle (III)
 - 9.10.1. Aperçu du projet (Entreprise C)
 - 9.10.2. Application de la conception de l'architecture Logicielle
 - 9.10.3. Exercices Proposés
 - 9.10.4. Exercices Proposés. *Feedback*

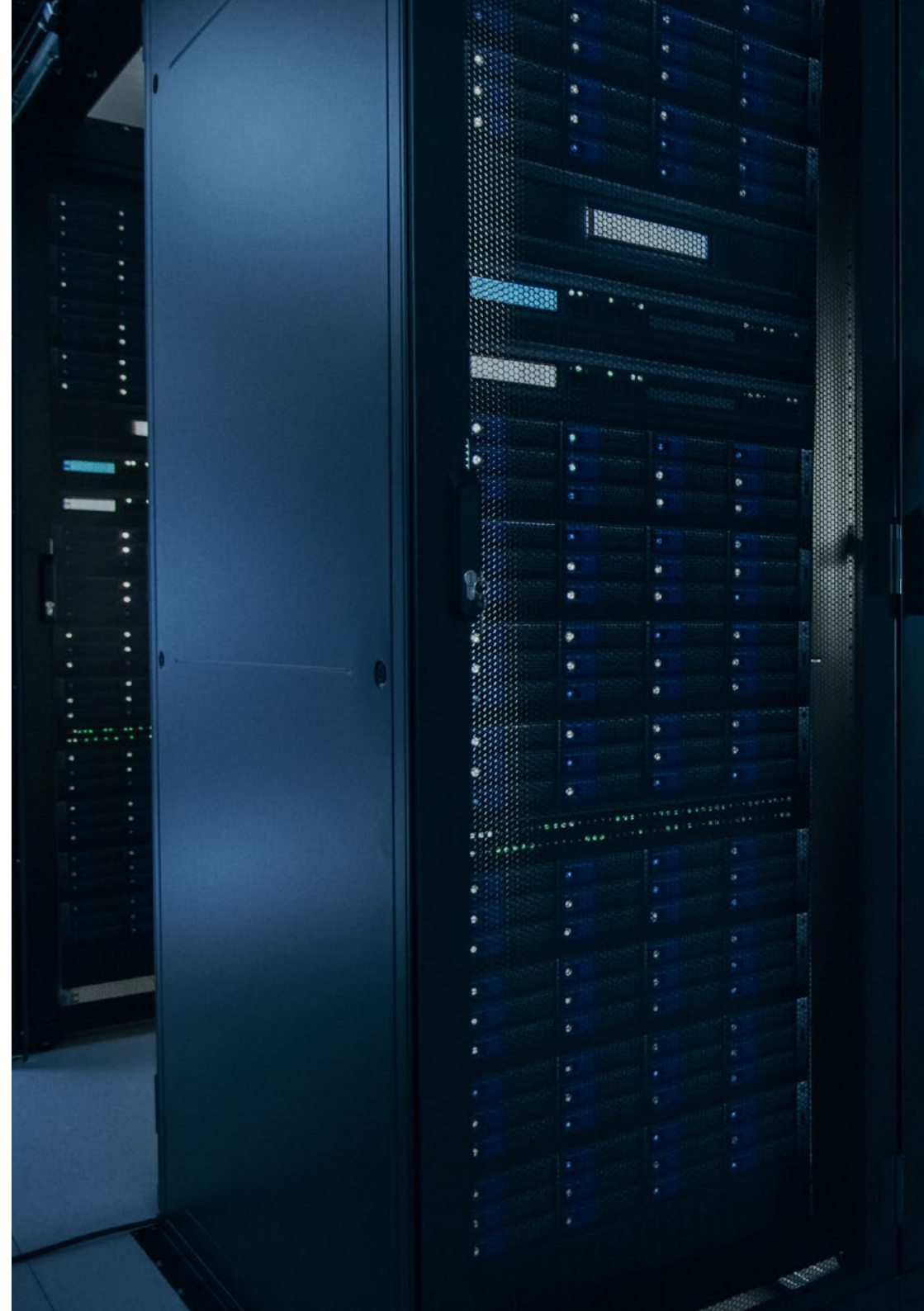
Module 10. Critères de Qualité ISO, IEC 9126. Mesures de Qualité du Software

- 10.1. Critères de qualité Norme ISO, IEC 9126
 - 10.1.1. Critères de qualité
 - 10.1.2. Qualité du Software. Justification. Norme ISO, IEC 9126
 - 10.1.3. La Mesure de la qualité des Logiciels en tant qu'Indicateur clé
- 10.2. Critères de qualité des Logiciels. Caractéristiques
 - 10.2.1. Fiabilité
 - 10.2.2. Fonctionnalité
 - 10.2.3. Efficacité
 - 10.2.4. Utilisabilité
 - 10.2.5. Maintenance
 - 10.2.6. Portabilité
- 10.3. Norme ISO, CEI 9126 (I). Présentation
 - 10.3.1. Description de la Norme ISO, IEC 9126
 - 10.3.2. Fonctionnalité
 - 10.3.3. Fiabilité
 - 10.3.4. Utilisabilité
 - 10.3.5. Maintenance
 - 10.3.6. Portabilité
 - 10.3.7. Qualité de l'utilisation
 - 10.3.8. Mesures de la qualité des Logiciels
 - 10.3.9. Les mesures de la qualité dans la norme ISO 9126
- 10.4. Norme ISO, CEI 9126 (II). Modèles de McCall et Boehm
 - 10.4.1. Modèle McCall: Facteurs de qualité
 - 10.4.2. Modèle Boehm
 - 10.4.3. Niveau intermédiaire. Caractéristiques
- 10.5. Mesures de la qualité des Logiciels (I). Éléments
 - 10.5.1. Mesure
 - 10.5.2. Métriques
 - 10.5.3. Indicateur
 - 10.5.3.1. Types d'indicateurs
 - 10.5.4. Mesures et modèles
 - 10.5.5. Portée des métriques Logicielles
 - 10.5.6. Classification des métriques Logicielles

- 10.6. Mesure de la qualité des Logiciels (II). Pratique de la mesure
 - 10.6.1. Collecte de données métriques
 - 10.6.2. Mesure des attributs internes du produit
 - 10.6.3. Mesure des attributs externes du produit
 - 10.6.4. Mesure des ressources
 - 10.6.5. Métriques pour les systèmes orientés objet
- 10.7. Conception d'un indicateur unique de qualité des Logiciels
 - 10.7.1. Indicateur unique en tant que scoreur global
 - 10.7.2. Développement d'indicateurs, justification et Application
 - 10.7.3. Exemples d'application. Besoin de connaître le détail
- 10.8. Simulation d'un projet réel pour la mesure de la qualité (I)
 - 10.8.1. Description générale du projet (entreprise A)
 - 10.8.2. Application de la mesure de la qualité
 - 10.8.3. Exercices proposés
 - 10.8.4. Exercices proposés. *Feedback*
- 10.9. Simulation d'un projet réel pour la mesure de la qualité (II)
 - 10.9.1. Aperçu du projet (entreprise B)
 - 10.9.2. Application de la mesure de la qualité
 - 10.9.3. Exercices proposés
 - 10.9.4. Exercices proposés. *Feedback*
- 10.10. Simulation d'un projet réel pour la mesure de la qualité (III)
 - 10.10.1. Aperçu du projet (entreprise C)
 - 10.10.2. Application de la mesure de la qualité
 - 10.10.3. Exercices proposés
 - 10.10.4. Exercices proposés. *Feedback*

Module 11. Méthodologies, développement et qualité du Génie Logiciel

- 11.1. Développement de Software basé sur des modèles
 - 11.1.1. La nécessité de
 - 11.1.3. Modélisation des objets
 - 11.1.4. UML
 - 11.1.5. Outils CASE





- 11.2. Modélisation des applications et patrons de conception avec UML
 - 11.2.1. Modélisation avancée des exigences
 - 11.2.2. Modélisation statique avancée
 - 11.2.3. Modélisation dynamique avancée
 - 11.2.4. Modélisation des composants
 - 11.2.5. Introduction aux patrons de conception avec UML
 - 11.2.6. Adapter
 - 11.2.7. Factory
 - 11.2.8. Singleton
 - 11.2.9. Strategy
 - 11.2.10. Composite
 - 11.2.11. Façade
 - 11.2.12. Observer
- 11.3. Ingénierie dirigée par les modèles
 - 11.3.1. Introduction
 - 11.3.2. Métamodélisation des systèmes
 - 11.3.3. MDA
 - 11.3.4. DSL
 - 11.3.5. Raffinements de modèles avec OCL
 - 11.3.6. Transformations de modèles
- 11.4. Ontologies en ingénierie Software
 - 11.4.1. Introduction
 - 11.4.2. Ingénierie ontologique
 - 11.4.3. Application des ontologies en génie Logiciel

Module 12. Gestion de projets de Software

- 12.1. Gestion des *stakeholders* et du champ d'application
 - 12.1.1. Identifier les parties prenantes
 - 12.1.2. Développer le plan de gestion des parties prenantes
 - 12.1.3. Gérer l'engagement des parties prenantes
 - 12.1.4. Suivre l'engagement des parties prenantes
 - 12.1.5. Objectif du projet
 - 12.1.6. La gestion de la portée et son plan
 - 12.1.7. Recueillir les besoins

- 12.1.8. Définir l'énoncé de la portée
- 12.1.9. Créer l'OTP
- 12.1.10. Vérifier et contrôle de la portée
- 12.2. Élaboration du calendrier
 - 12.2.1. La gestion du temps et son plan
 - 12.2.2. Définir les activités
 - 12.2.3. Établir la séquence des activités
 - 12.2.4. Estimation des ressources pour les activités
 - 12.2.5. Estimation de la durée des activités
 - 12.2.6. Élaboration du calendrier et calcul du chemin critique
 - 12.2.7. Contrôle des horaires
- 12.3. Élaboration du budget et réponse aux risques
 - 12.3.1. Estimation des coûts
 - 12.3.2. Élaboration du budget et de la courbe en S
 - 12.3.3. Contrôle des coûts et méthode de la valeur acquise
 - 12.3.4. Concepts de risque
 - 12.3.5. Comment faire une analyse de risque
 - 12.3.6. L'élaboration du plan d'intervention
- 12.4. Communication et ressources humaines
 - 12.4.1. Planification de la gestion des communications
 - 12.4.2. Analyse des besoins en communication
 - 12.4.3. Technologie des communications
 - 12.4.4. Modèles de communication
 - 12.4.5. Méthodes de communication
 - 12.4.6. Plan de gestion de la communication
 - 12.4.7. Gestion des communications
 - 12.4.8. Gestion des ressources humaines
 - 12.4.9. Principaux acteurs et leurs rôles dans les projets
 - 12.4.10. Types d'organisations
 - 12.4.11. Organisation par projet
 - 12.4.12. L'équipe de travail

