

# Privater Masterstudiengang Software Engineering und Computersysteme



## Privater Masterstudiengang Software Engineering und Computersysteme

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Internetzugang: [www.techtute.com/de/informatik/masterstudiengang/masterstudiengang-software-engineering-computersysteme](http://www.techtute.com/de/informatik/masterstudiengang/masterstudiengang-software-engineering-computersysteme)

# Index

01

Präsentation

---

Seite 4

02

Ziele

---

Seite 8

03

Kompetenzen

---

Seite 14

04

Struktur und Inhalt

---

Seite 18

05

Methodik

---

Seite 32

06

Qualifizierung

---

Seite 40

# 01 Präsentation

Dieses umfassende Programm in *Software Engineering* und *Computersystemen* ermöglicht es Fachleuten in der IT-Branche, die Prozesse des *Managements*, des *Entwurfs*, der *Entwicklung* und der *Implementierung* von qualitativ hochwertiger und sicherer *Software* zu vertiefen und zu trainieren, die die vorgegebenen Ziele sowohl auf *Desktop-* als auch auf *Web-Ebene* erfüllt.



“

*Spezialisieren Sie sich auf Computersysteme mit Fachleuten, die über umfassende Erfahrung in diesem Bereich verfügen”*

Dieser Masterstudiengang spezialisiert die Studenten auf Software Engineering und Computersysteme, um das Wissen und die Werkzeuge zu vermitteln, die für die Konzeption und Entwicklung komplexer Systeme erforderlich sind, die auf die gestellten Probleme reagieren.

Das Hauptziel dieser Fortbildung besteht darin, dass die Studenten die Fähigkeit erlangen, wesentliche qualitative Verbesserungen einzubringen und neue Lösungen für die spezifischen Probleme zu finden, die sich bei *Software* oder Computersystemen ergeben. Es zielt auch darauf ab, Fachleute auszubilden, die in der Lage sind, einen systematischen und quantifizierbaren Ansatz für die Entwicklung und Wartung von *Software* zu verwenden, so dass sie auch ein tiefes Wissen über Computerprogrammierung, Implementierung und Planung von Computersystemen erhalten, und zwar aus einer praktischen Perspektive und angepasst an die aktuelle Realität.

Im Rahmen dieser Fortbildung werden den Studenten die fortschrittlichsten Lehrmittel zur Verfügung gestellt und sie haben die Möglichkeit, ein Lehrprogramm zu studieren, das die fundiertesten Kenntnisse in diesem Bereich vereint. Eine Gruppe von Dozenten mit hoher wissenschaftlicher Strenge und umfassender internationaler Erfahrung wird ihnen die vollständigsten und aktuellsten Informationen über die neuesten Fortschritte und Techniken im Bereich *Software Engineering* und Computersysteme vermitteln.

Der Lehrplan deckt die wichtigsten aktuellen Themen des *Software Engineering* und der Computersysteme so ab, dass diejenigen, die sie beherrschen, auf die Arbeit in diesem Bereich vorbereitet werden. Es ist also nicht nur ein weiterer Titel im Rucksack, sondern ein echtes Lernmittel, um sich den Themen des Fachgebiets auf moderne, objektive Weise und mit der Fähigkeit zur Beurteilung auf der Grundlage der aktuellsten Informationen zu nähern.

Da es sich um einen 100%igen Online-Masterstudiengang handelt, ist der Student nicht an einen festen Zeitplan oder die Notwendigkeit, sich an einen anderen Ort zu begeben, gebunden, sondern kann zu jeder Tageszeit auf die Inhalte zugreifen und so sein Arbeits- oder Privatleben mit seinem akademischen Leben in Einklang bringen.

Wenn Sie sich von anderen abheben wollen und in der Lage sein wollen, komplexe systemtechnische Projekte zu entwerfen und zu entwickeln, ist dieses Programm genau das Richtige für Sie.

Dieser **Privater Masterstudiengang in Software Engineering und Computersysteme** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- ◆ Die Entwicklung von Fallstudien, die von Experten für *Software Engineering* und Computersysteme vorgestellt werden
- ◆ Der anschauliche, schematische und äußerst praxisnahe Inhalt soll wissenschaftliche und praktische Informationen zu den für die berufliche Praxis wesentlichen Disziplinen vermitteln
- ◆ Er enthält praktische Übungen in denen der Selbstbewertungsprozess durchgeführt werden kann um das Lernen zu verbessern
- ◆ Ein besonderer Schwerpunkt liegt auf innovativen Methoden in *Software Engineering* und Computersystemen
- ◆ Theoretische Vorträge, 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



*Mit dem Abschluss dieses privaten Masterstudiengangs werden Fachleute für Software Engineering und Computersysteme an der Spitze der neuesten Entwicklungen in diesem Sektor stehen*



“

*Dieser privater Masterstudiengang ist die beste Investition, die Sie bei der Auswahl eines Weiterbildungsprogramms im Bereich Software Engineering und Computersysteme tätigen können. Wir bieten Ihnen Qualität und freien Zugang zu den Inhalten"*

