Weiterbildender Masterstudiengang mit Spezialisierung

Softwaretechnik und -qualität

Akkredditierung/Mitgliedschaft

ASEE American Society for Engineering Education

tech global university

tech global university

Weiterbildender Masterstudiengang mit Spezialisierung

Softwaretechnik und -qualität

- Modalität: **online**
- Dauer: **2 Jahre**
- Qualifizierung: **TECH Global University**
- Akkreditierung: **120 ECTS**
- Zeitplan: **in Ihrem eigenen Tempo**
- Prüfungen: **online**

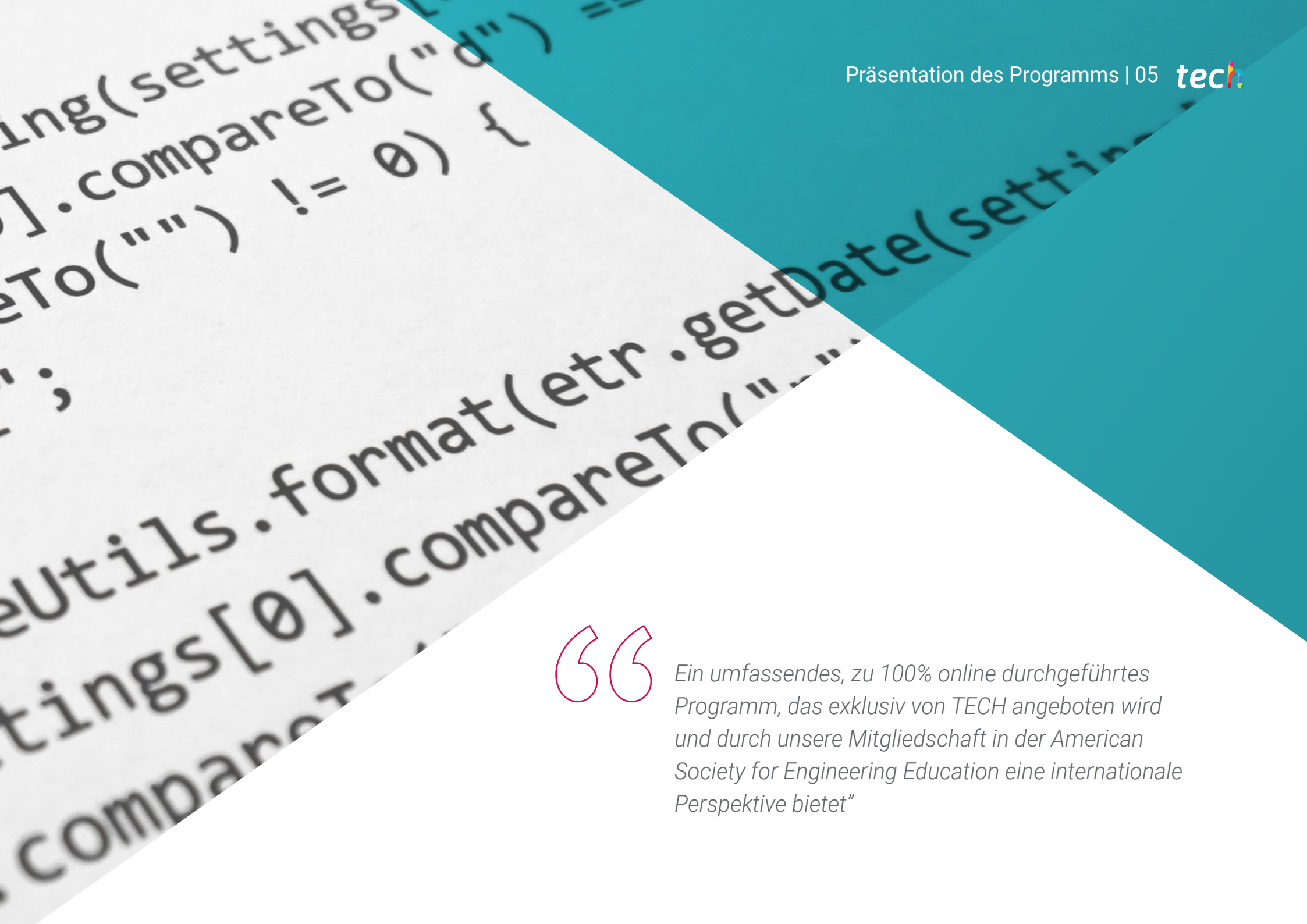
Internetzugang: www.techtitute.com/de/informatik/weiterbildender-masterstudiengang-spezialisierung/weiterbildender-masterstudiengang-spezialisierung-softwaretechnik-qualitat

Index

01

Präsentation des Programms

Softwaretechnik ist zum Eckpfeiler der digitalen Transformation geworden. Heute sind alle Branchen auf Technologielösungen angewiesen, um Prozesse zu optimieren, das Kundenerlebnis zu verbessern und wettbewerbsfähig zu bleiben. Die Softwarequalität wiederum sorgt dafür, dass diese Lösungen zuverlässig, skalierbar und sicher sind. Diese Disziplin ist ein Zweig des Ingenieurwesens, der technisches und Managementwissen kombiniert, um sicherzustellen, dass die entwickelten Produkte und Systeme funktional und nachhaltig sind. Aus diesem Grund geht dieser Studiengang über die reine Programmierung hinaus und konzentriert sich auf den gesamten Lebenszyklus der Software, von der ersten Konzeption bis zur Wartung und Weiterentwicklung des Systems. Das Hauptziel besteht darin, den Studenten eine einzigartige akademische Möglichkeit zu bieten, die ihnen Informationen auf dem neuesten Stand der Technik vermittelt. TECH hat diesen multidisziplinären und 100%igen Online-Studiengang entwickelt, der von den Grundlagen der Softwaretechnik bis hin zu den neuesten Trends in der agilen Methodik reicht.



Ein umfassendes, zu 100% online durchgeführtes Programm, das exklusiv von TECH angeboten wird und durch unsere Mitgliedschaft in der American Society for Engineering Education eine internationale Perspektive bietet"

Softwarequalität stellt sicher, dass Systeme nicht nur die funktionalen Anforderungen erfüllen, sondern auch intuitiv, sicher und langfristig tragfähig sind. Dies ist besonders in kritischen Sektoren wie dem Finanzwesen, dem Gesundheitswesen oder dem Verkehrswesen von Bedeutung, wo Ausfälle schwerwiegende Folgen haben können. Darüber hinaus wird durch die Priorisierung von Qualität sichergestellt, dass sich Unternehmen schnell an den ständigen technologischen Fortschritt anpassen und effektiv auf die wachsenden Marktanforderungen reagieren können.

Durch den Einsatz von Methoden wie der agilen Entwicklung, *DevOps* und der Umsetzung internationaler Qualitätsstandards garantiert die Softwaretechnik die Lieferung von Produkten in kürzerer Zeit. Darüber hinaus werden Kostenkontrolle und ein Qualitätsniveau, das kritische Fehler minimiert, durch die Integration neuer Technologien wie künstliche Intelligenz, *Cloud Computing* und Cybersicherheit noch verstärkt. In diesem Zusammenhang zielt das von TECH konzipierte Programm auf die Fortbildung hochqualifizierter Fachleute in den Bereichen Konzeption, Entwicklung, Management und Qualitätssicherung von Software ab. Um die erforderlichen Kompetenzen zu erwerben, umfasst der Lehrplan des weiterbildenden Masterstudiengangs die aktuellsten Konzepte des technologischen Projektmanagements und des strategischen Managements. Dieser Ansatz stellt einen Mehrwert für Ingenieure dar, die bereits eine verantwortungsvolle Position innehaben und ihr Wissen aktualisieren möchten, sowie für diejenigen, die zum ersten Mal in diesem Bereich Teams und Projekte leiten wollen.

Einer der Hauptvorteile dieses Programms besteht darin, dass es zu 100% online durchgeführt wird, wodurch die Notwendigkeit entfällt, zu reisen und sich an bestimmte Zeitpläne anzupassen. Darüber hinaus profitieren die Studenten von der *Relearning*-Lernmethode, die sich an ihr Lerntempo anpasst. Dieser flexible Ansatz ist sehr nützlich, da er es den Studenten ermöglicht, ihre täglichen Verpflichtungen, ob beruflich oder familiär, effizient zu organisieren und so eine vollständige Entwicklung zu erreichen.

Da TECH Mitglied der **American Society for Engineering Education (ASEE)** ist, haben ihre Studenten kostenlosen Zugang zu jährlichen Konferenzen und regionalen Workshops, die ihre Fortbildung im Ingenieurwesen bereichern. Darüber hinaus genießen sie Online-Zugang zu Fachpublikationen wie Prism und dem Journal of Engineering Education, wodurch sie ihre akademische Entwicklung stärken und ihr berufliches Netzwerk auf internationaler Ebene erweitern können.

Dieser **Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik und -qualität** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- Die Entwicklung von Fallstudien, die von Experten in Informatik präsentiert werden
- Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren wissenschaftlichen und praktischen Informationen
- Praktische Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens genutzt werden kann
- Sein besonderer Schwerpunkt liegt auf innovativen Methoden im Bereich der Softwaretechnik und -qualität
- Theoretische Lektionen, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- Die Verfügbarkeit des Zugangs zu Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss

> *Bei TECH werden Sie nicht nur lernen, Software zu entwickeln, sondern auch Systeme zu schaffen, die das Leben von Menschen und Unternehmen verändern“*

Beherrschen Sie die fortschrittlichsten technischen Fähigkeiten und Werkzeuge mit der innovativsten Lehrmethodik in der aktuellen akademischen Szene“

Steigern Sie Ihre beruflichen Erwartungen, indem Sie zu 100% online lernen, ohne Ihre persönlichen und familiären Verpflichtungen zu beeinträchtigen.

Werden Sie ein führender Ingenieur, der bereit ist, von jedem Ort der Welt aus zu lernen.

Das Lehrteam besteht aus Fachleuten aus dem Bereich der Informatik, die ihre Berufserfahrung in dieses Programm einbringen, sowie aus anerkannten Spezialisten aus führenden Gesellschaften und renommierten Universitäten.

Die multimedialen Inhalte, die mit den neuesten Bildungstechnologien entwickelt wurden, ermöglichen der Fachkraft ein situiertes und kontextbezogenes Lernen, d. h. eine simulierte Umgebung, die eine immersive Fortbildung bietet, die auf die Ausführung von realen Situationen ausgerichtet ist.

Das Konzept dieses Programms konzentriert sich auf problemorientiertes Lernen, bei dem der Student versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Dabei wird die Fachkraft durch ein innovatives interaktives Videosystem unterstützt, das von anerkannten Experten entwickelt wurde.

02

Warum an der TECH studieren?

TECH ist die größte digitale Universität der Welt. Mit einem beeindruckenden Katalog von über 14.000 Hochschulprogrammen, die in 11 Sprachen angeboten werden, ist sie mit einer Vermittlungsquote von 99% führend im Bereich der Beschäftigungsfähigkeit. Darüber hinaus verfügt sie über einen beeindruckenden Lehrkörper mit mehr als 6.000 Professoren von höchstem internationalem Prestige.

Studieren Sie an der größten digitalen Universität der Welt und sichern Sie sich Ihren beruflichen Erfolg. Die Zukunft beginnt bei TECH“

Die beste Online-Universität der Welt laut FORBES

Das renommierte, auf Wirtschaft und Finanzen spezialisierte Magazin Forbes hat TECH als „beste Online-Universität der Welt" ausgezeichnet. Dies wurde kürzlich in einem Artikel in der digitalen Ausgabe des Magazins festgestellt, in dem die Erfolgsgeschichte dieser Einrichtung „dank ihres akademischen Angebots, der Auswahl ihrer Lehrkräfte und einer innovativen Lernmethode, die auf die Ausbildung der Fachkräfte der Zukunft abzielt", hervorgehoben wird.

Die besten internationalen Top-Lehrkräfte

Der Lehrkörper der TECH besteht aus mehr als 6.000 Professoren von höchstem internationalen Ansehen. Professoren, Forscher und Führungskräfte multinationaler Unternehmen, darunter Isaiah Covington, Leistungstrainer der Boston Celtics, Magda Romanska, leitende Forscherin am Harvard MetaLAB, Ignacio Wistumba, Vorsitzender der Abteilung für translationale Molekularpathologie am MD Anderson Cancer Center, und D.W. Pine, Kreativdirektor des TIME Magazine, um nur einige zu nennen.

Die größte digitale Universität der Welt

TECH ist die weltweit größte digitale Universität. Wir sind die größte Bildungseinrichtung mit dem besten und umfangreichsten digitalen Bildungskatalog, der zu 100% online ist und die meisten Wissensgebiete abdeckt. Wir bieten weltweit die größte Anzahl eigener Abschlüsse sowie offizieller Grund- und Aufbaustudiengänge an. Insgesamt sind wir mit mehr als 14.000 Hochschulabschlüssen in zehn verschiedenen Sprachen die größte Bildungseinrichtung der Welt.

Die umfassendsten Lehrpläne in der Universitätslandschaft

TECH bietet die vollständigsten Lehrpläne in der Universitätslandschaft an, mit Lehrplänen, die grundlegende Konzepte und gleichzeitig die wichtigsten wissenschaftlichen Fortschritte in ihren spezifischen wissenschaftlichen Bereichen abdecken. Darüber hinaus werden diese Programme ständig aktualisiert, um den Studenten die akademische Avantgarde und die gefragtesten beruflichen Kompetenzen zu garantieren. Auf diese Weise verschaffen die Abschlüsse der Universität ihren Absolventen einen bedeutenden Vorteil, um ihre Karriere erfolgreich voranzutreiben.

Eine einzigartige Lernmethode

TECH ist die erste Universität, die *Relearning* in allen ihren Studiengängen einsetzt. Es handelt sich um die beste Online-Lernmethodik, die mit internationalen Qualitätszertifikaten renommierter Bildungseinrichtungen ausgezeichnet wurde. Darüber hinaus wird dieses disruptive akademische Modell durch die „Fallmethode" ergänzt, wodurch eine einzigartige Online-Lehrstrategie entsteht. Es werden auch innovative Lehrmittel eingesetzt, darunter ausführliche Videos, Infografiken und interaktive Zusammenfassungen.

Die offizielle Online-Universität der NBA

TECH ist die offizielle Online-Universität der NBA. Durch eine Vereinbarung mit der größten Basketball-Liga bietet sie ihren Studenten exklusive Universitätsprogramme sowie eine breite Palette von Bildungsressourcen, die sich auf das Geschäft der Liga und andere Bereiche der Sportindustrie konzentrieren. Jedes Programm hat einen einzigartig gestalteten Lehrplan und bietet außergewöhnliche Gastredner: Fachleute mit herausragendem Sporthintergrund, die ihr Fachwissen zu den wichtigsten Themen zur Verfügung stellen.

Führend in Beschäftigungsfähigkeit

TECH ist es gelungen, die führende Universität im Bereich der Beschäftigungsfähigkeit zu werden. 99% der Studenten finden innerhalb eines Jahres nach Abschluss eines Studiengangs der Universität einen Arbeitsplatz in dem von ihnen studierten Fachgebiet. Ähnlich viele erreichen einen unmittelbaren Karriereaufstieg. All dies ist einer Studienmethodik zu verdanken, die ihre Wirksamkeit auf den Erwerb praktischer Fähigkeiten stützt, die für die berufliche Entwicklung absolut notwendig sind.

Google Partner Premier

Der amerikanische Technologieriese hat TECH mit dem Logo Google Partner Premier ausgezeichnet. Diese Auszeichnung, die nur 3% der Unternehmen weltweit erhalten, unterstreicht die effiziente, flexible und angepasste Erfahrung, die diese Universität den Studenten bietet. Die Anerkennung bestätigt nicht nur die maximale Präzision, Leistung und Investition in die digitalen Infrastrukturen der TECH, sondern positioniert diese Universität auch als eines der modernsten Technologieunternehmen der Welt.

Die von ihren Studenten am besten bewertete Universität

Die Studenten haben TECH auf den wichtigsten Bewertungsportalen als die am besten bewertete Universität der Welt eingestuft, mit einer Höchstbewertung von 4,9 von 5 Punkten, die aus mehr als 1.000 Bewertungen hervorgeht. Diese Ergebnisse festigen die Position der TECH als internationale Referenzuniversität und spiegeln die Exzellenz und die positiven Auswirkungen ihres Bildungsmodells wider.

03

Lehrplan

Der Lehrplan des Weiterbildenden Masterstudiengangs in Softwaretechnik und -qualität ist so konzipiert, dass er eine umfassende und fortgeschrittene Spezialisierung in allen Schlüsselbereichen der Softwaretechnik ermöglicht. Die ersten Module konzentrieren sich auf die Grundlagen, die von Softwaredesign und Anforderungsmanagement bis hin zu Technologiearchitekturen und agilen Methoden reichen. Im weiteren Verlauf des Studiums befassen sich die Studenten mit spezielleren Bereichen wie Testautomatisierung, kontinuierliche Integration und Qualitätssicherung. Darüber hinaus werden Themen aus dem Bereich des Technologie-Projektmanagements behandelt, bei denen die Teilnehmer lernen, multidisziplinäre Teams zu leiten.