- 12.5. Approvisionnement
 - 12.5.1. Le processus de passation de marchés
 - 12.5.2. Planification
 - 12.5.3. Recherche de fournisseurs et appels d'offres
 - 12.5.4. Attribution du contrat
 - 12.5.5. Administration du contrat
 - 12.5.6. Contrats
 - 12.5.7. Types de contrats
 - 12.5.8. Négociation de contrats
- 12.6. Exécution, suivi et contrôle et clôture
 - 12.6.1. Les groupes de processus
 - 12.6.2. Mise en œuvre du projet
 - 12.6.3. Suivi et contrôle des projets
 - 12.6.4. Clôture du projet
- 12.7. Responsabilité professionnelle
 - 12.7.1. Responsabilité professionnelle
 - 12.7.2. Caractéristiques de la responsabilité sociale et professionnelle
 - 12.7.3. Code d'éthique du chef de projet
 - 12.7.4. Responsabilité vs. PMP®
 - 12.7.5. Exemples de responsabilité
 - 12.7.6. Avantages de la professionnalisation

Module 13. Plateformes de développement du Logiciel

- 13.1. Introduction au développement d'applications
 - 13.1.1. Applications de bureau
 - 13.1.2. Langages de programmation
 - 13.1.3. Environnements de développement intégrés
 - 13.1.4. Applications web
 - 13.1.5. Applications mobiles
 - 13.1.6. Applications du Cloud
- 13.2. Développement d'applications et d'interfaces graphiques en Java
 - 13.2.1. Environnements de développement intégrés pour Java
 - 13.2.2. Principaux IDE pour Java
 - 13.2.3. Introduction à la plate-forme de développement Eclipse

- 13.2.4. Introduction à la plate-forme de développement NetBeans
- 13.2.5. Modèle Vue-Contrôleur pour les interfaces utilisateur graphiques
- 13.2.6. Conception d'une interface graphique dans Eclipse
- 13.2.7. Conception d'une interface graphique dans NetBeans
- 13.3. Débogage et tests en Java
 - 13.3.1. Test et débogage des programmes Java
 - 13.3.2. Débogage dans Eclipse
 - 13.3.3. Débogage dans NetBeans
- 13.4. Développement d'applications et d'interfaces graphiques en .NET
 - 13.4.1. Net Framework
 - 13.4.2. Composants de la plate-forme de développement .NET
 - 13.4.3. Visual Studio .NET
 - 13.4.4. Outils .NET pour GUI
 - 13.4.5. GUI avec Windows Presentation Foundation
 - 13.4.6. Débogage et compilation d'une application WPF
- 13.5. Programmation pour les réseaux .NET
 - 13.5.1. Introduction à la programmation en réseau .NET
 - 13.5.2. Requêtes et réponses en .NET
 - 13.5.3. Utilisation de protocoles d'application dans .NET
 - 13.5.4. Sécurité dans la programmation de réseaux .NET
- 13.6. Environnements de développement d'applications mobiles
 - 13.6.1. Applications mobiles
 - 13.6.2. Applications mobiles Android
 - 13.6.3. Étapes du développement d'Android
 - 13.6.4. L'IDE Android Studio
- 13.7. Développement d'applications dans l'environnement Android Studio
 - 13.7.1. Installer et lancer Android Studio
 - 13.7.2. Exécution d'une application Android
 - 13.7.3. Développement d'une interface graphique dans Android Studio
 - 13.7.4. Démarrer des activités dans Android Studio
- 13.8. Débogage et publication d'applications Android
 - 13.8.1. Débogage d'une application dans Android Studio
 - 13.8.2. Mémoriser des applications dans Android Studio
 - 13.8.3. Publication d'une application sur Google Play

- 13.9. Développement d'applications pour le nuage
 - 13.9.1. *Cloud computing*
 - 13.9.2. Niveaux des *nuages*: SaaS, PaaS, IaaS
 - 13.9.3. Principales plates-formes de développement en nuage
 - 13.9.4. Références bibliographiques
- 13.10. Introduction à Google Cloud Platform
 - 13.10.1. Principes de base de Google Cloud Platform
 - 13.10.2. Services Google Cloud Platform
 - 13.10.3. Outils de Google Cloud Platform

Module 14. Informatique client web

- 14.1. Introduction à HTML
 - 14.1.1. Structure d'un document
 - 14.1.2. Couleur
 - 14.1.3. Texte
 - 14.1.4. Liens hypertextes
 - 14.1.5. Images
 - 14.1.6. Listes
 - 14.1.7. Tableaux
 - 14.1.8. Cadres (*frames*)
 - 14.1.9. Formulaires
 - 14.1.10. Éléments spécifiques aux technologies mobiles
 - 14.1.11. Articles désaffectés
- 14.2. Feuilles de style web (CSS)
 - 14.2.1. Éléments et structure d'une feuille de style
 - 14.2.1.1. Création de feuilles de style
 - 14.2.1.2. Application des styles. Sélecteurs
 - 14.2.1.3. Héritage de style et cascade
 - 14.2.1.4. Mise en page à l'aide de styles
 - 14.2.1.5. Structure de la page à l'aide de styles. Le modèle de la boîte
 - 14.2.2. Conception de styles pour différents appareils
 - 14.2.3. Types de feuilles de style: statiques et dynamiques. Les Pseudoclasses
 - 14.2.4. Bonnes pratiques dans l'utilisation des feuilles de style

- 14.3. Introduction et histoire de JavaScript
 - 14.3.1. Introduction
 - 14.3.2. Histoire de JavaScript
 - 14.3.3. Environnement de développement à utiliser
- 14.4. Notions de base de la programmation web
 - 14.4.1. Syntaxe de base de JavaScript
 - 14.4.2. Types de données primitives et opérateurs
 - 14.4.3. Variables et champs d'application
 - 14.4.4. Chaînes de texte et *template literals*
 - 14.4.5. Nombres et booléens
 - 14.4.6. Comparaisons
- 14.5. Structures complexes en JavaScript
 - 14.5.1. Vecteurs ou *arrays* et objets
 - 14.5.2. Sets
 - 14.5.3. Cartes
 - 14.5.4. Disjonctions
 - 14.5.5. Boucles
- 14.6. Fonctions et objectifs
 - 14.6.1. Définition et invocation des fonctions
 - 14.6.2. Argumentation
 - 14.6.3. Fonctions de la flèche
 - 14.6.4. Fonctions de rappel ou *callback*
 - 14.6.5. Fonctions d'ordre supérieur
 - 14.6.6. Objets littéraux
 - 14.6.7. L'objet *this*
 - 14.6.8. Les objets en tant qu'espaces de noms: l'objet Math et l'objet Date
- 14.7. Le modèle objet du document (DOM)
 - 14.7.1. Qu'est-ce que le DOM?
 - 14.7.2. Un peu d'histoire
 - 14.7.3. Navigation et obtention d'éléments
 - 14.7.4. Un DOM virtuel avec JSDOM
 - 14.7.5. Sélecteurs de requête ou *query selectors*
 - 14.7.6. Navigation par les propriétés
 - 14.7.7. Attribution d'attributs aux éléments
 - 14.7.8. Création et modification de nœuds
 - 14.7.9. Mise à jour du style des éléments du DOM
- 14.8. Développement web moderne
 - 14.8.1. Flux événementiel et *listeners*
 - 14.8.2. *Toolkits* web modernes et systèmes d'alignement
 - 14.8.3. Mode strict JavaScript
 - 14.8.4. Plus sur les fonctions
 - 14.8.5. Promesses et fonctions asynchrones
 - 14.8.6. *Fermetures*
 - 14.8.7. Programmation fonctionnelle
 - 14.8.8. POO en JavaScript
- 14.9. Utilisabilité du web
 - 14.9.1. Introduction à l'utilisabilité
 - 14.9.2. Définition de l'utilisabilité
 - 14.9.3. Importance de la conception de sites web centrés sur l'utilisateur
 - 14.9.4. Différences entre l'accessibilité et la facilité d'utilisation
 - 14.9.5. Avantages et problèmes liés à la combinaison de l'accessibilité et de la facilité d'utilisation
 - 14.9.6. Avantages et difficultés de la mise en œuvre de sites web utilisables
 - 14.9.7. Méthodes d'utilisabilité
 - 14.9.8. Analyse des besoins des utilisateurs
 - 14.9.9. Principes du design conceptuel. Création de prototypes orientés vers l'utilisateur
 - 14.9.10. Lignes directrices pour la création de sites web utilisables
 - 14.9.10.1. Lignes directrices d'utilisabilité de Jakob Nielsen
 - 14.9.10.2. Lignes directrices d'utilisabilité par Bruce Tognazzini
 - 14.9.11. Évaluation de la convivialité
- 14.10. Accessibilité du web
 - 14.10.1. Introduction
 - 14.10.2. Définition de l'accessibilité du web
 - 14.10.3. Types de handicaps
 - 14.10.3.1. Handicaps temporaires ou permanents
 - 14.10.3.2. Déficience visuelle
 - 14.10.3.3. Déficience auditive
 - 14.10.3.4. Handicaps moteurs
 - 14.10.3.5. Handicaps neurologiques ou cognitifs
 - 14.10.3.6. Difficultés liées au vieillissement
 - 14.10.3.7. Contraintes liées à l'environnement
 - 14.10.3.8. Obstacles à l'accès au web

- 14.10.4. Aides techniques et produits d'assistance pour surmonter les obstacles
 - 14.10.4.1. Aides pour les aveugles
 - 14.10.4.2. Aides pour les personnes malvoyantes
 - 14.10.4.3. Aides pour les daltoniens
 - 14.10.4.4. Aides pour les personnes malentendantes
 - 14.10.4.5. Aides pour les personnes handicapées moteur
 - 14.10.4.6. Aides pour les personnes souffrant de handicaps cognitifs et neurologiques
- 14.10.5. Avantages et difficultés de la mise en œuvre de l'accessibilité du web
- 14.10.6. Réglementation et normes en matière d'accessibilité du web
- 14.10.7. Organismes de réglementation de l'accessibilité du web
- 14.10.8. Comparaison des normes et des standards
- 14.10.9. Lignes directrices pour le respect des règlements et des normes
 - 14.10.9.1. Description des principales lignes directrices (images, liens, vidéos, etc.)
 - 14.10.9.2. Lignes directrices pour une navigation accessible
 - 14.10.9.2.1. Perceptibilité
 - 14.10.9.2.2. Opérationnalité
 - 14.10.9.2.3. Compréhensibilité
 - 14.10.9.2.4. Robustesse
- 14.10.10. Description du processus de mise en conformité de l'accessibilité du web
- 14.10.11. Niveaux de conformité
- 14.10.12. Critères de conformité
- 14.10.13. Exigences de conformité
- 14.10.14. Méthodologie d'évaluation de l'accessibilité des sites web