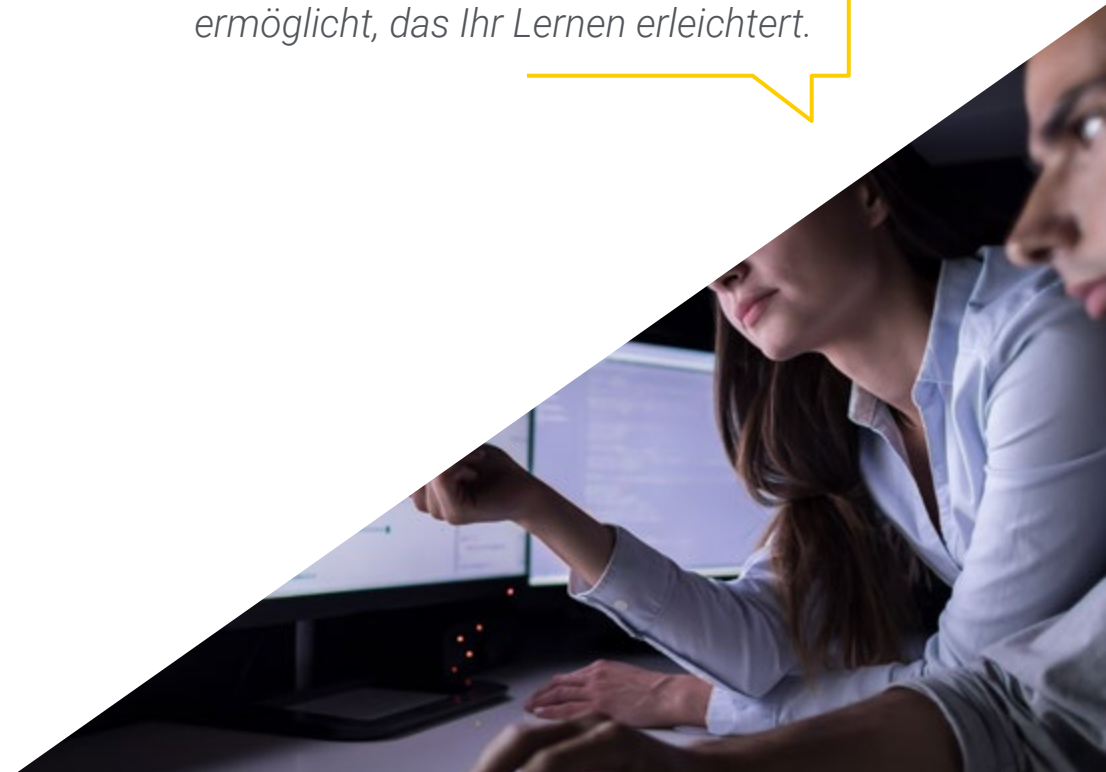
Das Dozententeam besteht aus Fachleuten aus den Bereichen *Software Engineering* und *Computersysteme*, die ihre Berufserfahrung in die Fortbildung einbringen, sowie aus anerkannten Fachleuten aus führenden Unternehmen und renommierten Universitäten.

Die multimedialen Inhalte, die mit den neuesten Bildungstechnologien entwickelt wurden, ermöglichen den Fachleuten ein situiertes und kontextbezogenes Lernen, d. h. eine simulierte Umgebung, die ein immersives Training ermöglicht, das auf reale Situationen ausgerichtet ist.

Das Konzept dieses Studiengangs konzentriert sich auf problemorientiertes Lernen, bei dem die Fachkräfte versuchen müssen, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Dabei wird die Fachkraft von einem innovativen interaktiven Videosystem unterstützt, das von anerkannten und erfahrenen Experten für *Software Engineering* und *Computersysteme* entwickelt wurde.

*Dieser 100%ige Online-Masterstudiengang ermöglicht es Ihnen, Ihr Studium mit Ihrer beruflichen Tätigkeit zu vereinen. Sie entscheiden, wo und wann Sie lernen möchten.*

*Diese Fortbildung verfügt über das beste didaktische Material, das Ihnen ein kontextbezogenes Studium ermöglicht, das Ihr Lernen erleichtert.*



# 02 Ziele

Der Private Masterstudiengang in *Software Engineering* und *Computersysteme* zielt darauf ab, den Fachleuten den Erwerb und die Kenntnis der wichtigsten Entwicklungen in diesem Bereich zu erleichtern, damit sie ihren Beruf mit höchster Qualität und Professionalität ausüben können.







“

*Unser Ziel ist es, dass Sie die beste Fachkraft in Ihrem Bereich werden. Und dafür haben wir die beste Methodik und den besten Inhalt"*