Dieser weiterbildende Masterstudiengang bereitet Sie darauf vor, der Experte zu sein, der den Unterschied im Bereich Softwaretechnik und -qualität ausmacht“

Modul 1. Softwarequalität. TRL-Entwicklungsstufen

1.1. Elemente, die die Softwarequalität beeinflussen (I). Die technische Schuld
- 1.1.1. Die technische Schuld. Ursachen und Folgen
- 1.1.2. Softwarequalität. Allgemeine Grundsätze
- 1.1.3. Software mit und ohne Qualitätsprinzipien
  - 1.1.3.1. Konsequenzen
  - 1.1.3.2. Notwendigkeit der Anwendung von Qualitätsprinzipien bei Software
- 1.1.4. Softwarequalität. Typologie
- 1.1.5. Qualitätssoftware. Besondere Features

1.2. Elemente, die die Softwarequalität beeinflussen (II). Zugehörige Kosten
- 1.2.1. Softwarequalität. Beeinflussende Elemente
- 1.2.2. Softwarequalität. Missverständnisse
- 1.2.3. Softwarequalität. Zugehörige Kosten

1.3. Software-Qualitätsmodelle (I). Wissensmanagement
- 1.3.1. Allgemeine Qualitätsmodelle
  - 1.3.1.1. *Total Quality Management*
  - 1.3.1.2. Europäisches Modell für *Business Excellence* (EFQM)
  - 1.3.1.3. Six-Sigma-Modell
- 1.3.2. Wissensmanagement-Modelle
  - 1.3.2.1. Dyba-Modell
  - 1.3.2.2. Seks-Modell
- 1.3.3. Erlebnisfabrik und QIP-Paradigma
- 1.3.4. Modelle mit hoher Nutzungsqualität (25010)

1.4. Software-Qualitätsmodelle (III). Qualität bei Daten, Prozessen und SEI-Modellen
- 1.4.1. Modell der Datenqualität
- 1.4.2. Software zur Prozessmodellierung
- 1.4.3. Software & *Systems Process Engineering Metamodel Specification* (SPEM)
- 1.4.4. SEI-Modelle
  - 1.4.4.1. CMMI
  - 1.4.4.2. SCAMPI
  - 1.4.4.3. IDEAL

1.5. ISO-Normen für Softwarequalität (I). Analyse der Standards
- 1.5.1. ISO 9000-Norm
  - 1.5.1.1. ISO 9000-Norm
  - 1.5.1.2. ISO-Familie von Qualitätsstandards (9000)
- 1.5.2. Andere ISO-Normen zum Thema Qualität
- 1.5.3. Normen zur Qualitätsmodellierung (ISO 2501)
- 1.5.4. Normen zur Qualitätsmessung (ISO 2502n)

1.6. ISO-Normen für Softwarequalität (II). Anforderungen und Bewertung
- 1.6.1. Normen für Qualitätsanforderungen (2503n)
- 1.6.2. Normen zur Qualitätsbewertung (2504n)
- 1.6.3. ISO/IEC 24744:2007

1.7. TRL-Entwicklungsstufen (I). Stufen 1 bis 4
- 1.7.1. TRL-Stufen
- 1.7.2. Stufe 1: Grundlegende Prinzipien
- 1.7.3. Stufe 2: Konzept und/oder Anwendung
- 1.7.4. Stufe 3: kritische analytische Funktion
- 1.7.5. Stufe 4: Komponentenvalidierung in einer Laborumgebung

1.8. TRL-Entwicklungsstufen (II). Stufen 5 bis 9
- 1.8.1. Stufe 5: Komponentenvalidierung in der entsprechenden Umgebung
- 1.8.2. Stufe 6: System/Subsystem-Modell
- 1.8.3. Stufe 7: Demonstration in realer Umgebung
- 1.8.4. Stufe 8: Vollständiges und zertifiziertes System
- 1.8.5. Stufe 9: Erfolg in realer Umgebung

1.9. TRL-Entwicklungsstufen. Verwendungen
- 1.9.1. Beispiel für ein Unternehmen mit Laborumgebung
- 1.9.2. Beispiel für ein FuEuI-Unternehmen
- 1.9.3. Beispiel für ein industrielles FuEuI-Unternehmen
- 1.9.4. Beispiel für ein *Joint-Venture*-Unternehmen zwischen Labor und Technik

1.10. Softwarequalität. Wichtige Details
- 1.10.1. Methodische Details
- 1.10.2. Technische Details
- 1.10.3. Details zum Software-Projektmanagement
  - 1.10.3.1. Qualität der IT-Systeme
  - 1.10.3.2. Qualität von Softwareprodukten
  - 1.10.3.3. Qualität der Softwareprozesse

Modul 2. Software-Projektentwicklung. Funktionelle und technische Dokumentation

2.1. Projektmanagement
- 2.1.1. Projektmanagement für Softwarequalität
- 2.1.2. Projektmanagement. Vorteile
- 2.1.3. Projektmanagement. Typologie

2.2. Methodik des Projektmanagements
- 2.2.1. Methodik des Projektmanagements
- 2.2.2. Projekt-Methoden. Typologie
- 2.2.3. Methodik des Projektmanagements. Anwendung

2.3. Phase der Anforderungsermittlung
- 2.3.1. Identifizierung der Projektanforderungen
- 2.3.2. Verwaltung von Projekttreffen
- 2.3.3. Vorzulegende Dokumentation

2.4. Modell
- 2.4.1. Anfangsphase
- 2.4.2. Analysephase
- 2.4.3. Bauphase
- 2.4.4. Testphase
- 2.4.5. Lieferung

2.5. Zu verwendendes Datenmodell
- 2.5.1. Festlegung des neuen Datenmodells
- 2.5.2. Identifizierung des Datenmigrationsplans
- 2.5.3. Datensatz

2.6. Auswirkungen auf andere Projekte
- 2.6.1. Auswirkungen eines Projekts. Beispiele

2.7. *MUST* des Projekts
- 2.7.1. *MUST* des Projekts
- 2.7.2. Identifizierung des *MUST* des Projekts
- 2.7.3. Identifizierung der Ausführungspunkte für die Lieferung eines Projekts

2.8. Das Konstruktionsteam des Projekts
- 2.8.1. Rollen, die je nach Projekt zu spielen sind
- 2.8.2. Kontakt mit der Personalabteilung für die Rekrutierung
- 2.8.3. Leistungen und Projektzeitplan

2.9. Technische Aspekte eines Softwareprojekts
- 2.9.1. Projekt-Architekt. Technische Aspekte
- 2.9.2. Technische Leiter
- 2.9.3. Aufbau des Projekts Software
- 2.9.4. Bewertung der Codequalität, Sonar

2.10. Projektleistungen
- 2.10.1. Funktionsanalyse
- 2.10.2. Datenmodell
- 2.10.3. Zustandsdiagramm
- 2.10.4. Technische Dokumentation

Modul 3. Software-*Testing*. Testautomatisierung

3.1. Software-Qualitätsmodelle
- 3.1.1. Produktqualität
- 3.1.2. Prozessqualität
- 3.1.3. Qualität der Nutzung

3.2. Prozessqualität
- 3.2.1. Prozessqualität
- 3.2.2. Reifegradmodelle
- 3.2.3. ISO 15504-Norm
  - 3.2.3.1. Verwendungszwecke
  - 3.2.3.2. Kontext
  - 3.2.3.3. Phasen

3.3. ISO/IEC 15504-Norm
- 3.3.1. Prozess-Kategorien
- 3.3.2. Entwicklungsprozess. Beispiel
- 3.3.3. Profil-Fragment
- 3.3.4. Phasen

3.4. CMMI (*Capability Maturity Model Integration*)
- 3.4.1. CMMI. Integration des *Capability Maturity Model*
- 3.4.2. Modelle und Bereiche. Typologie
- 3.4.3. Prozessbereiche
- 3.4.4. Kapazitätsstufen
- 3.4.5. Prozessmanagement
- 3.4.6. Projektleitung

3.5. Verwaltung von Änderungen und Repositorien
- 3.5.1. Management von Softwareänderungen
  - 3.5.1.1. Konfigurationselement. Kontinuierliche Integration
  - 3.5.1.2. Zeilen
  - 3.5.1.3. Flussdiagramme
  - 3.5.1.4. *Branches*
- 3.5.2. *Repository*
  - 3.5.2.1. Versionskontrolle
  - 3.5.2.2. Arbeitsteam und Nutzung des *Repository*
  - 3.5.2.3. Kontinuierliche Integration in das *Repository*

3.6. *Team Foundation Server* (TFS)
- 3.6.1. Installation und Konfiguration
- 3.6.2. Ein Team-Projekt erstellen
- 3.6.3. Hinzufügen von Inhalten zur Versionskontrolle
- 3.6.4. *TFS on Cloud*

3.7. *Testing*
- 3.7.1. Motivation für Tests
- 3.7.2. Verifikationsprüfung
- 3.7.3. Beta-Tests
- 3.7.4. Implementierung und Wartung

3.8. Belastungstests
- 3.8.1. *Load Testing*
- 3.8.2. *LoadView*-Tests
- 3.8.3. Testen mit K6 *Cloud*
- 3.8.4. Testen mit *Loader*

3.9. Belastungs- und Dauertests für Module
- 3.9.1. Motivation für Unit-Tests
- 3.9.2. Tools für *Unit Testing*
- 3.9.3. Motivation für Belastungstests
- 3.9.4. Testen mit *StressTesting*
- 3.9.5. Motivation für Stresstests
- 3.9.6. Testen mit *LoadRunner*

3.10. Skalierbarkeit. Skalierbares Software-Design
- 3.10.1. Skalierbarkeit und Softwarearchitektur
- 3.10.2. Unabhängigkeit zwischen den Ebenen
- 3.10.3. Kopplung zwischen Schichten. Architektur-Muster

Modul 4. Methodologien des Software-Projektmanagements. Waterfall-Methoden versus agile Methoden

4.1. Waterfall-Methode
- 4.1.1. Waterfall-Methode
- 4.1.2. Waterfall-Methode. Einfluss auf die Softwarequalität
- 4.1.3. Waterfall-Methode. Beispiele

4.2. Agile Methodik
- 4.2.1. Agile Methodik
- 4.2.2. Agile Methodik. Einfluss auf die Softwarequalität
- 4.2.3. Agile Methodik. Beispiele

4.3. SCRUM-Methodik
- 4.3.1. SCRUM-Methodik
- 4.3.2. SCRUM-Manifest
- 4.3.3. Einführung von SCRUM

4.4. Kanban-Panel
- 4.4.1. Kanban-Methode
- 4.4.2. Kanban-Panel
- 4.4.3. Kanban-Panel. Beispiel einer Anwendung

4.5. Projektmanagement in Waterfall
- 4.5.1. Phasen eines Projekts
- 4.5.2. Projektvision in Waterfall
- 4.5.3. Zu berücksichtigende Leistungen

4.6. Projektmanagement in SCRUM
- 4.6.1. Phasen eines Projekts SCRUM
- 4.6.2. Projektvision in SCRUM
- 4.6.3. Zu berücksichtigende Leistungen

4.7. Waterfall vs SCRUM. Vergleich
- 4.7.1. Ansatz des Pilotprojekts
- 4.7.2. Projekt nach dem Prinzip Waterfall. Beispiel
- 4.7.3. Projekt nach dem Prinzip SCRUM. Beispiel

4.8. Kundenvision
- 4.8.1. Dokumente in Waterfall
- 4.8.2. Dokumente in SCRUM
- 4.8.3. Vergleich

4.9. Kanban Struktur
- 4.9.1. Anwenderberichte
- 4.9.2. Backlog
- 4.9.3. Kanban-Analyse

4.10. Hybride Projekte
- 4.10.1. Projekt-Konstruktion
- 4.10.2. Projektleitung
- 4.10.3. Zu berücksichtigende Leistungen

Modul 5. TDD (*Test Driven Developement*). Testgetriebener Softwareentwurf

5.1. TDD. *Test Driven Development*
- 5.1.1. TDD. *Test Driven Development*
- 5.1.2. TDD. Einfluss von TDD auf die Qualität
- 5.1.3. Testgesteuertes Design und Entwicklung. Beispiele

5.2. TDD-Zyklus
- 5.2.1. Auswahl einer Anforderung
- 5.2.2. Testen. Typologien
  - 5.2.2.1. Einheitstests
  - 5.2.2.2. Integrationstests
  - 5.2.2.3. *End To End*-Tests
- 5.2.3. Prüfung des Tests. Misserfolge
- 5.2.4. Erstellung der Implementierung
- 5.2.5. Ausführung von automatisierten Tests
- 5.2.6. Eliminierung von Doppelarbeit
- 5.2.7. Aktualisierung der Liste der Anforderungen
- 5.2.8. Wiederholung des TDD-Zyklus
- 5.2.9. TDD-Zyklus. Theoretisches und praktisches Beispiel

5.3. TDD-Implementierungsstrategien
- 5.3.1. Mock-Implementierung
- 5.3.2. Dreieckige Implementierung
- 5.3.3. Offensichtliche Implementierung

5.4. TDD. Nutzung. Vorteile und Nachteile
- 5.4.1. Vorteile der Nutzung
- 5.4.2. Beschränkungen der Nutzung
- 5.4.3. Qualitätsbilanz in der Implementierung

5.5. TDD. Bewährte Verfahren
- 5.5.1. TDD-Regeln
- 5.5.2. Regel 1: Durchführung eines früheren Tests, falls dieser fehlschlägt, bevor in der Produktion programmiert wird
- 5.5.3. Regel 2: Nicht mehr als einen Einheitstest schreiben
- 5.5.4. Regel 3: Nicht mehr Code schreiben als nötig
- 5.5.5. Zu vermeidende Fehler und Anti-Patterns bei TDD

5.6. Simulation eines realen Projekts zur Anwendung von TDD (I)
- 5.6.1. Allgemeine Beschreibung des Projekts (Unternehmen A)
- 5.6.2. Implementierung von TDD
- 5.6.3. Vorgeschlagene Übungen
- 5.6.4. Übungen. *Feedback*

5.7. Simulation eines realen Projekts zur Anwendung von TDD (II)
- 5.7.1. Allgemeine Beschreibung des Projekts (Unternehmen B)
- 5.7.2. Implementierung von TDD
- 5.7.3. Vorgeschlagene Übungen
- 5.7.4. Übungen. *Feedback*

5.8. Simulation eines realen Projekts zur Anwendung von TDD (III)
- 5.8.1. Allgemeine Beschreibung des Projekts (Unternehmen C)
- 5.8.2. Implementierung von TDD
- 5.8.3. Vorgeschlagene Übungen
- 5.8.4. Übungen. *Feedback*

5.9. Alternativen zu 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. Theoretischer Vergleich

5.10. TDD TCR, BDD und ATDD. Praktischer Vergleich
- 5.10.1. Problemstellung
- 5.10.2. Lösen mit TCR
- 5.10.3. Lösen mit BDD
- 5.10.4. Lösen mit ATDD

Modul 6. DevOps. Management der Softwarequalität

6.1. DevOps. Management der Softwarequalität
- 6.1.1. DevOps
- 6.1.2. DevOps und Softwarequalität
- 6.1.3. DevOps. Vorteile der DevOps-Kultur

6.2. DevOps. Beziehung zu Agile
- 6.2.1. Beschleunigte Lieferung
- 6.2.2. Qualität
- 6.2.3. Kostensenkung

6.3. Implementierung von DevOps
- 6.3.1. Identifizierung des Problems
- 6.3.2. Implementierung in einem Unternehmen
- 6.3.3. Metriken zur Implementierung

6.4. Software-Lieferzyklus
- 6.4.1. Design-Methoden
- 6.4.2. Abkommen
- 6.4.3. Roadmap

6.5. Entwicklung von fehlerfreiem Code
- 6.5.1. Wartbarer Code
- 6.5.2. Entwicklungsmuster
- 6.5.3. Code-*Testing*
- 6.5.4. Softwareentwicklung auf Code-Ebene. *Best Practices*

6.6. Automatisierung
- 6.6.1. Automatisierung. Arten von Tests
- 6.6.2. Kosten für Automatisierung und Wartung
- 6.6.3. Automatisierung. Fehler abmildern

6.7. Einsätze
- 6.7.1. Zielbewertung
- 6.7.2. Entwurf eines automatischen und angepassten Prozesses
- 6.7.3. Feedback und Reaktionsfähigkeit

6.8. Management von Vorfällen
- 6.8.1. Bereitschaft für Vorfälle
- 6.8.2. Analyse und Lösung von Vorfällen
- 6.8.3. Künftige Fehler vermeiden

6.9. Automatisierung des Einsatzes
- 6.9.1. Vorbereitungen für automatisierte Einsätze
- 6.9.2. Automatische Bewertung des Prozesszustands
- 6.9.3. Metriken und *Rollback*-Fähigkeit

6.10. *Best Practices*. Entwicklung von DevOps
- 6.10.1. DevOps-Leitfaden für *Best Practices*
- 6.10.2. DevOps. Methodik für das Team
- 6.10.3. Nischen meiden

Modul 7. DevOps und kontinuierliche Integration. Fortgeschrittene praktische Lösungen in der Softwareentwicklung

7.1. Ablauf der Software-Lieferung
- 7.1.1. Identifizierung von Akteuren und Artefakten
- 7.1.2. Entwurf des Software-Lieferflusses
- 7.1.3. Ablauf der Softwarelieferung. Anforderungen zwischen den Etappen

7.2. Prozessautomatisierung
- 7.2.1. Kontinuierliche Integration
- 7.2.2. Kontinuierliche Bereitstellung
- 7.2.3. Konfiguration von Umgebungen und Verwaltung von Geheimnissen

