

Weiterbildender Masterstudiengang mit Spezialisierung Softwaretechnik



Weiterbildender Masterstudiengang mit Spezialisierung Softwaretechnik

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

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

Index

01

Präsentation des Programms

Seite 4

02

Warum an der TECH studieren?

Seite 8

03

Lehrplan

Seite 12

04

Lehrziele

Seite 36

05

Karrieremöglichkeiten

Seite 42

06

Studienmethodik

Seite 46

07

Lehrkörper

Seite 56

08

Qualifizierung

Seite 60

01

Präsentation des Programms

Die Nachfrage nach Software ist in den letzten Jahren exponentiell gestiegen. Das Aufkommen neuer digitaler Plattformen, fortschrittlicher Hardware und die zunehmende Virtualisierung alltäglicher Prozesse haben die Softwareingenieure vor ständige Herausforderungen gestellt. Angesichts eines Publikums, das sich zunehmend an neue Technologien gewöhnt und höhere Erwartungen hat, müssen sich die Entwickler anpassen und Lösungen schaffen, die den Anforderungen des Marktes gerecht werden. Dies erfordert technische Kenntnisse in mehreren Bereichen des IT-Wissens. Aus diesem Grund hat TECH dieses Programm konzipiert, das die wesentlichen Fähigkeiten zur Bewältigung der heutigen Herausforderungen vermittelt, darunter Softwareplanung und -design, Implementierung von Sicherheitsmaßnahmen, Methoden zur Wiederverwendung von Software und Webserververwaltung.



“

Spielen Sie eine Schlüsselrolle in der technologischen Zukunft. Spezialisieren Sie sich auf die Softwaretechnik und schaffen Sie transformative Systeme“

Die Softwaretechnik ist eine der wichtigsten Disziplinen in der heutigen technologischen Ära, da praktisch jeder Mensch täglich mit digitalen Geräten interagiert und diese einen erheblichen Einfluss auf die globale Wirtschaft haben. In diesem Zusammenhang kommt der Rolle der Softwareingenieure große Bedeutung zu, da die Kombination von technischem Wissen und innovativen Methoden die Entwicklung effizienter Lösungen ermöglicht, die an die Bedürfnisse des Marktes angepasst sind. Eine hohe Spezialisierung in diesem Bereich ist unerlässlich, um den Erfolg technologischer Projekte zu gewährleisten. Aus diesem Grund hat TECH dieses komplette akademische Programm entwickelt, das den Fachleuten die fortschrittlichsten und modernsten Werkzeuge bietet.

Der Studiengang befasst sich mit den Schlüsselkonzepten der Softwaretechnik und geht dabei auf grundlegende Aspekte wie Anwendungsentwicklung, technologische Plattformen und die fortschrittlichsten Methoden des Projektmanagements ein. Der technische Teil nimmt eine wichtige Rolle ein, indem er die Studenten von der Konzeption von Ideen bis zu deren Umsetzung in funktionale Systeme führt. Eine einzigartige Gelegenheit, die Softwarearchitektur und die Schaffung innovativer Lösungen zu verstehen und gleichzeitig die neuesten Trends in der Programmierung, der mobilen Entwicklung und den Cloud-Anwendungen zu erkunden. Um die Führungsqualitäten zu stärken, umfasst der Inhalt außerdem die aktuellsten Konzepte des Projektmanagements und der agilen Methodik, ein Mehrwert für erfahrene Fachleute, die ihre Fähigkeiten aktualisieren möchten, oder für diejenigen, die zum ersten Mal strategische Positionen erreichen wollen.

Einer der Hauptvorteile dieses Programms ist sein 100%iger Online-Modus, der Fahrten oder die Anpassung an starre Zeitpläne überflüssig macht. So können die Studenten ihr Studium selbst verwalten und ihre Zeit und ihr Lerntempo nach ihren eigenen Bedürfnissen einteilen, was ideal ist, um es mit anderen täglichen Verpflichtungen zu kombinieren.

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

- ♦ Die Entwicklung von Fallstudien, die von Experten der Softwaretechnik vorgestellt werden
- ♦ Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren wissenschaftlichen und praktischen Informationen
- ♦ Die praktischen Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens durchgeführt werden kann
- ♦ Sein besonderer Schwerpunkt liegt auf innovativen Methoden im Bereich der Softwaretechnik
- ♦ 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



Treiben Sie die technologische Entwicklung mit effizientem Management in der Softwaretechnik voran, dem Schlüssel zum Unternehmenserfolg“



Studieren Sie in Ihrem eigenen Tempo mit einem 100%igen Online-Programm, das jederzeit und von jedem Ort der Welt aus verfügbar ist“

Das Dozententeam besteht aus Fachleuten aus dem Bereich der Softwaretechnik, die ihre Erfahrungen in dieses Programm einbringen, sowie aus anerkannten Fachleuten 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.

Erhalten Sie Zugang zur innovativsten und aktuellsten Methodik für das Erlernen der Softwaretechnik, die von TECH entwickelt wurde.

Vertiefen Sie Ihr technisches Wissen durch eine Vielzahl von praktischen Ressourcen, die in diesem Programm enthalten sind.



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 elf 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.

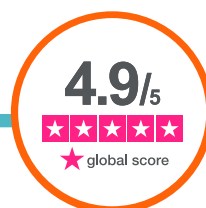
Führend in Beschäftigungsfähigkeit

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 dieses Studiengangs in Softwaretechnik wurde von einem Expertenteam für Informatik und technologische Entwicklung sorgfältig ausgearbeitet. Darüber hinaus werden während des gesamten Studiengangs die grundlegenden Aspekte der Softwaretechnik umfassend behandelt, von der Anwendungsplanung, dem Entwurf und der Entwicklung bis hin zu fortgeschrittener Computersicherheit und effizienter Serververwaltung. Die Studenten werden spezialisierte technische Fähigkeiten erwerben und lernen, komplexe Projekte zu leiten, die den Anforderungen eines sich ständig weiterentwickelnden Marktes entsprechen.





“

*Tragen Sie zum technologischen Fortschritt bei,
indem Sie innovative und wirksame Lösungen
im Bereich der Softwaretechnik entwickeln“*

Modul 1. Methodik, Entwicklung und Qualität in der Softwaretechnik

- 1.1. Einführung in die Softwaretechnik
 - 1.1.1. Einführung
 - 1.1.2. Die Softwarekrise
 - 1.1.3. Unterschiede zwischen Softwaretechnik und Informatik
 - 1.1.4. Ethik und berufliche Verantwortung in der Softwaretechnik
 - 1.1.5. Software-Fabriken
- 1.2. Der Softwareentwicklungsprozess
 - 1.2.1. Definition
 - 1.2.2. Softwareprozessmodell
 - 1.2.3. Der einheitliche Softwareentwicklungsprozess
- 1.3. Objektorientierte Softwareentwicklung
 - 1.3.1. Einführung
 - 1.3.2. Grundsätze der Objektorientierung
 - 1.3.3. Definition von Objekt
 - 1.3.4. Definition der Klasse
 - 1.3.5. Objektorientierte Analyse vs. objektorientiertes Design
- 1.4. Modellgestützte Softwareentwicklung
 - 1.4.1. Die Notwendigkeit der Modellierung
 - 1.4.2. Modellierung von Softwaresystemen
 - 1.4.3. Modellierung von Objekten
 - 1.4.4. UML
 - 1.4.5. CASE-Tools
- 1.5. Anwendungsmodellierung und Entwurfsmuster mit UML
 - 1.5.1. Fortgeschrittene Anforderungsmodellierung
 - 1.5.2. Erweiterte statische Modellierung
 - 1.5.3. Erweiterte dynamische Modellierung
 - 1.5.4. Modellierung von Bauteilen
 - 1.5.5. Einführung in Entwurfsmuster mit UML
 - 1.5.6. Adapter
 - 1.5.7. Factory
 - 1.5.8. Singleton
 - 1.5.9. Strategy
 - 1.5.10. Composite
 - 1.5.11. Facade
 - 1.5.12. Observer
- 1.6. Modellgestütztes Engineering
 - 1.6.1. Einführung
 - 1.6.2. Metamodellierung von Systemen
 - 1.6.3. MDA
 - 1.6.4. DSL
 - 1.6.5. Modellverfeinerungen mit OCL
 - 1.6.6. Modellumwandlungen
- 1.7. Ontologien in der Softwaretechnik
 - 1.7.1. Einführung
 - 1.7.2. Ontologietechnik
 - 1.7.3. Anwendung von Ontologien in der Softwaretechnik
- 1.8. Agile Methoden für die Software-Entwicklung, Scrum
 - 1.8.1. Was ist Software-Agilität?
 - 1.8.2. Das agile Manifest
 - 1.8.3. Der agile Projektfahrplan
 - 1.8.4. Der Product Owner
 - 1.8.5. Anwenderberichte
 - 1.8.6. Agile Planung und Schätzung
 - 1.8.7. Messungen in der agilen Entwicklung
 - 1.8.8. Einführung in Scrum
 - 1.8.9. Rollen
 - 1.8.10. Product Backlog
 - 1.8.11. Sprint
 - 1.8.12. Die Sitzungen
- 1.9. Die Methodik der Entwicklung der *Lean-Software*
 - 1.9.1. Einführung
 - 1.9.2. Kanban

- 1.10. Qualität und Verbesserung von Softwareprozessen
 - 1.10.1. Einführung
 - 1.10.2. Software-Messung
 - 1.10.3. Software-Tests
 - 1.10.4. Modell der Software-Prozessqualität: CMMI

Modul 2. Software-Projektmanagement

- 2.1. Grundlegende Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
 - 2.1.1. Was ist ein Projekt?
 - 2.1.2. Gemeinsame Methodik
 - 2.1.3. Was ist Projektmanagement?
 - 2.1.4. Was ist ein Projektplan?
 - 2.1.5. Vorteile
 - 2.1.6. Projektlebenszyklus
 - 2.1.7. Prozessgruppen oder Lebenszyklus des Projektmanagements
 - 2.1.8. Die Beziehung zwischen Prozessgruppen und Wissensgebieten
 - 2.1.9. Beziehung zwischen Produkt- und Projektlebenszyklus
- 2.2. Inbetriebnahme und Planung
 - 2.2.1. Von der Idee zum Projekt
 - 2.2.2. Entwicklung der Projektcharta
 - 2.2.3. Projekt-Kick-off-Meeting
 - 2.2.4. Aufgaben, Kenntnisse und Fähigkeiten im Gründungsprozess
 - 2.2.5. Der Projektplan
 - 2.2.6. Entwicklung des Basisplans. Schritte
 - 2.2.7. Aufgaben, Kenntnisse und Fähigkeiten im Planungsprozess
- 2.3. Management von *Stakeholdern* und Umfang
 - 2.3.1. Identifizierung von *Stakeholdern*
 - 2.3.2. Entwicklung des *Stakeholder*-Management-Plans
 - 2.3.3. Management der Einbindung von *Stakeholdern*
 - 2.3.4. Überwachung des Engagements der *Stakeholder*
 - 2.3.5. Das Projektziel
 - 2.3.6. Umfangsmanagement und sein Plan
 - 2.3.7. Erfassen von Anforderungen
 - 2.3.8. Definieren Sie den Geltungsbereich
 - 2.3.9. Erstellen des Projektstrukturplans
 - 2.3.10. Überprüfung und Kontrolle des Umfangs
- 2.4. Die Entwicklung des Zeitplans
 - 2.4.1. Zeitmanagement und sein Plan
 - 2.4.2. Definieren der Aktivitäten
 - 2.4.3. Festlegung der Reihenfolge der Aktivitäten
 - 2.4.4. Schätzung der Ressourcen für die Aktivitäten
 - 2.4.5. Geschätzte Dauer der Aktivitäten
 - 2.4.6. Entwicklung des Zeitplans und Berechnung des kritischen Pfades
 - 2.4.7. Zeitplan-Kontrolle
- 2.5. Budgetentwicklung und Risikobewältigung
 - 2.5.1. Schätzung der Kosten
 - 2.5.2. Entwicklung des Budgets und der S-Kurve
 - 2.5.3. Kostenkontrolle und *Earned-Value*-Methode
 - 2.5.4. Risikokonzepte
 - 2.5.5. Wie man eine Risikoanalyse durchführt
 - 2.5.6. Die Entwicklung des Reaktionsplans
- 2.6. Qualitätsmanagement
 - 2.6.1. Planung der Qualität
 - 2.6.2. Qualitätssicherung
 - 2.6.3. Qualitätskontrolle
 - 2.6.4. Grundlegende statistische Konzepte
 - 2.6.5. Instrumente des Qualitätsmanagements
- 2.7. Kommunikation und Personalwesen
 - 2.7.1. Planung des Kommunikationsmanagements
 - 2.7.2. Analyse der Kommunikationsanforderungen
 - 2.7.3. Technologie der Kommunikation
 - 2.7.4. Kommunikationsmodelle