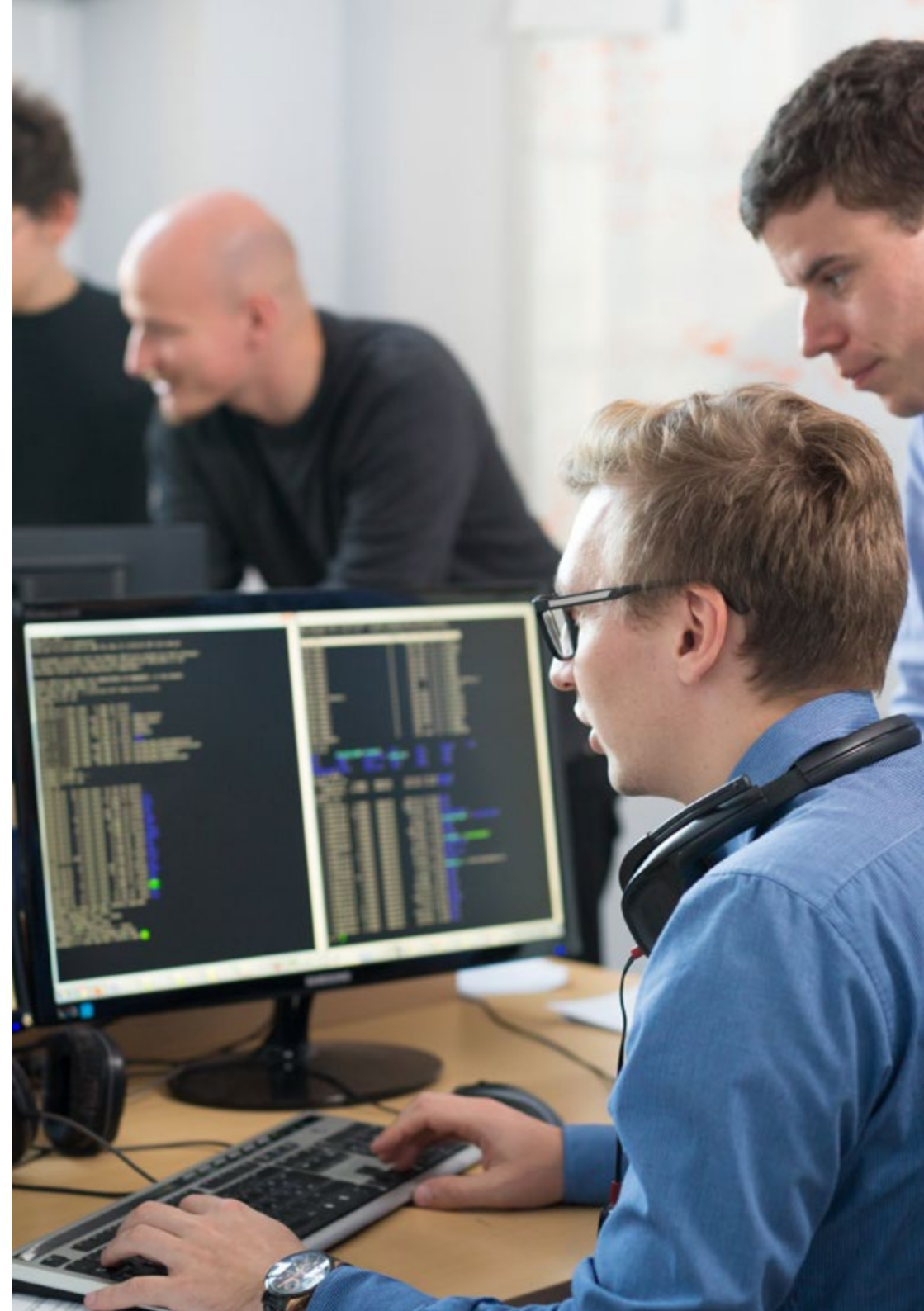
## Allgemeine Ziele

---

- ◆ Erwerb neuer Kenntnisse in den Bereichen *Software Engineering* und *Computersysteme*
- ◆ Erwerb neuer Kenntnisse in Bezug auf neue Technologien und die neuesten *Software-Entwicklungen*
- ◆ Umgang mit den Daten, die bei den Tätigkeiten im Bereich *Software Engineering* und *Computersysteme* anfallen



*Schließen Sie sich uns an, und wir werden Ihnen helfen, berufliche Spitzenleistungen zu erbringen"*





## Spezifische Ziele

---

### Modul 1. Methodik, Entwicklung und Qualität im Software Engineering

- ◆ Kenntnis der Grundlagen des Software Engineering sowie des Regelwerks oder der ethischen Grundsätze und der beruflichen Verantwortung während und nach der Entwicklung
- ◆ Verstehen des Softwareentwicklungsprozesses unter Berücksichtigung der verschiedenen Programmiermodelle und des objektorientierten Programmierparadigmas
- ◆ Die verschiedenen Arten der Anwendungsmodellierung und Entwurfsmuster in der Unified Modelling Language (UML) verstehen
- ◆ Aneignung der erforderlichen Kenntnisse für die korrekte Anwendung agiler Methoden in der Softwareentwicklung, einschließlich *Scrum*
- ◆ Kenntnis der *Lean*-Entwicklungsmethodik zur Unterscheidung der Aktivitäten, die keinen Mehrwert im Prozess bringen, um eine höhere Softwarequalität zu erreichen

### Modul 2. Software-Projektmanagement

- ◆ Kenntnis der grundlegenden Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
- ◆ Die verschiedenen Phasen des Projektmanagements wie Initiierung, Planung, *Stakeholder*-Management und Scoping zu verstehen
- ◆ Erlernen der Entwicklung von Zeitplänen für Zeitmanagement, Budgetentwicklung und Risikobewältigung
- ◆ Verständnis der Funktionsweise des Qualitätsmanagements in Projekten, einschließlich Planung, Sicherung, Kontrolle, statistischer Konzepte und verfügbarer Instrumente
- ◆ Verstehen, wie die Prozesse der Projektbeschaffung, -durchführung, -überwachung, -kontrolle und -abschluss funktionieren
- ◆ Aneignung der wesentlichen Kenntnisse im Zusammenhang mit der beruflichen Verantwortung, die sich aus dem Projektmanagement ergibt

### Modul 3. Plattformen für die Software-Entwicklung

- ◆ Die verschiedenen Softwareentwicklungsplattformen verstehen
- ◆ Erwerb der notwendigen Kenntnisse für die Entwicklung von Anwendungen und grafischen Oberflächen in den Sprachen Java und .NET
- ◆ Erlernen der notwendigen Techniken zur Fehlersuche und zum Testen der durchgeführten Entwicklungen
- ◆ Erlernen der Entwicklungsumgebungen für mobile Anwendungen in Android und der Prozesse des Debuggens und der Veröffentlichung
- ◆ Die Entwicklung von Cloud-basierten Anwendungen verstehen und die richtigen Verfahren für deren Implementierung festlegen
- ◆ Beherrschung der grundlegenden Konzepte, Dienste und Tools der Google Clouds-Plattform

### Modul 4. Web-Client-Computing

- ◆ Den Prozess der Erstellung von Webinhalten mit Hilfe der Auszeichnungssprache HTML verstehen
- ◆ Die Verfahren und Techniken zur Verbesserung des Erscheinungsbildes eines in HTML geschriebenen Dokuments verstehen
- ◆ Die Entwicklung der JavaScript-Sprache kennen
- ◆ Erwerb der notwendigen Kenntnisse für die Entwicklung von Anwendungen auf der Web-Client-Seite
- ◆ Entwicklung von Anwendungen mit komplexen Strukturen unter Verwendung der verschiedenen Verfahren, Funktionen und Objekte, aus denen JavaScript besteht
- ◆ Lernen, wie man die DOM-Programmierschnittstelle für HTML- und XML-Dokumente verwendet, um sowohl deren Stilstruktur als auch deren Inhalt zu ändern

- ◆ Verständnis für die Verwendung von ereignisgesteuerten Abläufen und *Listeners* sowie die Verwendung moderner Toolkits und Ausrichtungssysteme
- ◆ Verstehen des Konzepts der Web-Usability, seiner Vorteile, Prinzipien, Methoden und Techniken, um eine Website für den Benutzer benutzbar zu machen
- ◆ Erwerb von Kenntnissen über die Barrierefreiheit im Internet, ihre Bedeutung für die heutigen digitalen Plattformen, Methoden, Normen und Standards sowie die Festlegung von Konformitätsmaßstäben

### Modul 5. Web-Server-Computing

- ◆ Verstehen der grundlegenden, mittleren und fortgeschrittenen Konzepte der Sprache PHP für die Implementierung von serverseitigen Anwendungen
- ◆ Aneignung der erforderlichen Kenntnisse über Datenmodellierung, Beziehungen, Schlüssel und Normalisierungen
- ◆ Verstehen des Aufbaus des logischen Datenmodells, der Spezifikation von Tabellen, Spalten, Schlüsseln und Abhängigkeiten sowie der notwendigen Kenntnisse für die physische Handhabung von Daten, Dateitypen, Zugriffsmodi und deren Organisation
- ◆ Erlernen der Integration von in PHP entwickelten Anwendungen mit MariaDB- und MySQL-Datenbanken
- ◆ Beherrschung des Prozesses der Interaktion mit dem Kunden durch den Einsatz von: Formularen, *Cookies* und Sitzungsmanagement
- ◆ Verstehen der Model-View-Controller-View (MVC)-Softwarearchitektur, die die Daten, die Benutzeroberfläche und die Steuerlogik einer Anwendung in drei verschiedene Komponenten aufteilt
- ◆ Erwerb von Fähigkeiten zur Nutzung von Webservices unter Verwendung von XML, SOA und REST