7.3. Deklarative Pipelines
- 7.3.1. Unterschiede zwischen traditionellen, codeähnlichen und deklarativen Pipelines
- 7.3.2. Deklarative Pipelines
- 7.3.3. Deklarative Pipelines in Jenkins
- 7.3.4. Vergleich der Anbieter von kontinuierlicher Integration

7.4. Qualitätsprüfpunkte und erweitertes Feedback
- 7.4.1. Qualitätsprüfpunkte
- 7.4.2. Qualitätsstandards mit Qualitätsprüfpunkten. Instandhaltung
- 7.4.3. Geschäftsanforderungen für Integrationsanfragen

7.5. Verwaltung von Artefakten
- 7.5.1. Artefakte und Lebenszyklus
- 7.5.2. Systeme zur Aufbewahrung und Verwaltung von Artefakten
- 7.5.3. Sicherheit bei der Verwaltung von Artefakten

7.6. Kontinuierliche Bereitstellung
- 7.6.1. Kontinuierliche Bereitstellung in Containern
- 7.6.2. Kontinuierliche Bereitstellung mit PaaS

7.7. Verbesserung der Pipeline-Laufzeit: statische Analyse und *Git Hooks*
- 7.7.1. Statische Analyse
- 7.7.2. Code-Stilregeln
- 7.7.3. *Git Hooks* und Einheitstests
- 7.7.4. Die Auswirkungen der Infrastruktur

7.8. Container-Schwachstellen
- 7.8.1. Container-Schwachstellen
- 7.8.2. Scannen von Bildern
- 7.8.3. Regelmäßige Berichte und Warnmeldungen

Modul 8. Datenbankdesign. Standardisierung und Leistung. Softwarequalität

8.1. Entwurf von Datenbanken
- 8.1.1. Datenbanken. Typologie
- 8.1.2. Derzeit verwendete Datenbanken
  - 8.1.2.1. Relational
  - 8.1.2.2. Schlüssel-Wert
  - 8.1.2.3. Netzwerkbasiert
- 8.1.3. Die Datenqualität

8.2. Entwurf eines *Entity-Relationship*-Modells (I)
- 8.2.1. *Entity-Relationship*-Modell. Qualität und Dokumentation
- 8.2.2. Einheiten
  - 8.2.2.1. Starke Einheit
  - 8.2.2.2. Schwache Einheit
- 8.2.3. Attribute
- 8.2.4. Beziehungsset
  - 8.2.4.1. 1 zu 1
  - 8.2.4.2. 1 zu vielen
  - 8.2.4.3. Viele zu 1
  - 8.2.4.4. Viele zu viele
- 8.2.5. Schlüssel
  - 8.2.5.1. Primärschlüssel
  - 8.2.5.2. Fremdschlüssel
  - 8.2.5.3. Schwacher Primärschlüssel der Einheit
- 8.2.6. Beschränkungen
- 8.2.7. Kardinalität
- 8.2.8. Vererbung
- 8.2.9. Aggregation

8.3. *Entity-Relationship*-Modell (II). Werkzeuge
- 8.3.1. *Entity-Relationship*-Modell. Werkzeuge
- 8.3.2. *Entity-Relationship*-Modell. Praktisches Beispiel
- 8.3.3. Durchführbares *Entity-Relationship*-Modell
  - 8.3.3.1. Visuelles Beispiel
  - 8.3.3.2. Beispiel in tabellarischer Darstellung

8.4. Standardisierung von Datenbanken (I). Erwägungen zur Softwarequalität
- 8.4.1. DB-Standardisierung und Qualität
- 8.4.2. Abhängigkeit
  - 8.4.2.1. Funktionsabhängigkeit
  - 8.4.2.2. Eigenschaften der Funktionsabhängigkeit
  - 8.4.2.3. Abgeleitete Eigenschaften
- 8.4.3. Schlüssel

8.5. Standardisierung von Datenbanken (II). Normalformen und Codd-Regeln
- 8.5.1. Normalformen
  - 8.5.1.1. Erste Normalform (1NF)
  - 8.5.1.2. Zweite Normalform (2NF)
  - 8.5.1.3. Dritte Normalform (3NF)
  - 8.5.1.4. Boyce-Codd-Normalform (BCNF)
  - 8.5.1.5. Vierte Normalform (4NF)
  - 8.5.1.6. Fünfte Normalform (5NF)
- 8.5.2. Codd's Regeln
  - 8.5.2.1. Regel 1: Information
  - 8.5.2.2. Regel 2: Garantierter Zugang
  - 8.5.2.3. Regel 3: Systematische Behandlung von Nullwerten
  - 8.5.2.4. Regel 4: Beschreibung der Datenbank
  - 8.5.2.5. Regel 5: Integrale Untersprache
  - 8.5.2.6. Regel 6: Ansicht aktualisieren
  - 8.5.2.7. Regel 7: Einfügen und Aktualisieren
  - 8.5.2.8. Regel 8: Körperliche Unabhängigkeit
  - 8.5.2.9. Regel 9: Logische Unabhängigkeit
  - 8.5.2.10. Regel 10: Unabhängigkeit der Integrität
    - 8.5.2.10.1. Integritätsregeln
  - 8.5.2.11. Regel 11: Verteilung
  - 8.5.2.12. Regel 12: Nicht-Subversion
- 8.5.3. Praktisches Beispiel

8.6. Datenlager / OLAP-System
- 8.6.1. *Data Warehouse*
- 8.6.2. Faktentabelle
- 8.6.3. Tabelle der Abmessungen
- 8.6.4. Erstellung des OLAP-Systems. Werkzeuge

8.7. Leistung der Datenbank
- 8.7.1. Index-Optimierung
- 8.7.2. Optimierung von Abfragen
- 8.7.3. Tabelle Partitionierung

8.8. Simulation des realen Projekts für DB-Design (I)
- 8.8.1. Allgemeine Beschreibung des Projekts (Unternehmen A)
- 8.8.2. Anwendung von Datenbankdesign
- 8.8.3. Vorgeschlagene Übungen
- 8.8.4. Vorgeschlagene Übungen. *Feedback*

8.9. Simulation des realen Projekts für DB-Design (II)
- 8.9.1. Allgemeine Beschreibung des Projekts (Unternehmen B)
- 8.9.2. Anwendung von Datenbankdesign
- 8.9.3. Vorgeschlagene Übungen
- 8.9.4. Vorgeschlagene Übungen. *Feedback*

8.10. Relevanz der DB-Optimierung für die Softwarequalität
- 8.10.1. Design-Optimierung
- 8.10.2. Optimierung des Abfragecodes
- 8.10.3. Optimierung von gespeichertem Prozedur-Code
- 8.10.4. Der Einfluss von *Triggers* auf die Softwarequalität. Empfehlungen für die Verwendung

Modul 9. Entwurf skalierbarer Architekturen. Architektur im Lebenszyklus der Software

9.1. Entwurf skalierbarer Architekturen (I)
- 9.1.1. Skalierbare Architekturen
- 9.1.2. Grundsätze einer skalierbaren Architektur
  - 9.1.2.1. Zuverlässig
  - 9.1.2.2. Skalierbar
  - 9.1.2.3. Wartbar
- 9.1.3. Arten der Skalierbarkeit
  - 9.1.3.1. Vertikal
  - 9.1.3.2. Horizontal
  - 9.1.3.3. Kombiniert

9.2. DDD-Architekturen (*Domain-Driven Design*)
- 9.2.1. DDD-Modell. Domain-Ausrichtung
- 9.2.2. Schichten, Aufteilung der Verantwortung und Entwurfsmuster
- 9.2.3. Entkopplung als Grundlage für Qualität

9.3. Entwurf skalierbarer Architekturen (II). Vorteile, Einschränkungen und Designstrategien
- 9.3.1. Skalierbare Architektur. Vorteile
- 9.3.2. Skalierbare Architektur. Beschränkungen
- 9.3.3. Strategien für die Entwicklung skalierbarer Architekturen (Beschreibende Tabelle)

9.4. Lebenszyklus der Software (I). Phasen
- 9.4.1. Lebenszyklus der Software
  - 9.4.1.1. Planungsphase
  - 9.4.1.2. Analysephase
  - 9.4.1.3. Entwurfsphase
  - 9.4.1.4. Phase der Umsetzung
  - 9.4.1.5. Testphase
  - 9.4.1.6. Phase der Installation/Einrichtung
  - 9.4.1.7. Phase der Nutzung und Pflege

9.5. Software-Lebenszyklus-Modelle
- 9.5.1. Wasserfall-Modell
- 9.5.2. Wiederholtes Modell
- 9.5.3. Spiralförmiges Modell
- 9.5.4. Big-Bang-Modell

9.6. Lebenszyklus der Software (II). Automatisierung
- 9.6.1. Lebenszyklus der Softwareentwicklung. Lösungen
  - 9.6.1.1. Kontinuierliche Integration und Entwicklung (CI/CD)
  - 9.6.1.2. Agile Methoden
  - 9.6.1.3. DevOps / Produktionsbetrieb
- 9.6.2. Zukünftige Trends
- 9.6.3. Praktische Beispiele

9.7. Softwarearchitektur im Software-Lebenszyklus
- 9.7.1. Vorteile
- 9.7.2. Beschränkungen
- 9.7.3. Werkzeuge

9.8. Simulation eines realen Projekts zum Entwurf einer Softwarearchitektur (I)
- 9.8.1. Allgemeine Beschreibung des Projekts (Unternehmen A)
- 9.8.2. Anwendung des Entwurfs einer Softwarearchitektur
- 9.8.3. Vorgeschlagene Übungen
- 9.8.4. Vorgeschlagene Übungen. *Feedback*

9.9. Simulation eines realen Projekts zum Entwurf einer Softwarearchitektur (II)
- 9.9.1. Allgemeine Beschreibung des Projekts (Unternehmen B)
- 9.9.2. Anwendung des Entwurfs einer Softwarearchitektur
- 9.9.3. Vorgeschlagene Übungen
- 9.9.4. Vorgeschlagene Übungen. *Feedback*

9.10. Simulation eines realen Projekts zum Entwurf einer Softwarearchitektur (III)
- 9.10.1. Allgemeine Beschreibung des Projekts (Unternehmen C)
- 9.10.2. Anwendung des Entwurfs einer Softwarearchitektur
- 9.10.3. Vorgeschlagene Übungen
- 9.10.4. Vorgeschlagene Übungen. *Feedback*

Modul 10. ISO-, IEC 9126-Qualitätskriterien. Metriken zur Softwarequalität

10.1. Qualitätskriterien. ISO/IEC 9126-Norm
- 10.1.1. Qualitätskriterien
- 10.1.2. Softwarequalität. Gründe dafür. ISO/IEC 9126 Norm
- 10.1.3. Die Messung der Softwarequalität als Schlüsselindikator

10.2. Qualitätskriterien für Software. Merkmale
- 10.2.1. Verlässlichkeit
- 10.2.2. Funktionsweise
- 10.2.3. Effizienz
- 10.2.4. Benutzerfreundlichkeit
- 10.2.5. Instandhaltbarkeit
- 10.2.6. Tragbarkeit

10.3. ISO, IEC 9126 (I). Präsentation
- 10.3.1. Beschreibung von ISO, IEC 9126
- 10.3.2. Funktionsweise
- 10.3.3. Verlässlichkeit
- 10.3.4. Benutzerfreundlichkeit
- 10.3.5. Instandhaltbarkeit
- 10.3.6. Tragbarkeit
- 10.3.7. Qualität im Einsatz
- 10.3.8. Metriken zur Softwarequalität
- 10.3.9. Qualitätsmetriken in ISO 9126

10.4. ISO, IEC 9126 (II). McCall und Boehm Modelle
- 10.4.1. McCall Modell: Qualitätsfaktoren
- 10.4.2. Böhm-Modell
- 10.4.3. Mittleres Niveau. Merkmale

10.5. Metriken zur Softwarequalität (I). Elemente
- 10.5.1. Messung
- 10.5.2. Metrik
- 10.5.3. Indikator
  - 10.5.3.1. Arten von Indikatoren
- 10.5.4. Maßnahmen und Modelle
- 10.5.5. Umfang der Software Metriken
- 10.5.6. Klassifizierung von Softwaremetriken

10.6. Messung der Softwarequalität (II). Praxis der Messung
- 10.6.1. Metrische Datenerfassung
- 10.6.2. Messung der internen Produkteigenschaften
- 10.6.3. Messung von externen Produktattributen
- 10.6.4. Messung der Ressourcen
- 10.6.5. Metriken für objektorientierte Systeme

10.7. Design eines einzigen Indikators für Softwarequalität
- 10.7.1. Einzelner Indikator als *Global Scorer*
- 10.7.2. Entwicklung, Rechtfertigung und Anwendung von Indikatoren
- 10.7.3. Beispiel für eine Anwendung. Notwendigkeit, die Details zu kennen

10.8. Simulation eines realen Projekts zur Qualitätsmessung (I)
- 10.8.1. Allgemeine Beschreibung des Projekts (Unternehmen A)
- 10.8.2. Anwendung der Qualitätsmessung
- 10.8.3. Vorgeschlagene Übungen
- 10.8.4. Vorgeschlagene Übungen. *Feedback*

10.9. Simulation eines realen Projekts zur Qualitätsmessung (II)
- 10.9.1. Allgemeine Beschreibung des Projekts (Unternehmen B)
- 10.9.2. Anwendung der Qualitätsmessung
- 10.9.3. Vorgeschlagene Übungen
- 10.9.4. Vorgeschlagene Übungen. *Feedback*

10.10. Simulation eines realen Projekts zur Qualitätsmessung (III)
- 10.10.1. Allgemeine Beschreibung des Projekts (Unternehmen C)
- 10.10.2. Anwendung der Qualitätsmessung
- 10.10.3. Vorgeschlagene Übungen
- 10.10.4. Vorgeschlagene Übungen. *Feedback*

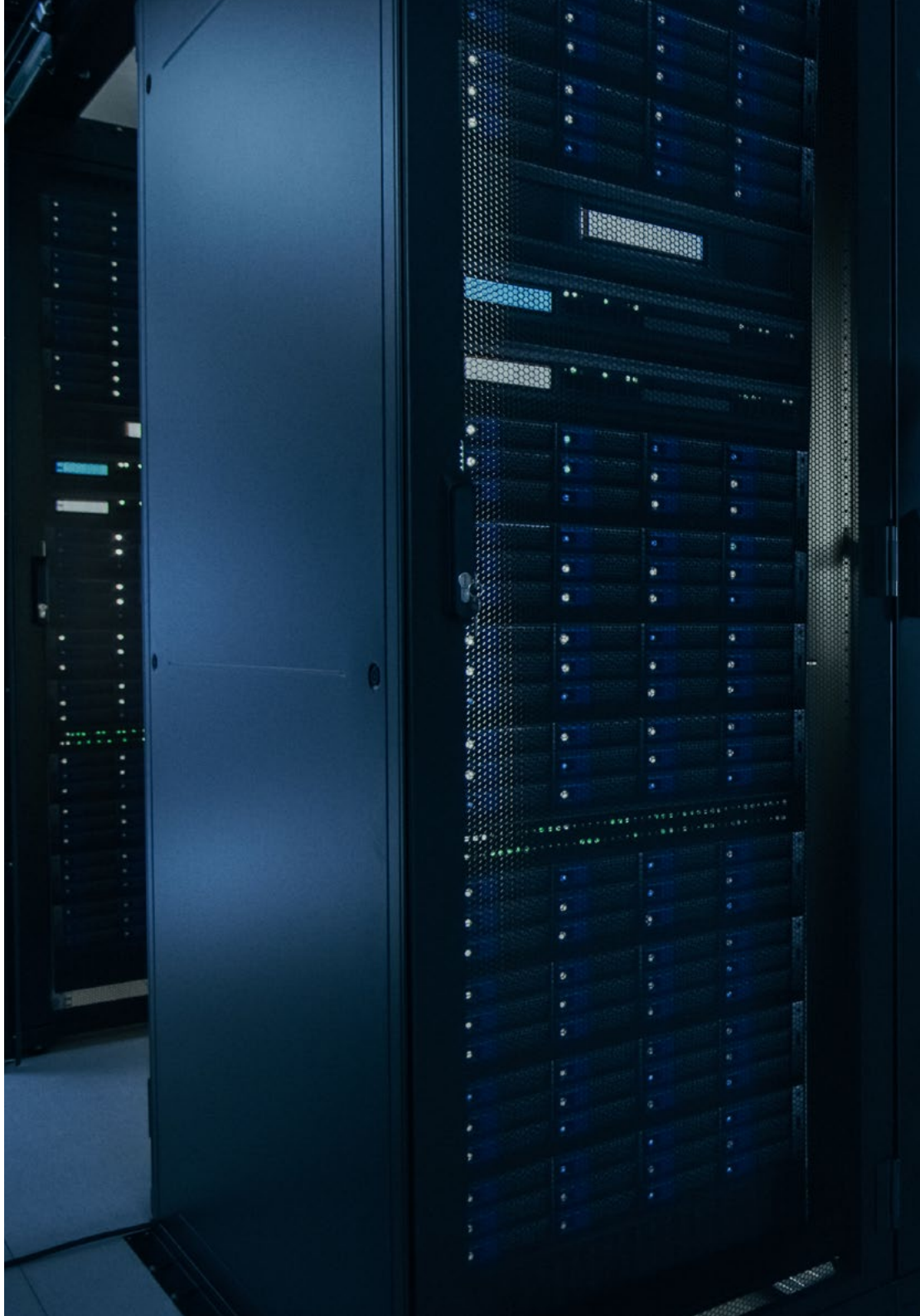
Modul 11. Methodik, Entwicklung und Qualität in der Softwaretechnik