Module 15. Informatique des Serveurs Web

- 15.1. Introduction à la programmation côté serveur: PHP
 - 15.1.1. Principes de base de la programmation côté serveur
 - 15.1.2. Syntaxe de base de PHP
 - 15.1.3. Génération de contenu HTML avec PHP
 - 15.1.4. Environnements de développement et de test: XAMPP
- 15.2. PHP avancé
 - 15.2.1. Structures de contrôle avec PHP
 - 15.2.2. Fonctions de PHP
 - 15.2.3. Gestion des *arrays* en PHP
 - 15.2.4. Traitement des chaînes de caractères avec PHP
 - 15.2.5. Orientation vers l'objet en PHP
- 15.3. Modèles de données
 - 15.3.1. Concept de données. Cycle de vie des données
 - 15.3.2. Types de données
 - 15.3.2.1. Bases
 - 15.3.2.2. Enregistrements
 - 15.3.2.3. Dynamisme
- 15.4. Le modèle relationnel
 - 15.4.1. Description
 - 15.4.2. Entités et types d'entités
 - 15.4.3. Éléments de données. Attributs
 - 15.4.4. Relations: types, sous-types, cardinalité
 - 15.4.5. Clés. Types de clés
 - 15.4.6. Normalisation Formes normales
- 15.5. Construction du modèle logique de données
 - 15.5.1. Spécification des tableaux
 - 15.5.2. Définition des colonnes
 - 15.5.3. Spécifications clés
 - 15.5.4. Conversion en formes normales. Dépendances
- 15.6. Le modèle physique de données. Fichiers de données
 - 15.6.1. Description des fichiers de données
 - 15.6.2. Types de fichiers
 - 15.6.3. Modes d'accès
 - 15.6.4. Organisation des dossiers
- 15.7. Accès aux bases de données à partir de PHP
 - 15.7.1. Introduction à MariaDB
 - 15.7.2. Travailler avec une base de données MariaDB: le langage SQL
 - 15.7.3. Accéder à la base de données MariaDB depuis PHP
 - 15.7.4. Introduction à MySQL
 - 15.7.5. Travailler avec une base de données MySQL: le langage SQL
 - 15.7.6. Accéder à la base de données MySQL depuis PHP

- 15.8. Interaction avec le client à partir de PHP
 - 15.8.1. Formulaires PHP
 - 15.8.2. Cookies
 - 15.8.3. Gestion des sessions
- 15.9. Architecture des applications web
 - 15.9.1. Le modèle Modèle Vue Contrôleur
 - 15.9.2. Contrôleur
 - 15.9.3. Modèle
 - 15.9.4. Afficher
- 15.10. Introduction aux services web
 - 15.10.1. Introduction à XML
 - 15.10.2. Architectures orientées services (SOA): services web
 - 15.10.3. Création de services web SOAP et REST
 - 15.10.4. Le protocole SOAP
 - 15.10.5. Le protocole REST

Module 16. Gestion de la sécurité

- 16.1. La sécurité de l'information
 - 16.1.1. Introduction
 - 16.1.2. La sécurité de l'information implique la confidentialité, l'intégrité et la disponibilité
 - 16.1.3. La sécurité est une question économique
 - 16.1.4. La sécurité est un processus
 - 16.1.5. La classification des informations
 - 16.1.6. La sécurité de l'information implique une gestion des risques
 - 16.1.7. La sécurité s'articule avec les contrôles de sécurité
 - 16.1.8. La sécurité est à la fois physique et logique
 - 16.1.9. La sécurité implique des personnes
- 16.2. Le professionnel de la sécurité de l'information
 - 16.2.1. Introduction
 - 16.2.2. La sécurité de l'information en tant que profession
 - 16.2.3. Certifications (ISC)2
 - 16.2.4. La norme ISO 27001
 - 16.2.5. Bonnes pratiques de sécurité dans la gestion des services IT
 - 16.2.6. Modèles de maturité pour la sécurité de l'information
 - 16.2.7. Autres certifications, normes et ressources professionnelles
- 16.3. Contrôle d'accès
 - 16.3.1. Introduction
 - 16.3.2. Exigences en matière de contrôle d'accès
 - 16.3.3. Mécanismes d'authentification
 - 16.3.4. Méthodes d'autorisation
 - 16.3.5. Comptabilité d'accès et audit
 - 16.3.6. Technologies Triple A
- 16.4. Programmes, processus et politiques en matière de sécurité de l'information
 - 16.4.1. Introduction
 - 16.4.2. Programmes de gestion de la sécurité
 - 16.4.3. Gestion des risques
 - 16.4.4. Conception de la politique de sécurité
- 16.5. Plans de continuité des activités
 - 16.5.1. Introduction aux PCN
 - 16.5.2. Phases I et II
 - 16.5.3. Phases III et IV
 - 16.5.4. Maintenance du PCN
- 16.6. Procédures pour la protection adéquate de l'entreprise
 - 16.6.1. Réseaux DMZ
 - 16.6.2. Systèmes de détection d'intrusion
 - 16.6.3. Listes de contrôle d'accès
 - 16.6.4. Apprendre de l'attaquant: Honeypot
- 16.7. Architecture de sécurité. Prévention
 - 16.7.1. Aperçu général Activités et modèle de couche
 - 16.7.2. Défense du périmètre (pare-feu, WAF, IPS, etc.)
 - 16.7.3. Défense des points finaux (équipements, serveurs et services)
- 16.8. Architecture de sécurité. Détection
 - 16.8.1. Aperçu de la détection et de la surveillance
 - 16.8.2. Logs, rupture de trafic cryptée, enregistrement et Siems
 - 16.8.3. Alertes et renseignements
- 16.9. Architecture de sécurité. Réaction
 - 16.9.1. Réaction. Produits, services et ressources
 - 16.9.2. Gestion des Incidents de
 - 16.9.3. CERTS et CSIRT

- 16.10. Architecture de sécurité. Récupération
 - 16.10.1. Résilience, Concepts, Exigences Commerciales et Réglementations
 - 16.10.2. Solutions IT de Résilience
 - 16.10.3. Gestion et Gouvernance des Crises

Module 17. Sécurité du Software

- 17.1. Questions relatives à la sécurité des Softwares
 - 17.1.1. Introduction au problème de la sécurité des Softwares
 - 17.1.2. Les vulnérabilités et leur classification
 - 17.1.3. Propriétés des Softwares sécurisés
 - 17.1.4. Références
- 17.2. Principes de conception de la sécurité des Softwares
 - 17.2.1. Introduction
 - 17.2.2. Principes de conception de la sécurité des Softwares
 - 17.2.3. Types de S-SDLC
 - 17.2.4. Sécurité des Softwares dans les phases S-SDLC
 - 17.2.5. Méthodologies et normes
 - 17.2.6. Références
- 17.3. Sécurité dans le cycle de vie du Software dans les phases d'exigences et de conception
 - 17.3.1. Introduction
 - 17.3.2. Modélisation des attaques
 - 17.3.3. Cas d'abus
 - 17.3.4. Ingénierie des exigences de sécurité
 - 17.3.5. Analyse des risques. Architectural
 - 17.3.6. Modèles de conception
 - 17.3.7. Références
- 17.4. Sécurité du cycle de vie des Softwares dans les phases de codage, de test et d'exploitation
 - 17.4.1. Introduction
 - 17.4.2. Tests de sécurité fondés sur le risque
 - 17.4.3. Examen du code
 - 17.4.4. Test de pénétration
 - 17.4.5. Opérations de sécurité
 - 17.4.6. Examen externe
 - 17.4.7. Références
- 17.5. Applications de codage sécurisé I
 - 17.5.1. Introduction
 - 17.5.2. Pratiques de codage sécurisées
 - 17.5.3. Traitement et validation des entrées
 - 17.5.4. Débordement de mémoire
 - 17.5.5. Références
- 17.6. Applications de codage sécurisé II
 - 17.6.1. Introduction
 - 17.6.2. Integers overflows, erreurs de troncature et problèmes de conversion de type entre entiers
 - 17.6.3. Erreurs et exceptions
 - 17.6.4. Vie privée et confidentialité
 - 17.6.5. Programmes privilégiés
 - 17.6.6. Références
- 17.7. La sécurité dans le développement et dans le cloud
 - 17.7.1. La sécurité dans le développement ; méthodologie et pratique
 - 17.7.2. Modèles PaaS, IaaS, CaaS et SaaS
 - 17.7.3. Sécurité dans le nuage et pour les services Cloud
- 17.8. Cryptage
 - 17.8.1. Principes fondamentaux de la Cryptologie
 - 17.8.2. Cryptage Symétrique et asymétrique
 - 17.8.3. Cryptage au repos et en transit
- 17.9. Orchestration et automatisation de la sécurité (SOAR)
 - 17.9.1. Complexité du traitement manuel ; nécessité d'automatiser les tâches
 - 17.9.2. Produits et services
 - 17.9.3. Architecture du SOAR
- 17.10. La sécurité dans le Télétravail
 - 17.10.1. Besoin et scénarios
 - 17.10.2. Produits et services
 - 17.10.3. La sécurité dans le télétravail