### Modul 6. Sicherheitsmanagement

- ◆ Erlernen der Anwendung bewährter Sicherheitspraktiken bei der Verwaltung von Informationstechnologiediensten
- ◆ Aneignung von Kenntnissen für die ordnungsgemäße Zertifizierung von Sicherheitsprozessen
- ◆ Authentifizierungsmechanismen und -methoden für die Zugangskontrolle sowie den Prozess der Zugangsprüfung verstehen
- ◆ Verständnis von Sicherheitsmanagementprogrammen, Risikomanagement und der Gestaltung von Sicherheitsstrategien
- ◆ Erlernen von Geschäftskontinuitätsplänen, ihren Phasen und ihrem Wartungsprozess
- ◆ Kenntnis der Verfahren für den korrekten Schutz des Unternehmens durch DMZ-Netze, den Einsatz von Systemen zur Erkennung von Eindringlingen und andere Methoden

### Modul 7. Software-Sicherheit

- ◆ Verstehen der Probleme im Zusammenhang mit der Softwaresicherheit, ihrer Schwachstellen und deren Klassifizierung
- ◆ Kenntnis der Entwurfsgrundsätze, Methoden und Standards der Softwaresicherheit
- ◆ Verstehen der Anwendung von Sicherheit in den verschiedenen Phasen des Software-Lebenszyklus
- ◆ Aneignung der erforderlichen Kenntnisse für sichere Softwarecodierung und Validierungstechniken
- ◆ Aneignung der Methoden und Verfahren zur Gewährleistung der Sicherheit bei der Entwicklung und Bereitstellung von Cloud-Diensten
- ◆ Verstehen der Grundlagen der Kryptologie und der verschiedenen derzeit verfügbaren Verschlüsselungstechniken

### Modul 8. Web-Server-Verwaltung

- ◆ Verstehen des Konzepts, der Funktionsweise, der Architektur, der Ressourcen und des Inhalts eines Webservers
- ◆ Verstehen der Funktionsweise, Struktur und Handhabung des HTTP-Protokolls
- ◆ Aneignung des Konzepts der verteilten Architekturen auf mehreren Servern
- ◆ Beherrschung der Funktionsweise eines Anwendungsservers und eines Proxyservers
- ◆ Analyse der verschiedenen Webserver, die heute auf dem Markt sind
- ◆ Verständnis des Prozesses der Nutzungsstatistik und des Lastausgleichs auf Webservern
- ◆ Erwerb der notwendigen Kenntnisse für die Installation, Administration, Konfiguration und Sicherheit des *Microsoft Internet Information Services* (IIS) Webservers sowie des freien Apache Webservers

### Modul 9. Sicherheitsprüfung

- ◆ Aneignung der Kenntnisse, die für die korrekte Durchführung des internen IT-Prüfungs- und Kontrollprozesses erforderlich sind
- ◆ Verständnis der für die Sicherheitsüberprüfung von Systemen und Netzen durchzuführenden Verfahren
- ◆ Verstehen der verschiedenen Hilfsmittel, Methoden und anschließenden Analysen während der Sicherheitsprüfung von Internet und mobilen Geräten
- ◆ Die Eigenschaften und Einflussfaktoren von Unternehmensrisiken kennenlernen und die richtige Umsetzung eines angemessenen Risikomanagements bestimmen
- ◆ Erlernen von Maßnahmen zur Risikominderung sowie von Methoden zur Umsetzung eines Informationssicherheitsmanagementsystems und der zu verwendenden Vorschriften und Normen
- ◆ Verstehen der Verfahren zur Durchführung eines Sicherheitsaudits, seiner Nachvollziehbarkeit und der Präsentation der Ergebnisse

### Modul 10. Sicherheit bei Online-Anwendungen

- ◆ Aneignung der erforderlichen Kenntnisse zur Bewertung und Erkennung von Schwachstellen in Online-Anwendungen
- ◆ Erlernen der Verfahren, die bei der Entwicklung von Web-Applikationen und ihrer anschließenden Validierung durch Analyse und Sicherheitstests anzuwenden sind
- ◆ Erlernen der Sicherheitsmaßnahmen für die Bereitstellung und Produktion von Web-Applikationen
- ◆ Verstehen der Konzepte, Funktionen und Technologien, die bei der Sicherheit von Webdiensten anzuwenden sind, sowie der Sicherheitstests und Schutzmaßnahmen
- ◆ Aneignung der Verfahren für ethisches *Hacking*, *Malware*-Analyse und forensische Analyse
- ◆ Kenntnis der Maßnahmen zur Abschwächung und Eindämmung von Zwischenfällen bei Webdiensten
- ◆ Aneignung von Kenntnissen über die Anwendung von Best-Practice-Techniken für die Entwicklung und Implementierung von Online-Anwendungen sowie über die häufigsten Fehler



# 03

# Kompetenzen

Nach Bestehen der Prüfungen des Masterstudiengangs in Software Engineering und Computersystemen wird die Fachkraft die notwendigen Kompetenzen für eine qualitativ hochwertige und aktualisierte Praxis auf der Grundlage der innovativsten didaktischen Methodik erworben haben.