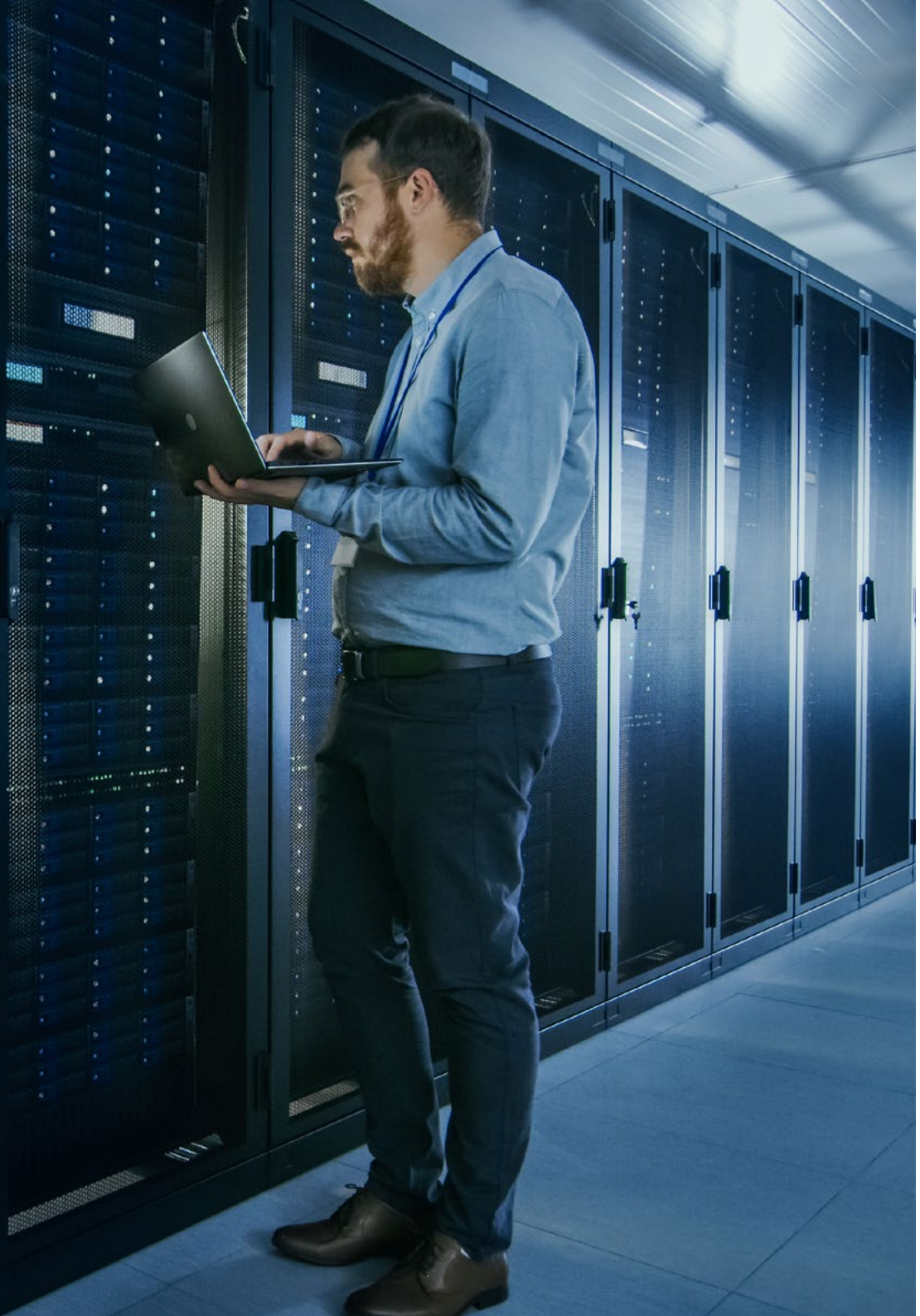
11.1. Modellgestützte Softwareentwicklung
- 11.1.1. Der Bedarf an
- 11.1.3. Modellierung von Objekten
- 11.1.4. UML
- 11.1.5. CASE-Tools

11.2. Anwendungsmodellierung und Entwurfsmuster mit UML
- 11.2.1. Fortgeschrittene Anforderungsmodellierung
- 11.2.2. Erweiterte statische Modellierung
- 11.2.3. Erweiterte dynamische Modellierung
- 11.2.4. Modellierung von Bauteilen
- 11.2.5. Einführung in Entwurfsmuster mit UML
- 11.2.6. Adapter
- 11.2.7. Factory
- 11.2.8. Singleton
- 11.2.9. Strategy
- 11.2.10. Composite
- 11.2.11. Facade
- 11.2.12. Observer

11.3. Modellgestütztes Engineering
- 11.3.1. Einführung
- 11.3.2. Metamodellierung von Systemen
- 11.3.3. MDA
- 11.3.4. DSL
- 11.3.5. Modellverfeinerungen mit OCL
- 11.3.6. Modellumwandlungen

11.4. Ontologien in der Softwaretechnik
- 11.4.1. Einführung
- 11.4.2. Ontologietechnik
- 11.4.3. Anwendung der Ontologien in der Softwaretechnik





Modul 12. Software-Projektmanagement

12.1. Management von *Stakeholdern* und Umfang
- 12.1.1. Identifizierung von Interessengruppen
- 12.1.2. Entwicklung des *Stakeholder*-Management-Plans
- 12.1.3. Management der Einbindung von *Stakeholdern*
- 12.1.4. Überwachung des Engagements der *Stakeholder*
- 12.1.5. Das Projektziel
- 12.1.6. Umfangsmanagement und sein Plan
- 12.1.7. Erfassen von Anforderungen
- 12.1.8. Definieren Sie den Geltungsbereich
- 12.1.9. Erstellen des Projektstrukturplans
- 12.1.10. Überprüfung und Kontrolle des Umfangs

12.2. Die Entwicklung des Zeitplans
- 12.2.1. Zeitmanagement und sein Plan
- 12.2.2. Definieren der Aktivitäten
- 12.2.3. Festlegung der Reihenfolge der Aktivitäten
- 12.2.4. Schätzung der Ressourcen für die Aktivitäten
- 12.2.5. Geschätzte Dauer der Aktivitäten
- 12.2.6. Entwicklung des Zeitplans und Berechnung des kritischen Pfades
- 12.2.7. Zeitplan-Kontrolle

12.3. Budgetentwicklung und Risikobewältigung
- 12.3.1. Schätzung der Kosten
- 12.3.2. Entwicklung des Budgets und der S-Kurve
- 12.3.3. Kostenkontrolle und *Earned-Value*-Methode
- 12.3.4. Risikokonzepte
- 12.3.5. Wie man eine Risikoanalyse durchführt
- 12.3.6. Die Entwicklung des Reaktionsplans

12.4. Kommunikation und Personalwesen
- 12.4.1. Planung des Kommunikationsmanagements
- 12.4.2. Analyse der Kommunikationsanforderungen
- 12.4.3. Technologie der Kommunikation
- 12.4.4. Kommunikationsmodelle
- 12.4.5. Kommunikationsmethoden
- 12.4.6. Plan für das Kommunikationsmanagement
- 12.4.7. Verwaltung der Kommunikation
- 12.4.8. Verwaltung des Personalwesens
- 12.4.9. Hauptakteure und ihre Rolle in den Projekten
- 12.4.10. Arten von Organisationen
- 12.4.11. Projektorganisation
- 12.4.12. Das Projektteam

12.5. Beschaffung
- 12.5.1. Der Beschaffungsprozess
- 12.5.2. Planung
- 12.5.3. Beschaffung von Lieferanten und Einholung von Angeboten
- 12.5.4. Vergabe des Auftrags
- 12.5.5. Vertragsverwaltung
- 12.5.6. Verträge
- 12.5.7. Arten von Verträgen
- 12.5.8. Vertragsverhandlungen

12.6. Durchführung, Überwachung und Kontrolle sowie Abschluss
- 12.6.1. Prozessgruppen
- 12.6.2. Projektdurchführung
- 12.6.3. Projektüberwachung und -kontrolle
- 12.6.4. Abschluss des Projekts

12.7. Berufliche Verantwortung
- 12.7.1. Berufliche Verantwortung
- 12.7.2. Merkmale der sozialen und beruflichen Verantwortung
- 12.7.3. Ethischer Kodex für Projektleiter
- 12.7.4. Verantwortung vs. PMP®
- 12.7.5. Beispiele für Rechenschaftspflicht
- 12.7.6. Vorteile der Professionalisierung

Modul 13. Plattformen für die Softwareentwicklung

13.1. Einführung in die Entwicklung von Applikationen
- 13.1.1. Desktop-Applikationen
- 13.1.2. Programmiersprache
- 13.1.3. Integrierte Entwicklungsumgebungen
- 13.1.4. Webanwendungen
- 13.1.5. Mobile Anwendungen
- 13.1.6. *Cloud*-Anwendungen

13.2. Anwendungsentwicklung und grafische Oberfläche in Java
- 13.2.1. Integrierte Entwicklungsumgebungen für Java
- 13.2.2. Wichtigste IDEs für Java
- 13.2.3. Einführung in die Eclipse-Entwicklungsplattform
- 13.2.4. Einführung in die NetBeans-Entwicklungsplattform
- 13.2.5. *Controller-View*-Modell für grafische Benutzeroberflächen
- 13.2.6. Entwerfen einer grafischen Benutzeroberfläche in Eclipse
- 13.2.7. Entwerfen einer grafischen Benutzeroberfläche in NetBeans

13.3. Fehlersuche und Testen in Java
- 13.3.1. Testen und Debuggen von Java-Programmen
- 13.3.2. Fehlersuche in Eclipse
- 13.3.3. Fehlersuche in NetBeans

13.4. Anwendungsentwicklung und grafische Oberfläche in .NET
- 13.4.1. Net Framework
- 13.4.2. Komponenten der .NET-Entwicklungsplattform
- 13.4.3. Visual Studio .NET
- 13.4.4. .NET GUI-Werkzeuge
- 13.4.5. Die grafische Benutzeroberfläche mit Windows Presentation Foundation
- 13.4.6. Debuggen und Kompilieren einer WPF-Anwendung

13.5. Programmierung für .NET-Netzwerke
- 13.5.1. Einführung in die .NET-Netzwerkprogrammierung
- 13.5.2. .NET-Anfragen und -Antworten
- 13.5.3. Verwendung von .NET-Anwendungsprotokollen
- 13.5.4. Sicherheit in der .NET-Netzwerkprogrammierung

13.6. Entwicklungsumgebungen für mobile Anwendungen
- 13.6.1. Mobile Anwendungen
- 13.6.2. Mobile Android-Anwendungen
- 13.6.3. Schritte für die Android-Entwicklung
- 13.6.4. Die Android Studio IDE

13.7. Entwicklung von Anwendungen in der Android Studio-Umgebung
- 13.7.1. Installieren und Starten von Android Studio
- 13.7.2. Ausführen einer Android-Anwendung
- 13.7.3. Entwicklung der grafischen Oberfläche in Android Studio
- 13.7.4. Starten von Aktivitäten in Android Studio

13.8. Debuggen und Veröffentlichen von Android-Anwendungen
- 13.8.1. Fehlersuche in einer Anwendung in Android Studio
- 13.8.2. Speichern von Anwendungen in Android Studio
- 13.8.3. Veröffentlichung einer Anwendung auf Google Play

13.9. Entwicklung von Anwendungen für die *Cloud*
- 13.9.1. *Cloud Computing*
- 13.9.2. *Cloud*-Ebenen: SaaS, PaaS, IaaS
- 13.9.3. Wichtigste *Cloud*-Entwicklungsplattformen
- 13.9.4. Bibliografische Referenzen

13.10. Einführung in die Google Cloud Platform
- 13.10.1. Grundlagen der Google Cloud Platform
- 13.10.2. Google Cloud Platform-Dienste
- 13.10.3. Google Cloud Platform-Werkzeuge

Modul 14. Web-Client-Computing

14.1. Einführung in HTML
- 14.1.1. Aufbau eines Dokuments
- 14.1.2. Farbe
- 14.1.3. Text
- 14.1.4. Hypertext-Links
- 14.1.5. Bilder
- 14.1.6. Listen
- 14.1.7. Tabellen
- 14.1.8. Rahmen (*Frames*)
- 14.1.9. Formulare
- 14.1.10. Spezifische Elemente für mobile Technologien
- 14.1.11. Ausgediente Elemente

14.2. Cascading Style Sheets (CSS)
- 14.2.1. Elemente und Struktur einer Formatvorlage
  - 14.2.1.1. Erstellung von Stilvorlagen
  - 14.2.1.2. Anwendung von Stilen. Selektoren
  - 14.2.1.3. Stilvererbung und Kaskadierung
  - 14.2.1.4. Seitenformatierung mit Formatvorlagen
  - 14.2.1.5. Seitenstruktur mit Hilfe von Stilen. Das Kastenmodell
- 14.2.2. Gestaltung von Stilen für verschiedene Geräte
- 14.2.3. Arten von Formatvorlagen: statisch und dynamisch. Pseudoklassen
- 14.2.4. Bewährte Verfahren bei der Verwendung von Formatvorlagen

14.3. Einführung und Geschichte von JavaScript
- 14.3.1. Einführung
- 14.3.2. Geschichte von JavaScript
- 14.3.3. Entwicklungsumgebung, die wir verwenden werden

14.4. Grundbegriffe der Webprogrammierung
- 14.4.1. Grundlegende JavaScript-Syntax
- 14.4.2. Primitive Datentypen und Operatoren
- 14.4.3. Variablen und Domänen
- 14.4.4. *Textstrings* und *Template Literals*
- 14.4.5. Zahlen und Boolesche Werte
- 14.4.6. Vergleiche

14.5. Komplexe JavaScript-Strukturen
- 14.5.1. Vektoren oder *Arrays* und Objekte
- 14.5.2. Sets
- 14.5.3. Karten
- 14.5.4. Disjunktionen
- 14.5.5. Schleifen

14.6. Funktionen und Objekte
- 14.6.1. Funktionsdefinition und -aufruf
- 14.6.2. Argumente
- 14.6.3. Pfeil-Funktionen
- 14.6.4. Rückruf-Funktionen oder *Callback*
- 14.6.5. Funktionen höherer Ordnung
- 14.6.6. Wörtliche Objekte
- 14.6.7. Das Objekt *this*
- 14.6.8. Objekte als Namensräume: das Math-Objekt und das Date-Objekt

14.7. Das Dokumentenobjektmodell (DOM)
- 14.7.1. Was ist DOM?
- 14.7.2. Ein bisschen Geschichte
- 14.7.3. Navigieren und Abrufen von Elementen
- 14.7.4. Ein virtuelles DOM mit JSDOM
- 14.7.5. Abfrage-Selektoren oder *Query Selectors*
- 14.7.6. Navigation mittels Eigenschaften
- 14.7.7. Zuweisung von Attributen zu Elementen
- 14.7.8. Erstellen und Ändern von Knoten
- 14.7.9. Aktualisieren des Stils von DOM-Elementen

14.8. Moderne Webentwicklung
- 14.8.1. Ereignisgesteuerter Ablauf und *Listeners*
- 14.8.2. Moderne Web-*Toolkits* und Ausrichtungssysteme
- 14.8.3. Strikter JavaScript-Modus
- 14.8.4. Mehr über Funktionen
- 14.8.5. Asynchrone Funktionen und Versprechen
- 14.8.6. *Closures*
- 14.8.7. Funktionale Programmierung
- 14.8.8. OOP in JavaScript

14.9. Web-Benutzbarkeit
- 14.9.1. Einführung in die Benutzerfreundlichkeit
- 14.9.2. Definition von Benutzerfreundlichkeit
- 14.9.3. Bedeutung des nutzerzentrierten Webdesigns
- 14.9.4. Unterschiede zwischen Barrierefreiheit und Benutzerfreundlichkeit
- 14.9.5. Vorteile und Probleme bei der Kombination von Barrierefreiheit und Benutzerfreundlichkeit
- 14.9.6. Vorteile und Schwierigkeiten bei der Umsetzung von nutzbaren Websites
- 14.9.7. Methoden zur Benutzerfreundlichkeit
- 14.9.8. Analyse der Benutzeranforderungen
- 14.9.9. Konzeptionelle Gestaltungsgrundsätze. Benutzerorientiertes *Prototyping*
- 14.9.10. Leitlinien für die Erstellung von nutzbaren Websites
  - 14.9.10.1. Jakob Nielsens Leitlinien zur Benutzerfreundlichkeit
  - 14.9.10.2. Bruce Tognazzinis Leitlinien zur Benutzerfreundlichkeit
- 14.9.11. Bewertung der Benutzbarkeit