- 2.7.5. Kommunikationsmethoden
- 2.7.6. Plan für das Kommunikationsmanagement
- 2.7.7. Verwaltung der Kommunikation
- 2.7.8. Verwaltung des Personalwesens
- 2.7.9. Hauptakteure und ihre Rolle in den Projekten
- 2.7.10. Arten von Organisationen
- 2.7.11. Projektorganisation
- 2.7.12. Das Projektteam
- 2.8. Beschaffung
 - 2.8.1. Der Beschaffungsprozess
 - 2.8.2. Planung
 - 2.8.3. Beschaffung von Lieferanten und Einholung von Angeboten
 - 2.8.4. Vergabe des Auftrags
 - 2.8.5. Vertragsverwaltung
 - 2.8.6. Verträge
 - 2.8.7. Arten von Verträgen
 - 2.8.8. Vertragsverhandlungen
- 2.9. Durchführung, Überwachung und Kontrolle sowie Abschluss
 - 2.9.1. Prozessgruppen
 - 2.9.2. Projektdurchführung
 - 2.9.3. Projektüberwachung und -kontrolle
 - 2.9.4. Abschluss des Projekts
- 2.10. Berufliche Verantwortung
 - 2.10.1. Berufliche Verantwortung
 - 2.10.2. Merkmale der sozialen und beruflichen Verantwortung
 - 2.10.3. Ethischer Kodex für Projektleiter
 - 2.10.4. Verantwortung vs. PMP®
 - 2.10.5. Beispiele für Rechenschaftspflicht
 - 2.10.6. Vorteile der Professionalisierung

Modul 3. Plattformen für die Softwareentwicklung

- 3.1. Einführung in die Entwicklung von Applikationen
 - 3.1.1. Desktop-Applikationen
 - 3.1.2. Programmiersprache
 - 3.1.3. Integrierte Entwicklungsumgebungen
 - 3.1.4. Webanwendungen
 - 3.1.5. Mobile Anwendungen
 - 3.1.6. *Cloud*-Anwendungen
- 3.2. Anwendungsentwicklung und grafische Oberfläche in Java
 - 3.2.1. Integrierte Entwicklungsumgebungen für Java
 - 3.2.2. Wichtigste IDEs für Java
 - 3.2.3. Einführung in die Eclipse-Entwicklungsplattform
 - 3.2.4. Einführung in die NetBeans-Entwicklungsplattform
 - 3.2.5. *Controller-View*-Modell für grafische Benutzeroberflächen
 - 3.2.6. Entwerfen einer grafischen Benutzeroberfläche in Eclipse
 - 3.2.7. Entwerfen einer grafischen Benutzeroberfläche in NetBeans
- 3.3. Fehlersuche und Testen in Java
 - 3.3.1. Testen und Debuggen von Java-Programmen
 - 3.3.2. Fehlersuche in Eclipse
 - 3.3.3. Fehlersuche in NetBeans
- 3.4. Anwendungsentwicklung und grafische Oberfläche in .NET
 - 3.4.1. Net Framework
 - 3.4.2. Komponenten der .NET-Entwicklungsplattform
 - 3.4.3. Visual Studio .NET
 - 3.4.4. .NET GUI-Werkzeuge
 - 3.4.5. Die grafische Benutzeroberfläche mit Windows Presentation Foundation
 - 3.4.6. Debuggen und Kompilieren einer WPF-Anwendung

- 3.5. Programmierung für .NET-Netzwerke
 - 3.5.1. Einführung in die .NET-Netzwerkprogrammierung
 - 3.5.2. .NET-Anfragen und -Antworten
 - 3.5.3. Verwendung von .NET-Anwendungsprotokollen
 - 3.5.4. Sicherheit in der .NET-Netzwerkprogrammierung
- 3.6. Entwicklungsumgebungen für mobile Anwendungen
 - 3.6.1. Mobile Anwendungen
 - 3.6.2. Mobile Android-Anwendungen
 - 3.6.3. Schritte für die Android-Entwicklung
 - 3.6.4. Die Android Studio IDE
- 3.7. Entwicklung von Anwendungen in der Android Studio-Umgebung
 - 3.7.1. Installieren und Starten von Android Studio
 - 3.7.2. Ausführen einer Android-Anwendung
 - 3.7.3. Entwicklung der grafischen Oberfläche in Android Studio
 - 3.7.4. Starten von Aktivitäten in Android Studio
- 3.8. Debuggen und Veröffentlichen von Android-Anwendungen
 - 3.8.1. Fehlersuche in einer Anwendung in Android Studio
 - 3.8.2. Speichern von Anwendungen in Android Studio
 - 3.8.3. Veröffentlichung einer Anwendung auf Google Play
- 3.9. Entwicklung von Anwendungen für die *Cloud*
 - 3.9.1. Cloud Computing
 - 3.9.2. Cloud-Ebenen: SaaS, PaaS, IaaS
 - 3.9.3. Wichtigste *Cloud*-Entwicklungsplattformen
 - 3.9.4. Bibliografische Referenzen
- 3.10. Einführung in die Google Cloud Platform
 - 3.10.1. Grundlagen der Google Cloud Platform
 - 3.10.2. Google Cloud Platform-Dienste
 - 3.10.3. Google Cloud Platform-Werkzeuge

Modul 4. Web-Client-Computing

- 4.1. Einführung in HTML
 - 4.1.1. Aufbau eines Dokuments
 - 4.1.2. Farbe
 - 4.1.3. Text
 - 4.1.4. Hypertext-Links
 - 4.1.5. Bilder
 - 4.1.6. Listen
 - 4.1.7. Tabellen
 - 4.1.8. Rahmen (*Frames*)
 - 4.1.9. Formulare
 - 4.1.10. Spezifische Elemente für mobile Technologien
 - 4.1.11. Ausgediente Elemente
- 4.2. *Cascading Style Sheets* (CSS)
 - 4.2.1. Elemente und Struktur einer Formatvorlage
 - 4.2.1.1. Erstellung von Stilvorlagen
 - 4.2.1.2. Anwendung von Stilen. Selektoren
 - 4.2.1.3. Stilvererbung und Kaskadierung
 - 4.2.1.4. Seitenformatierung mit Formatvorlagen
 - 4.2.1.5. Seitenstruktur mit Hilfe von Stilen. Das Kastenmodell
 - 4.2.2. Gestaltung von Stilen für verschiedene Geräte
 - 4.2.3. Arten von Formatvorlagen: statisch und dynamisch. Pseudo-Klassen
 - 4.2.4. Bewährte Verfahren bei der Verwendung von Formatvorlagen
- 4.3. Einführung und Geschichte von JavaScript
 - 4.3.1. Einführung
 - 4.3.2. Geschichte von JavaScript
 - 4.3.3. Entwicklungsumgebung, die wir verwenden werden

- 4.4. Grundbegriffe der Webprogrammierung
 - 4.4.1. Grundlegende JavaScript-Syntax
 - 4.4.2. Primitive Datentypen und Operatoren
 - 4.4.3. Variablen und Domänen
 - 4.4.4. *Textstrings* und *Template Literals*
 - 4.4.5. Zahlen und Boolesche Werte
 - 4.4.6. Vergleiche
- 4.5. Komplexe JavaScript-Strukturen
 - 4.5.1. Vektoren oder Arrays und Objekte
 - 4.5.2. Sets
 - 4.5.3. Karten
 - 4.5.4. Disjunktionen
 - 4.5.5. Schleifen
- 4.6. Funktionen und Objekte
 - 4.6.1. Funktionsdefinition und -aufruf
 - 4.6.2. Argumente
 - 4.6.3. Pfeil-Funktionen
 - 4.6.4. Rückruf-Funktionen oder *Callback*
 - 4.6.5. Funktionen höherer Ordnung
 - 4.6.6. Wörtliche Objekte
 - 4.6.7. Das Objekt *this*
 - 4.6.8. Objekte als Namensräume: das *Math*-Objekt und das *Date*-Objekt
- 4.7. Das Dokumentenobjektmodell (DOM)
 - 4.7.1. Was ist DOM?
 - 4.7.2. Ein bisschen Geschichte
 - 4.7.3. Navigieren und Abrufen von Elementen
 - 4.7.4. Ein virtuelles DOM mit JSDOM
 - 4.7.5. Abfrage-Selektoren oder *Query Selectors*
 - 4.7.6. Navigation mittels Eigenschaften
 - 4.7.7. Zuweisung von Attributen zu Elementen
 - 4.7.8. Erstellen und Ändern von Knoten
 - 4.7.9. Aktualisieren des Stils von DOM-Elementen
- 4.8. Moderne Webentwicklung
 - 4.8.1. Ereignisgesteuerter Ablauf und *Listeners*
 - 4.8.2. Moderne Web-Toolkits und Ausrichtungssysteme
 - 4.8.3. Strikter JavaScript-Modus
 - 4.8.4. Mehr über Funktionen
 - 4.8.5. Asynchrone Funktionen und Versprechen
 - 4.8.6. Closures
 - 4.8.7. Funktionale Programmierung
 - 4.8.8. OOP in JavaScript
- 4.9. Web-Benutzbarkeit
 - 4.9.1. Einführung in die Benutzerfreundlichkeit
 - 4.9.2. Definition von Benutzerfreundlichkeit
 - 4.9.3. Bedeutung des nutzerzentrierten Webdesigns
 - 4.9.4. Unterschiede zwischen Barrierefreiheit und Benutzerfreundlichkeit
 - 4.9.5. Vorteile und Probleme bei der Kombination von Barrierefreiheit und Benutzerfreundlichkeit
 - 4.9.6. Vorteile und Schwierigkeiten bei der Umsetzung von nutzbaren Websites
 - 4.9.7. Methoden zur Benutzerfreundlichkeit
 - 4.9.8. Analyse der Benutzeranforderungen
 - 4.9.9. Konzeptionelle Gestaltungsgrundsätze. Benutzerorientiertes *Prototyping*
 - 4.9.10. Leitlinien für die Erstellung von nutzbaren Websites
 - 4.9.10.1. Jakob Nielsens Leitlinien zur Benutzerfreundlichkeit
 - 4.9.10.2. Bruce Tognazzinis Leitlinien zur Benutzerfreundlichkeit
 - 4.9.11. Bewertung der Benutzbarkeit
- 4.10. Barrierefreiheit im Internet
 - 4.10.1. Einführung
 - 4.10.2. Definition von Barrierefreiheit im Internet
 - 4.10.3. Arten von Behinderungen
 - 4.10.3.1. Vorübergehende oder dauerhafte Behinderungen
 - 4.10.3.2. Visuelle Beeinträchtigungen
 - 4.10.3.3. Beeinträchtigungen des Hörvermögens
 - 4.10.3.4. Motorische Behinderungen

- 4.10.3.5. Neurologische oder kognitive Behinderungen
- 4.10.3.6. Altersbedingte Schwierigkeiten
- 4.10.3.7. Umweltbedingte Einschränkungen
- 4.10.3.8. Hindernisse für den Zugang zum Internet
- 4.10.4. Technische Hilfsmittel und unterstützende Produkte zur Überwindung von Barrieren
 - 4.10.4.1. Hilfsmittel für Blinde
 - 4.10.4.2. Hilfsmittel für Menschen mit Sehschwäche
 - 4.10.4.3. Hilfsmittel für Menschen mit Farbenblindheit
 - 4.10.4.4. Hilfsmittel für Hörgeschädigte
 - 4.10.4.5. Hilfsmittel für Menschen mit motorischen Behinderungen
 - 4.10.4.6. Hilfsmittel für Menschen mit kognitiven und neurologischen Behinderungen
- 4.10.5. Vorteile und Schwierigkeiten bei der Umsetzung der Barrierefreiheit im Internet
- 4.10.6. Vorschriften und Normen für die Barrierefreiheit im Internet
- 4.10.7. Regulierungsstellen für die Barrierefreiheit im Internet
- 4.10.8. Vergleich von Normen und Standards
- 4.10.9. Leitlinien für die Einhaltung von Vorschriften und Normen
 - 4.10.9.1. Beschreibung der wichtigsten Leitlinien (Bilder, Links, Videos usw.)
 - 4.10.9.2. Leitlinien für eine barrierefreie Navigation
 - 4.10.9.2.1. Wahrnehmbarkeit
 - 4.10.9.2.2. Operationalität
 - 4.10.9.2.3. Nachvollziehbarkeit
 - 4.10.9.2.4. Robustheit
- 4.10.10. Beschreibung des Prozesses der Webzugänglichkeitskonformität
- 4.10.11. Konformitätsstufen
- 4.10.12. Konformitätskriterien
- 4.10.13. Anforderungen an die Konformität
- 4.10.14. Methodik zur Bewertung der Zugänglichkeit von Websites