“

*Dieses Programm wird es Ihnen ermöglichen,  
die notwendigen Kompetenzen zu erwerben,  
um in Ihrer täglichen Arbeit effizienter zu sein”*



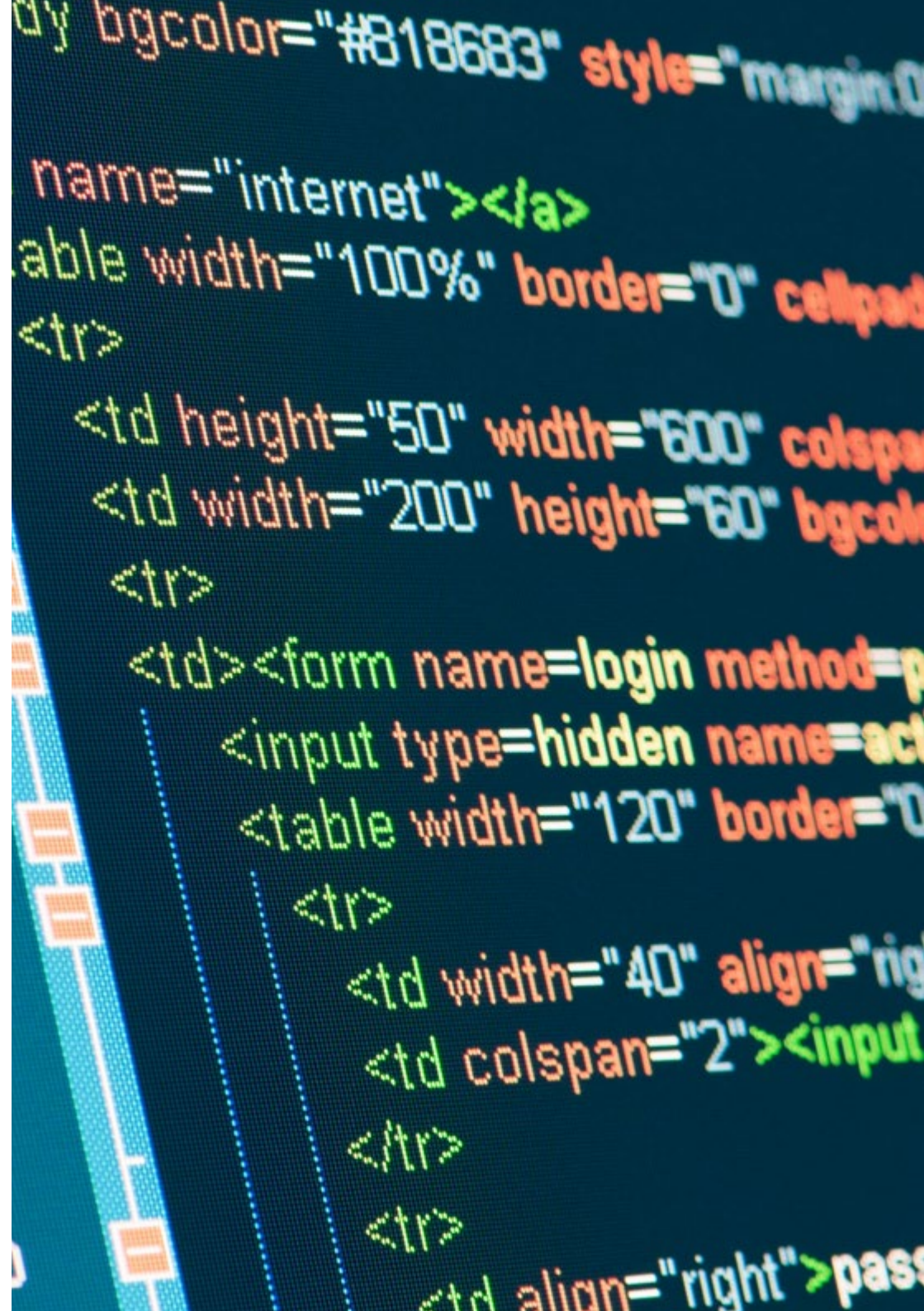


## Allgemeine Kompetenz

- ♦ Entwurf, Verwaltung und Durchführung von Projekten in den Bereichen Software Engineering und Computersysteme



*Die Verbesserung Ihrer Fähigkeiten auf dem Gebiet der Software- und Informationssystemtechnik wird es Ihnen ermöglichen, wettbewerbsfähiger zu sein. Bilden Sie sich weiter und bringen Sie Ihre Karriere in Schwung"*





## Spezifische Kompetenzen

---

- ◆ Die verschiedenen Arten der Anwendungsmodellierung und Entwurfsmuster in der Unified Modelling Language (UML) verstehen
- ◆ Verständnis der Funktionsweise des Qualitätsmanagements in Projekten, einschließlich Planung, Sicherung, Kontrolle, statistischer Konzepte und verfügbarer Instrumente
- ◆ Erwerb der notwendigen Kenntnisse für die Entwicklung von Anwendungen und grafischen Oberflächen in den Sprachen Java und .NET
- ◆ Die Verfahren und Techniken zur Verbesserung des Erscheinungsbildes eines in HTML geschriebenen Dokuments verstehen
- ◆ Beherrschen Sie den Prozess der Kundeninteraktion durch den Einsatz von: Formulare, Cookies und Sitzungsmanagement
- ◆ Authentifizierungsmechanismen und -methoden für die Zugangskontrolle sowie den Prozess der Zugangsprüfung verstehen
- ◆ Verstehen der Anwendung von Sicherheit in den verschiedenen Phasen des *Software-Lebenszyklus*
- ◆ Verstehen des Konzepts, der Funktionsweise, der Architektur, der Ressourcen und des Inhalts eines Webservers
- ◆ Verstehen der verschiedenen Hilfsmittel, Methoden und anschließenden Analysen während der Sicherheitsprüfung von Internet und mobilen Geräten
- ◆ Verständnis der Sicherheitsrichtlinien und -standards, die auf Online-Anwendungen anzuwenden sind



04

# Struktur und Inhalt

Die Struktur der Inhalte wurde von den besten Fachleuten des Sektors *Software Engineering* und *Computersysteme* entworfen, die über umfangreiche Erfahrung und anerkanntes Ansehen in diesem Beruf verfügen und sich der Vorteile bewusst sind, die die neueste Bildungstechnologie für die Hochschulbildung bringen kann.