14.10. Barrierefreiheit im Internet
- 14.10.1. Einführung
- 14.10.2. Definition von Barrierefreiheit im Internet
- 14.10.3. Arten von Behinderungen
  - 14.10.3.1. Vorübergehende oder dauerhafte Behinderungen
  - 14.10.3.2. Visuelle Beeinträchtigungen
  - 14.10.3.3. Beeinträchtigungen des Hörvermögens
  - 14.10.3.4. Motorische Behinderungen
  - 14.10.3.5. Neurologische oder kognitive Behinderungen
  - 14.10.3.6. Altersbedingte Schwierigkeiten
  - 14.10.3.7. Umweltbedingte Einschränkungen
  - 14.10.3.8. Hindernisse für den Zugang zum Internet
- 14.10.4. Technische Hilfsmittel und unterstützende Produkte zur Überwindung von Barrieren
  - 14.10.4.1. Hilfsmittel für Blinde
  - 14.10.4.2. Hilfsmittel für Menschen mit Sehschwäche
  - 14.10.4.3. Hilfsmittel für Menschen mit Farbenblindheit
  - 14.10.4.4. Hilfsmittel für Hörgeschädigte
  - 14.10.4.5. Hilfsmittel für Menschen mit motorischen Behinderungen
  - 14.10.4.6. Hilfsmittel für Menschen mit kognitiven und neurologischen Behinderungen
- 14.10.5. Vorteile und Schwierigkeiten bei der Umsetzung der Barrierefreiheit im Internet
- 14.10.6. Vorschriften und Normen für die Barrierefreiheit im Internet
- 14.10.7. Regulierungsstellen für die Barrierefreiheit im Internet
- 14.10.8. Vergleich von Normen und Standards
- 14.10.9. Leitlinien für die Einhaltung von Vorschriften und Normen
  - 14.10.9.1. Beschreibung der wichtigsten Leitlinien (Bilder, Links, Videos usw.)
  - 14.10.9.2. Leitlinien für eine barrierefreie Navigation
    - 14.10.9.2.1. Wahrnehmbarkeit
    - 14.10.9.2.2. Operationalität
    - 14.10.9.2.3. Nachvollziehbarkeit
    - 14.10.9.2.4. Robustheit
- 14.10.10. Beschreibung des Prozesses der Webzugänglichkeitskonformität
- 14.10.11. Konformitätsstufen
- 14.10.12. Konformitätskriterien
- 14.10.13. Anforderungen an die Konformität
- 14.10.14. Methodik zur Bewertung der Zugänglichkeit von Websites

Modul 15. Web-Server-Computing

15.1. Einführung in die Programmierung im Server: PHP
- 15.1.1. Grundlagen der Programmierung im Server
- 15.1.2. Grundlegende PHP-Syntax
- 15.1.3. Generierung von HTML-Inhalten mit PHP
- 15.1.4. Entwicklungs- und Testumgebungen: XAMPP

15.2. PHP für Fortgeschrittene
- 15.2.1. PHP-Kontrollstrukturen
- 15.2.2. PHP-Funktionen
- 15.2.3. *Array*-Verarbeitung in PHP
- 15.2.4. String-Verarbeitung in PHP
- 15.2.5. Objektorientierung in PHP

15.3. Datenmodelle
- 15.3.1. Begriff der Daten. Lebenszyklus der Daten
- 15.3.2. Datentypen
  - 15.3.2.1. Grundlegend
  - 15.3.2.2. Register
  - 15.3.2.3. Dynamisch

15.4. Das relationale Modell
- 15.4.1. Beschreibung
- 15.4.2. Entitäten und Entitätstypen
- 15.4.3. Datenelemente. Attribute
- 15.4.4. Beziehungen: Typen, Untertypen, Kardinalität
- 15.4.5. Schlüssel. Schlüsselarten
- 15.4.6. Normalisierung. Normale Formen

15.5. Aufbau des logischen Datenmodells
- 15.5.1. Spezifikation der Tabelle
- 15.5.2. Definition von Spalten
- 15.5.3. Wichtige Spezifikation
- 15.5.4. Umwandlung in Normalformen. Abhängigkeit

15.6. Das physische Datenmodell. Dateien
- 15.6.1. Beschreibung der Datendateien
- 15.6.2. Datentypen
- 15.6.3. Zugriffsmodi
- 15.6.4. Organisation von Dateien

15.7. Zugriff auf Datenbanken von PHP aus
- 15.7.1. Einführung in MariaDB
- 15.7.2. Arbeiten mit einer MariaDB-Datenbank: die SQL-Sprache
- 15.7.3. Zugriff auf die MariaDB-Datenbank von PHP aus
- 15.7.4. Einführung in MySql
- 15.7.5. Arbeiten mit einer MySql-Datenbank: die SQL-Sprache
- 15.7.6. Zugriff auf die MySql-Datenbank über PHP

15.8. Interaktion mit dem Client über PHP
- 15.8.1. PHP-Formulare
- 15.8.2. Cookies
- 15.8.3. Handhabung von Sitzungen

15.9. Architektur von Webanwendungen
- 15.9.1. Das Model-View-Controller-Muster
- 15.9.2. Controller
- 15.9.3. Modell
- 15.9.4. Ansicht

15.10. Einführung in Webdienste
- 15.10.1. Einführung in XML
- 15.10.2. Dienstorientierte Architekturen (SOA): Webdienste
- 15.10.3. Erstellung von SOAP- und REST-Webdiensten
- 15.10.4. Das SOAP-Protokoll
- 15.10.5. Das REST-Protokoll

Modul 16. Sicherheitsmanagement

16.1. Informationssicherheit
- 16.1.1. Einführung
- 16.1.2. Die Sicherheit von Informationen setzt Vertraulichkeit, Integrität und Verfügbarkeit voraus
- 16.1.3. Sicherheit ist eine wirtschaftliche Frage
- 16.1.4. Sicherheit ist ein Prozess
- 16.1.5. Die Klassifizierung von Informationen
- 16.1.6. Informationssicherheit ist Risikomanagement
- 16.1.7. Sicherheit ist mit Sicherheitskontrollen verbunden
- 16.1.8. Sicherheit ist sowohl physisch als auch logisch
- 16.1.9. Sicherheit betrifft Menschen

16.2. Die Fachkraft für Informationssicherheit
- 16.2.1. Einführung
- 16.2.2. Informationssicherheit als Beruf
- 16.2.3. Zertifizierungen (ISC)2
- 16.2.4. Die Norm ISO 27001
- 16.2.5. Bewährte Sicherheitspraktiken im IT-Service-Management
- 16.2.6. Reifegradmodelle für die Informationssicherheit
- 16.2.7. Andere Zertifizierungen, Standards und professionelle Ressourcen

16.3. Zugangskontrolle
- 16.3.1. Einführung
- 16.3.2. Anforderungen an die Zugangskontrolle
- 16.3.3. Authentifizierungsmechanismen
- 16.3.4. Genehmigungsverfahren
- 16.3.5. Zugang zu Buchhaltung und Rechnungsprüfung
- 16.3.6. Triple A-Technologien

16.4. Programme, Verfahren und Richtlinien zur Informationssicherheit
- 16.4.1. Einführung
- 16.4.2. Programme für das Sicherheitsmanagement
- 16.4.3. Risikomanagement
- 16.4.4. Gestaltung der Sicherheitspolitik

16.5. Pläne zur Aufrechterhaltung des Geschäftsbetriebs
- 16.5.1. Einführung in BCPs
- 16.5.2. Phase I und II
- 16.5.3. Phase III und IV
- 16.5.4. Aufrechterhaltung der BCP

16.6. Verfahren für den korrekten Schutz des Unternehmens
- 16.6.1. DMZ-Netzwerke
- 16.6.2. Systeme zur Erkennung von Eindringlingen
- 16.6.3. Zugriffskontrolllisten
- 16.6.4. Vom Angreifer lernen: Honeypot

16.7. Sicherheitsarchitektur. Prävention
- 16.7.1. Überblick. Aktivitäten und Schichtenmodell
- 16.7.2. Perimeter-Verteidigung (Firewalls, WAFs, IPS usw.)
- 16.7.3. Endpunktschutz (Geräte, Server und Dienste)

16.8. Sicherheitsarchitektur. Erkennung
- 16.8.1. Überblick über Erkennung und Überwachung
- 16.8.2. Logs, verschlüsselte Verkehrsunterbrechung, Aufzeichnung und Siems
- 16.8.3. Warnungen und Informationen

16.9. Sicherheitsarchitektur. Reaktion
- 16.9.1. Reaktion. Produkte, Dienstleistungen und Ressourcen
- 16.9.2. Management von Vorfällen
- 16.9.3. CERTS und CSIRTs

16.10. Sicherheitsarchitektur. Wiederherstellung
- 16.10.1. Ausfallsicherheit, Konzepte, Geschäftsanforderungen und Vorschriften
- 16.10.2. IT-Resilienz-Lösungen
- 16.10.3. Krisenmanagement und *Governance*

Modul 17. Softwaresicherheit

17.1. Software-Sicherheitsprobleme
- 17.1.1. Einführung in das Problem der Softwaresicherheit
- 17.1.2. Schwachstellen und ihre Klassifizierung
- 17.1.3. Sichere Softwareeigenschaften
- 17.1.4. Referenzen

17.2. Grundsätze des Software-Sicherheitsdesigns
- 17.2.1. Einführung
- 17.2.2. Grundsätze des Software-Sicherheitsdesigns
- 17.2.3. Arten von S-SDLC
- 17.2.4. Softwaresicherheit in den S-SDLC-Phasen
- 17.2.5. Methodologien und Normen
- 17.2.6. Referenzen

17.3. Sicherheit im Software-Lebenszyklus in der Anforderungs- und Entwurfsphase
- 17.3.1. Einführung
- 17.3.2. Angriffsmodellierung
- 17.3.3. Missbrauchsfälle
- 17.3.4. Entwicklung von Sicherheitsanforderungen
- 17.3.5. Risikoanalyse. Architektonisch
- 17.3.6. Entwurfsmuster
- 17.3.7. Referenzen

17.4. Sicherheit im Software-Lebenszyklus in den Phasen Kodierung, Test und Betrieb
- 17.4.1. Einführung
- 17.4.2. Risikobasierte Sicherheitsprüfungen
- 17.4.3. Code-Überprüfung
- 17.4.4. Penetrationstests
- 17.4.5. Sicherheitsmaßnahmen
- 17.4.6. Externe Überprüfung
- 17.4.7. Referenzen

17.5. Sichere Kodierungsanwendungen I
- 17.5.1. Einführung
- 17.5.2. Sichere Kodierungspraktiken
- 17.5.3. Eingabeverarbeitung und Validierung
- 17.5.4. Speicherüberlauf
- 17.5.5. Referenzen

17.6. Sichere Kodierungsanwendungen II
- 17.6.1. Einführung
- 17.6.2. Integers Overflows, Abbruchfehler und Probleme mit Typkonvertierungen zwischen Ganzzahlen
- 17.6.3. Fehler und Ausnahmen
- 17.6.4. Datenschutz und Vertraulichkeit
- 17.6.5. Privilegierte Programme
- 17.6.6. Referenzen

17.7. Sicherheit in der Entwicklung und in der *Cloud*
- 17.7.1. Entwicklungssicherheit; Methodik und Praxis
- 17.7.2. PaaS, IaaS, CaaS und SaaS-Modelle
- 17.7.3. Sicherheit in der *Cloud* und für *Cloud*-Dienste

17.8. Verschlüsselung
- 17.8.1. Grundlagen der Kryptologie
- 17.8.2. Symmetrische und asymmetrische Verschlüsselung
- 17.8.3. Verschlüsselung im Ruhezustand und bei der Übermittlung

17.9. Orchestrierung und Automatisierung der Sicherheit (SOAR)
- 17.9.1. Komplexität der manuellen Verarbeitung; Notwendigkeit der Automatisierung von Aufgaben
- 17.9.2. Produkte und Dienstleistungen
- 17.9.3. SOAR-Architektur

17.10. Sicherheit der Telearbeit
- 17.10.1. Bedarf und Szenarien
- 17.10.2. Produkte und Dienstleistungen
- 17.10.3. Sicherheit der Telearbeit

Modul 18. Webserver-Verwaltung

18.1. Einführung in Webserver
- 18.1.1. Was ist ein Webserver?
- 18.1.2. Architektur und Funktionsweise eines Webservers
- 18.1.3. Ressourcen und Inhalte auf einem Webserver
- 18.1.4. Anwendungsserver
- 18.1.5. Proxy-Server
- 18.1.6. Die wichtigsten Webserver auf dem Markt
- 18.1.7. Statistiken zur Webserver-Nutzung
- 18.1.8. Sicherheit des Webservers
- 18.1.9. Lastausgleich bei Webservern
- 18.1.10. Referenzen

18.2. Umgang mit dem HTTP-Protokoll
- 18.2.1. Betrieb und Struktur
- 18.2.2. Beschreibung der Abfragemethoden oder *Request Methods*
- 18.2.3. Status-Codes
- 18.2.4. Kopfzeilen
- 18.2.5. Codierung des Inhalts. Code-Seiten
- 18.2.6. Durchführung von HTTP-Anfragen im Internet mit Hilfe eines Proxys, livehttpheaders oder einer ähnlichen Methode, Analyse des verwendeten Protokolls

18.3. Beschreibung von verteilten Architekturen auf mehreren Servern
- 18.3.1. 3-Schichten-Modell
- 18.3.2. Fehlertoleranz
- 18.3.3. Lastverteilung
- 18.3.4. Sitzungsstatus-Speicher
- 18.3.5. Cache-Speicher

18.4. Internet Information Services (IIS)
- 18.4.1. Was ist IIS?
- 18.4.2. Geschichte und Entwicklung des IIS
- 18.4.3. Die wichtigsten Vorteile und Funktionen von IIS7 und darüber hinaus
- 18.4.4. IIS7 und neuere Architektur

18.5. IIS-Installation, -Verwaltung und -Konfiguration
- 18.5.1. Präambel
- 18.5.2. Installation der Internet Information Services (IIS)
- 18.5.3. IIS-Verwaltungstools
- 18.5.4. Erstellen, Konfigurieren und Verwalten von Websites
- 18.5.5. Installieren und Verwalten von IIS-Erweiterungen

18.6. Erweiterte Sicherheit im IIS
- 18.6.1. Präambel
- 18.6.2. IIS-Authentifizierung, Autorisierung und Zugriffskontrolle
- 18.6.3. Konfigurieren einer sicheren Website auf IIS mit SSL
- 18.6.4. In IIS 8.x implementierte Sicherheitsrichtlinien

18.7. Einführung in Apache
- 18.7.1. Was ist Apache?
- 18.7.2. Die wichtigsten Vorteile von Apache
- 18.7.3. Hauptmerkmale von Apache
- 18.7.4. Architektur

18.8. Apache-Installation und -Konfiguration
- 18.8.1. Apache-Erstinstallation
- 18.8.2. Apache-Konfiguration

18.9. Installieren und Konfigurieren der verschiedenen Apache-Module
- 18.9.1. Installation von Apache-Modulen
- 18.9.2. Arten von Modulen
- 18.9.3. Sichere Apache-Konfiguration

18.10. Erweiterte Sicherheit
- 18.10.1. Authentifizierung, Autorisierung und Zugangskontrolle
- 18.10.2. Authentifizierungsmethoden
- 18.10.3. Sichere Apache-Konfiguration mit SSL

Modul 19. Sicherheitsprüfung

19.1. Einführung in Informationssysteme und deren Prüfung
- 19.1.1. Einführung in Informationssysteme und die Rolle der IT-Prüfung
- 19.1.2. Definitionen von IT-Audit und interner IT-Kontrolle
- 19.1.3. Funktionen und Ziele der IT-Prüfung
- 19.1.4. Unterschiede zwischen interner Kontrolle und IT-Audit

19.2. Interne Kontrolle von Informationssystemen
- 19.2.1. Funktionsschema eines Datenverarbeitungszentrums
- 19.2.2. Klassifizierung der Kontrollen von Informationssystemen
- 19.2.3. Die Goldene Regel

19.3. Der Prozess und die Phasen der Prüfung von Informationssystemen
- 19.3.1. Risikobewertung (EDR) und andere IT-Prüfungsmethoden
- 19.3.2. Durchführung einer Prüfung der Informationssysteme. Prüfungsphasen
- 19.3.3. Grundlegende Fähigkeiten des Auditors für Informationssysteme

19.4. Technische Sicherheitsüberprüfung von Systemen und Netzen
- 19.4.1. Technische Sicherheitsaudits. Penetrationstests. Vorläufige Konzepte
- 19.4.2. Audits der Systemsicherheit. Hilfsmittel
- 19.4.3. Audits der Netzwerksicherheit. Hilfsmittel

19.5. Technische Prüfung der Sicherheit von Internet und mobilen Geräten
- 19.5.1. Internet-Sicherheitsaudit. Hilfsmittel
- 19.5.2. Prüfung der Sicherheit von mobilen Geräten. Hilfsmittel
- 19.5.3. Anhang 1. Aufbau des Kurzberichts und des technischen Berichts
- 19.5.4. Anhang 2. Inventar der Werkzeuge
- 19.5.5. Anhang 3. Methoden

19.6. Managementsystem für die Informationssicherheit
- 19.6.1. IS-Sicherheit: Eigenschaften und Einflussfaktoren
- 19.6.2. Unternehmensrisiken und Risikomanagement: Implementierung von Kontrollen
- 19.6.3. Informationssicherheits-Managementsystem (ISMS): Konzept und kritische Erfolgsfaktoren
- 19.6.4. ISMS - PDCA-Modell
- 19.6.5. ISMS ISO-IEC 27001: organisatorischer Kontext
- 19.6.6. Organisatorischer Kontext
- 19.6.7. Führung
- 19.6.8. Planung
- 19.6.9. Support
- 19.6.10. Operation
- 19.6.11. Leistungsbewertung
- 19.6.12. Verbesserung
- 19.6.13. Anhang zu ISO 27001/ISO-IEC 27002: Zielsetzungen und Kontrollen
- 19.6.14. ISMS-Audit