Module 18. Administration des Serveurs Web

- 18.1. Introduction aux serveurs web
 - 18.1.1. Qu'est-ce qu'un serveur web?
 - 18.1.2. Architecture et fonctionnement d'un serveur web
 - 18.1.3. Ressources et contenu d'un serveur web
 - 18.1.4. Serveurs d'application
 - 18.1.5. Serveurs Proxy
 - 18.1.6. Principaux serveurs web du marché
 - 18.1.7. Statistiques d'utilisation du serveur web
 - 18.1.8. Sécurité des serveurs web
 - 18.1.9. Équilibrage de la charge sur les serveurs web
 - 18.1.10. Références
- 18.2. Traitement du protocole HTTP
 - 18.2.1. Fonctionnement et structure
 - 18.2.2. Description des demandes ou des request methods
 - 18.2.3. Codes d'état
 - 18.2.4. En-têtes
 - 18.2.5. Codage du contenu. Pages de code
 - 18.2.6. Effectuer des requêtes HTTP sur l'internet à l'aide d'un proxy, de livehttpheaders ou d'une méthode similaire, en analysant le protocole utilisé
- 18.3. Description des architectures distribuées multi-serveurs
 - 18.3.1. Modèle à trois couches
 - 18.3.2. Tolérance aux pannes
 - 18.3.3. Partage des charges
 - 18.3.4. Stockage de l'état de la session
 - 18.3.5. Magasins de stockage
- 18.4. Internet Information Services (IIS)
 - 18.4.1. Qu'est-ce que l'IIS?
 - 18.4.2. Histoire et évolution de l'IIS
 - 18.4.3. Principaux avantages et caractéristiques d'IIS7 et des versions ultérieures
 - 18.4.4. Architecture IIS7 et ultérieures
- 18.5. Installation, administration et configuration de IIS
 - 18.5.1. Préambule
 - 18.5.2. Installation d'Internet Information Services (IIS)
 - 18.5.3. Outils d'administration IIS
 - 18.5.4. Création, configuration et administration de sites web
 - 18.5.5. Installation et gestion des extensions dans IIS
- 18.6. Sécurité avancée dans IIS
 - 18.6.1. Préambule
 - 18.6.2. Authentification, Autorisation et Contrôle d'accès dans IIS
 - 18.6.3. Mise en place d'un site web sécurisé sur IIS avec SSL
 - 18.6.4. Politiques de sécurité mises en œuvre dans IIS 8.x
- 18.7. Introduction à Apache
 - 18.7.1. Qu'est-ce qu'Apache?
 - 18.7.2. Principaux avantages d'Apache
 - 18.7.3. Principales caractéristiques d'Apache
 - 18.7.4. Architecture
- 18.8. Installation et configuration d'Apache
 - 18.8.1. Installation initiale d'Apache
 - 18.8.2. Configuration d'Apache
- 18.9. Installation et configuration des différents modules d'Apache
 - 18.9.1. Installation des modules Apache
 - 18.9.2. Types de modules
 - 18.9.3. Configuration sécurisée d'Apache
- 18.10. Sécurité avancée
 - 18.10.1. Authentification, autorisation et contrôle d'accès
 - 18.10.2. Méthodes d'authentification
 - 18.10.3. Configuration sécurisée d'Apache avec SSL

Module 19. Audit de Sécurité

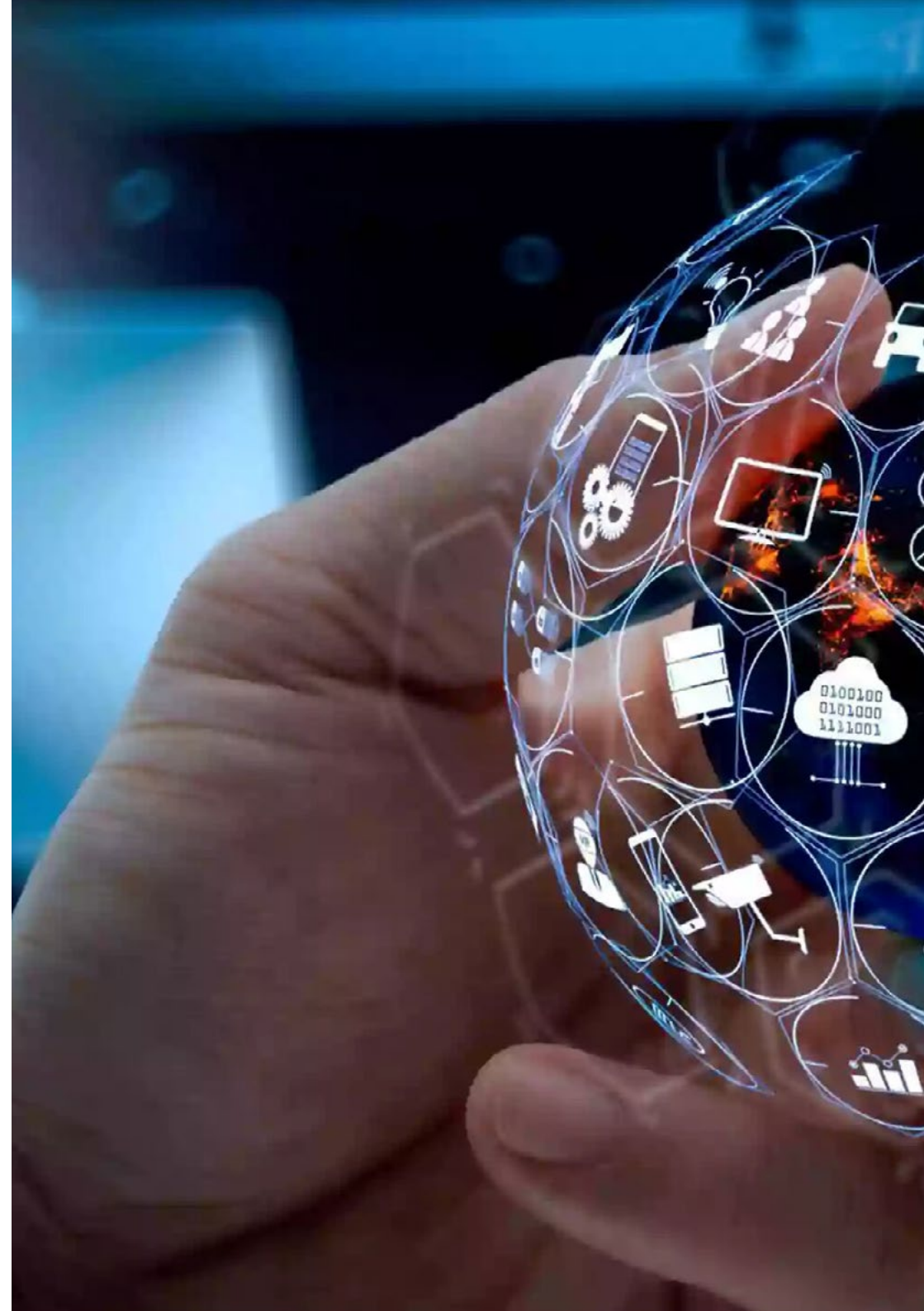
- 19.1. Introduction aux systèmes d'information et à leur audit
 - 19.1.1. Introduction aux systèmes d'information et au rôle de l'audit informatique
 - 19.1.2. Définitions d'audit informatique et contrôle interne informatique
 - 19.1.3. Fonctions et objectifs de l'audit informatique
 - 19.1.4. Différences entre le contrôle interne et l'audit informatique
- 19.2. Contrôles internes des systèmes d'information
 - 19.2.1. Organigramme fonctionnel d'un centre de traitement des données
 - 19.2.2. Classification des contrôles des systèmes d'information
 - 19.2.3. La règle d'Or
- 19.3. Le processus et les phases de l'audit des Systèmes d'Information
 - 19.3.1. Évaluation des risques (RRA) et autres méthodologies d'audit informatique
 - 19.3.2. Réalisation d'un audit des Systèmes d'Information. Phases d'audit
 - 19.3.3. Compétences fondamentales de l'auditeur de Systèmes d'Information
- 19.4. Audit de sécurité technique des systèmes et des réseaux
 - 19.4.1. Audits techniques de sécurité. Tests de pénétration. Concepts préliminaires
 - 19.4.2. Audits de sécurité du système. Outils de soutien
 - 19.4.3. Audits de sécurité des réseaux. Outils de soutien
- 19.5. Audits de sécurité technique sur l'internet et les appareils mobiles
 - 19.5.1. Audits de sécurité Internet. Outils de soutien
 - 19.5.2. Audit de sécurité des dispositifs mobiles. Outils de soutien
 - 19.5.3. Annexe 1. Structure du rapport exécutif et du rapport technique
 - 19.5.4. Annexe 2. Inventaire des outils
 - 19.5.5. Annexe 3. Méthodologie
- 19.6. Système de gestion de la sécurité de l'information
 - 19.6.1. Sécurité des SI: propriétés et facteurs d'influence
 - 19.6.2. Risques d'entreprise et gestion des risques: mise en œuvre des contrôles
 - 19.6.3. Système de gestion de la sécurité de l'information (SGSI): concept et facteurs critiques de succès
 - 19.6.4. ISMS - Modèle PDCA
 - 19.6.5. ISMS ISO-IEC 27001: Contexte organisationnel
 - 19.6.6. Contexte de l'organisation
 - 19.6.7. Leadership
 - 19.6.8. Planification