“

*Wir verfügen über das vollständigste und aktuellste wissenschaftliche Programm auf dem Markt. Wir streben nach Exzellenz und wollen, dass auch Sie sie erreichen”*

## Modul 1. Methodik, Entwicklung und Qualität im *Software Engineering*

- 1.1. Einführung in das *Software Engineering*
  - 1.1.1. Einführung
  - 1.1.2. Die *Software-Krise*
  - 1.1.3. Unterschiede zwischen *Software Engineering* und Informatik
  - 1.1.4. Ethik und berufliche Verantwortung in der *Software Engineering*
  - 1.1.5. *Software-Fabriken*
- 1.2. Der *Software-Entwicklungsprozess*
  - 1.2.1. Definition
  - 1.2.2. *Software-Prozessmodell*
  - 1.2.3. Der einheitliche *Software-Entwicklungsprozess*
- 1.3. Objektorientierte *Software-Entwicklung*
  - 1.3.1. Einführung
  - 1.3.2. Grundsätze der Objektorientierung
  - 1.3.3. Objekt-Definition
  - 1.3.4. Definition der Klasse
  - 1.3.5. Objektorientierte Analyse vs. Objektorientierter Entwurf
- 1.4. Modellgestützte *Software-Entwicklung*
  - 1.4.1. Die Notwendigkeit der Modellierung
  - 1.4.2. Modellierung von *Software-Systemen*
  - 1.4.3. Modellierung von Objekten
  - 1.4.4. UML
  - 1.4.5. CASE-Werkzeuge
- 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 im *Software-Engineering*
  - 1.7.1. Einführung
  - 1.7.2. Ontologie-Technik
  - 1.7.3. Anwendung der Ontologien im *Software-Engineering*
- 1.8. Agile Methoden für die *Software-Entwicklung, Scrum*
  - 1.8.1. Was bedeutet Agilität in der *Software*?
  - 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. Das *Product Backlog*
  - 1.8.11. Der *Sprint*
  - 1.8.12. Die Sitzungen
- 1.9. Die Methodik der Entwicklung der *software Lean*
  - 1.9.1. Einführung
  - 1.9.2. *Kanban*
- 1.10. Qualität und Verbesserung von *Software-Prozessen*
  - 1.10.1. Einführung
  - 1.10.2. *Software-Messung*
  - 1.10.3. *Software-Prüfung*
  - 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 Interessengruppen
  - 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. Festlegung der Aktivitäten
  - 2.4.3. Festlegung der Reihenfolge der Aktivitäten
  - 2.4.4. Schätzung der Ressourcen für die Aktivitäten
  - 2.4.5. Schätzung der Dauer von 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. Sicherung der Qualität
  - 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. Kommunikationstechnik
  - 2.7.4. Kommunikationsmodelle
  - 2.7.5. Methoden der Kommunikation
  - 2.7.6. Managementplan für Kommunikation
  - 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 Arbeitsteam
- 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, Kontrolle und 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 *Software*-Entwicklung

- 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. Verzeichnisse
  - 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. Web 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 Praktiken 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 Geltungsbereiche
  - 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. Sätze
  - 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 das 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. JavaScript OOP
- 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 Richtlinien zur Benutzerfreundlichkeit
  - 4.9.11. Bewertung der Benutzerfreundlichkeit
- 4.10. Zugänglichkeit des Internets
  - 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. Beschreibung des Prozesses der Webzugänglichkeitskonformität
  - 4.10.7. Konformitätsstufen
  - 4.10.8. Konformitätskriterien
  - 4.10.9. Anforderungen an die Konformität

## Modul 5. Web-Server-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. Grundlegend
    - 5.3.2.2. Register
    - 5.3.2.3. Dynamisch
- 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. Normalformen
- 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. Abhängigkeiten
- 5.6. Das physische Datenmodell. Dateien
  - 5.6.1. Beschreibung der Datendateien
  - 5.6.2. Arten von Dateien
  - 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 von PHP aus
- 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. Model
  - 5.9.4. Ansicht
- 5.10. Einführung in Webdienste
  - 5.10.1. Einführung in XML
  - 5.10.2. Service-orientierte 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)<sup>2</sup>
  - 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<sup>1</sup>-Technologien
- 6.4. Programme, Verfahren und Richtlinien zur Informationssicherheit
  - 6.4.1. Einführung
  - 6.4.2. Programme für das Sicherheitsmanagement
  - 6.4.3. Risikomanagement
- 6.5. 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*, etc...)
  - 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 Zwischenfällen
  - 6.9.3. CERTS und CSIRTs
- 6.10. Sicherheitsarchitektur. Erholung
  - 6.10.1. IT-Resilienz-Lösungen
  - 6.10.2. Krisenmanagement und Governance

## Modul 7. Software-Sicherheit

- 7.1. Software-Sicherheitsprobleme
  - 7.1.1. Einführung in das Problem der Softwaresicherheit
  - 7.1.2. Schwachstellen und ihre Klassifizierung
  - 7.1.3. Sichere Software-Eigenschaften
  - 7.1.4. Referenzen
- 7.2. Grundsätze des Software-Sicherheitsdesigns
  - 7.2.1. Einführung
  - 7.2.2. Grundsätze des Software-Sicherheitsdesigns
  - 7.2.3. Arten von S-SDLC
  - 7.2.4. Software-Sicherheit in den S-SDLC-Phasen
  - 7.2.5. Methodologien und Normen
  - 7.2.6. Referenzen
- 7.3. Sicherheit im Software-Lebenszyklus in der Anforderungs- und Entwurfsphase
  - 7.3.1. Einführung
  - 7.3.2. Angriffsmodellierung
  - 7.3.3. Missbrauchsfälle
  - 7.3.4. Entwicklung von Sicherheitsanforderungen
  - 7.3.5. Risikoanalyse. Architektonisch
  - 7.3.6. Entwurfsmuster
  - 7.3.7. Referenzen
- 7.4. Sicherheit im Software-Lebenszyklus in den Phasen Kodierung, Test und Betrieb
  - 7.4.1. Einführung
  - 7.4.2. Risikobasierte Sicherheitsprüfungen
  - 7.4.3. Code-Überprüfung
  - 7.4.4. Penetrationstests
  - 7.4.5. Sicherheitsmaßnahmen
  - 7.4.6. Externe Überprüfung
  - 7.4.7. Referenzen
- 7.5. Sichere Kodierungsanwendungen I
  - 7.5.1. Einführung
  - 7.5.2. Sichere Kodierungspraktiken
  - 7.5.3. Eingabeverarbeitung und Validierung
  - 7.5.4. Speicherüberlauf
  - 7.5.5. Referenzen
- 7.6. Sichere Kodierungsanwendungen II
  - 7.6.1. Einführung
  - 7.6.2. *Integers overflows*, Abschneidefehler und Probleme mit Typkonvertierungen zwischen ganzen Zahlen
  - 7.6.3. Fehler und Ausnahmen
  - 7.6.4. Datenschutz und Vertraulichkeit
  - 7.6.5. Privilegierte Programme
  - 7.6.6. Referenzen
- 7.7. Sicherheit in der Entwicklung und in der Cloud
  - 7.7.1. Entwicklungssicherheit; Methodik und Praxis
  - 7.7.2. PaaS, IaaS, CaaS und SaaS-Modelle
  - 7.7.3. Sicherheit in der Cloud und für Cloud-Dienste
- 7.8. Verschlüsselung
  - 7.8.1. Grundlagen der Kryptologie
  - 7.8.2. Symmetrische und asymmetrische Verschlüsselung
  - 7.8.3. Verschlüsselung im Ruhezustand und bei der Übermittlung
- 7.9. Orchestrierung und Automatisierung der Sicherheit (SOAR)
  - 7.9.1. Komplexität der manuellen Verarbeitung; Notwendigkeit der Automatisierung von Aufgaben
  - 7.9.2. Produkte und Dienstleistungen
  - 7.9.3. SOAR-Architektur
- 7.10. Sicherheit der Telearbeit
  - 7.10.1. Bedarf und Szenarien
  - 7.10.2. Produkte und Dienstleistungen
  - 7.10.3. Sicherheit bei der Telearbeit