Modul 5. Webserver-Computing

- 5.1. Einführung in die Programmierung im Server: PHP
 - 5.1.1. Grundlagen der Programmierung im Server
 - 5.1.2. Grundlegende PHP-Syntax
 - 5.1.3. Generierung von HTML-Inhalten mit PHP
 - 5.1.4. Entwicklungs- und Testumgebungen: XAMPP
- 5.2. PHP für Fortgeschrittene
 - 5.2.1. PHP-Kontrollstrukturen
 - 5.2.2. PHP-Funktionen
 - 5.2.3. *Array*-Verarbeitung in PHP
 - 5.2.4. *String*-Verarbeitung in PHP
 - 5.2.5. Objektorientierung in PHP
- 5.3. Datenmodelle
 - 5.3.1. Begriff der Daten. Lebenszyklus der Daten
 - 5.3.2. Datentypen
 - 5.3.2.1. Grundlegende
 - 5.3.2.2. Register
 - 5.3.2.3. Dynamische
- 5.4. Das relationale Modell
 - 5.4.1. Beschreibung
 - 5.4.2. Entitäten und Entitätstypen
 - 5.4.3. Datenelemente. Attribute
 - 5.4.4. Beziehungen: Typen, Untertypen, Kardinalität
 - 5.4.5. Schlüssel. Schlüsselarten
 - 5.4.6. Normalisierung. Normale Formen
- 5.5. Aufbau des logischen Datenmodells
 - 5.5.1. Spezifikation der Tabelle
 - 5.5.2. Definition von Spalten
 - 5.5.3. Wichtige Spezifikation
 - 5.5.4. Umwandlung in Normalformen. Abhängigkeit

- 5.6. Das physische Datenmodell. Dateien
 - 5.6.1. Beschreibung der Datendateien
 - 5.6.2. Datentypen
 - 5.6.3. Zugriffsmodi
 - 5.6.4. Organisation von Dateien
- 5.7. Zugriff auf Datenbanken von PHP aus
 - 5.7.1. Einführung in MariaDB
 - 5.7.2. Arbeiten mit einer MariaDB-Datenbank: die SQL-Sprache
 - 5.7.3. Zugriff auf die MariaDB-Datenbank von PHP aus
 - 5.7.4. Einführung in MySQL
 - 5.7.5. Arbeiten mit einer MySQL-Datenbank: die SQL-Sprache
 - 5.7.6. Zugriff auf die MySQL-Datenbank über PHP
- 5.8. Interaktion mit dem Client über PHP
 - 5.8.1. PHP-Formulare
 - 5.8.2. Cookies
 - 5.8.3. Handhabung von Sitzungen
- 5.9. Architektur von Webanwendungen
 - 5.9.1. Das *Model-View-Controller*-Muster
 - 5.9.2. Controller
 - 5.9.3. Modell
 - 5.9.4. Ansicht
- 5.10. Einführung in Webdienste
 - 5.10.1. Einführung in XML
 - 5.10.2. Dienstorientierte Architekturen (SOA): Webdienste
 - 5.10.3. Erstellung von SOAP- und REST-Webdiensten
 - 5.10.4. Das SOAP-Protokoll
 - 5.10.5. Das REST-Protokoll

Modul 6. Sicherheitsmanagement

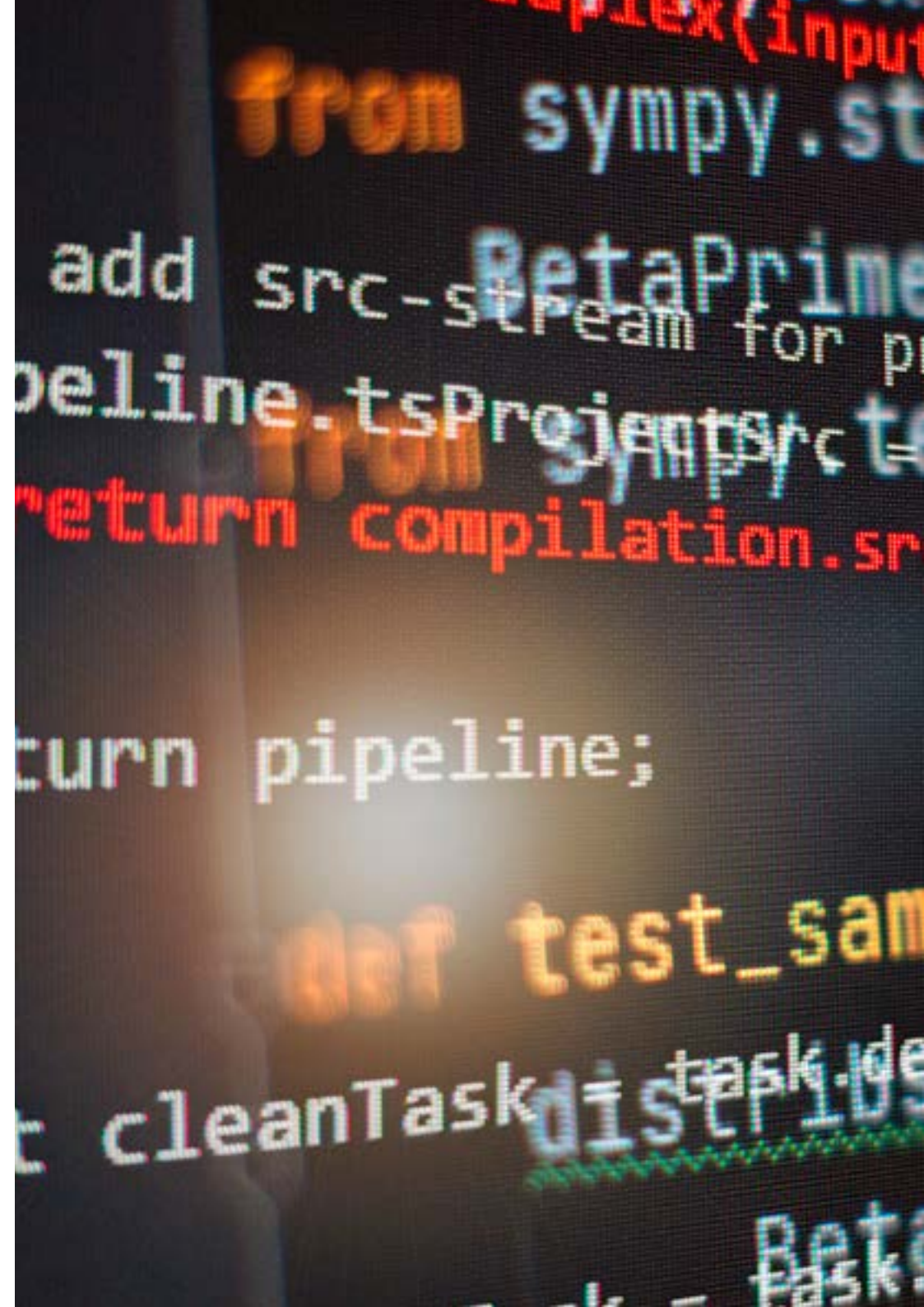
- 6.1. Informationssicherheit
 - 6.1.1. Einführung
 - 6.1.2. Die Sicherheit von Informationen setzt Vertraulichkeit, Integrität und Verfügbarkeit voraus
 - 6.1.3. Sicherheit ist eine wirtschaftliche Frage
 - 6.1.4. Sicherheit ist ein Prozess
 - 6.1.5. Die Klassifizierung von Informationen
 - 6.1.6. Informationssicherheit ist Risikomanagement
 - 6.1.7. Sicherheit ist mit Sicherheitskontrollen verbunden
 - 6.1.8. Sicherheit ist sowohl physisch als auch logisch
 - 6.1.9. Sicherheit betrifft Menschen
- 6.2. Die Fachkraft für Informationssicherheit
 - 6.2.1. Einführung
 - 6.2.2. Informationssicherheit als Beruf
 - 6.2.3. Zertifizierungen (ISC)2
 - 6.2.4. Die Norm ISO 27001
 - 6.2.5. Bewährte Sicherheitspraktiken im IT-Service-Management
 - 6.2.6. Reifegradmodelle für die Informationssicherheit
 - 6.2.7. Andere Zertifizierungen, Standards und professionelle Ressourcen
- 6.3. Zugangskontrolle
 - 6.3.1. Einführung
 - 6.3.2. Anforderungen an die Zugangskontrolle
 - 6.3.3. Authentifizierungsmechanismen
 - 6.3.4. Genehmigungsverfahren
 - 6.3.5. Zugang zu Buchhaltung und Rechnungsprüfung
 - 6.3.6. *Triple A*-Technologien

- 6.4. Programme, Verfahren und Richtlinien zur Informationssicherheit
 - 6.4.1. Einführung
 - 6.4.2. Programme für das Sicherheitsmanagement
 - 6.4.3. Risikomanagement
 - 6.4.4. Gestaltung der Sicherheitspolitik
- 6.5. Pläne zur Aufrechterhaltung des Geschäftsbetriebs
 - 6.5.1. Einführung in BCPs
 - 6.5.2. Phase I und II
 - 6.5.3. Phase III und IV
 - 6.5.4. Aufrechterhaltung der BCP
- 6.6. Verfahren für den korrekten Schutz des Unternehmens
 - 6.6.1. DMZ-Netzwerke
 - 6.6.2. Systeme zur Erkennung von Eindringlingen
 - 6.6.3. Zugriffskontrolllisten
 - 6.6.4. Vom Angreifer lernen: Honeypot
- 6.7. Sicherheitsarchitektur. Prävention
 - 6.7.1. Überblick. Aktivitäten und Schichtenmodell
 - 6.7.2. Perimeter-Verteidigung (Firewalls, WAFs, IPS usw.)
 - 6.7.3. Endpunktschutz (Geräte, Server und Dienste)
- 6.8. Sicherheitsarchitektur. Erkennung
 - 6.8.1. Überblick über Erkennung und Überwachung
 - 6.8.2. Logs, verschlüsselte Verkehrsunterbrechung, Aufzeichnung und Siems
 - 6.8.3. Warnungen und Informationen
- 6.9. Sicherheitsarchitektur. Reaktion
 - 6.9.1. Reaktion. Produkte, Dienstleistungen und Ressourcen
 - 6.9.2. Management von Vorfällen
 - 6.9.3. CERTS und CSIRTs
- 6.10. Sicherheitsarchitektur. Wiederherstellung
 - 6.10.1. Ausfallsicherheit, Konzepte, Geschäftsanforderungen und Vorschriften
 - 6.10.2. IT-Resilienz-Lösungen
 - 6.10.3. Krisenmanagement und *Governance*

Modul 7. Sicherheit von Informationssystemen

- 7.1. Ein Überblick über Sicherheit, Kryptographie und klassische Kryptoanalyse
 - 7.1.1. Computersicherheit: Historische Perspektive
 - 7.1.2. Aber was genau ist mit Sicherheit gemeint?
 - 7.1.3. Geschichte der Kryptographie
 - 7.1.4. Substitutions-Chiffren
 - 7.1.5. Fallstudie: Die Enigma-Maschine
- 7.2. Symmetrische Kryptographie
 - 7.2.1. Einführung und grundlegende Terminologie
 - 7.2.2. Symmetrische Verschlüsselung
 - 7.2.3. Betriebsarten
 - 7.2.4. DES
 - 7.2.5. Der neue AES-Standard
 - 7.2.6. Stream-Verschlüsselung
 - 7.2.7. Kryptoanalyse
- 7.3. Asymmetrische Kryptographie
 - 7.3.1. Die Ursprünge der *Public-Key*-Kryptographie
 - 7.3.2. Grundlegende Konzepte und Bedienung
 - 7.3.3. Der RSA-Algorithmus
 - 7.3.4. Digitale Zertifikate
 - 7.3.5. Speicherung und Verwaltung von Schlüsseln
- 7.4. Netzwerk-Angriffe
 - 7.4.1. Bedrohungen und Angriffe aus dem Netzwerk
 - 7.4.2. Aufzählung
 - 7.4.3. Verkehrsüberwachung: *Sniffers*
 - 7.4.4. *Denial-of-Service*-Angriffe
 - 7.4.5. *ARP-Poisoning*-Angriffe