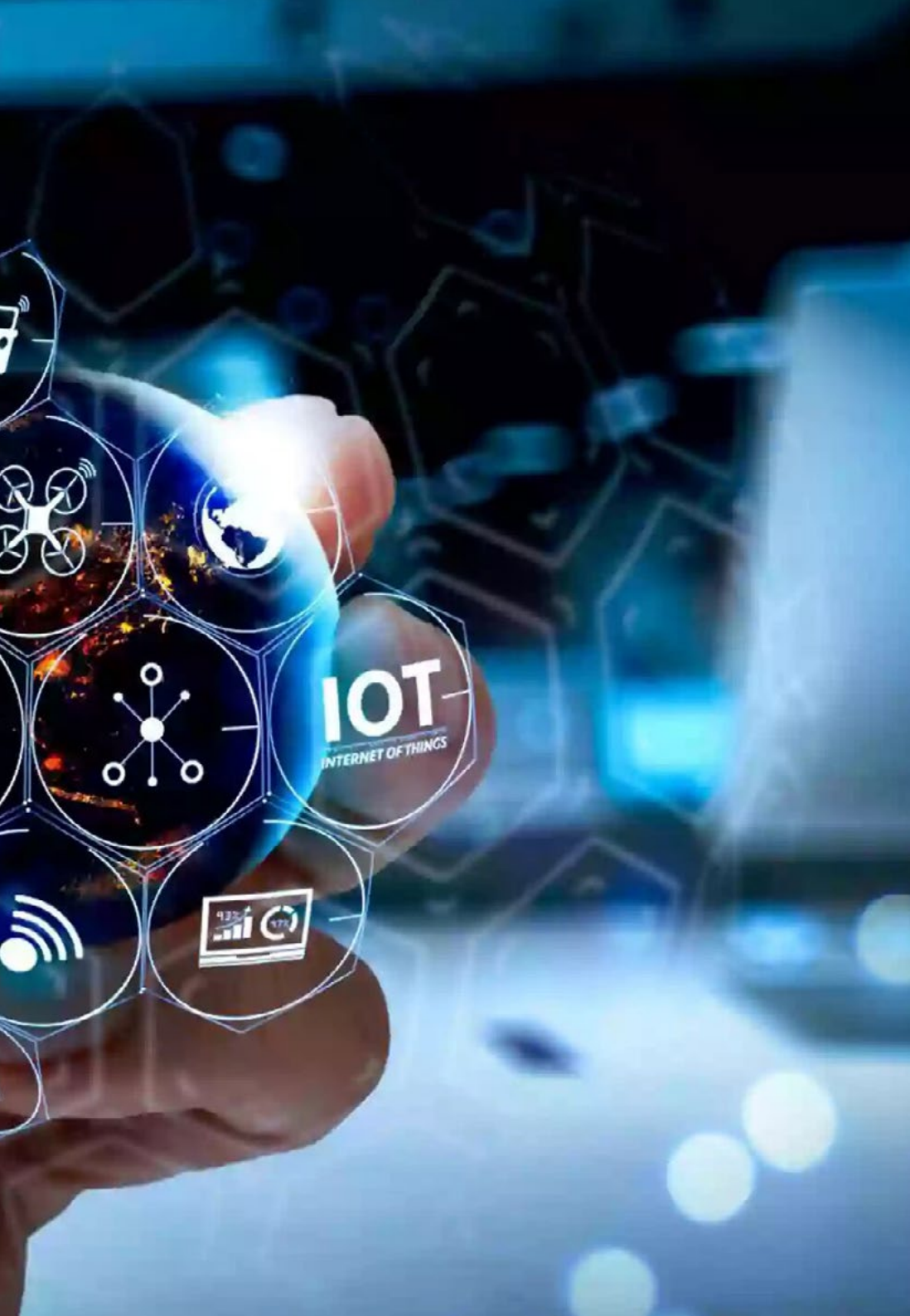
- 19.6.9. Soutien
- 19.6.10. Opération
- 19.6.11. Évaluation des performances
- 19.6.12. Amélioration
- 19.6.13. Annexe à l'ISO 27001/ISO-IEC 27002: objectifs et contrôles
- 19.6.14. Audit ISMS
- 19.7. Réalisation de l'audit
 - 19.7.1. Procédures
 - 19.7.2. Techniques
- 19.8. Traçabilité
 - 19.8.1. Méthodologie
 - 19.8.2. Analyse
- 19.9. Garde d'enfants
 - 19.9.1. Techniques
 - 19.9.2. Résultats
- 19.10. Rapport et présentation des preuves
 - 19.10.1. Types de rapports
 - 19.10.2. Analyse des données
 - 19.10.3. Présentation des preuves

Module 20. Sécurité des Applications Online

- 20.1. Vulnérabilités et problèmes de sécurité dans les applications online
 - 20.1.1. Introduction à la sécurité dans les applications online
 - 20.1.2. Les failles de sécurité dans la conception des applications web
 - 20.1.3. Les failles de sécurité dans la mise en œuvre des applications web
 - 20.1.4. Les failles de sécurité dans le déploiement des applications web
 - 20.1.5. Listes officielles de failles de sécurité
- 20.2. Politiques et normes relatives à la sécurité des applications online
 - 20.2.1. Piliers de la sécurité des applications online
 - 20.2.2. Politique de sécurité
 - 20.2.3. Système de gestion de la sécurité de l'information
 - 20.2.4. Cycle de développement sécurisé des Logiciels
 - 20.2.5. Normes pour la sécurité des applications

- 20.3. La sécurité dans la conception des applications web
 - 20.3.1. Introduction à la sécurité des applications web
 - 20.3.2. La sécurité dans la conception des applications web
- 20.4. Tester la sécurité online et la sûreté des applications web
 - 20.4.1. Analyse et test de la sécurité des applications web
 - 20.4.2. Sécurité dans le déploiement et la production d'applications web
- 20.5. Sécurité des services web
 - 20.5.1. Introduction à la sécurité des applications web
 - 20.5.2. Fonctions et technologies de sécurité des services web
- 20.6. Tester la sécurité et la protection en ligne des services web
 - 20.6.1. Évaluation de la sécurité des services web
 - 20.6.2. Protection en ligne. Firewalls et gateways XML
- 20.7. Hacking éthique, malware y forensic
 - 20.7.1. Piratage éthique
 - 20.7.2. Analyse de Malware
 - 20.7.3. Analyse Forense
- 20.8. Bonnes pratiques pour assurer la sécurité des applications
 - 20.8.1. Manuel de bonnes pratiques pour le développement d'applications online
 - 20.8.2. Manuel de bonnes pratiques pour la mise en œuvre des applications online
- 20.9. Erreurs courantes qui compromettent la sécurité des applications
 - 20.9.1. Erreurs de développement courantes
 - 20.9.2. Erreurs courantes en matière d'hébergement
 - 20.9.3. Erreurs courantes dans la production





“

*Un programme d'études complet
qui vous permettra de maîtriser
le domaine du Big Data et de
devenir un architecte de stratégie
d'entreprise performant”*

04

Objectifs pédagogiques

Le Mastère Avancé en Ingénierie et Qualité du Software vise à former des professionnels hautement qualifiés dans la conception, le développement et la gestion de systèmes Logiciels de haute qualité. En mettant l'accent sur la qualité et la gestion stratégique des projets technologiques, le programme favorise également la capacité à s'adapter à l'évolution rapide du secteur. Cela garantit que les étudiants sont non seulement préparés à relever les défis de l'avenir, mais qu'ils sont également capables de mener l'innovation dans le domaine des Logiciels.



“

Transformez vos idées en solutions technologiques efficaces grâce à l'art du génie Logiciel”



Objectifs généraux

- ♦ Développer des compétences avancées dans la conception, le développement et la maintenance de systèmes du Software complexes et évolutifs, en appliquant les meilleures pratiques et méthodologies de l'ingénierie du Software
- ♦ Former les étudiants à l'assurance qualité des logiciels, en leur fournissant des outils et des techniques pour garantir la fiabilité, la sécurité et la performance des solutions technologiques
- ♦ Favoriser le leadership dans la gestion de projets technologiques, en développant des compétences dans la gestion d'équipes pluridisciplinaires, la planification stratégique et la prise de décision dans des environnements dynamiques
- ♦ Promouvoir la capacité à s'adapter aux progrès technologiques rapides, en se spécialisant dans les nouveaux outils, techniques et tendances qui leur permettent de rester à la pointe de l'ingénierie du Software
- ♦ Développer des compétences en gestion de la qualité tout au long du cycle de vie du Software, de la planification initiale à la maintenance et à l'amélioration continue des systèmes
- ♦ Renforcer les compétences en matière de communication et de travail en équipe, essentielles pour collaborer efficacement avec différents stakeholders, gérer les attentes et garantir la réussite des projets technologiques



Améliorez vos compétences et devenez un leader dans la création de solutions technologiques de pointe





Objectifs spécifiques

Module 1. Qualité du Software. Niveaux de développement TRL

- ♦ Comprendre les différents niveaux de maturité technologique et leur relation avec la qualité du Software
- ♦ Évaluer le développement du Software à chaque étape du TRL et son impact sur la qualité du produit final

Module 2. Développement de projets Software. Documentation fonctionnelle et technique

- ♦ Développer des compétences pour créer une documentation fonctionnelle et technique claire et détaillée dans le cadre de projets de Software
- ♦ Analyser l'importance d'une documentation précise pour la gestion des projets et la qualité du Software

Module 3. Testing de Software. Automatisation des tests

- ♦ Développer des compétences pour concevoir et exécuter des tests automatisés sur des applications Software
- ♦ Mettre en œuvre des solutions de *testing* efficaces à l'aide d'outils d'automatisation des tests

Module 4. Méthodologies de gestion de projets de Software. Méthodologies Waterfall et Méthodologies Agile

- ♦ Analyser les différences entre les méthodologies Waterfall et Agile dans la gestion de projets Software
- ♦ Évaluer les avantages et les limites de chaque méthodologie en fonction du type de projet

Module 5. TDD (*Test Driven Development*). Conception de Logiciels pilotés par les tests

- ♦ Développer des compétences pour écrire des tests unitaires avant d'écrire le code de production
- ♦ Améliorer la qualité du Software en mettant en œuvre le TDD dans le processus de développement

Module 6. DevOps. Gestion de Qualité du Software

- ♦ Explorer le concept de DevOps et son impact sur l'amélioration continue de la qualité du Software
- ♦ Apprenez à intégrer les pratiques de développement et d'exploitation pour obtenir un cycle de vie du Software plus agile et plus efficace

Module 7. DevOps et intégration continue. Solutions pratiques avancées en matière de développement de Software

- ♦ Approfondir les techniques avancées d'intégration continue dans le cadre DevOps
- ♦ Mettre en œuvre des solutions pratiques d'intégration continue pour automatiser le processus de développement et de déploiement du Software

Module 8. Conception de Bases de Données (DB). Standardisation et performance. Qualité du Software

- ♦ Analyser les principes de conception des bases de données, y compris la normalisation et l'optimisation des performances
- ♦ Comprendre comment une conception correcte de la base de données contribue à la qualité du Software

Module 9. Conception d'architectures évolutives. L'architecture dans le cycle de vie des Logiciels

- ♦ Découvrez les principes de conception des architectures évolutives et leur impact sur la qualité et les performances du Software
- ♦ Évaluer différents modèles d'architecture pour des applications du Software évolutives

Module 10. Critères de Qualité ISO, IEC 9126. Mesures de Qualité du Software

- ♦ Comprendre les critères de Qualité du Logiciel selon ces normes et comment les appliquer
- ♦ Mettre en œuvre des mesures de qualité afin d'évaluer et d'améliorer en permanence les applications de Software

Module 11. Méthodologies, développement et qualité du Génie Logiciel

- ♦ Étudier en profondeur les méthodologies les plus utilisées en Ingénierie du Software et leur relation avec la Qualité
- ♦ Développer une approche intégrée combinant le développement, les tests et la qualité dans les projets de Software

Module 12. Gestion de projets de Software

- ♦ Développer des compétences en matière de gestion de projets de Software, de la planification à l'exécution
- ♦ Gérer les ressources, le temps et les risques associés aux projets de développement de Software

Module 13. Plateformes de développement du Logiciel

- ♦ Explorer les différentes plates-formes de développement de logiciels et leurs caractéristiques
- ♦ Évaluer les plates-formes de développement en fonction de leurs capacités, de leur flexibilité et de leur compatibilité avec différents projets

Module 14. Informatique client web

- ♦ Analyser comment l'informatique côté client est utilisée dans le développement d'applications web
- ♦ Développer des applications qui exploitent l'informatique côté client pour améliorer l'interaction et les performances