## Modul 8. Web-Server-Verwaltung

- 8.1. Einführung in Webserver
  - 8.1.1. Was ist ein Webserver?
  - 8.1.2. Architektur und Funktionsweise eines Webserver
  - 8.1.3. Ressourcen und Inhalte auf einem Webserver
  - 8.1.4. Anwendungsserver
  - 8.1.5. Proxy-Server
  - 8.1.6. Die wichtigsten Webserver auf dem Markt
  - 8.1.7. Statistiken zur Webserver-Nutzung
  - 8.1.8. Sicherheit des Webserver
  - 8.1.9. Lastausgleich bei Webservern
  - 8.1.10. Referenzen
- 8.2. Umgang mit dem HTTP-Protokoll
  - 8.2.1. Betrieb und Struktur
  - 8.2.2. Beschreibung der Abfragemethoden oder *Request Methods*
  - 8.2.3. Status-Codes
  - 8.2.4. Kopfzeilen
  - 8.2.5. Codierung des Inhalts. Code-Seiten
  - 8.2.6. Durchführung von HTTP-Anfragen im Internet mit Hilfe eines Proxys, *livehttpheaders* oder einer ähnlichen Methode, Analyse des verwendeten Protokolls
- 8.3. Beschreibung von verteilten Architekturen auf mehreren Servern
  - 8.3.1. 3-Schichten-Modell
  - 8.3.2. Fehlertoleranz
  - 8.3.3. Lastverteilung
  - 8.3.4. Sitzungsstatus-Speicher
  - 8.3.5. Cache-Speicher
- 8.4. Internet-Informationdienste (IIS)
  - 8.4.1. Was ist IIS?
  - 8.4.2. Geschichte und Entwicklung des IIS
  - 8.4.3. Die wichtigsten Vorteile und Funktionen von IIS7 und darüber hinaus
  - 8.4.4. IIS7 und neuere Architektur
- 8.5. IIS-Installation, -Verwaltung und -Konfiguration
  - 8.5.1. Präambel
  - 8.5.2. Installation der Internet-Informationdienste (IIS)
  - 8.5.3. IIS-Verwaltungstools
  - 8.5.4. Erstellen, Konfigurieren und Verwalten von Websites
  - 8.5.5. Installieren und Verwalten von IIS-Erweiterungen
- 8.6. Erweiterte Sicherheit im IIS
  - 8.6.1. Präambel
  - 8.6.2. IIS-Authentifizierung, Autorisierung und Zugriffskontrolle
  - 8.6.3. Konfigurieren einer sicheren Website auf IIS mit SSL
  - 8.6.4. In IIS 8.x implementierte Sicherheitsrichtlinien
- 8.7. Einführung in Apache
  - 8.7.1. Was ist Apache?
  - 8.7.2. Die wichtigsten Vorteile von Apache
  - 8.7.3. Hauptmerkmale von Apache
  - 8.7.4. Architektur
- 8.8. Apache-Installation und -Konfiguration
  - 8.8.1. Apache-Erstinstallation
  - 8.8.2. Apache-Konfiguration
- 8.9. Installieren und Konfigurieren der verschiedenen Apache-Module
  - 8.9.1. Installation von Apache-Modulen
  - 8.9.2. Arten von Modulen
  - 8.9.3. Sichere Apache-Konfiguration
- 8.10. Erweiterte Sicherheit
  - 8.10.1. Authentifizierung, Autorisierung und Zugangskontrolle
  - 8.10.2. Authentifizierungsmethoden
  - 8.10.3. Sichere Apache-Konfiguration mit SSL