- 7.5. Sicherheitsarchitekturen
 - 7.5.1. Traditionelle Sicherheitsarchitekturen
 - 7.5.2. Secure Socket Layer: SSL
 - 7.5.3. SSH-Protokoll
 - 7.5.4. Virtuelle private Netzwerke (VPNs)
 - 7.5.5. Schutzmechanismen für externe Speicherlaufwerke
 - 7.5.6. Hardware-Schutzmechanismen
- 7.6. Systemschutztechniken und Entwicklung von sicherem Code
 - 7.6.1. Sicherheit im Betrieb
 - 7.6.2. Ressourcen und Kontrollen
 - 7.6.3. Überwachung
 - 7.6.4. Systeme zur Erkennung von Angriffen
 - 7.6.5. Host IDS
 - 7.6.6. Netzwerk-IDS
 - 7.6.7. Signatur-basiertes IDS
 - 7.6.8. Decoy-Systeme
 - 7.6.9. Grundlegende Sicherheitsprinzipien bei der Code-Entwicklung
 - 7.6.10. Störungsmanagement
 - 7.6.11. Staatsfeind Nummer 1: Der *Buffer Overflow*
 - 7.6.12. Kryptographische Botschaften
- 7.7. Botnets und Spam
 - 7.7.1. Ursprung des Problems
 - 7.7.2. Spam-Prozess
 - 7.7.3. Spam verschicken
 - 7.7.4. Verfeinerung der Verteilerlisten
 - 7.7.5. Methoden zum Schutz
 - 7.7.6. Von Dritten angebotener Anti-Spam-Service
 - 7.7.7. Fallstudien
 - 7.7.8. Exotischer Spam



- 7.8. Web-Auditing und Angriffe
 - 7.8.1. Sammeln von Informationen
 - 7.8.2. Angriffs-Techniken
 - 7.8.3. Werkzeuge
- 7.9. *Malware* und böartiger Code
 - 7.9.1. Was ist *Malware*?
 - 7.9.2. Arten von *Malware*
 - 7.9.3. Virus
 - 7.9.4. Kryptoviren
 - 7.9.5. Würmer
 - 7.9.6. Adware
 - 7.9.7. Spyware
 - 7.9.8. Hoaxes
 - 7.9.9. Phishing
 - 7.9.10. Trojaner
 - 7.9.11. Die *Malware*-Wirtschaft
 - 7.9.12. Mögliche Lösungen
- 7.10. Forensische Analyse
 - 7.10.1. Sammeln von Beweisen
 - 7.10.2. Analyse der Beweise
 - 7.10.3. Anti-Forensische Techniken
 - 7.10.4. Praktische Fallstudie

Modul 8. Softwaresicherheit

- 8.1. Software-Sicherheitsprobleme
 - 8.1.1. Einführung in das Problem der Softwaresicherheit
 - 8.1.2. Schwachstellen und ihre Klassifizierung
 - 8.1.3. Sichere Softwareeigenschaften
 - 8.1.4. Referenzen

- 8.2. Grundsätze des Software-Sicherheitsdesigns
 - 8.2.1. Einführung
 - 8.2.2. Grundsätze des Software-Sicherheitsdesigns
 - 8.2.3. Arten von S-SDLC
 - 8.2.4. Softwaresicherheit in den S-SDLC-Phasen
 - 8.2.5. Methodologien und Normen
 - 8.2.6. Referenzen
- 8.3. Sicherheit im Software-Lebenszyklus in der Anforderungs- und Entwurfsphase
 - 8.3.1. Einführung
 - 8.3.2. Angriffsmodellierung
 - 8.3.3. Missbrauchsfälle
 - 8.3.4. Entwicklung von Sicherheitsanforderungen
 - 8.3.5. Risikoanalyse. Architektonisch
 - 8.3.6. Entwurfsmuster
 - 8.3.7. Referenzen
- 8.4. Sicherheit im Software-Lebenszyklus in den Phasen Kodierung, Test und Betrieb
 - 8.4.1. Einführung
 - 8.4.2. Risikobasierte Sicherheitsprüfungen
 - 8.4.3. Code-Überprüfung
 - 8.4.4. Penetrationstests
 - 8.4.5. Sicherheitsmaßnahmen
 - 8.4.6. Externe Überprüfung
 - 8.4.7. Referenzen
- 8.5. Sichere Kodierungsanwendungen I
 - 8.5.1. Einführung
 - 8.5.2. Sichere Kodierungspraktiken
 - 8.5.3. Eingabeverarbeitung und Validierung
 - 8.5.4. Speicherüberlauf
 - 8.5.5. Referenzen

- 8.6. Sichere Kodierungsanwendungen II
 - 8.6.1. Einführung
 - 8.6.2. *Integers Overflows*, Abbruchfehler und Probleme mit Typkonvertierungen zwischen Ganzzahlen
 - 8.6.3. Fehler und Ausnahmen
 - 8.6.4. Datenschutz und Vertraulichkeit
 - 8.6.5. Privilegierte Programme
 - 8.6.6. Referenzen
- 8.7. Sicherheit in der Entwicklung und in der *Cloud*
 - 8.7.1. Entwicklungssicherheit; Methodik und Praxis
 - 8.7.2. PaaS, IaaS, CaaS und SaaS-Modelle
 - 8.7.3. Sicherheit in der *Cloud* und für *Cloud*-Dienste
- 8.8. Orchestrierung und Automatisierung der Sicherheit (SOAR)
 - 8.8.1. Komplexität der manuellen Verarbeitung; Notwendigkeit der Automatisierung von Aufgaben
 - 8.8.2. Produkte und Dienstleistungen
 - 8.8.3. SOAR-Architektur
- 8.9. Sicherheit der Telearbeit
 - 8.9.1. Bedarf und Szenarien
 - 8.9.2. Produkte und Dienstleistungen
 - 8.9.3. Sicherheit der Telearbeit

Modul 9. Qualität und Prüfung von Informationssystemen

- 9.1. Einführung in Informationssicherheits-Managementsysteme
 - 9.1.1. Grundlegende Prinzipien des ISMS
 - 9.1.2. Goldene Regeln des ISMS
 - 9.1.3. Die Rolle der IT-Auditierung in ISMS
- 9.2. Planung im Sicherheitsmanagement
 - 9.2.1. Konzepte für das Sicherheitsmanagement
 - 9.2.2. Klassifizierung von Informationen: Ziele, Konzepte und Rollen
 - 9.2.3. Umsetzung der Sicherheitspolitik: Sicherheitsrichtlinien, Standards und Verfahren
 - 9.2.4. Risikomanagement: Grundsätze und Analyse des Risikos von Informationsgütern

- 9.3. Wichtigste Mechanismen für den Schutz von Informationsgütern (I)
 - 9.3.1. Überblick über die wichtigsten kryptografischen Werkzeuge zum Schutz der CID-Trias
 - 9.3.2. Berücksichtigung der Erfordernisse des Datenschutzes, der Anonymität und der ordnungsgemäßen Verwaltung der Rückverfolgbarkeit der Nutzer
- 9.4. Wichtigste Mechanismen für den Schutz von Informationsgütern (II)
 - 9.4.1. Kommunikationssicherheit: Protokolle, Geräte und Sicherheitsarchitekturen
 - 9.4.2. Sicherheit der Betriebssysteme
- 9.5. Interne ISMS-Kontrollen
 - 9.5.1. Taxonomie der ISMS-Kontrollen: administrative, logische und physische Kontrollen
 - 9.5.2. Klassifizierung der Kontrollen nach der Art und Weise, wie sie die Bedrohung angehen: Kontrollen zur Vorbeugung, Aufdeckung und Korrektur von Bedrohungen
 - 9.5.3. Implementierung von internen Kontrollsystemen in ISMS
- 9.6. Arten von Audits
 - 9.6.1. Unterschied zwischen Audit und interner Prüfung
 - 9.6.2. Internes vs. externes Audit
 - 9.6.3. Klassifizierung des Audits nach dem Ziel und der Art der Analyse
- 9.7. Drehbuchautor und Drehbuch: Subjekt und Objekt geschützt durch das Recht des geistigen Eigentums
 - 9.7.1. Einführung in Penetrationstests und forensische Analyse
 - 9.7.2. Definition und Relevanz der Konzepte des *Fingerprinting* und *Footprinting*
- 9.8. Scannen auf Schwachstellen und Überwachung des Netzwerkverkehrs
 - 9.8.1. Werkzeuge für die Schwachstellenanalyse in Systemen
 - 9.8.2. Die wichtigsten Schwachstellen im Zusammenhang mit Web-Applikationen
 - 9.8.3. Analyse der Kommunikationsprotokollen
- 9.9. Der Prozess des IT-Audits
 - 9.9.1. Lebenszykluskonzept in der Systementwicklung
 - 9.9.2. Tätigkeits- und Prozessüberwachung: Sammlung und Verarbeitung von Beweisen
 - 9.9.3. Methodik des IT-Audits
 - 9.9.4. Prozess eines IT-Audits
 - 9.9.5. Identifizierung der wichtigsten Straftaten und Ordnungswidrigkeiten im Zusammenhang mit der Informationstechnologie
 - 9.9.6. Untersuchung von Internetkriminalität: eine Einführung in die forensische Analyse und ihre Beziehung zum IT-Audits

- 9.10. Planung von Betriebskontinuität und Wiederherstellung im Katastrophenfall
 - 9.10.1. Definition des Betriebskontinuitätsplans und des Konzepts der Betriebsunterbrechung
 - 9.10.2. NIST-Empfehlung zur Planung der Betriebskontinuität
 - 9.10.3. Plan zur Wiederherstellung im Katastrophenfall
 - 9.10.4. Prozess des Plan zur Wiederherstellung im Katastrophenfall

Modul 10. Verwaltung von Webservern

- 10.1. Einführung in Webserver
 - 10.1.1. Was ist ein Webserver?
 - 10.1.2. Architektur und Funktionsweise eines Webservers
 - 10.1.3. Ressourcen und Inhalte auf einem Webserver
 - 10.1.4. Anwendungsserver
 - 10.1.5. Proxy-Server
 - 10.1.6. Die wichtigsten Webserver auf dem Markt
 - 10.1.7. Statistiken zur Webserver-Nutzung
 - 10.1.8. Sicherheit des Webservers
 - 10.1.9. Lastausgleich bei Webservern
 - 10.1.10. Referenzen
- 10.2. Umgang mit dem HTTP-Protokoll
 - 10.2.1. Betrieb und Struktur
 - 10.2.2. Beschreibung der Abfragemethoden oder *Request Methods*
 - 10.2.3. Status-Codes
 - 10.2.4. Kopfzeilen
 - 10.2.5. Codierung des Inhalts. Code-Seiten
 - 10.2.6. Durchführung von HTTP-Anfragen im Internet mit Hilfe eines Proxys, livehttpheaders oder einer ähnlichen Methode, Analyse des verwendeten Protokolls
- 10.3. Beschreibung von verteilten Architekturen auf mehreren Servern
 - 10.3.1. 3-Schichten-Modell
 - 10.3.2. Fehlertoleranz
 - 10.3.3. Lastverteilung
 - 10.3.4. Sitzungsstatus-Speicher
 - 10.3.5. Cache-Speicher

- 10.4. Internet Information Services (IIS)
 - 10.4.1. Was ist IIS?
 - 10.4.2. Geschichte und Entwicklung des IIS
 - 10.4.3. Die wichtigsten Vorteile und Funktionen von IIS7 und darüber hinaus
 - 10.4.4. IIS7 und neuere Architektur
- 10.5. IIS-Installation, -Verwaltung und -Konfiguration
 - 10.5.1. Präambel
 - 10.5.2. Installation der Internet Information Services (IIS)
 - 10.5.3. IIS-Verwaltungstools
 - 10.5.4. Erstellen, Konfigurieren und Verwalten von Websites
 - 10.5.5. Installieren und Verwalten von IIS-Erweiterungen
- 10.6. Erweiterte Sicherheit im IIS
 - 10.6.1. Präambel
 - 10.6.2. IIS-Authentifizierung, Autorisierung und Zugriffskontrolle
 - 10.6.3. Konfigurieren einer sicheren Website auf IIS mit SSL
 - 10.6.4. In IIS 8.x implementierte Sicherheitsrichtlinien
- 10.7. Einführung in Apache
 - 10.7.1. Was ist Apache?
 - 10.7.2. Die wichtigsten Vorteile von Apache
 - 10.7.3. Hauptmerkmale von Apache
 - 10.7.4. Architektur
- 10.8. Apache-Installation und -Konfiguration
 - 10.8.1. Apache-Erstinstallation
 - 10.8.2. Apache-Konfiguration
- 10.9. Installieren und Konfigurieren der verschiedenen Apache-Module
 - 10.9.1. Installation von Apache-Modulen
 - 10.9.2. Arten von Modulen
 - 10.9.3. Sichere Apache-Konfiguration
- 10.10. Erweiterte Sicherheit
 - 10.10.1. Authentifizierung, Autorisierung und Zugangskontrolle
 - 10.10.2. Authentifizierungsmethoden
 - 10.10.3. Sichere Apache-Konfiguration mit SSL