19.7. Die Durchführung der Prüfung
- 19.7.1. Verfahren
- 19.7.2. Techniken

19.8. Rückverfolgbarkeit
- 19.8.1. Methoden
- 19.8.2. Analyse

19.9. Gewahrsam
- 19.9.1. Techniken
- 19.9.2. Ergebnisse

19.10. Berichterstattung und Präsentation von Beweisen
- 19.10.1. Arten von Berichten
- 19.10.2. Analyse der Daten
- 19.10.3. Vorlage von Beweismitteln

Modul 20. Sicherheit bei Online-Anwendungen

20.1. Schwachstellen und Sicherheitsprobleme in Online-Anwendungen
- 20.1.1. Einführung in die Sicherheit von Online-Anwendungen
- 20.1.2. Sicherheitsschwachstellen beim Entwurf von Webanwendungen
- 20.1.3. Sicherheitsschwachstellen bei der Implementierung von Webanwendungen
- 20.1.4. Sicherheitsschwachstellen bei der Bereitstellung von Webanwendungen
- 20.1.5. Offizielle Listen von Sicherheitslücken

20.2. Richtlinien und Standards für die Sicherheit von Online-Anwendungen
- 20.2.1. Säulen der Sicherheit von Online-Anwendungen
- 20.2.2. Sicherheitspolitik
- 20.2.3. Managementsystem für die Informationssicherheit
- 20.2.4. Sicherer Lebenszyklus der Software Entwicklung
- 20.2.5. Standards für die Anwendungssicherheit

20.3. Sicherheit beim Entwurf von Webanwendungen
- 20.3.1. Einführung in die Sicherheit von Webanwendungen
- 20.3.2. Sicherheit beim Entwurf von Webanwendungen

20.4. Prüfung der Online-Sicherheit von Webanwendungen
- 20.4.1. Analyse und Prüfung der Sicherheit von Webanwendungen
- 20.4.2. Sicherheit bei der Bereitstellung und Produktion von Webanwendungen

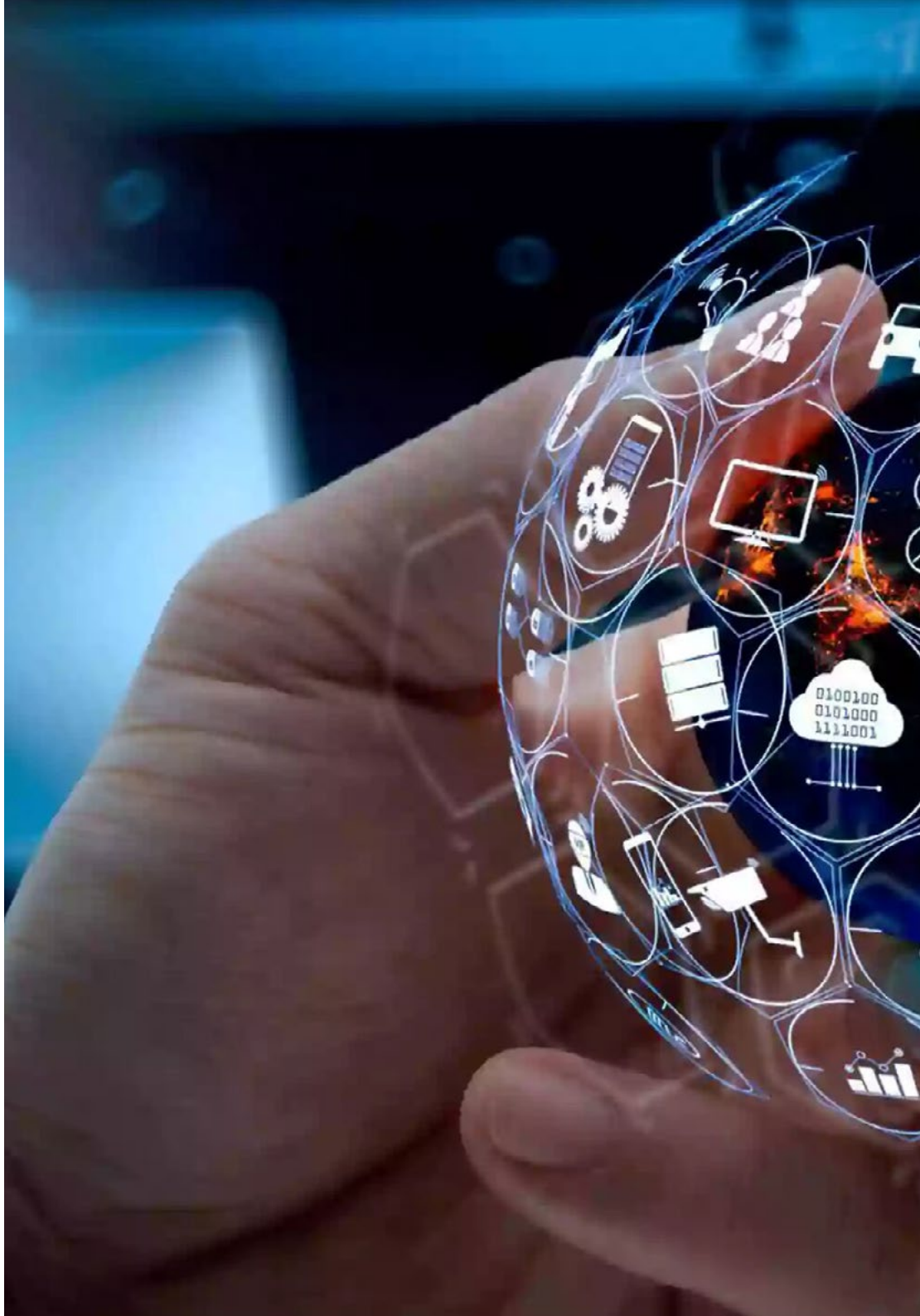
20.5. Sicherheit von Webdiensten
- 20.5.1. Einführung in die Sicherheit von Webdiensten
- 20.5.2. Sicherheitsfunktionen und -technologien für Webdienste

20.6. Prüfung der Online-Sicherheit und des Schutzes von Webdiensten
- 20.6.1. Bewertung der Sicherheit von Webdiensten
- 20.6.2. Online-Schutz. XML-Firewalls und -Gateways

20.7. Ethisches *Hacking*, *Malware* und Forensic
- 20.7.1. Ethisches *Hacking*
- 20.7.2. *Malware*-Analyse
- 20.7.3. Forensische Analyse

20.8. Bewährte Verfahren zur Gewährleistung der Anwendungssicherheit
- 20.8.1. Handbuch für bewährte Praktiken bei der Entwicklung von Online-Anwendungen
- 20.8.2. Handbuch für bewährte Praktiken bei der Umsetzung von Online-Anwendungen

20.9. Häufige Fehler, die die Anwendungssicherheit untergraben
- 20.9.1. Häufige Entwicklungsfehler
- 20.9.2. Häufige Fehler beim *Hosting*
- 20.9.3. Häufige Fehler in der Produktion



Ein umfassender Lehrplan, der Sie dazu bringt, den Bereich Big Data zu beherrschen und ein erfolgreicher Business Strategy Architect zu werden“

04

Lehrziele

Der Weiterbildende Masterstudiengang in Softwaretechnik und -qualität zielt auf die Fortbildung hochqualifizierter Fachleute im Bereich der Konzeption, Entwicklung und Verwaltung hochwertiger Softwaresysteme ab. Mit einem besonderen Schwerpunkt auf Qualität und strategischem Management technologischer Projekte fördert das Programm auch die Fähigkeit, sich an die rasanten Entwicklungen in diesem Sektor anzupassen. So wird sichergestellt, dass die Studenten nicht nur auf die Herausforderungen der Zukunft vorbereitet sind, sondern auch in der Lage sind, Innovationen im Bereich Software anzuführen.

Verwandeln Sie Ideen in effektive Technologielösungen mit der Kunst der Softwareentwicklung“

Allgemeine Ziele

- Entwickeln fortgeschrittener Fähigkeiten in der Konzeption, Entwicklung und Wartung komplexer und skalierbarer Softwaresysteme unter Anwendung der besten Praktiken und Methoden der Softwaretechnik
- Fortbilden der Studenten in der Software-Qualitätssicherung, indem ihnen Werkzeuge und Techniken an die Hand gegeben werden, die die Zuverlässigkeit, Sicherheit und Leistung technologischer Lösungen garantieren
- Fördern von Führungsqualitäten beim Management technologischer Projekte, Entwicklung von Kompetenzen in der Leitung multidisziplinärer Teams, strategischer Planung und Entscheidungsfindung in dynamischen Umgebungen
- Fördern der Fähigkeit zur Anpassung an den raschen technologischen Fortschritt durch Spezialisierung auf neue Tools, Techniken und Trends, die es Ihnen ermöglichen, an der Spitze der Softwaretechnik zu bleiben
- Entwickeln von Kompetenzen im Qualitätsmanagement während des gesamten Software-Lebenszyklus, von der anfänglichen Planung bis zur Wartung und kontinuierlichen Verbesserung von Systemen
- Stärken der Kommunikations- und Teamworkfähigkeiten, die für eine effektive Zusammenarbeit mit verschiedenen *Stakeholdern*, das Management von Erwartungen und den Erfolg von Technologieprojekten unerlässlich sind

Verbessern Sie Ihre Fähigkeiten und werden Sie führend bei der Entwicklung modernster Technologielösungen“

Spezifische Ziele

Modul 1. Softwarequalität. TRL-Entwicklungsstufen

- Verstehen der verschiedenen Stufen der technologischen Reife und ihrer Beziehung zur Softwarequalität
- Bewerten der Softwareentwicklung in jeder TRL-Stufe und wie sie sich auf die Qualität des Endprodukts auswirkt

Modul 2. Software-Projektentwicklung. Funktionelle und technische Dokumentation

- Entwickeln von Fähigkeiten zur Erstellung klarer und detaillierter funktionaler und technischer Dokumentation für Softwareprojekte
- Analysieren der Bedeutung einer genauen Dokumentation für das Projektmanagement und die Softwarequalität

Modul 3. Software-*Testing*. Testautomatisierung

- Entwickeln von Fähigkeiten zum Entwerfen und Ausführen automatisierter Tests für Softwareanwendungen
- Implementieren effizienter Testlösungen unter Verwendung von Tools zur Testautomatisierung

Modul 4. Methodologien des Software-Projektmanagements. Waterfall-Methoden versus agile Methoden

- Analysieren der Unterschiede zwischen Waterfall- und Agile-Methoden im Software-Projektmanagement
- Bewerten der Vorteile und Grenzen der einzelnen Methoden je nach Projekttyp

Modul 5. TDD (*Test Driven Developement*). Testgetriebener Softwareentwurf

- Entwickeln von Fähigkeiten zum Schreiben von *Unit-Tests* vor dem Schreiben von Produktionscode
- Verbessern der Softwarequalität durch die Implementierung von TDD im Entwicklungsprozess

Modul 6. DevOps. Management der Softwarequalität

- Untersuchen des Konzepts von DevOps und seiner Auswirkungen auf die kontinuierliche Verbesserung der Softwarequalität
- Lernen, wie man Entwicklungs- und Betriebspraktiken integriert, um einen agileren und effizienteren Software-Lebenszyklus zu erreichen

Modul 7. DevOps und kontinuierliche Integration. Fortgeschrittene praktische Lösungen in der Softwareentwicklung

- Vertiefen der fortgeschrittenen kontinuierlichen Integrationstechniken im Rahmen von DevOps
- Implementieren praktischer Lösungen zur kontinuierlichen Integration, um den Softwareentwicklungs- und -bereitstellungsprozess zu automatisieren

Modul 8. Datenbankdesign. Standardisierung und Leistung. Softwarequalität

- Analysieren der Grundsätze des Datenbankdesigns, einschließlich Normalisierung und Leistungsoptimierung
- Verstehen, wie ein angemessenes Datenbankdesign zur Softwarequalität beiträgt

Modul 9. Entwurf skalierbarer Architekturen. Architektur im Lebenszyklus der Software

- Vertiefen der Entwurfsprinzipien skalierbarer Architekturen und ihrer Auswirkungen auf die Softwarequalität und -leistung
- Bewerten verschiedener Architekturmuster für skalierbare Softwareanwendungen

Modul 10. ISO-, IEC 9126-Qualitätskriterien. Metriken zur Softwarequalität

- Verstehen der Kriterien für die Softwarequalität gemäß diesen Standards und deren Anwendung
- Implementieren von Qualitätsmetriken zur Bewertung und kontinuierlichen Verbesserung von Softwareanwendungen

Modul 11. Methodik, Entwicklung und Qualität in der Softwaretechnik

- Vertiefen der am häufigsten verwendeten Methoden in der Softwaretechnik und ihrer Beziehung zur Qualität
- Entwickeln eines ganzheitlichen Ansatzes, der Entwicklung, Testen und Qualität in Softwareprojekten kombiniert

Modul 12. Software-Projektmanagement

- Entwickeln von Fähigkeiten im Software-Projektmanagement, von der Planung bis zur Ausführung
- Verwalten von Ressourcen, Zeit und Risiken im Zusammenhang mit Projekten zur Softwareentwicklung

Modul 13. Plattformen für die Softwareentwicklung

- Untersuchen verschiedener Plattformen für die Softwareentwicklung und ihrer Merkmale
- Bewerten von Entwicklungsplattformen in Bezug auf ihre Fähigkeiten, Flexibilität und Kompatibilität mit verschiedenen Projekten

Modul 14. Web-Client-Computing

- Analysieren, wie clientseitiges Computing bei der Entwicklung von Webanwendungen durchgeführt wird
- Entwickeln von Anwendungen, die das clientseitige Computing für eine verbesserte Interaktion und Leistung nutzen

Modul 15. Web-Server-Computing

- Erforschen der Technologien und Techniken, die für die Datenverarbeitung auf dem Webserver verwendet werden
- Verstehen von Datenverarbeitung, Geschäftslogik und Benutzerverwaltung auf dem Server

Modul 16. Sicherheitsmanagement

- Bewerten der Sicherheitsrisiken von Anwendungen und Anwenden von Präventivmaßnahmen
- Implementieren von Sicherheitskontrollen in allen Phasen des Software-Lebenszyklus

Modul 17. Softwaresicherheit

- Erkunden der besten Sicherheitsverfahren in der Softwareentwicklung
- Analysieren der häufigsten Schwachstellen in Software und lernen, wie man sie entschärfen kann

Modul 18. Webserver-Verwaltung

- Verstehen der Rolle von Webservern bei der Anwendungsentwicklung und -bereitstellung
- Entwickeln von Fähigkeiten in der Verwaltung und Wartung von Webservern

Modul 19. Sicherheitsprüfung

- Bewerten der Systemsicherheit durch Audits und Penetrationstests
- Implementieren kontinuierlicher Audit-Prozesse zur Verbesserung der Softwaresicherheit

Modul 20. Sicherheit bei Online-Anwendungen

- Implementieren von Lösungen zum Schutz von Online-Anwendungen vor externen und internen Bedrohungen
- Einrichten von Sicherheits- und Prüfungsrichtlinien zur Gewährleistung der Integrität von Online-Anwendungen

05

Karrieremöglichkeiten

Der Absolvent wird bestens fortgebildet sein, um Führungsaufgaben bei der Entwicklung, Implementierung und Verwaltung von hochwertiger Software zu übernehmen. Dank seiner fortgeschrittenen Spezialisierung in Schlüsselbereichen wie Softwarearchitektur, Sicherheit von Online-Anwendungen, technologisches Projektmanagement und agile Methoden wird er in der Lage sein, Entwicklungsteams zu leiten. Darüber hinaus wird er aufgrund seiner Vorbereitung in der Lage sein, Führungspositionen in Softwareprojekten zu übernehmen, während seine Fähigkeit zur Innovation und zur Leitung multidisziplinärer Teams ihn in die Lage versetzen wird, sich den Herausforderungen des digitalen Umfelds zu stellen und einen wichtigen Beitrag zum Erfolg eines jeden Unternehmens zu leisten.

TECH bietet Ihnen die Möglichkeit, Ihre Träume in der aufregendsten Disziplin zu verwirklichen, die Ideen in greifbare Produkte verwandelt, die das Leben der Menschen verbessern können“

Profil des Absolventen

Der Weiterbildende Masterstudiengang in Softwaretechnik und -qualität zielt auf die Fortbildung hochqualifizierter Fachkräfte ab, die in der Lage sind, technologisch anspruchsvolle Projekte zu leiten und zu verwalten. Der Absolvent wird sowohl agile als auch traditionelle Methoden beherrschen, um Qualität, Sicherheit und Effizienz in allen Phasen der Softwareentwicklung zu gewährleisten. Darüber hinaus wird er in der Lage sein, skalierbare, effiziente und sichere Softwaresysteme zu entwerfen und zu entwickeln und dabei internationale Qualitätsstandards und fortschrittliche Methoden wie DevOps und kontinuierliche Integration anzuwenden.

Werden Sie ein Experte, der den Erfolg von Unternehmen an der größten digitalen Universität der Welt garantiert.

- **Software- und Systemsicherheit:** Kompetenz bei der Umsetzung fortgeschrittener Sicherheitspraktiken, einschließlich Datenschutz und Schwachstellenmanagement in Online-Anwendungen
- **Software-Qualitätssicherung:** Fähigkeit zur Anwendung internationaler Normen (ISO, IEC 9126) und automatisierter Testwerkzeuge zur Gewährleistung der Zuverlässigkeit und Leistung von Software
- **Entwicklung skalierbarer Architekturen:** Fähigkeit, Softwaresysteme zu entwerfen und zu erstellen, die wachsen und sich an die Marktanforderungen anpassen können, ohne die Qualität und Sicherheit zu beeinträchtigen.
- **Kontinuierliche Integration und DevOps:** Fähigkeit zur Implementierung und Verwaltung kontinuierlicher Integrationsprozesse, die eine effiziente und nahtlose Bereitstellung neuer Softwarefunktionen gewährleisten.