Module 15. Informatique des Serveurs Web

- ♦ Exploration des technologies et des techniques utilisées pour les serveurs web
- ♦ Comprendre le traitement des données, la logique commerciale et la gestion des utilisateurs sur le serveur





Module 16. Gestion de la sécurité

- ♦ Évaluer les risques de sécurité dans les applications et mettre en œuvre des mesures préventives
- ♦ Mettre en œuvre des contrôles de sécurité à tous les stades du cycle de vie du Software

Module 17. Sécurité du Software

- ♦ Explorer les meilleures pratiques en matière de sécurité dans le développement de Software
- ♦ Analyser les vulnérabilités les plus courantes des logiciels et apprendre à les atténuer

Module 18. Administration des Serveurs Web

- ♦ Comprendre le rôle des serveurs web dans le développement et le déploiement des applications
- ♦ Développer des compétences dans l'administration et la maintenance des serveurs web

Module 19. Audit de Sécurité

- ♦ Évaluer la sécurité des systèmes au moyen d'audits et de tests de pénétration
- ♦ Mettre en œuvre des processus d'audit continu pour améliorer la sécurité du Software

Module 20. Sécurité des Applications Online

- ♦ Mise en œuvre de solutions pour protéger les applications en ligne contre les menaces externes et internes
- ♦ Mettre en place des politiques de sécurité et d'audit pour garantir l'intégrité des applications en ligne

05

Opportunités de carrière

Le diplômé sera hautement qualifié pour assumer des rôles de leadership dans le développement, la mise en œuvre et la gestion de Logiciels de haute qualité. Grâce à votre spécialisation avancée dans des domaines clés tels que l'architecture Logicielle, la sécurité des applications en ligne, la gestion de projets technologiques et les méthodologies agiles, vous serez en mesure de diriger des équipes de développement. En outre, votre préparation vous permettra d'occuper des postes de direction dans des projets de Software, tandis que votre capacité à innover et à diriger des équipes pluridisciplinaires vous permettra de relever les défis de l'environnement numérique et de contribuer de manière significative au succès de toute organisation.



“

TECH Euromed University vous donne la possibilité de réaliser vos rêves dans la discipline la plus passionnante, qui transforme les idées en produits tangibles capables d'améliorer la vie des gens”

Profil des diplômés

Le profil du Mastère Avancé en Ingénierie et Qualité du Software est orienté vers la formation de professionnels hautement qualifiés, capables de diriger et de gérer des projets technologiques à fort impact. Assurant la qualité, la sécurité et l'efficacité dans toutes les phases du développement du Software, vous maîtriserez à la fois les méthodologies agiles et traditionnelles. En outre, vous serez en mesure de concevoir et de développer des systèmes de Software évolutifs, efficaces et sécurisés, en appliquant des normes de qualité internationales et des méthodologies avancées telles que DevOps et l'intégration continue.

Devenez un expert en réussite professionnelle grâce à la plus grande université numérique du monde.

- ♦ **Sécurité du Software et des systèmes:** Compétence dans la mise en œuvre de pratiques de sécurité avancées, y compris la protection des données et la gestion de la vulnérabilité dans les applications en ligne.
- ♦ **Assurance qualité du Software:** Capacité à appliquer les normes internationales (ISO, IEC 9126) et les outils de test automatisés pour garantir la fiabilité et la performance du Software
- ♦ **Développement d'architectures évolutives:** Capacité à concevoir et à construire des systèmes de Software qui peuvent évoluer et s'adapter aux demandes du marché sans compromettre la qualité et la sécurité
- ♦ **Intégration continue et DevOps:** Capacité à mettre en œuvre et à gérer des processus d'intégration continue, garantissant une livraison efficace et transparente de nouvelles fonctionnalités du Software



Après avoir obtenu le titre de Mastère Avancé, vous serez en mesure d'utiliser vos connaissances et vos compétences dans les postes suivants:

1. **Directeur de la Technologie (CTO):** Responsable de l'orientation stratégique de la technologie dans une entreprise, il dirige les équipes de développement et supervise la mise en œuvre de solutions technologiques innovantes.
2. **Responsable de la Qualité du Logiciel:** Il est chargé de superviser et de veiller à ce que les processus et les produits de Software respectent les normes de qualité établies, de diriger les tests de Software et les initiatives d'amélioration continue
3. **Architecte Logiciel:** Concepteur principal de la structure et de l'architecture de systèmes de Software complexes, en veillant à ce qu'ils soient évolutifs, sûrs et efficaces
4. **Leader d'un Projet de Software:** Responsable de la planification, de l'exécution et de la livraison de projets de Software, de la gestion d'équipes pluridisciplinaires et de la garantie que les projets sont achevés dans les délais, dans le respect du budget et des normes de qualité appropriées
5. **Spécialiste de la Sécurité Informatique:** Responsable de la protection des applications, de l'infrastructure et des données contre les cybermenaces, en mettant en œuvre des stratégies et des politiques de sécurité efficaces
6. **Auditeur de la Sécurité du Logiciel:** Effectuer des audits complets afin d'identifier les vulnérabilités des applications et des systèmes et proposer des améliorations et des solutions pour garantir la sécurité du Software



Si vous voulez faire la différence dans le monde numérique, choisissez cette voie qui vous spécialise en tant qu'expert dans la création de Software de qualité"

06

Méthodologie d'étude

TECH Euromed University est la première au monde à combiner la méthodologie des **case studies** avec **Relearning**, un système d'apprentissage 100% en ligne basé sur la répétition guidée.

Cette stratégie d'enseignement innovante est conçue pour offrir aux professionnels la possibilité d'actualiser leurs connaissances et de développer leurs compétences de manière intensive et rigoureuse. Un modèle d'apprentissage qui place l'étudiant au centre du processus académique et lui donne le rôle principal, en s'adaptant à ses besoins et en laissant de côté les méthodologies plus conventionnelles.



“

*TECH Euromed University vous prépare
à relever de nouveaux défis dans des
environnements incertains et à réussir
votre carrière”*

L'étudiant: la priorité de tous les programmes de TECH Euromed University

Dans la méthodologie d'étude de TECH Euromed University, l'étudiant est le protagoniste absolu.

Les outils pédagogiques de chaque programme ont été sélectionnés en tenant compte des exigences de temps, de disponibilité et de rigueur académique que demandent les étudiants d'aujourd'hui et les emplois les plus compétitifs du marché.

Avec le modèle éducatif asynchrone de TECH Euromed University, c'est l'étudiant qui choisit le temps qu'il consacre à l'étude, la manière dont il décide d'établir ses routines et tout cela dans le confort de l'appareil électronique de son choix. L'étudiant n'a pas besoin d'assister à des cours en direct, auxquels il ne peut souvent pas assister. Les activités d'apprentissage se dérouleront à votre convenance. Vous pouvez toujours décider quand et où étudier.

“

À TECH Euromed University, vous n'aurez PAS de cours en direct (auxquelles vous ne pourrez jamais assister)”



Les programmes d'études les plus complets au niveau international

TECH Euromed University se caractérise par l'offre des itinéraires académiques les plus complets dans l'environnement universitaire. Cette exhaustivité est obtenue grâce à la création de programmes d'études qui couvrent non seulement les connaissances essentielles, mais aussi les dernières innovations dans chaque domaine.

Grâce à une mise à jour constante, ces programmes permettent aux étudiants de suivre les évolutions du marché et d'acquérir les compétences les plus appréciées par les employeurs. Ainsi, les diplômés de TECH Euromed University reçoivent une préparation complète qui leur donne un avantage concurrentiel significatif pour progresser dans leur carrière.

De plus, ils peuvent le faire à partir de n'importe quel appareil, PC, tablette ou smartphone.

“

Le modèle de TECH Euromed University est asynchrone, de sorte que vous pouvez étudier sur votre PC, votre tablette ou votre smartphone où vous voulez, quand vous voulez et aussi longtemps que vous le voulez”

Case studies ou Méthode des cas

La méthode des cas est le système d'apprentissage le plus utilisé par les meilleures écoles de commerce du monde. Développée en 1912 pour que les étudiants en Droit n'apprennent pas seulement le droit sur la base d'un contenu théorique, sa fonction était également de leur présenter des situations réelles et complexes. De cette manière, ils pouvaient prendre des décisions en connaissance de cause et porter des jugements de valeur sur la manière de les résoudre. Elle a été établie comme méthode d'enseignement standard à Harvard en 1924.

Avec ce modèle d'enseignement, ce sont les étudiants eux-mêmes qui construisent leurs compétences professionnelles grâce à des stratégies telles que *Learning by doing* ou le *Design Thinking*, utilisées par d'autres institutions renommées telles que Yale ou Stanford.

Cette méthode orientée vers l'action sera appliquée tout au long du parcours académique de l'étudiant avec TECH Euromed University. Vous serez ainsi confronté à de multiples situations de la vie réelle et devrez intégrer des connaissances, faire des recherches, argumenter et défendre vos idées et vos décisions. Il s'agissait de répondre à la question de savoir comment ils agiraient lorsqu'ils seraient confrontés à des événements spécifiques complexes dans le cadre de leur travail quotidien.



Méthode Relearning

À TECH Euromed University, les *case studies* sont complétées par la meilleure méthode d'enseignement 100% en ligne: le *Relearning*.

Cette méthode s'écarte des techniques d'enseignement traditionnelles pour placer l'apprenant au centre de l'équation, en lui fournissant le meilleur contenu sous différents formats. De cette façon, il est en mesure de revoir et de répéter les concepts clés de chaque matière et d'apprendre à les appliquer dans un environnement réel.

Dans le même ordre d'idées, et selon de multiples recherches scientifiques, la répétition est le meilleur moyen d'apprendre. C'est pourquoi TECH Euromed University propose entre 8 et 16 répétitions de chaque concept clé au sein d'une même leçon, présentées d'une manière différente, afin de garantir que les connaissances sont pleinement intégrées au cours du processus d'étude.

Le Relearning vous permettra d'apprendre plus facilement et de manière plus productive tout en développant un esprit critique, en défendant des arguments et en contrastant des opinions: une équation directe vers le succès.



Un Campus Virtuel 100% en ligne avec les meilleures ressources didactiques

Pour appliquer efficacement sa méthodologie, TECH Euromed University se concentre à fournir aux diplômés du matériel pédagogique sous différents formats: textes, vidéos interactives, illustrations et cartes de connaissances, entre autres. Tous ces supports sont conçus par des enseignants qualifiés qui axent leur travail sur la combinaison de cas réels avec la résolution de situations complexes par la simulation, l'étude de contextes appliqués à chaque carrière professionnelle et l'apprentissage basé sur la répétition, par le biais d'audios, de présentations, d'animations, d'images, etc.