Modul 11. Sicherheit bei Online-Anwendungen

- 11.1. Schwachstellen und Sicherheitsprobleme in Online-Anwendungen
 - 11.1.1. Einführung in die Sicherheit von Online-Anwendungen
 - 11.1.2. Sicherheitsschwachstellen beim Entwurf von Webanwendungen
 - 11.1.3. Sicherheitsschwachstellen bei der Implementierung von Webanwendungen
 - 11.1.4. Sicherheitsschwachstellen bei der Bereitstellung von Webanwendungen
 - 11.1.5. Offizielle Listen von Sicherheitslücken
- 11.2. Richtlinien und Standards für die Sicherheit von Online-Anwendungen
 - 11.2.1. Säulen der Sicherheit von Online-Anwendungen
 - 11.2.2. Sicherheitspolitik
 - 11.2.3. Managementsystem für die Informationssicherheit
 - 11.2.4. Sicherer Lebenszyklus der Software Entwicklung
 - 11.2.5. Standards für die Anwendungssicherheit
- 11.3. Sicherheit beim Entwurf von Webanwendungen
 - 11.3.1. Einführung in die Sicherheit von Webanwendungen
 - 11.3.2. Sicherheit beim Entwurf von Webanwendungen
- 11.4. Prüfung der Online-Sicherheit von Webanwendungen
 - 11.4.1. Analyse und Prüfung der Sicherheit von Webanwendungen
 - 11.4.2. Sicherheit bei der Bereitstellung und Produktion von Webanwendungen
- 11.5. Sicherheit von Webdiensten
 - 11.5.1. Einführung in die Sicherheit von Webdiensten
 - 11.5.2. Sicherheitsfunktionen und -technologien für Webdienste
- 11.6. Prüfung der Online-Sicherheit und des Schutzes von Webdiensten
 - 11.6.1. Bewertung der Sicherheit von Webdiensten
 - 11.6.2. Online-Schutz. XML-Firewalls und -Gateways
- 11.7. Ethisches *Hacking*, *Malware* und Forensik
 - 11.7.1. Ethisches *Hacking*
 - 11.7.2. *Malware*-Analyse
 - 11.7.3. Forensische Analyse

- 11.8. Auflösung von Vorfällen bei Webdiensten
 - 11.8.1. Überwachung
 - 11.8.2. Instrumente zur Leistungsmessung
 - 11.8.3. Eindämmungsmaßnahmen
 - 11.8.4. Ursachenanalyse
 - 11.8.5. Proaktives Problemmanagement
- 11.9. Bewährte Verfahren zur Gewährleistung der Anwendungssicherheit
 - 11.9.1. Handbuch für bewährte Praktiken bei der Entwicklung von Online-Anwendungen
 - 11.9.2. Handbuch für bewährte Praktiken bei der Umsetzung von Online-Anwendungen
- 11.10. Häufige Fehler, die die Anwendungssicherheit untergraben
 - 11.10.1. Häufige Entwicklungsfehler
 - 11.10.2. Häufige Fehler beim *Hosting*
 - 11.10.3. Häufige Fehler in der Produktion

Modul 12. Softwaretechnik

- 12.1. Einführung in die Softwaretechnik und Modellierung
 - 12.1.1. Die Natur der Software
 - 12.1.2. Die Besonderheit von Webapps
 - 12.1.3. Softwaretechnik
 - 12.1.4. Der Software-Prozess
 - 12.1.5. Die Praxis der Softwaretechnik
 - 12.1.6. Software-Mythen
 - 12.1.7. Wie alles beginnt
 - 12.1.8. Objektorientierte Konzepte
 - 12.1.9. Einführung in UML
- 12.2. Der Software-Prozess
 - 12.2.1. Ein allgemeines Prozessmodell
 - 12.2.2. Vorgeschriebene Prozessmodelle
 - 12.2.3. Spezialisierte Prozessmodelle
 - 12.2.4. Der vereinheitlichte Prozess
 - 12.2.5. Personal- und Teamprozessmodelle
 - 12.2.6. Was ist Agilität?
 - 12.2.7. Was ist ein agiler Prozess?
 - 12.2.8. Scrum
 - 12.2.9. Werkzeugkasten für agile Prozesse
- 12.3. Prinzipien als Leitfaden für die Praxis der Softwareentwicklung
 - 12.3.1. Leitprinzipien des Prozesses
 - 12.3.2. Prinzipien als Leitfaden für die Praxis
 - 12.3.3. Grundsätze der Kommunikation
 - 12.3.4. Grundsätze der Planung
 - 12.3.5. Grundsätze der Modellierung
 - 12.3.6. Konstruktionsprinzipien
 - 12.3.7. Grundsätze für die Einführung
- 12.4. Verständnis der Anforderungen
 - 12.4.1. Anforderungsmanagement
 - 12.4.2. Schaffung der Grundlagen
 - 12.4.3. Bedarfsermittlung
 - 12.4.4. Entwicklung von Anwendungsfällen
 - 12.4.5. Ausarbeitung des Anforderungsmodells
 - 12.4.6. Aushandeln von Anforderungen
 - 12.4.7. Validierung der Anforderungen
- 12.5. Modellierung der Anforderungen: Szenarien, Informationen und Arten der Analyse
 - 12.5.1. Analyse der Anforderungen
 - 12.5.2. Szenario-basiertes Modell
 - 12.5.3. UML-Modelle, die den Anwendungsfall liefern
 - 12.5.4. Konzepte der Datenmodellierung
 - 12.5.5. Klassen-basiertes Modell
 - 12.5.6. Klassendiagramme
- 12.6. Modellierung der Anforderungen: Fluss, Verhalten und Muster
 - 12.6.1. Anforderungen die die Strategien gestalten
 - 12.6.2. Flussorientierte Modellierung
 - 12.6.3. Zustandsdiagramme
 - 12.6.4. Erstellung eines Verhaltensmodells
 - 12.6.5. Sequenzdiagramme
 - 12.6.6. Kommunikationsdiagramme
 - 12.6.7. Muster für die Modellierung von Anforderungen

- 12.7. Konzepte des Designs
 - 12.7.1. Design im Kontext der Softwaretechnik
 - 12.7.2. Der Entwurfsprozess
 - 12.7.3. Konzepte des Designs
 - 12.7.4. Objektorientierte Konzepte des Designs
 - 12.7.5. Das Designmodell
- 12.8. Design der Architektur
 - 12.8.1. Software-Architektur
 - 12.8.2. Architektonische Gattungen
 - 12.8.3. Architektonische Stile
 - 12.8.4. Architektonischer Design
 - 12.8.5. Entwicklung von alternativen Designs für die Architektur
 - 12.8.6. Abbildung der Architektur mit Hilfe von Datenflüssen
- 12.9. Design auf Komponentenebene und musterbasierter Entwurf
 - 12.9.1. Was ist eine Komponente?
 - 12.9.2. Klassenbasiertes Komponentendesign
 - 12.9.3. Verwirklichung des Designs auf Komponentenebene
 - 12.9.4. Design der traditionellen Komponenten
 - 12.9.5. Komponentenbasierte Entwicklung
 - 12.9.6. Entwurfsmuster
 - 12.9.7. Musterbasiertes Softwaredesign
 - 12.9.8. Architektonische Muster
 - 12.9.9. Musterdesign auf Komponentenebene
 - 12.9.10. Musterdesign für Benutzeroberflächen
- 12.10. Softwarequalität und Projektmanagement
 - 12.10.1. Qualität
 - 12.10.1. Softwarequalität
 - 12.10.2. Das Dilemma der Softwarequalität
 - 12.10.3. Erreichen von Softwarequalität

- 12.10.4. Software-Qualitätssicherung
- 12.10.5. Das administrative Spektrum
- 12.10.6. Personal
- 12.10.7. Das Produkt
- 12.10.8. Der Prozess
- 12.10.9. Das Projekt
- 12.10.10. Grundsätze und Praktiken

Modul 13. Fortgeschrittene Softwaretechnik

- 13.1. Extreme Programmierung
 - 13.1.1. Grundprinzip und Überblick über XP
 - 13.1.2. Der Lebenszyklus von XP
 - 13.1.3. Die fünf Grundwerte
 - 13.1.4. Die zwölf Kernpraktiken von XP
 - 13.1.5. Rollen der Teilnehmer
 - 13.1.6. Industrie XP
 - 13.1.7. Kritische Beurteilung von XP
- 13.2. Softwareentwicklung auf der Grundlage von Wiederverwendung
 - 13.2.1. Wiederverwendung von Software
 - 13.2.2. Stufen der Wiederverwendung von Codes
 - 13.2.3. Spezifische Techniken der Wiederverwendung
 - 13.2.4. Komponentenbasierte Entwicklung
 - 13.2.5. Vorteile und Probleme der Wiederverwendung
 - 13.2.6. Planung der Wiederverwendung
- 13.3. Systemarchitektur und Software-Entwurfsmuster
 - 13.3.1. Architektonisches Design
 - 13.3.2. Allgemeine architektonische Muster
 - 13.3.3. Fehlertolerante Architekturen
 - 13.3.4. Architekturen verteilter Systeme
 - 13.3.5. Entwurfsmuster
 - 13.3.6. Gamma Muster
 - 13.3.7. Muster für die Interaktionsgestaltung

- 13.4. Architektur von *Cloud*-Anwendungen
 - 13.4.1. Grundlagen des *Cloud Computing*
 - 13.4.2. Qualität von *Cloud*-Anwendungen
 - 13.4.3. Architektonische Stile
 - 13.4.4. Entwurfsmuster
- 13.5. Software-Tests: TDD, ATDD und BDD
 - 13.5.1. Überprüfung und Validierung von Software
 - 13.5.2. Software-Tests
 - 13.5.3. Test Driven Development (TDD)
 - 13.5.4. Acceptance Test Driven Development (ATDD)
 - 13.5.5. Behavior Driven Development (BDD)
 - 13.5.6. BDD und Cucumber
- 13.6. Verbesserung von Software-Prozessen
 - 13.6.1. Verbesserung von Software-Prozessen
 - 13.6.2. Der Prozess der Prozessverbesserung
 - 13.6.3. Reifegradmodelle
 - 13.6.4. CMMI-Modell
 - 13.6.5. CMMI V2.0.
 - 13.6.6. CMMI und Agilität
- 13.7. Qualität von Softwareprodukten: SQuaRE
 - 13.7.1. Qualität der Software
 - 13.7.2. Qualitätsmodelle für Softwareprodukte
 - 13.7.3. ISO/IEC 25000-Familie
 - 13.7.4. ISO/IEC 25010: Qualitätsmodell und Qualitätsmerkmale
 - 13.7.5. ISO/IEC 25012: Datenqualität
 - 13.7.6. ISO/IEC 25020: Messung der Softwarequalität
 - 13.7.7. ISO/IEC 25022, 25023 und 25024: Messgrößen für die Software- und Datenqualität
 - 13.7.8. ISO/IEC 25040: Bewertung der Software
 - 13.7.9. Der Prozess der Zertifizierung
- 13.8. Einführung in DevOps
 - 13.8.1. DevOps-Konzept
 - 13.8.2. Grundlegende Praktiken

Modul 14. Anforderungsmanagement

- 14.1. Einführung in das Anforderungsmanagement
 - 14.1.1. Die Bedeutung der Anforderungen
 - 14.1.2. Konzept der Anforderung
 - 14.1.3. Dimensionen der Anforderungen
 - 14.1.4. Stufen und Arten von Anforderungen
 - 14.1.5. Eigenschaften der Anforderungen
 - 14.1.6. Das Anforderungsmanagement
 - 14.1.7. Der Prozess des Anforderungsmanagement
 - 14.1.8. *Frameworks* für Anforderungsmanagement
 - 14.1.9. Bewährte Praktiken im Anforderungsmanagement
 - 14.1.10. Der Business-Analyst
- 14.2. Die Quellen der Anforderungen
 - 14.2.1. Das Netzwerk der Anforderungen
 - 14.2.2. Die *Stakeholder*
 - 14.2.3. Geschäftliche Anforderungen
 - 14.2.4. Dokument zur Vision und zum Anwendungsbereich
- 14.3. Techniken zur Erhebung von Anforderungen
 - 14.3.1. Die Erhebung von Anforderungen
 - 14.3.2. Probleme der Erhebung von Anforderungen
 - 14.3.3. Kontexte der Entdeckung
 - 14.3.4. Befragungen
 - 14.3.5. Beobachtung und „Lernen“
 - 14.3.6. Ethnographie
 - 14.3.7. Workshops
 - 14.3.8. Focus groups
 - 14.3.9. Umfragen
 - 14.3.10. *Brainstorming* und kreative Techniken
 - 14.3.11. Gruppenmedien
 - 14.3.12. Analyse der Systemschnittstellen
 - 14.3.13. Dokumentenanalyse und „Archäologie“
 - 14.3.14. Anwendungsbeispiele und Szenarien
 - 14.3.15. Prototypen
 - 14.3.16. Reverse Engineering
 - 14.3.17. Wiederverwendung von Anforderungen
 - 14.3.18. Bewährte Erhebungsmethoden