Nach Abschluss des weiterbildenden Masterstudiengangs werden Sie in der Lage sein, Ihre Kenntnisse und Fähigkeiten in den folgenden Positionen anzuwenden:

1. **Leiter der Technologieabteilung (CTO):** Verantwortlich für die strategische Ausrichtung der Technologie in einem Unternehmen, leitet Entwicklungsteams und beaufsichtigt die Umsetzung innovativer Technologielösungen
2. **Manager für Softwarequalität:** Verantwortlich für die Überwachung und Sicherstellung, dass Softwareprozesse und -produkte die festgelegten Qualitätsstandards erfüllen, sowie für die Leitung von Initiativen zur kontinuierlichen Verbesserung und Softwareprüfung
3. **Softwarearchitekt:** Hauptdesigner der Struktur und Architektur komplexer Softwaresysteme, der sicherstellt, dass diese skalierbar, sicher und effizient sind
4. **Leiter von Softwareprojekten:** Verantwortlich für die Planung, Durchführung und Lieferung von Softwareprojekten, wobei er multidisziplinäre Teams leitet und sicherstellt, dass die Projekte fristgerecht, im Rahmen des Budgets und unter Einhaltung der entsprechenden Qualitätsstandards abgeschlossen werden
5. **IT-Sicherheitsspezialist:** Verantwortlich für den Schutz von Anwendungen, Infrastruktur und Daten vor Cyber-Bedrohungen, Umsetzung wirksamer Sicherheitsstrategien und -richtlinien
6. **Auditor für Softwaresicherheit:** Führt umfassende Audits durch, um Schwachstellen in Anwendungen und Systemen zu ermitteln, und schlägt Verbesserungen und Lösungen vor, um die Softwaresicherheit zu gewährleisten

“Wenn Sie in der digitalen Welt etwas bewegen wollen, wählen Sie diesen Weg, der Sie zum Experten für die Entwicklung von Qualitätssoftware macht“

06

Studienmethodik

TECH ist die erste Universität der Welt, die die Methodik der ***case studies*** mit ***Relearning*** kombiniert, einem 100%igen Online-Lernsystem, das auf geführten Wiederholungen basiert.

Diese disruptive pädagogische Strategie wurde entwickelt, um Fachleuten die Möglichkeit zu bieten, ihr Wissen zu aktualisieren und ihre Fähigkeiten auf intensive und gründliche Weise zu entwickeln. Ein Lernmodell, das den Studenten in den Mittelpunkt des akademischen Prozesses stellt und ihm die Hauptrolle zuweist, indem es sich an seine Bedürfnisse anpasst und die herkömmlichen Methoden beiseite lässt.

TECH bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein“

Der Student: die Priorität aller Programme von TECH

Bei der Studienmethodik von TECH steht der Student im Mittelpunkt.
Die pädagogischen Instrumente jedes Programms wurden unter Berücksichtigung der Anforderungen an Zeit, Verfügbarkeit und akademische Genauigkeit ausgewählt, die heutzutage nicht nur von den Studenten, sondern auch von den am stärksten umkämpften Stellen auf dem Markt verlangt werden.

Beim asynchronen Bildungsmodell von TECH entscheidet der Student selbst, wie viel Zeit er mit dem Lernen verbringt und wie er seinen Tagesablauf gestaltet, und das alles bequem von einem elektronischen Gerät seiner Wahl aus. Der Student muss nicht an Präsenzveranstaltungen teilnehmen, die er oft nicht wahrnehmen kann.
Die Lernaktivitäten werden nach eigenem Ermessen durchgeführt. Er kann jederzeit entscheiden, wann und von wo aus er lernen möchte.

> *Bei TECH gibt es KEINE Präsenzveranstaltungen (an denen man nie teilnehmen kann)“*

Die international umfassendsten Lehrpläne

TECH zeichnet sich dadurch aus, dass sie die umfassendsten Studiengänge im universitären Umfeld anbietet. Dieser Umfang wird durch die Erstellung von Lehrplänen erreicht, die nicht nur die wesentlichen Kenntnisse, sondern auch die neuesten Innovationen in jedem Bereich abdecken.

Durch ihre ständige Aktualisierung ermöglichen diese Programme den Studenten, mit den Veränderungen des Marktes Schritt zu halten und die von den Arbeitgebern am meisten geschätzten Fähigkeiten zu erwerben. Auf diese Weise erhalten die Studenten, die ihr Studium bei TECH absolvieren, eine umfassende Vorbereitung, die ihnen einen bedeutenden Wettbewerbsvorteil verschafft, um in ihrer beruflichen Laufbahn voranzukommen.

Und das von jedem Gerät aus, ob PC, Tablet oder Smartphone.

> *Das Modell der TECH ist asynchron, d. h. Sie können an Ihrem PC, Tablet oder Smartphone studieren, wo immer Sie wollen, wann immer Sie wollen und so lange Sie wollen“*

Case studies oder Fallmethode

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Wirtschaftshochschulen der Welt. Sie wurde 1912 entwickelt, damit Studenten der Rechtswissenschaften das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernten, sondern auch mit realen komplexen Situationen konfrontiert wurden. Auf diese Weise konnten sie fundierte Entscheidungen treffen und Werturteile darüber fällen, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Bei diesem Lehrmodell ist es der Student selbst, der durch Strategien wie *Learning by doing* oder *Design Thinking*, die von anderen renommierten Einrichtungen wie Yale oder Stanford angewandt werden, seine berufliche Kompetenz aufbaut.

Diese handlungsorientierte Methode wird während des gesamten Studiengangs angewandt, den der Student bei TECH absolviert. Auf diese Weise wird er mit zahlreichen realen Situationen konfrontiert und muss Wissen integrieren, recherchieren, argumentieren und seine Ideen und Entscheidungen verteidigen. All dies unter der Prämisse, eine Antwort auf die Frage zu finden, wie er sich verhalten würde, wenn er in seiner täglichen Arbeit mit spezifischen, komplexen Ereignissen konfrontiert würde.

Relearning-Methode

Bei TECH werden die *case studies* mit der besten 100%igen Online-Lernmethode ergänzt: *Relearning*.

Diese Methode bricht mit traditionellen Lehrmethoden, um den Studenten in den Mittelpunkt zu stellen und ihm die besten Inhalte in verschiedenen Formaten zu vermitteln. Auf diese Weise kann er die wichtigsten Konzepte der einzelnen Fächer wiederholen und lernen, sie in einem realen Umfeld anzuwenden.

In diesem Sinne und gemäß zahlreicher wissenschaftlicher Untersuchungen ist die Wiederholung der beste Weg, um zu lernen. Aus diesem Grund bietet TECH zwischen 8 und 16 Wiederholungen jedes zentralen Konzepts innerhalb ein und derselben Lektion, die auf unterschiedliche Weise präsentiert werden, um sicherzustellen, dass das Wissen während des Lernprozesses vollständig gefestigt wird.

Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.

Ein 100%iger virtueller Online-Campus mit den besten didaktischen Ressourcen

Um seine Methodik wirksam anzuwenden, konzentriert sich TECH darauf, den Studenten Lehrmaterial in verschiedenen Formaten zur Verfügung zu stellen: Texte, interaktive Videos, Illustrationen und Wissenskarten, um nur einige zu nennen. Sie alle werden von qualifizierten Lehrkräften entwickelt, die ihre Arbeit darauf ausrichten, reale Fälle mit der Lösung komplexer Situationen durch Simulationen, dem Studium von Zusammenhängen, die für jede berufliche Laufbahn gelten, und dem Lernen durch Wiederholung mittels Audios, Präsentationen, Animationen, Bildern usw. zu verbinden.

Die neuesten wissenschaftlichen Erkenntnisse auf dem Gebiet der Neurowissenschaften weisen darauf hin, dass es wichtig ist, den Ort und den Kontext, in dem der Inhalt abgerufen wird, zu berücksichtigen, bevor ein neuer Lernprozess beginnt. Die Möglichkeit, diese Variablen individuell anzupassen, hilft den Menschen, sich zu erinnern und Wissen im Hippocampus zu speichern, um es langfristig zu behalten. Dies ist ein Modell, das als *Neurocognitive context-dependent e-learning* bezeichnet wird und in diesem Hochschulstudium bewusst angewendet wird.

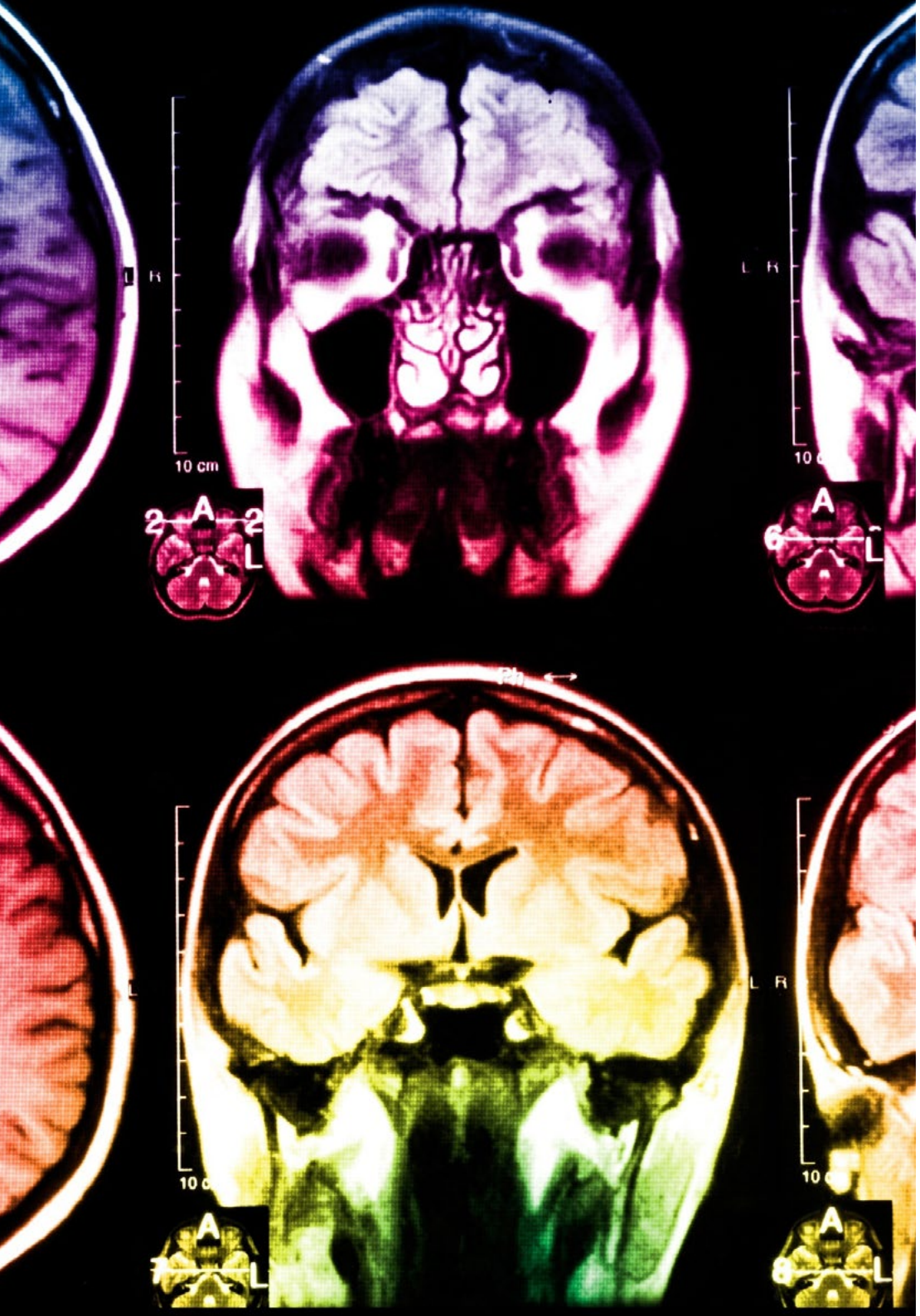
Zum anderen, auch um den Kontakt zwischen Mentor und Student so weit wie möglich zu begünstigen, wird eine breite Palette von Kommunikationsmöglichkeiten angeboten, sowohl in Echtzeit als auch zeitversetzt (internes Messaging, Diskussionsforen, Telefondienst, E-Mail-Kontakt mit dem technischen Sekretariat, Chat und Videokonferenzen).

Darüber hinaus wird dieser sehr vollständige virtuelle Campus den Studenten der TECH die Möglichkeit geben, ihre Studienzeiten entsprechend ihrer persönlichen Verfügbarkeit oder ihren beruflichen Verpflichtungen zu organisieren. Auf diese Weise haben sie eine globale Kontrolle über die akademischen Inhalte und ihre didaktischen Hilfsmittel, in Übereinstimmung mit ihrer beschleunigten beruflichen Weiterbildung.

> *Der Online-Studienmodus dieses Programms wird es Ihnen ermöglichen, Ihre Zeit und Ihr Lerntempo zu organisieren und an Ihren Zeitplan anzupassen“*

Die Wirksamkeit der Methode wird durch vier Schlüsselergebnisse belegt:

1. Studenten, die diese Methode anwenden, nehmen nicht nur Konzepte auf, sondern entwickeln auch ihre geistigen Fähigkeiten durch Übungen zur Bewertung realer Situationen und zur Anwendung ihres Wissens.
2. Das Lernen basiert auf praktischen Fähigkeiten, die es den Studenten ermöglichen, sich besser in die reale Welt zu integrieren.
3. Eine einfachere und effizientere Aufnahme von Ideen und Konzepten wird durch die Verwendung von Situationen erreicht, die aus der Realität entstanden sind.
4. Das Gefühl der Effizienz der investierten Anstrengung wird zu einem sehr wichtigen Anreiz für die Studenten, was sich in einem größeren Interesse am Lernen und einer Steigerung der Zeit, die für die Arbeit am Kurs aufgewendet wird, niederschlägt.



Die von ihren Studenten am besten bewertete Hochschulmethodik

Die Ergebnisse dieses innovativen akademischen Modells lassen sich an der Gesamtzufriedenheit der Absolventen der TECH ablesen.

Die Studenten bewerten die Qualität der Lehre, die Qualität der Materialien, die Kursstruktur und die Ziele als hervorragend. So überrascht es nicht, dass die Einrichtung von ihren Studenten auf der Bewertungsplattform Trustpilot mit 4,9 von 5 Punkten am besten bewertet wurde.

Sie können von jedem Gerät mit Internetanschluss (Computer, Tablet, Smartphone) auf die Studieninhalte zugreifen, da TECH in Sachen Technologie und Pädagogik führend ist.

Sie werden die Vorteile des Zugangs zu simulierten Lernumgebungen und des Lernens durch Beobachtung, d. h. Learning from an expert, nutzen können.

In diesem Programm stehen Ihnen die besten Lehrmaterialien zur Verfügung, die sorgfältig vorbereitet wurden:

Studienmaterial

Alle didaktischen Inhalte werden von den Fachkräfte, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf ein audiovisuelles Format übertragen, das unsere Online-Arbeitsweise mit den neuesten Techniken ermöglicht, die es uns erlauben, Ihnen eine hohe Qualität in jedem der Stücke zu bieten, die wir Ihnen zur Verfügung stellen werden.

Übungen für Fertigkeiten und Kompetenzen

Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.