Les dernières données scientifiques dans le domaine des Neurosciences soulignent l'importance de prendre en compte le lieu et le contexte d'accès au contenu avant d'entamer un nouveau processus d'apprentissage. La possibilité d'ajuster ces variables de manière personnalisée aide les gens à se souvenir et à stocker les connaissances dans l'hippocampe pour une rétention à long terme. Il s'agit d'un modèle intitulé *Neurocognitive context-dependent e-learning* qui est sciemment appliqué dans le cadre de ce diplôme d'université.

D'autre part, toujours dans le but de favoriser au maximum les contacts entre mentors et mentorés, un large éventail de possibilités de communication est offert, en temps réel et en différé (messagerie interne, forums de discussion, service téléphonique, contact par courrier électronique avec le secrétariat technique, chat et vidéoconférence).

De même, ce Campus Virtuel très complet permettra aux étudiants TECH Euromed University d'organiser leurs horaires d'études en fonction de leurs disponibilités personnelles ou de leurs obligations professionnelles. De cette manière, ils auront un contrôle global des contenus académiques et de leurs outils didactiques, mis en fonction de leur mise à jour professionnelle accélérée.



Le mode d'étude en ligne de ce programme vous permettra d'organiser votre temps et votre rythme d'apprentissage, en l'adaptant à votre emploi du temps"

L'efficacité de la méthode est justifiée par quatre acquis fondamentaux:

1. Les étudiants qui suivent cette méthode parviennent non seulement à assimiler les concepts, mais aussi à développer leur capacité mentale au moyen d'exercices pour évaluer des situations réelles et appliquer leurs connaissances.
2. L'apprentissage est solidement traduit en compétences pratiques ce qui permet à l'étudiant de mieux s'intégrer dans le monde réel.
3. L'assimilation des idées et des concepts est rendue plus facile et plus efficace, grâce à l'utilisation de situations issues de la réalité.
4. Le sentiment d'efficacité de l'effort investi devient un stimulus très important pour les étudiants, qui se traduit par un plus grand intérêt pour l'apprentissage et une augmentation du temps passé à travailler sur le cours.

La méthodologie universitaire la mieux évaluée par ses étudiants

Les résultats de ce modèle académique innovant sont visibles dans les niveaux de satisfaction générale des diplômés de TECH Euromed University.

L'évaluation par les étudiants de la qualité de l'enseignement, de la qualité du matériel, de la structure et des objectifs des cours est excellente. Sans surprise, l'institution est devenue l'université la mieux évaluée par ses étudiants sur la plateforme d'évaluation Global Score, avec une note de 4,9 sur 5.

Accédez aux contenus de l'étude depuis n'importe quel appareil disposant d'une connexion Internet (ordinateur, tablette, smartphone) grâce au fait que TECH Euromed University est à la pointe de la technologie et de l'enseignement.

Vous pourrez apprendre grâce aux avantages offerts par les environnements d'apprentissage simulés et à l'approche de l'apprentissage par observation: le Learning from an expert.



Ainsi, le meilleur matériel pédagogique, minutieusement préparé, sera disponible dans le cadre de ce programme:



Matériel didactique

Tous les contenus didactiques sont créés par les spécialistes qui enseignent les cours. Ils ont été conçus en exclusivité pour le programme afin que le développement didactique soit vraiment spécifique et concret.

Ces contenus sont ensuite appliqués au format audiovisuel afin de mettre en place notre mode de travail en ligne, avec les dernières techniques qui nous permettent de vous offrir une grande qualité dans chacune des pièces que nous mettrons à votre service.



Pratique des aptitudes et des compétences

Vous effectuerez des activités visant à développer des compétences et des aptitudes spécifiques dans chaque domaine. Pratiques et dynamiques permettant d'acquérir et de développer les compétences et les capacités qu'un spécialiste doit acquérir dans le cadre de la mondialisation dans laquelle nous vivons.



Résumés interactifs

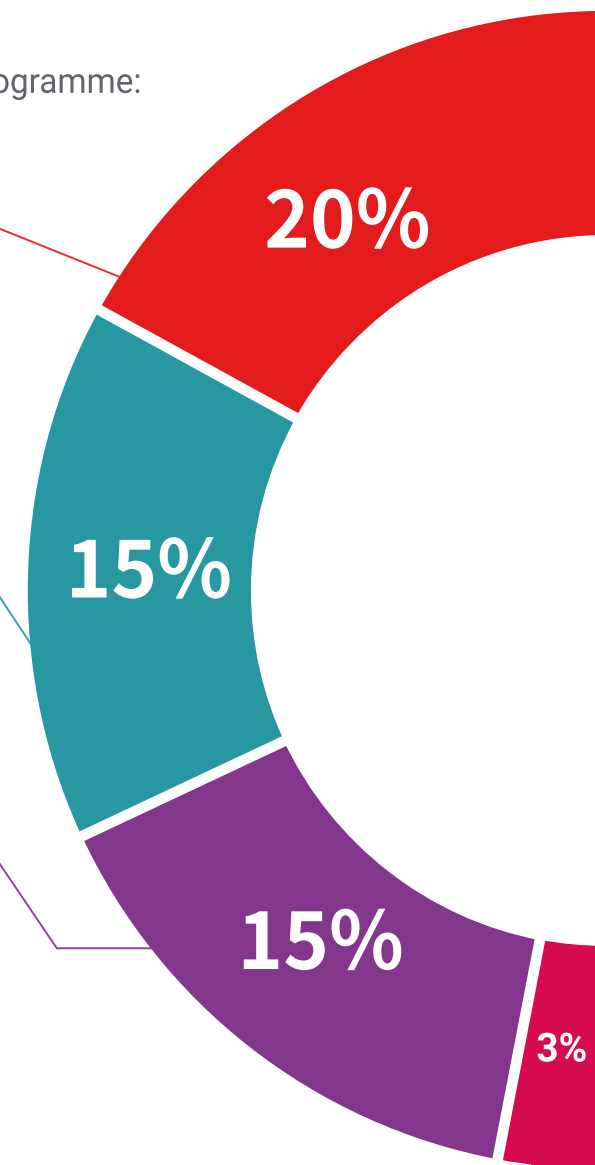
Nous présentons les contenus de manière attrayante et dynamique dans des dossiers multimédias qui incluent de l'audio, des vidéos, des images, des diagrammes et des cartes conceptuelles afin de consolider les connaissances.

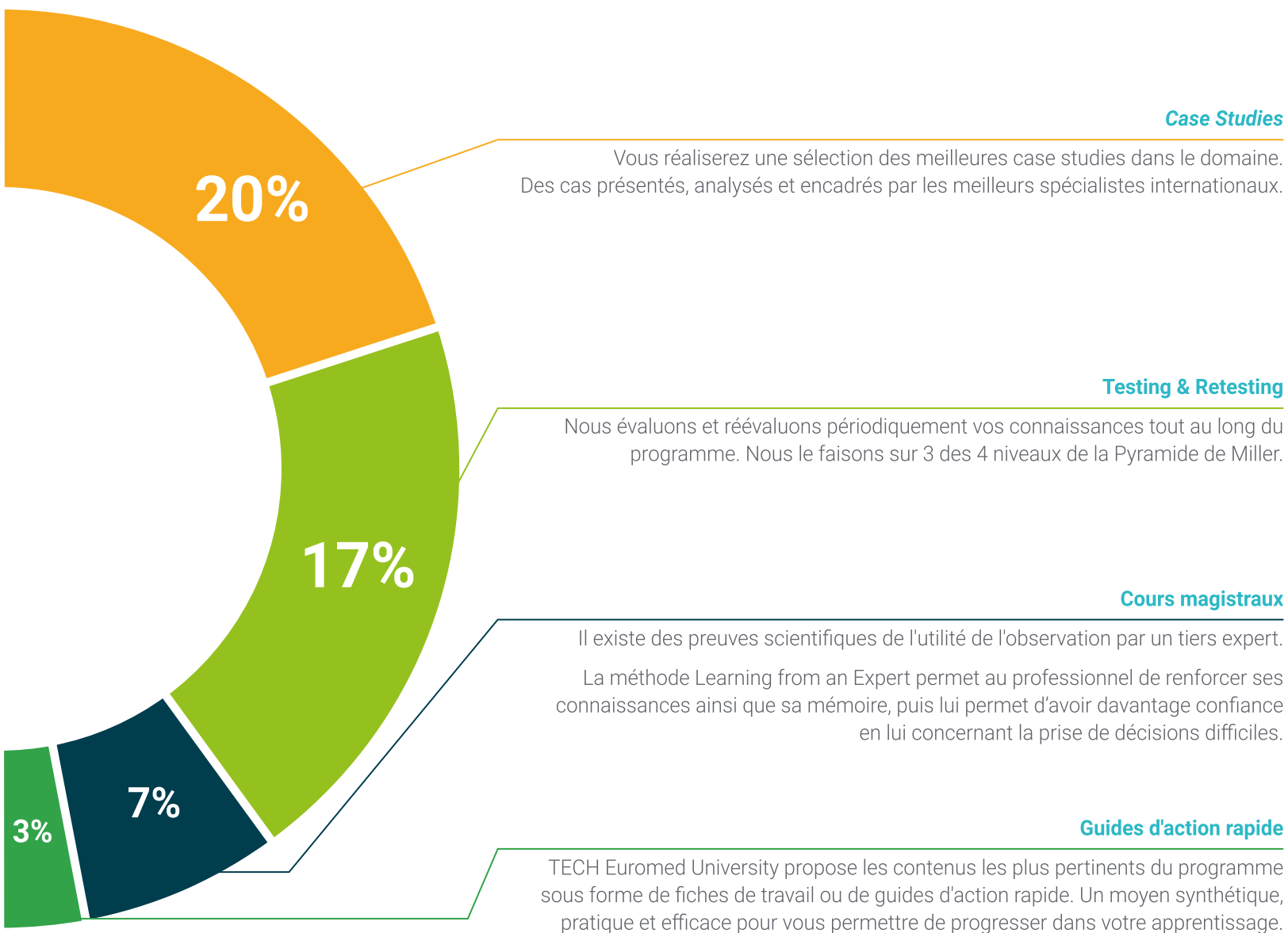
Ce système éducatif unique de présentation de contenu multimédia a été récompensé par Microsoft en tant que «European Success Story».



Lectures complémentaires

Articles récents, documents de consensus, guides internationaux, etc... Dans notre bibliothèque virtuelle, vous aurez accès à tout ce dont vous avez besoin pour compléter votre formation.