- 14.4. Benutzeranforderungen
 - 14.4.1. Personen
 - 14.4.2. Anwendungsfälle und Anwenderberichte
 - 14.4.3. Szenarien
 - 14.4.5. Arten von Szenarien
 - 14.4.6. Wie man Szenarien entdeckt
- 14.5. Prototyping-Techniken
 - 14.5.1. Prototyping
 - 14.5.2. Prototypen nach ihrem Anwendungsbereich
 - 14.5.3. Prototypen nach ihrer Aktualität
 - 14.5.4. Die Treue eines Prototyps
 - 14.5.5. Prototypen von Benutzeroberflächen
 - 14.5.6. Bewertung von Prototypen
- 14.6. Analyse der Anforderungen
 - 14.6.1. Die Analyse der Anforderungen
 - 14.6.2. Bewährte Praktiken bei der Anforderungsanalyse
 - 14.6.3. Das Wörterbuch der Daten
 - 14.6.4. Priorisierung der Anforderungen
- 14.7. Dokumentation der Anforderungen
 - 14.7.1. Das Dokument zur Anforderungsspezifikation
 - 14.7.2. Aufbau und Inhalt eines SRS
 - 14.7.3. Dokumentation in natürlicher Sprache
 - 14.7.4. EARS: Easy Approach to Requirements Syntax
 - 14.7.5. Nichtfunktionale Anforderungen
 - 14.7.6. Attribute und Vorlagen in Tabellenform
 - 14.7.7. Bewährte Praktiken bei Spezifikationen
- 14.8. Validierung und Verhandlung von Anforderungen
 - 14.8.1. Validierung der Anforderungen
 - 14.8.2. Techniken zur Validierung von Anforderungen
 - 14.8.3. Verhandlung von Anforderungen

- 14.9. Modellierung und Anforderungsmanagement
 - 14.9.1. Die Modellierung der Anforderungen
 - 14.9.2. Die Nutzerperspektive
 - 14.9.3. Die Perspektive von Daten
 - 14.9.4. Die funktionale oder flussorientierte Perspektive
 - 14.9.5. Die Verhaltensperspektive
 - 14.9.6. Die Volatilität der Anforderungen
 - 14.9.7. Prozess des Anforderungsmanagements
 - 14.9.8. Werkzeuge für das Anforderungsmanagement
 - 14.9.9. Bewährte Praktiken im Anforderungsmanagement
- 14.10. Kritische Systeme und formale Spezifikation
 - 14.10.1. Kritische Systeme
 - 14.10.2. Risikoorientierte Spezifikation
 - 14.10.3. Formale Spezifikation

Modul 15. Prozesse der Softwaretechnik

- 15.1. Rahmen der Softwaretechnik
 - 15.1.1. Eigenschaften der Software
 - 15.1.2. Die wichtigsten Prozesse der Softwaretechnik
 - 15.1.3. Prozessmodelle für die Softwareentwicklung
 - 15.1.4. Standard-Referenzrahmen für den Softwareentwicklungsprozess: Norm ISO/IEC 12207
- 15.2. Einheitlicher Softwareentwicklungsprozess
 - 15.2.1. Einheitlicher Prozess
 - 15.2.2. Dimensionen des einheitlichen Prozesses
 - 15.2.3. Anwendungsfallorientierter Entwicklungsprozess
 - 15.2.4. Grundlegende Unified Process Workflows
- 15.3. Planung im Rahmen der agilen Softwareentwicklung
 - 15.3.1. Eigenschaften der agilen Softwareentwicklung
 - 15.3.2. Unterschiedliche Planungszeithorizonte in der agilen Entwicklung
 - 15.3.3. Rahmen für die agile Entwicklung nach Scrum und Planung von Zeithorizonten
 - 15.3.4. Anwenderberichte als Planungs- und Schätzungseinheit

- 15.3.5. Übliche Techniken zur Ableitung einer Schätzung
- 15.3.6. Skalen für die Interpretation von Schätzungen
- 15.3.7. Planning Poker
- 15.3.8. Gängige Planungsarten: Lieferplanung und Iterationsplanung
- 15.4. Designstile für verteilte Software und dienstorientierte Softwarearchitekturen
 - 15.4.1. Kommunikationsmodelle in verteilten Softwaresystemen
 - 15.4.2. Zwischenschicht oder *Middleware*
 - 15.4.3. Architekturmuster für verteilte Systeme
 - 15.4.4. Allgemeiner Prozess zur Entwicklung von Softwarediensten
 - 15.4.5. Gestaltungsaspekte von Softwarediensten
 - 15.4.6. Zusammensetzung der Dienstleistungen
 - 15.4.7. Architektur der Webdienste
 - 15.4.8. Komponenten für Infrastruktur und SOA
- 15.5. Einführung in die modellgetriebene Softwareentwicklung
 - 15.5.1. Das Konzept des Modells
 - 15.5.2. Modellgestützte Softwareentwicklung
 - 15.5.3. MDA-Rahmen für modellbasierte Entwicklung
 - 15.5.4. Elemente eines Transformationsmodells
- 15.6. Gestaltung der grafischen Benutzeroberfläche
 - 15.6.1. Grundlagen der Gestaltung der Benutzeroberfläche
 - 15.6.2. Architektonische Entwurfsmuster für interaktive Systeme: *Model-View-Controller* (MVC)
 - 15.6.3. Benutzererfahrung (UX User Experience)
 - 15.6.4. Benutzerzentriertes Design
 - 15.6.5. Prozess der Analyse und Gestaltung grafischer Benutzeroberflächen
 - 15.6.6. Benutzerfreundlichkeit der Benutzeroberfläche
 - 15.6.7. Zugänglichkeit der Benutzeroberflächen
- 15.7. Gestaltung von Webanwendungen
 - 15.7.1. Eigenschaften von Webanwendungen
 - 15.7.2. Benutzeroberfläche einer Webanwendungen
 - 15.7.3. Gestaltung der Navigation
 - 15.7.4. Grundlegendes Interaktionsprotokoll für Webanwendungen
 - 15.7.5. Architekturstile für Webanwendungen

- 15.8. Strategien und Techniken der Softwareprüfung und Faktoren der Softwarequalität
 - 15.8.1. Prüfstrategien
 - 15.8.2. Gestaltung von Testfällen
 - 15.8.3. Preis-Leistungs-Verhältnis
 - 15.8.4. Qualitätsmodelle
 - 15.8.5. ISO/IEC 25000-Normenfamilie (SQuaRE)
 - 15.8.6. Modell der Produktqualität (ISO 2501n)
 - 15.8.7. Modelle der Datenqualität (ISO 2501n)
 - 15.8.8. Software-Qualitätsmanagement
- 15.9. Einführung in Messgrößen in der Softwaretechnik
 - 15.9.1. Grundlegende Konzepte: Messgrößen, Metriken und Indikatoren
 - 15.9.2. Arten von Messgrößen in der Softwaretechnik
 - 15.9.3. Der Messprozess
 - 15.9.4. ISO 25024. Externe und verwendete Qualitätsmessgrößen
 - 15.9.5. Objektorientierte Messgrößen
- 15.10. Software-Wartung und *Reengineering*
 - 15.10.1. Wartungsprozess
 - 15.10.2. Standardrahmen für Wartungsprozesse. ISO/IEC 14764
 - 15.10.3. Prozessmodell für das Software-*Reengineering*
 - 15.10.4. *Reverse Engineering*

Modul 16. Systemintegration

- 16.1. Einführung in Wirtschaftsinformationssysteme
 - 16.1.1. Die Rolle der Informationssysteme
 - 16.1.2. Was ist ein Informationssystem?
 - 16.1.3. Dimensionen von Informationssystemen
 - 16.1.4. Geschäftsprozesse und Informationssysteme
 - 16.1.5. Die IS/IT-Abteilung

- 16.2. Möglichkeiten und Bedürfnisse für Informationssysteme im Unternehmen
 - 16.2.1. Organisationen und Informationssysteme
 - 16.2.2. Merkmale von Organisationen
 - 16.2.3. Auswirkungen von Informationssystemen auf das Unternehmen
 - 16.2.4. Informationssysteme als Wettbewerbsvorteil
 - 16.2.5. Einsatz von Systemen in der Unternehmensverwaltung und im Management
- 16.3. Informationssysteme und technologische Grundlagen
 - 16.3.1. Daten, Informationen und Wissen
 - 16.3.2. Informationssysteme und Technologie
 - 16.3.3. Technologie-Komponenten
 - 16.3.4. Klassifizierung und Arten von Informationssystemen
 - 16.3.5. Service- und geschäftsprozessbasierte Architekturen
 - 16.3.6. Formen der Systemintegration
- 16.4. Integrierte Systeme zur Verwaltung von Unternehmensressourcen
 - 16.4.1. Geschäftliche Anforderungen
 - 16.4.2. Ein integriertes Informationssystem für das Unternehmen
 - 16.4.3. Akquisition vs. Entwicklung
 - 16.4.4. ERP-Implementierung
 - 16.4.5. Auswirkungen auf das Management
 - 16.4.6. Wichtigste ERP-Anbieter
- 16.5. Informationssysteme für die Verwaltung von Lieferketten und Kundenbeziehungen
 - 16.5.1. Definition der Lieferkette
 - 16.5.2. Effektives Management der Lieferkette
 - 16.5.3. Die Rolle der Informationssysteme
 - 16.5.4. Lösungen für das Lieferkettenmanagement
 - 16.5.5. Verwaltung von Kundenbeziehungen
 - 16.5.6. Die Rolle der Informationssysteme
 - 16.5.7. Einführung eines CRM-Systems
 - 16.5.8. Kritische Erfolgsfaktoren bei der CRM-Implementierung
 - 16.5.9. CRM, e-CRM und andere Trends
- 16.6. Entscheidungsfindung für IKT-Investitionen und Planung von Informationssystemen
 - 16.6.1. Kriterien für IKT-Investitionsentscheidungen
 - 16.6.2. Verknüpfung des Projekts mit dem Management- und Geschäftsplan
 - 16.6.3. Auswirkungen auf das Management
 - 16.6.4. Neugestaltung von Geschäftsprozessen
 - 16.6.5. Entscheidung des Managements über Implementierungsmethoden
 - 16.6.6. Notwendigkeit der Planung von Informationssystemen
 - 16.6.7. Zielsetzung, Teilnehmer und Zeitplan
 - 16.6.8. Aufbau und Entwicklung des Systemplans
 - 16.6.9. Überwachung und Aktualisierung
- 16.7. Sicherheitserwägungen bei der Nutzung von IKTs
 - 16.7.1. Risikoanalyse
 - 16.7.2. Sicherheit in Informationssystemen
 - 16.7.3. Praktische Ratschläge
- 16.8. Durchführbarkeit von IKT-Projekten und finanzielle Aspekte von Informationssystemprojekten
 - 16.8.1. Beschreibung und Ziele
 - 16.8.2. Teilnehmer an der Machbarkeitsstudie des Systems
 - 16.8.3. Techniken und Praktiken
 - 16.8.4. Kostenstruktur
 - 16.8.5. Finanzielle Projektion
 - 16.8.6. Budgets
- 16.9. Business Intelligence
 - 16.9.1. Was ist *Business Intelligence*?
 - 16.9.2. BI-Strategie und -Implementierung
 - 16.9.3. Gegenwart und Zukunft von BI
- 16.10. ISO/IEC 12207
 - 16.10.1. Was bedeutet „ISO/IEC 12207“?
 - 16.10.2. Analyse von Informationssystemen
 - 16.10.3. Entwurf eines Informationssystems
 - 16.10.4. Implementierung und Akzeptanz des Informationssystems