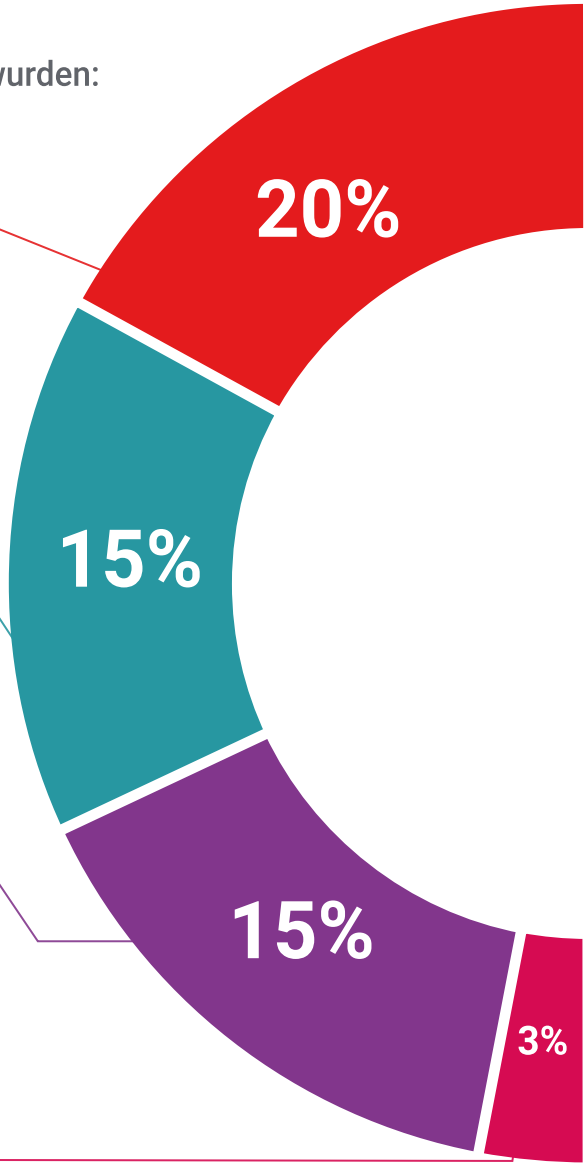
Interaktive Zusammenfassungen

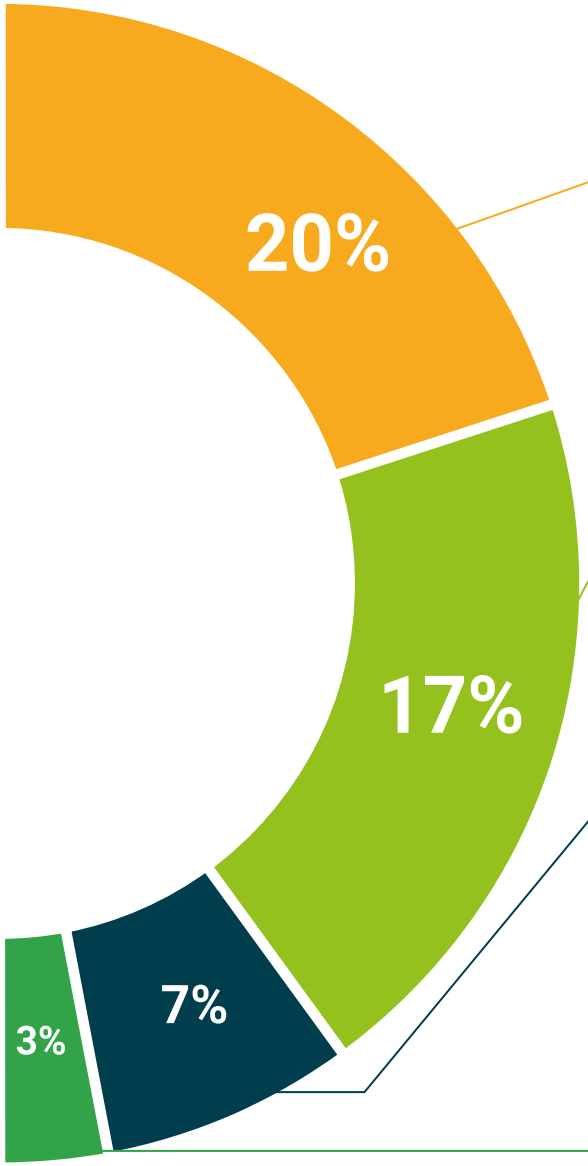
Wir präsentieren die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu festigen.

Dieses einzigartige System für die Präsentation multimedialer Inhalte wurde von Microsoft als „Europäische Erfolgsgeschichte" ausgezeichnet.

Weitere Lektüren

Aktuelle Artikel, Konsensdokumente, internationale Leitfäden... In unserer virtuellen Bibliothek haben Sie Zugang zu allem, was Sie für Ihre Ausbildung benötigen.





Case Studies

Sie werden eine Auswahl der besten *case studies* zu diesem Thema bearbeiten. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.

Testing & Retesting

Während des gesamten Programms werden Ihre Kenntnisse in regelmäßigen Abständen getestet und wiederholt. Wir tun dies auf 3 der 4 Ebenen der Millerschen Pyramide.

Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte *Learning from an Expert* stärkt das Wissen und das Gedächtnis und schafft Vertrauen in unsere zukünftigen schwierigen Entscheidungen.

Kurzanleitungen zum Vorgehen

TECH bietet die wichtigsten Inhalte des Kurses in Form von Arbeitsblättern oder Kurzanleitungen an. Ein synthetischer, praktischer und effektiver Weg, um dem Studenten zu helfen, in seinem Lernen voranzukommen.

07

Lehrkörper

Die Leitung und das Dozententeam dieses Weiterbildenden Masterstudiengangs in Softwaretechnik und -qualität werden von einem Team von Ingenieuren mit langjähriger Erfahrung im Management und in der Entwicklung von technischen und Spezialprojekten durchgeführt. Der professionelle Hintergrund verleiht dieser Fortbildung ein Plus an Qualität, das sich in einer besseren Kontextualisierung der Inhalte durch den Studenten sowie in der Implementierung realer und simulierter praktischer Fälle in die akademische Erfahrung widerspiegeln wird, aber immer mit dem Ziel, ein 100%iges Online-Programm anzubieten, das dynamisch, avantgardistisch und auf der unmittelbaren Realität des Sektors basiert.

Nutzen Sie die Chance, sich in einem der gefragtesten Berufsfelder für Softwareentwickler und Qualitätsexperten zu etablieren"

Internationaler Gastdirektor

Darren Pulsipher ist ein sehr erfahrener **Softwarearchitekt**, ein Innovator mit einer hervorragenden internationalen Erfolgsbilanz in der **Software- und Firmwareentwicklung**. Er verfügt über hoch entwickelte **Kommunikations-**, **Projektmanagement-** und **Geschäftsfähigkeiten**, die es ihm ermöglicht haben, große globale Initiativen zu leiten.

Im Laufe seiner Karriere hatte er auch leitende Positionen mit großer Verantwortung inne, wie z. B. die des **Chefarchitekten für Lösungen für den öffentlichen Sektor** bei der **Intel Corporation**, wo er **moderne Geschäfte, Prozesse und Technologien** für Kunden, Partner und Benutzer im **öffentlichen Sektor** vorantrieb. Darüber hinaus gründete er **Yoly Inc.**, wo er auch als **CEO** fungierte, und arbeitete an der Entwicklung eines **Tools zur Aggregation und Diagnose sozialer Netzwerke** auf der Grundlage von **Software as a Service (SaaS)**, das ***Big Data*** und **Web 2.0**-Technologien nutzt.

Darüber hinaus war er in anderen Unternehmen tätig, unter anderem als **leitender Ingenieur** bei **Dell Technologies**, wo er die **Abteilung *Big Data* in der Cloud** leitete und Teams in den **USA** und **China** führte, um große Projekte zu verwalten und Geschäftsbereiche für eine erfolgreiche Integration umzustrukturieren. Er war auch ***Chief Information Officer*** bei **XanGo**, wo er Projekte wie **Helpdesk-Support**, **Produktionssupport** und **Lösungsentwicklung** leitete.

Zu den vielen Spezialgebieten, in denen er Experte ist, gehören ***Edge-to-Cloud***-Technologie, **Cybersicherheit**, **generative künstliche Intelligenz**, **Softwareentwicklung**, **Netzwerktechnologie**, **Cloud-native Entwicklung** und das **Container-Ökosystem**. Sein Wissen gibt er über den **wöchentlichen Podcast** und die **Newsletter *„Embracing Digital Transformation"*** weiter, die er produziert und präsentiert hat und die Organisationen dabei helfen, die **digitale Transformation** durch den Einsatz von **Menschen**, **Prozessen** und **Technologie** erfolgreich zu meistern.

Hr. Pulsipher, Darren

- Chefarchitekt für Lösungen für den öffentlichen Sektor bei Intel, Kalifornien, USA
- Moderator und Produzent von *„Embracing Digital Transformation"*, Kalifornien
- Gründer und CEO von Yoly Inc., Arkansas
- Leitender Ingenieur bei Dell Technologies, Arkansas
- *Chief Information Officer* bei XanGo, Utah
- Leitender Architekt bei Cadence Design Systems, Kalifornien
- Leitender Projektprozessmanager bei Lucent Technologies, Kalifornien
- Software-Ingenieur bei Cemax-Icon, Kalifornien
- Software-Ingenieur bei ISG Technologies, Kanada
- MBA in Technologiemanagement von der Universität von Phoenix
- Hochschulabschluss in Informatik und Elektrotechnik von der Brigham Young University

Dank TECH werden Sie mit den besten Fachleuten der Welt lernen können"

Leitung

Hr. Molina Molina, Jerónimo

- Leiter der Abteilung Künstliche Intelligenz bei Helphone
- AI Engineer & Software Architect bei NASSAT, Internet Satélite en Movimiento
- Leitender Berater bei Hexa Ingeniero
- Einführung in die künstliche Intelligenz (ML und CV)
- Experte für auf künstlicher Intelligenz basierende Lösungen in den Bereichen Computer Vision, ML/DL und NLP
- Universitätsexperte für Unternehmensgründung und -entwicklung bei Bancaixa und Fundeun
- Computeringenieur von der Universität von Alicante
- Masterstudiengang in Künstliche Intelligenz an der Katholischen Universität von Avila
- MBA Executive beim Europäischen Business Campus Forum

Professoren

Fr. Rodríguez Míguez, Cándida

- Junior Application Developer bei Getronics
- Mitgründerin und City Leader des KI-Netzwerks von Galicien
- Junior Software-Ingenieurin bei Indra
- Web-Entwicklerin bei EDISA
- Hochschulabschluss in Informatik an der Universität von Vigo
- Masterstudiengang in Computertechnik an der Universität von Vigo

Hr. Pi Morell, Oriol

- Funktionsanalytiker bei Fihoca
- Produktverantwortlicher für Hosting und E-Mail bei CDmon
- Funktionsanalytiker und Softwareingenieur bei Atmira und CapGemini
- Dozent bei CapGemini, CapGemini Forms und Atmira
- Hochschulabschluss in technischem Ingenieurwesen in Computer Management von der Autonomen Universität von Barcelona
- Masterstudiengang in Künstliche Intelligenz an der Katholischen Universität von Avila
- Masterstudiengang MBA in Unternehmensführung und Verwaltung von IMF Smart Education
- Masterstudiengang in Management von Informationssystemen von IMF Smart Education
- Aufbaustudiengang in Design Patterns von der Offenen Universität von Katalonien (UOC)

Hr. Martínez Calvo, Francisco Javier

- Technischer Wirtschaftsingenieur mit Spezialisierung auf Elektrizität und Elektronik
- Software-Techniker bei HEXA Ingenieros
- Senior .Net-Entwickler/Architekt von .Net-Lösungen bei Everis
- Software-Analyst/Architekt bei LaLiga
- Microsoft On-Site-Ingenieur bei BBVA
- Freiberuflicher technischer IT-Berater
- Ausbilder für Visual Studio, SqlServer, CCNA (Cisco-Router und -Switch), PHP- und .NET-Webprogrammierung in mehreren Zentren (Salesianos, Maforem, Dreamsoft)
- Technischer Wirtschaftsingenieur mit Spezialisierung auf Elektrizität und Industrieelektronik
- Masterstudiengang Cibernos in .NET, MCAD
- Masterstudiengang Eidos in Fortgeschrittene Programmierung, Expertenniveau
- Masterstudiengang in Web mit Zertifizierungen für Dreamweaver, Fireworks, Flash und ActionScript, Version MX

Hr. Tenrero Morán, Marcos

- DevOps-Ingenieur bei Allot Communications
- Manager für das Management des Anwendungslebenszyklus bei Cegid Meta4
- QA-Automatisierungsingenieur bei Cegid Meta4
- Masterstudiengang in Professioneller Android-Anwendungsentwicklung an der Universität Galileo, Guatemala
- Masterstudiengang in Entwicklung von Cloud-Diensten, Node.Js, JavaScript, HTML5 an der Polytechnischen Universität von Madrid
- Web-Entwicklung mit Angular-CLI (4), Ionic und Node.Js, Meta4 von der Universität Rey Juan Carlos
- Hochschulabschluss in Computertechnik an der Universität Rey Juan Carlos

Fr. Acebes Tamargo, Patricia

- Beraterin mit Spezialisierung auf Big Data
- Abteilung Operations, Arbeit mit Elasticsearch und Kivana bei Sirt
- Online-Forscherin im Bereich *Human Factor* und *AI Applications* am CTIC-Technologiezentrum
- Online-Forscherin im Bereich Wirtschaft am CTIC-Technologiezentrum
- Abteilung für Digitale Gesundheit und Aktives Altern im CTIC-Technologiezentrum
- Abteilung Datenwissenschaft im CTIC-Technologiezentrum
- Promotion in Informatik im Bereich Künstliche Intelligenz an der Polytechnischen Universität von Valencia
- Hochschulabschluss in Wirtschaftswissenschaften an der Universität von Oviedo
- Masterstudiengang in Datenanalyse an der UCJC
- Masterstudiengang in Forschung im Bereich Künstliche Intelligenz an der UNED
- Masterstudiengang in *Blockchain, Smart Contracts* und Kryptowährungen an der Universität von Alcalá
- Aufbaustudiengang in *Blockchain Engineering* von der EADA
- Masterstudiengang in Wirtschaft, Instrumente, Wirtschaftsanalyse an der Universität von Oviedo
- Masterstudiengang in Steuerwesen vom Verband der Wirtschaftswissenschaftler

08
Qualifizierung

Der Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik und -qualität garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Global University ausgestellten Diplom.

Schließen Sie dieses Programm erfolgreich ab und erhalten Sie Ihren Universitätsabschluss ohne lästige Reisen oder Formalitäten"

Mit diesem Programm erwerben Sie den von **TECH Global University**, der größten digitalen Universität der Welt, bestätigten eigenen Titel **Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik und -qualität**

TECH Global University ist eine offizielle europäische Universität, die von der Regierung von Andorra (***Amtsblatt***) öffentlich anerkannt ist. Andorra ist seit 2003 Teil des Europäischen Hochschulraums (EHR). Der EHR ist eine von der Europäischen Union geförderte Initiative, die darauf abzielt, den internationalen Ausbildungsrahmen zu organisieren und die Hochschulsysteme der Mitgliedsländer dieses Raums zu vereinheitlichen. Das Projekt fördert gemeinsame Werte, die Einführung gemeinsamer Instrumente und die Stärkung der Mechanismen zur Qualitätssicherung, um die Zusammenarbeit und Mobilität von Studenten, Forschern und Akademikern zu verbessern.

Dieser eigene Abschluss der **TECH Global University** ist ein europäisches Programm zur kontinuierlichen Weiterbildung und beruflichen Fortbildung, das den Erwerb von Kompetenzen in seinem Wissensgebiet garantiert und dem Lebenslauf des Studenten, der das Programm absolviert, einen hohen Mehrwert verleiht.

TECH ist Mitglied der **American Society for Engineering Education (ASEE)**, einer Vereinigung, der die wichtigsten internationalen Experten im Bereich Ingenieurwesen angehören. Diese Auszeichnung stärkt ihre Führungsrolle in der akademischen und technologischen Entwicklung im Ingenieurwesen.

Akkredditierung/Mitgliedschaft ASEE

Titel: **Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik und -qualität**

Modalität: **online**

Dauer: **2 Jahre**

Akkreditierung: **120 ECTS**

tech global university

Hr./Fr. _____, mit der Ausweis-Nr. _____ hat erfolgreich bestanden und den folgenden Abschluss erworben:

Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik und -qualität

Es handelt sich um einen eigenen Abschluss mit einer Dauer von 3.000 Stunden, was 120 ECTS entspricht, mit Anfangsdatum am dd/mm/aaaa und Enddatum am dd/mm/aaaa.

TECH Global University ist eine von der Regierung Andorras am 31. Januar 2024 offiziell anerkannte Universität, die dem Europäischen Hochschulraum (EHR) angehört.

Andorra la Vella, den 28. Februar 2024

Dr. Pedro Navarro Illana
Rektor

einzigartiger Code TECH: AFWOR23S techtitute.com/titel

Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik und -qualität

Allgemeiner Aufbau des Lehrplans

Kurs	Modul	ECTS	Kategorie
1°	Softwarequalität. TRL-Entwicklungsstufen	6	OB
1°	Software-Projektentwicklung. Funktionelle und technische Dokumentation	6	OB
1°	Software-Testing. Testautomatisierung	6	OB
1°	Methodologien des Software-Projektmanagements. Waterfal-Methoden versus agile Methoden	6	OB
1°	TDD (*Test Driven Developement*). Testgetriebener Softwareentwurf	6	OB
1°	DevOps. Management der Softwarequalität	6	OB
1°	DevOps und kontinuierliche Integration. Fortgeschrittene praktische Lösungen in der Softwareentwicklung	6	OB
1°	Datenbankdesign. Standardisierung und Leistung. Softwarequalität	6	OB
1°	Entwurf skalierbarer Architekturen. Architektur im Lebenszyklus der Software	6	OB
1°	ISO-, IEC 9126-Qualitätskriterien. Metriken zur Softwarequalität	6	OB
2°	Methodik, Entwicklung und Qualität in der Softwaretechnik	6	OB
2°	Software-Projektmanagement	6	OB
2°	Plattformen für die Softwareentwicklung	6	OB
2°	Web-Client-Computing	6	OB
2°	Web-Server-Computing	6	OB
2°	Sicherheitsmanagement	6	OB
2°	Softwaresicherheit	6	OB
2°	Webserver-Verwaltung	6	OB
2°	Sicherheitsprüfung	6	OB
2°	Sicherheit bei Online-Anwendungen	6	OB

Dr. Pedro Navarro Illana
Rektor

tech global university

*Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH Global University die notwendigen Vorkehrungen treffen, um diese gegen eine zusätzliche Gebühr zu beschaffen.

tech global university

Weiterbildender Masterstudiengang mit Spezialisierung

Softwaretechnik und -qualität

- Modalität: **online**
- Dauer: **2 Jahre**
- Qualifizierung: **TECH Global University**
- Akkreditierung: **120 ECTS**
- Zeitplan: **in Ihrem eigenen Tempo**
- Prüfungen: **online**

Weiterbildender Masterstudiengang mit Spezialisierung

Softwaretechnik und -qualität

Akkredditierung/Mitgliedschaft

ASEE American Society for Engineering Education

tech global university