07

Corps Enseignant

La gestion et l'enseignement de ce Mastère Avancé en Ingénierie et Qualité du Software sont assurés par une équipe d'experts en ingénierie possédant de nombreuses années d'expérience dans la gestion et le développement de projets techniques et spécialisés. Son expérience professionnelle confère à ce diplôme un plus de qualité qui se traduira par une meilleure contextualisation du contenu par le diplômé, ainsi que par la mise en œuvre de cas pratiques réels et simulés dans l'expérience académique, mais toujours dans le but d'offrir un programme 100% en ligne qui soit dynamique, avant-gardiste et basé sur la réalité immédiate du secteur.



“

*Rejoignez la carrière la plus demandée
pour les experts en ingénierie et qualité
du Software et saisissez l'opportunité de
devenir l'un d'entre eux”*

Directeur Invité International

Darren Pulsipher est un **architecte logiciel** très expérimenté, un innovateur avec une expérience internationale distinguée dans le **développement de software et de firmware**. En fait, il possède des compétences très développées en matière de **communication**, de **gestion de projets et d'affaires**, qui lui ont permis de diriger d'importantes initiatives au niveau mondial.

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 modernes**, des **processus et des technologies** 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**, afin de développer un **outil d'agrégation et de diagnostic des réseaux sociaux** basé sur un **Software AS A Service (SaaS)**, utilisant les technologies du **Big Data** et du **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**. Il a dirigé des équipes aux **Etats-Unis** et en **Chine** dans la gestion de grands projets et a restructuré des divisions commerciales pour une intégration réussie. Il a également travaillé en tant que **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 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**. Des connaissances qu'il a partagées grâce au **podcast** et à la **lettre d'information hebdomadaires "Embracing Digital Transformation"**, qu'il a produits et présentés, et qui aident 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, Arkansas
- Directeur Senior de l'Ingénierie chez Dell Technologies, Arkansas
- Directeur des Technologies de l'Information (*Chief Information Officer*) chez XanGo, Utah
- Architecte Senior chez Cadence Design Systems, Californie
- Gestionnaire Principal des Processus de Projet chez Lucent Technologies, Californie
- Ingénieur Logiciel chez Cemax-Icon, Californie
- Ingénieur Logiciel à ISG Technologies, Canada
- MBA en Gestion des Technologies de l'Université de Phoenix
- Licence en Sciences Informatiques et Ingénierie Électrique de l'Université Brigham Young



Grâce à TECH Euromed University, vous pourrez apprendre avec les meilleurs professionnels du monde"

Direction



M. Molina Molina, Jerónimo

- ♦ Responsable de l'Intelligence Artificielle à Helphone
- ♦ AI Engineer & Software Architect en NASSAT Internet par Satellite en Mouvement
- ♦ Consultant Senior chez Hexa Engineer
- ♦ Introduceur de l'Intelligence Artificielle (ML et CV)
- ♦ Expert en Solutions Basées sur l'Intelligence Artificielle, dans le domaine de la Vision par Ordinateur, ML/DL et NLP
- ♦ Expert Universitaire en Création et Développement d'Entreprises à Bancaixa et Fundeun
- ♦ Ingénieur en Informatique de l'Université de Alicante
- ♦ Master en Intelligence artificielle, Université Catholique de Ávila
- ♦ MBA Executive au Forum Européen des Campus d'Affaires

Professeurs

Mme Rodríguez Míguez, Cándida

- ♦ Junior Application Developer à Getronics
- ♦ Cofondatrice et City Leader du réseau Galicia AI
- ♦ Ingénieure Logiciel Junior chez Indra
- ♦ Développeuse web chez EDISA
- ♦ Diplôme en Ingénierie Informatique de l'Université de Vigo
- ♦ Master Universitaire en Ingénierie Informatique de l'Université de Vigo

M. Pi Morell, Oriol

- ♦ Analyste Fonctionnel à Fihoca
- ♦ Product Owner Hébergement et Email chez CDmon
- ♦ Analyste Fonctionnel et Ingénieur Logiciel à Atmira et CapGemini
- ♦ Enseignant chez Capgemini, Forms Capgemini et Atmira
- ♦ Licence en Génie Technique en Informatique de Getion, Université Autonome de Barcelone
- ♦ Master en Intelligence artificielle, Université Catholique de Ávila
- ♦ MBA en Gestion et Administration des Entreprises par IMF Smart Education
- ♦ Master en Gestion des Systèmes d'Information par IMF Smart Education
- ♦ Diplôme en Modèles de Conception par l'Université Ouverte de Catalogne (UOC)

M. Martínez Calvo, Francisco Javier

- ♦ Ingénieur Technique Industriel Spécialisé dans l'Électricité et l'Électronique
- ♦ Technicien Logiciel chez HEXA Ingenieros
- ♦ Développeur Senior .Net / Architecte de Solutions .Net chez Everis
- ♦ Analyste/Architecte *Logiciel* dans LaLiga
- ♦ Ingénieur On-site de Microsoft chez BBVA
- ♦ Consultant IT-Technique *Freelance*
- ♦ Formateur en Visual Studio, SqlServer, CCNA (Routeurs et Switchs Cisco), Programmation Web PHP et .Net dans plusieurs centres (Salesianos, Maforem, Dreamsoft)
- ♦ Ingénieur Technique Industriel avec Spécialisation en Électricité, Électronique Industrielle
- ♦ Master Cibernos en .NET, MCAD
- ♦ Master Eidos en Programmation Avancée, Niveau Expert
- ♦ Master Web avec les Certifications Dreamweaver, Fireworks, Flash et ActionScript, Versions MX

M. Tenrero Morán, Marcos

- ♦ Ingénieur DevOps chez Allot Communications
- ♦ Manager en Gestion du Cycle de Vie des Applications chez Cegid Meta4
- ♦ Ingénieur en Automatisation QA chez Cegid Meta4
- ♦ Master en Développement d'Applications Professionnelles pour Android par l'Université Galileo. Guatemala
- ♦ Master en Développement de Services Cloud, Node.Js, JavaScript, HTML5 par l'Université Polytechnique de Madrid
- ♦ Développement Web avec Angular-CLI (4), Ionic et Node.Js, Meta4 de l'Université Rey Juan Carlos
- ♦ Diplôme en Ingénierie de ordinateur de l'Université Rey Juan Carlos

Mme Acebes Tamargo, Patricia

- ♦ Consultante Spécialisée en Big Data
- ♦ Département des Opérations, travaille avec Elasticsearch et Kivana dans Sirt
- ♦ Chercheuse en Ligne *Facteur Humain* et *Applications IA* à CTIC Centre Technologique
- ♦ Chercheuse en Ligne Business Unit au CTIC Centre Technologique
- ♦ Département Santé Numérique et Vieillessement Actif au CTIC Centre Technologique
- ♦ Département de Science des Données au CTIC Centre Technologique
- ♦ Doctorat en Informatique et Intelligence Artificielle de l'Université Polytechnique de Valence
- ♦ Licence en Économie de l'Université d'Oviedo
- ♦ Master en Analyse de Données UCJC
- ♦ Master en Recherche en Intelligence Artificielle de l'UNED
- ♦ Master en *Blockchain*, *Smart Contracts* et Cryptocurrencies de l'Université d'Alcalá
- ♦ Diplôme d'Études Supérieures en Ingénierie de la *Blockchain* de l'EADA
- ♦ Master en Économie, Instruments, Analyse Économique de l'Université d'Oviedo
- ♦ Master en Fiscalité du Collège des Économistes

08 Diplôme

Le Mastère Avancé en Ingénierie et Qualité du Software garantit, outre la formation la plus rigoureuse et la plus actualisée, l'accès à un diplôme de Mastère Avancé délivré par TECH Global University.



“

*Terminez ce programme avec succès
et recevez votre diplôme sans avoir à
vous soucier des déplacements ou des
formalités administratives”*

Le programme du **Mastère Avancé en Ingénierie et Qualité du Software** est le programme le plus complet sur la scène académique actuelle. Après avoir obtenu leur diplôme, les étudiants recevront un diplôme d'université délivré par TECH Global University et un autre par Université Euromed de Fès.

Ces diplômes de formation continue et d'actualisation professionnelle de TECH Global University et d'Université Euromed de Fès garantissent l'acquisition de compétences dans le domaine de la connaissance, en accordant une grande valeur curriculaire à l'étudiant qui réussit les évaluations et accrédite le programme après l'avoir suivi dans son intégralité.

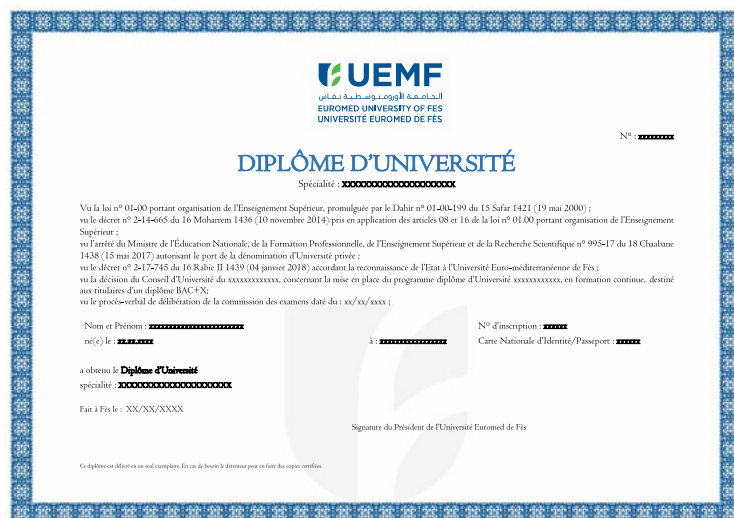
Ce double certificat, de la part de deux institutions universitaires de premier plan, représente une double récompense pour une formation complète et de qualité, assurant à l'étudiant l'obtention d'une certification reconnue au niveau national et international. Ce mérite académique vous positionnera comme un professionnel hautement qualifié, prêt à relever les défis et à répondre aux exigences de votre secteur professionnel.

Diplôme: **Mastère Avancé en Ingénierie et Qualité du Software**

Modalité: **en ligne**

Durée: **2 ans**

Accréditation: **120 ECTS**



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



Mastère Avancé Ingénierie et Qualité du Software

- » Modalité: en ligne
- » Durée: 2 ans
- » Qualification: TECH Euromed University
- » Accréditation: 120 ECTS
- » Horaire: à votre rythme
- » Examens: en ligne

Mastère Avancé

Ingénierie et Qualité du Software