Modul 17. Wiederverwendung von Software

- 17.1. Überblick über die Wiederverwendung von Software
 - 17.1.1. Was ist die Wiederverwendung von Software?
 - 17.1.2. Vor- und Nachteile der Wiederverwendung von Software
 - 17.1.3. Wichtigste Techniken der Wiederverwendung von Software
- 17.2. Einführung in die Entwurfsmuster
 - 17.2.1. Was ist ein Entwurfsmuster?
 - 17.2.2. Katalog der wichtigsten Entwurfsmuster
 - 17.2.3. Wie man Muster zur Lösung von Designproblemen einsetzt
 - 17.2.4. Wie man das beste Entwurfsmuster auswählt
- 17.3. Erzeugungsmuster
 - 17.3.1. Erzeugungsmuster
 - 17.3.2. *Abstract-Factory*-Muster
 - 17.3.3. Beispielimplementierung des *Abstract-Factory*-Musters
 - 17.3.4. *Builder*-Muster
 - 17.3.5. Beispielimplementierung des *Builder*-Musters
 - 17.3.6. *Abstract-Factory*-Muster vs. *Builder*
- 17.4. Erzeugungsmuster (II)
 - 17.4.1. *Factory-Method*-Muster
 - 17.4.2. *Factory Method* vs. *Abstract Factory*
 - 17.4.3. *Singleton*-Muster
- 17.5. Strukturmuster
 - 17.5.1. Strukturmuster
 - 17.5.2. *Adapter*-Muster
 - 17.5.3. *Bridge*-Muster
- 17.6. Strukturmuster (II)
 - 17.6.1. *Composite*-Muster
 - 17.6.2. *Decorator*-Muster
- 17.7. Strukturmuster (III)
 - 17.7.1. *Facade*-Muster
 - 17.7.2. *Proxy*-Muster

- 17.8. Verhaltensmuster
 - 17.8.1. Konzept der Verhaltensmuster
 - 17.8.2. Verhaltensmuster: Kette der Verantwortung
 - 17.8.3. Verhaltensmuster Ordnung
- 17.9. Verhaltensmuster (II)
 - 17.9.1. *Interpreter*-Muster
 - 17.9.2. *Iterator*-Muster
 - 17.9.3. *Observer*-Muster
 - 17.9.4. *Strategie*-Muster
- 17.10. *Frameworks*
 - 17.10.1. *Konzepte des Framework*
 - 17.10.2. Entwicklung mit *Frameworks*
 - 17.10.3. *Model View Controller*-Muster
 - 17.10.4. *Frameworks* für die Gestaltung grafischer Benutzeroberflächen
 - 17.10.5. *Frameworks* für die Entwicklung Web-Applikationen
 - 17.10.6. *Frameworks* für die Verwaltung der Objektpersistenz in Datenbanken

Modul 18. Dienstleistungen der Informationstechnologie

- 18.1. Digitale Transformation (I)
 - 18.1.1. *Business Innovation*
 - 18.1.2. Management der Produktion
 - 18.1.3. Finanzielle Verwaltung
- 18.2. Digitale Transformation (II)
 - 18.2.1. Marketing
 - 18.2.2. Personalmanagement
 - 18.2.3. Ein integriertes Informationssystem
- 18.3. Fallstudie
 - 18.3.1. Präsentation des Unternehmens
 - 18.3.2. Methoden zur Analyse der IT-Beschaffung
 - 18.3.3. Bestimmung von Kosten, Nutzen und Risiken
 - 18.3.4. Wirtschaftliche Bewertung der Investition
- 18.4. IKT-Governance und -Verwaltung
 - 18.4.1. Definition der *Governance* von Informationstechnologien und -systemen
 - 18.4.2. Unterschied zwischen IKT-Governance und Management
 - 18.4.3. IKT-Governance und Managementrahmen
 - 18.4.4. Standards und die Leitung und Verwaltung von IKTs

- 18.5. IKT-Unternehmensführung
 - 18.5.1. Was ist gute Unternehmensführung?
 - 18.5.2. Hintergrund zur IKT-Governance
 - 18.5.3. ISO/IEC 38500:2008-Norm
 - 18.5.4. Umsetzung einer guten IKT-Governance
 - 18.5.5. IKT-Governance und bewährte Praktiken
 - 18.5.6. Unternehmensführung. Überblick und Trends
- 18.6. Kontrollziele für Informations- und verwandte Technologien (COBIT)
 - 18.6.1. Rahmen für die Umsetzung
 - 18.6.2. Bereich: Planung und Organisation
 - 18.6.3. Bereich: Akquisition und Implementierung
 - 18.6.4. Bereich: Lieferung und Unterstützung
 - 18.6.5. Bereich: Überwachung und Bewertung
 - 18.6.6. Anwendung des COBIT-Leitfadens
- 18.7. Die Informationstechnologie-Infrastruktur-Bibliothek (ITIL)
 - 18.7.1. Einführung in ITIL
 - 18.7.2. Service-Strategie
 - 18.7.3. Service-Design
 - 18.7.4. Service-Übergang
 - 18.7.5. Service-Betrieb
 - 18.7.6. Service-Verbesserung
- 18.8. Das *Service Management System*
 - 18.8.1. Grundlegende Prinzipien von UNE-ISO/IEC 20000-1
 - 18.8.2. Die Struktur der Normenreihe ISO/IEC 20000
 - 18.8.3. Anforderungen an das *Service Management System* (SMS)
 - 18.8.4. Gestaltung und Umstellung neuer oder geänderter Dienste
 - 18.8.5. Prozesse der Dienstleistungserbringung
 - 18.8.6. Prozessgruppen
- 18.9. Das *Software Asset Management System*
 - 18.9.1. Rechtfertigung des Bedarfs
 - 18.9.2. Hintergrund
 - 18.9.3. Präsentation der Norm 19770
 - 18.9.4. Umsetzung der Verwaltung
- 18.10. Management der Geschäftskontinuität
 - 18.10.1. Plan zur Aufrechterhaltung des Geschäftsbetriebs
 - 18.10.2. Implementierung eines BCM



“

Nutzen Sie Ihre Vorkenntnisse und bewerben Sie sich für die Validierung von Fächern, die Sie bereits absolviert haben, um Ihre Erfahrungen in diesem Programm zu optimieren“

04

Lehrziele

Dieser Studiengang in Softwaretechnik wurde entwickelt, um IT-Fachleuten die fortschrittlichen Werkzeuge an die Hand zu geben, die sie benötigen, um ihre Karriere in Richtung der Entwicklung moderner Software voranzutreiben, die an die sich ändernden Anforderungen des Marktes angepasst ist. Durch die Vermittlung von technischem Fachwissen werden die Studenten ihre beruflichen Aufstiegschancen erheblich verbessern und Schlüsselpositionen in führenden Unternehmen des Sektors einnehmen können.



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

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

```
mirror_ob.select= 1
```

```
modifier_ob.select=1
```

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

```
print("Selected" + str(modifier_ob.name))
```

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

```
key = bpy.context.selected_objects[0].name
```

```
print(key)
```

```
print(modifier_ob.name)
```

```
print(modifier_ob.name)
```

```
print(modifier_ob.name)
```

“

Ein Programm, das den ultimativen
Anstoß für eine erfolgreiche Karriere in
der Softwaretechnik gibt“



Allgemeine Ziele

- ♦ Erwerben neuer Kompetenzen, die im Hinblick auf neue Technologien und neueste Softwareentwicklungen erforderlich und gefragt sind
- ♦ Ergänzen der erworbenen Kenntnisse durch Fähigkeiten auf dem Gebiet der Informatik und der Computerstruktur, einschließlich der mathematischen, statistischen und physikalischen Grundlagen, die für das Ingenieurwesen wesentlich sind
- ♦ Erweitern der Kenntnisse in der Softwaretechnik und in Computersystemen mit den neuesten Entwicklungen und innovativsten Methoden
- ♦ Bewältigen komplexer Software-Projekte und -Umgebungen mit dem Wissen, wie man intelligente Lösungen für unterschiedliche Probleme bietet



Steigern Sie Ihre Karriere mit einem Spezialisierungsprogramm, das auf die strategische Leitung der Softwaretechnik ausgerichtet ist“





Spezifische Ziele

Modul 1. Methodik, Entwicklung und Qualität in der Softwaretechnik

- ♦ Fortbilden in der Anwendung agiler und traditioneller Methoden in der Softwareentwicklung
- ♦ Sicherstellen der Qualität des Endprodukts durch die Anwendung von Standards und bewährten Verfahren in jeder Phase des Software-Lebenszyklus

Modul 2. Management von Softwareprojekten

- ♦ Entwickeln der notwendigen Kompetenzen zur Planung, Durchführung und Verwaltung von Softwareprojekten
- ♦ Anwenden von Projektmanagement-Methoden und Planungswerkzeugen, um eine termingerechte und budgetkonforme Lieferung zu gewährleisten

Modul 3. Plattformen für die Softwareentwicklung

- ♦ Vermitteln von Kenntnissen über verschiedene Plattformen zur Softwareentwicklung
- ♦ Nutzen von Entwicklungsumgebungen, *Frameworks* und Tools, die die Erstellung robuster und skalierbarer Anwendungen optimieren

Modul 4. Web-Client-Computing

- ♦ Fortbilden in der Entwicklung von clientseitigen Webanwendungen unter Verwendung von Technologien wie HTML, CSS und JavaScript
- ♦ Erstellen interaktiver Schnittstellen, die für die Benutzererfahrung optimiert sind

Modul 5. Webserver-Computing

- ♦ Entwickeln der für die Verwaltung und Entwicklung serverseitiger Webanwendungen erforderlichen Fähigkeiten
- ♦ Handhaben von Technologien wie PHP, Node.js oder Python und Verwaltung von Datenbanken und Backend-Diensten

Modul 6. Sicherheitsmanagement

- ♦ Vermitteln der Kenntnisse und Fähigkeiten für das Sicherheitsmanagement in Softwareentwicklungsumgebungen
- ♦ Anwenden von Richtlinien und Strategien für Datenschutz, Zugang und Sicherheit der technologischen Infrastruktur

Modul 7. Sicherheit von Informationssystemen

- ♦ Vermitteln eines eingehenden Verständnisses der Sicherheit von Informationssystemen
- ♦ Analysieren des Schutzes von Netzwerken und Servern bis hin zur Implementierung von Sicherheitsprotokollen und -standards in technologischen Infrastrukturen

Modul 8. Softwaresicherheit

- ♦ Fortbilden in der Entwicklung von sicherer Software
- ♦ Identifizieren, Entschärfen und Verhindern von Schwachstellen in Code und in der Anwendungsentwicklung

Modul 9. Qualität und Prüfung von Informationssystemen

- ♦ Entwickeln von Kompetenzen zur Durchführung von Qualitätsaudits von Informationssystemen
- ♦ Sicherstellen der Wirksamkeit von Sicherheitskontrollen, Einhalten von Vorschriften und Optimieren von technologischen Prozessen

Modul 10. Verwaltung von Webservern

- ♦ Fortbilden in der Verwaltung und Administration von Webservern
- ♦ Vertiefen der Wartung von Servern, die Online-Anwendungen und -Dienste hosten

Modul 11. Sicherheit bei Online-Anwendungen

- ♦ Vermitteln der notwendigen Kenntnisse zur Sicherung von Online-Anwendungen
- ♦ Behandeln von Aspekten wie Authentifizierung, Datenschutz, Zugangsverwaltung und Angriffsvermeidung in Webumgebungen

Modul 12. Softwaretechnik

- ♦ Vermitteln eines umfassenden Verständnisses der Grundsätze und Praktiken der Softwaretechnik
- ♦ Entwerfen, Entwickeln und Warten effizienter, skalierbarer und qualitativ hochwertiger Softwareanwendungen

Modul 13. Fortgeschrittene Softwaretechnik

- ♦ Eingehen auf fortgeschrittene Methoden der Softwaretechnik
- ♦ Vertiefen der anspruchsvollen Entwicklungsmethoden, Entwurfsmuster, Softwarearchitektur und Prozessoptimierung für komplexe Projekte

Modul 14. Anforderungsmanagement

- ♦ Entwickeln von Fähigkeiten zur Ermittlung, Analyse und Dokumentation von Softwareanforderungen
- ♦ Sicherstellen, dass die Endprodukte die Bedürfnisse und Erwartungen der Benutzer erfüllen

Modul 15. Prozesse der Softwaretechnik

- ♦ Fortbilden in den grundlegenden Prozessen der Softwaretechnik, von der Planung und dem Entwurf bis hin zu Implementierung, Test und Wartung
- ♦ Verbessern der Qualität und Effizienz des Software-Lebenszyklus



Modul 16. Systemintegration

- ♦ Integrieren verschiedener Softwaresysteme, Datenbanken und Anwendungen
- ♦ Erstellen kohärenter und effizienter Technologielösungen zur Lösung komplexer Geschäftsprobleme

Modul 17. Wiederverwendung von Software

- ♦ Anwenden von Techniken und Prinzipien für die Wiederverwendung von Softwarekomponenten und -modulen in neuen Projekten
- ♦ Optimieren von Entwicklungszeit und -kosten durch Erstellen wiederverwendbarer Bibliotheken oder *Frameworks*

Modul 18. Dienstleistungen der Informationstechnologie

- ♦ Entwerfen, Implementieren und Verwalten von Dienstleistungen der Informationstechnologie
- ♦ Verbessern der technologischen Infrastruktur zur Gewährleistung der Geschäftskontinuität und der betrieblichen Effizienz

05

Karrieremöglichkeiten

Nach Abschluss dieses Hochschulprogramms verfügen die Fachkräfte über ein solides Verständnis der Techniken, Methoden und Werkzeuge, die erforderlich sind, um moderne, an die aktuellen Marktbedürfnisse angepasste Softwareentwicklungsprojekte zu bewältigen. Die Absolventen werden in der Lage sein, fortschrittliche technologische Lösungen zu entwerfen, zu implementieren und zu verwalten und dabei Qualität, Sicherheit und Effizienz in allen Entwicklungsphasen zu gewährleisten. Sie werden auch in der Lage sein, multidisziplinäre Teams zu leiten, in agilen Umgebungen zusammenzuarbeiten und strategische Entscheidungen auf der Grundlage internationaler Standards zu treffen.



Sie werden innovative Ansätze und effiziente Lösungen anwenden, die die Entwicklung und das Management von Software in komplexen technologischen Projekten optimieren“

Profil des Absolventen

Der Absolvent des Studiengangs in Softwaretechnik ist hoch fortgebildet, um die technologischen Herausforderungen der modernen Softwareentwicklung zu meistern. Darüber hinaus verfügt er über fundierte Kenntnisse fortschrittlicher Methoden, Entwicklungswerkzeuge und internationaler Standards, die für die Entwicklung innovativer und effektiver Lösungen erforderlich sind. Außerdem ist er in der Lage, hochwirksame technologische Projekte zu konzipieren, umzusetzen und zu leiten, mit multidisziplinären Teams zusammenzuarbeiten und strategische Initiativen in einem dynamischen und wettbewerbsorientierten Umfeld zu leiten, wobei er die Qualität, Sicherheit und Skalierbarkeit der entwickelten Anwendungen gewährleistet.

Sie werden theoretische Kenntnisse mit praktischen Fähigkeiten im Bereich Softwaredesign und fortgeschrittene Entwicklungsmethoden kombinieren.

- ♦ **Softwareentwurf und -entwicklung:** Fähigkeit, fortgeschrittene Anwendungen zu entwerfen und zu implementieren, unter Einbeziehung von Entwurfsmustern, UML-Modellierung und Entwicklungsplattformen wie Java und .NET
- ♦ **Management von Technologieprojekten:** Fähigkeit zur Planung, Überwachung und Leitung von Technologieprojekten unter Verwendung agiler Methoden wie Scrum und Anwendung internationaler ISO/IEC-Normen
- ♦ **Sicherheit in der Entwicklung:** Gründliche Kenntnisse über bewährte Verfahren der IT-Sicherheit, des Risikomanagements und des Datenschutzes in allen Phasen des Software-Lebenszyklus
- ♦ **Implementierung in komplexen Umgebungen:** Fähigkeit zur Entwicklung von mobilen, web- und cloud-basierten Anwendungen unter Verwendung von Technologien wie REST, SOA und MVC
- ♦ **Problemlösung:** Anwendung von kritischem Denken und innovativen Strategien zur Identifizierung und Lösung technologischer Herausforderungen in Softwareprojekten
- ♦ **Digitale Kompetenz:** Beherrschung fortschrittlicher technologischer Werkzeuge und digitaler Plattformen für die Entwicklung, Prüfung und Verwaltung von Systemen



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

1. **Leiter von Softwareprojekten:** Verantwortlich für die Planung und Durchführung komplexer Projekte, um sicherzustellen, dass die Qualitäts-, Zeit- und Budgetziele eingehalten werden.
2. **Softwarearchitekt:** Spezialist für die Konzeption und den Aufbau komplexer und skalierbarer Systeme.
3. **IT-Sicherheitsingenieur:** Verantwortlich für den Schutz von Systemen, Anwendungen und Daten vor Schwachstellen und Angriffen.
4. **Full-Stack-Entwickler:** Fachkraft, die in der Lage ist, umfassende Lösungen sowohl auf der Client- als auch auf der Serverseite zu erstellen.
5. **Technologieberater:** Fachkundiger Berater bei der Umsetzung fortschrittlicher Technologien und Strategien für die digitale Transformation.
6. **Spezialist für Cloud-Anwendungen:** Designer und Implementierer von SaaS-, PaaS- und IaaS-basierten Lösungen.
7. **Systemadministrator:** Verwalter von Servern, Netzwerken und Technologieplattformen, der deren Leistung und Sicherheit optimiert.
8. **Software-Qualitätsanalyst:** Bewerter der Funktionalität, Benutzerfreundlichkeit und Leistung von technologischen Anwendungen.
9. **Berater für Anforderungsmanagement:** Spezialist für die Analyse, Dokumentation und Validierung von Anforderungen in Softwareprojekten.



Nach Abschluss dieses Programms haben Sie Zugang zu Schlüsselpositionen in der Technologiebranche und können sich als Führungskraft in der Softwaretechnik profilieren“

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 ihre 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.

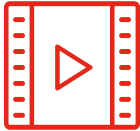
Die Studenten bewerten die pädagogische Qualität, die Qualität der Materialien, die Struktur und die Ziele der Kurse als ausgezeichnet. Es überrascht nicht, dass die Einrichtung im global score Index mit 4,9 von 5 Punkten die von ihren Studenten am besten bewertete Universität ist.

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 Fachkräften, 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

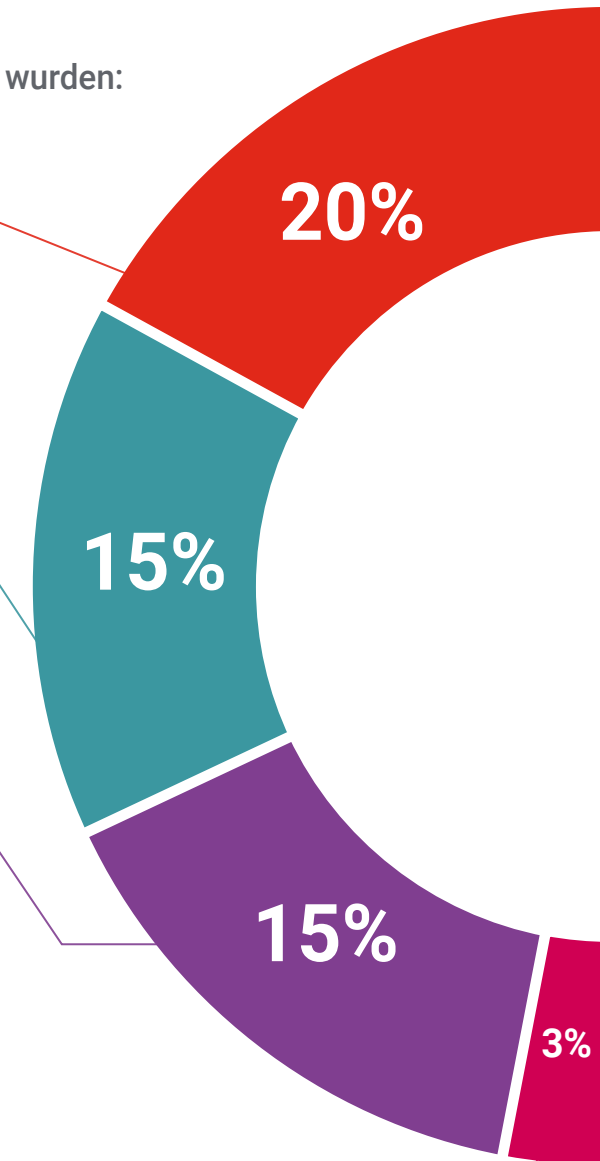
Wir präsentieren die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, Audios, Videos, Bildern, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu festigen.

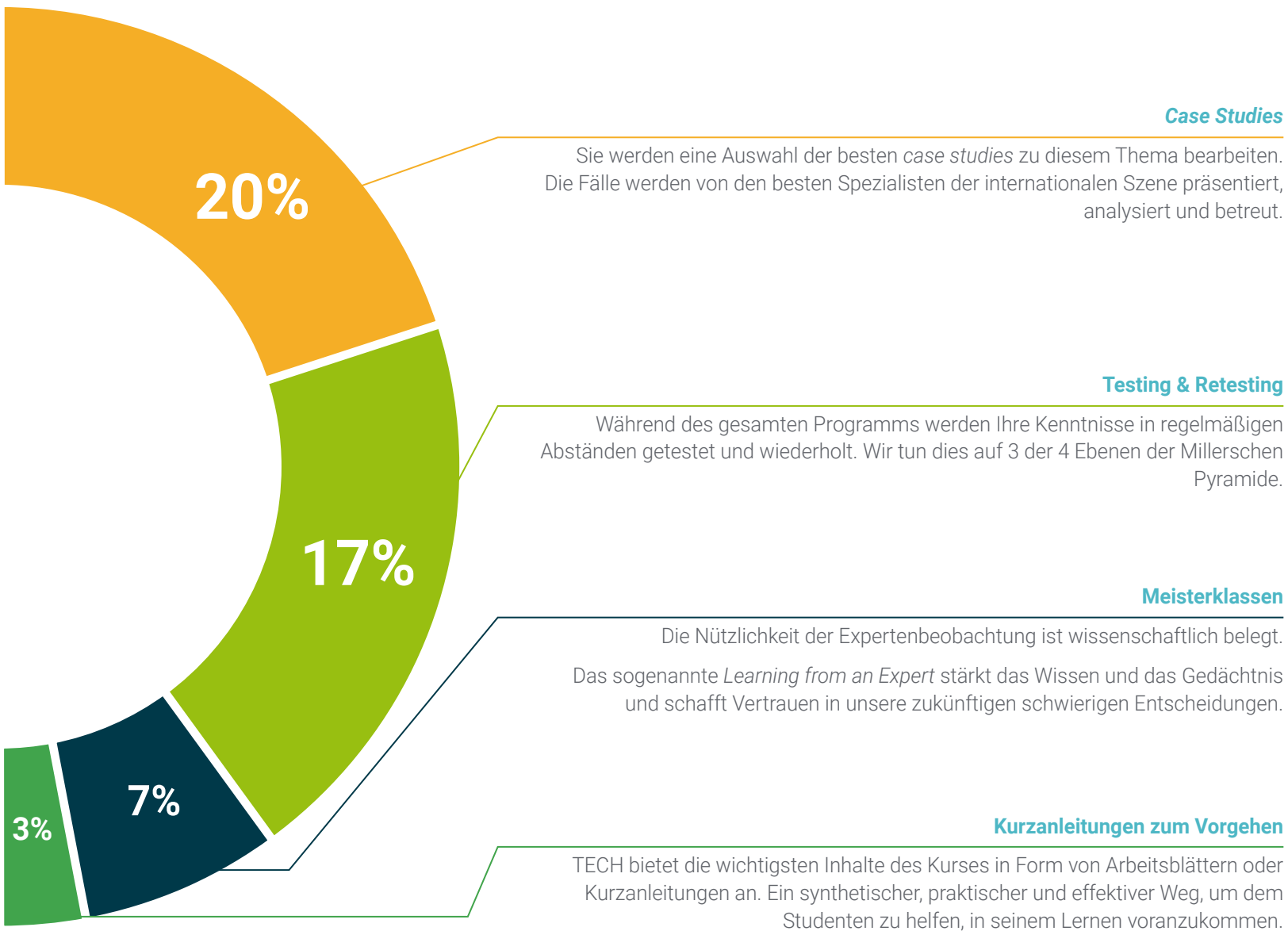
Dieses einzigartige System für die Präsentation multimedialer Inhalte wurde von Microsoft als „Europäische Erfolgsgeschichte“ ausgezeichnet.



Weitere Lektüren

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





07

Lehrkörper

Die Dozenten dieses Studiengangs in Softwaretechnik zeichnen sich durch ihr tiefes technisches Wissen und ihre praktische Erfahrung in der Softwarebranche aus. Sie haben in der Tat bedeutende Projekte in führenden Unternehmen der Branche geleitet, was es ihnen ermöglicht, nicht nur fortschrittliche Theorien, sondern auch praktische Anwendungen und reale Fälle weiterzugeben, die die Vorbereitung der Studenten bereichern werden. Darüber hinaus sind sie durch ihr Engagement für Innovation und Forschung stets auf dem Laufenden über die neuesten Trends und Entwicklungsmethoden.

“

Lernen Sie von Lehrkräften, die Experten auf dem Gebiet der Informatik und der Softwaretechnik sind und über fortgeschrittene Kenntnisse und Erfahrungen verfügen“

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"

08

Qualifizierung

Der Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik 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**

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.

Titel: **Weiterbildender Masterstudiengang mit Spezialisierung in Softwaretechnik**

Modalität: **online**

Dauer: **2 Jahre**

Akkreditierung: **120 ECTS**





Weiterbildender Masterstudiengang mit Spezialisierung Softwaretechnik

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

Weiterbildender Masterstudiengang mit Spezialisierung Softwaretechnik