## Modul 9. Sicherheitsprüfung

- 9.1. Einführung in Informationssysteme und deren Prüfung
  - 9.1.1. Einführung in Informationssysteme und die Rolle der IT-Prüfung
  - 9.1.2. Definitionen der Begriffe "IT-Audit" und "interne IT-Kontrolle"
  - 9.1.3. Funktionen und Ziele der IT-Prüfung
  - 9.1.4. Unterschiede zwischen interner Kontrolle und IT-Audit
- 9.2. Interne Kontrolle von Informationssystemen
  - 9.2.1. Funktionsschema eines Datenverarbeitungszentrums
  - 9.2.2. Klassifizierung der Kontrollen von Informationssystemen
  - 9.2.3. Die Goldene Regel
- 9.3. Der Prozess und die Phasen der Prüfung von Informationssystemen
  - 9.3.1. Risikobewertung (RRA) und andere IT-Prüfungsmethoden
  - 9.3.2. Durchführung einer Prüfung der Informationssysteme. Prüfungsphasen
  - 9.3.3. Grundlegende Fähigkeiten des Wirtschaftsprüfers für Informationssysteme
- 9.4. Technische Sicherheitsüberprüfung von Systemen und Netzen
  - 9.4.1. Technische Sicherheitsaudits. Penetrationstests. Vorläufige Konzepte
  - 9.4.2. Audits der Systemsicherheit. Hilfsmittel
  - 9.4.3. Audits der Netzwerksicherheit. Hilfsmittel
- 9.5. Technische Prüfung der Sicherheit von Internet und mobilen Geräten
  - 9.5.1. Internet-Sicherheitsaudit. Hilfsmittel
  - 9.5.2. Prüfung der Sicherheit von mobilen Geräten. Hilfsmittel
  - 9.5.3. Anhang 1. Aufbau des Kurzberichts und des technischen Berichts
  - 9.5.4. Anhang 2. Inventar der Werkzeuge
  - 9.5.5. Anhang 3. Methoden
- 9.6. Managementsystem für die Informationssicherheit
  - 9.6.1. IS-Sicherheit: Eigenschaften und Einflussfaktoren
  - 9.6.2. Unternehmensrisiken und Risikomanagement: Implementierung von Kontrollen
  - 9.6.3. Informationssicherheitsmanagementsystem (ISMS): Konzept und kritische Erfolgsfaktoren
  - 9.6.4. ISMS - PDCA-Modell
  - 9.6.5. ISMS ISO-IEC 27001: organisatorischer Kontext
  - 9.6.6. Abschnitt 4. Organisatorischer Kontext
  - 9.6.7. Abschnitt 5. Führungsrolle
  - 9.6.8. Abschnitt 6. Planung
  - 9.6.9. Abschnitt 7. Unterstützung
  - 9.6.10. Abschnitt 8. Betrieb
  - 9.6.11. Abschnitt 9. Leistungsbeurteilung
  - 9.6.12. Abschnitt 10. Verbesserung
  - 9.6.13. Anhang zu ISO 27001/ISO-IEC 27002: Zielsetzungen und Kontrollen
  - 9.6.14. ISMS-Audit
- 9.7. Durchführung der Prüfung
  - 9.7.1. Verfahren
  - 9.7.2. Techniken
- 9.8. Rückverfolgbarkeit
  - 9.8.1. Methodologien
  - 9.8.2. Analyse
- 9.9. Gewahrsam
  - 9.9.1. Techniken
  - 9.9.2. Ergebnisse
- 9.10. Berichterstattung und Präsentation von Beweisen
  - 9.10.1. Arten von Berichten
  - 9.10.2. Analyse der Daten
  - 9.10.3. Vorlage von Beweismitteln

## Modul 10. Sicherheit bei Online-Anwendungen

- 10.1. Schwachstellen und Sicherheitsprobleme in Online-Anwendungen
  - 10.1.1. Einführung in die Sicherheit von Online-Anwendungen
  - 10.1.2. Sicherheitsschwachstellen beim Entwurf von Webanwendungen
  - 10.1.3. Sicherheitsschwachstellen Implementierung von Webanwendungen
  - 10.1.4. Sicherheitsschwachstellen bei der Bereitstellung von Webanwendungen
  - 10.1.5. Offizielle Listen von Sicherheitslücken
- 10.2. Richtlinien und Standards für die Sicherheit von Online-Anwendungen
  - 10.2.1. Säulen der Sicherheit von Online-Anwendungen
  - 10.2.2. Managementsystem für die Informationssicherheit
  - 10.2.3. Sicherer Lebenszyklus der *Software* Entwicklung
  - 10.2.4. Standards für die Anwendungssicherheit
- 10.3. Sicherheit beim Entwurf von Webanwendungen
  - 10.3.1. Einführung in die Sicherheit von Webanwendungen
  - 10.3.2. Sicherheit beim Entwurf von Webanwendungen
- 10.4. Prüfung der Online-Sicherheit von Webanwendungen
  - 10.4.1. Analyse und Prüfung der Sicherheit von Webanwendungen
  - 10.4.2. Sicherheit bei der Bereitstellung und Produktion von Webanwendungen
- 10.5. Sicherheit von Webdiensten
  - 10.5.1. Einführung in die Sicherheit von Webdiensten
  - 10.5.2. Sicherheitsfunktionen und -technologien für Webdienste
- 10.6. Prüfung der Online-Sicherheit und des Schutzes von Webdiensten
  - 10.6.1. Bewertung der Sicherheit von Webdiensten
  - 10.6.2. Online-Schutz. *Firewalls* und *Gateways XML*
- 10.7. Ethisches *Hacking*, *Malware* und *Forensik*
  - 10.7.1. Ethisches *Hacking*
  - 10.7.2. Analyse von *Malware*
  - 10.7.3. Forensische Analyse
- 10.8. Auflösung von Vorfällen bei Webdiensten
  - 10.8.1. Überwachung
  - 10.8.2. Instrumente zur Leistungsmessung
  - 10.8.3. Eindämmungsmaßnahmen
  - 10.8.4. Analyse der Grundursache
  - 10.8.5. Proaktives Problemmanagement
- 10.9. Bewährte Verfahren zur Gewährleistung der Anwendungssicherheit
  - 10.9.1. Handbuch für bewährte Praktiken bei der Entwicklung von Online-Anwendungen
  - 10.9.2. Handbuch für bewährte Praktiken bei der Umsetzung von Online-Anwendungen
- 10.10. Häufige Fehler, die die Anwendungssicherheit untergraben
  - 10.10.1. Häufige Entwicklungsfehler
  - 10.10.2. Häufige Fehler beim Hosting
  - 10.10.3. Häufige Fehler in der Produktion



*Eine einzigartige, wichtige und entscheidende Fortbildungserfahrung, die Ihre berufliche Entwicklung fördert"*

# 05 Methodik

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: **das Relearning**.

Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem **New England Journal of Medicine** als eines der effektivsten angesehen.





“

*Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen aufgibt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"*

## Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

“

*Mit TECH werden Sie eine Art des Lernens erleben, die die Grundlagen der traditionellen Universitäten in der ganzen Welt verschiebt”*



*Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.*



*Die Studenten lernen durch gemeinschaftliche Aktivitäten und reale Fälle die Lösung komplexer Situationen in realen Geschäftsumgebungen.*

## Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

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

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit die Jurastudenten das Recht nicht nur anhand theoretischer Inhalte erlernen, sondern ihnen reale, komplexe Situationen vorlegen, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen können, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard eingeführt.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode. Während des gesamten Kurses werden die Studierenden mit mehreren realen Fällen konfrontiert. Sie müssen Ihr gesamtes Wissen integrieren, recherchieren, argumentieren und Ihre Ideen und Entscheidungen verteidigen.

## Relearning Methodik

TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

*Im Jahr 2019 erzielten wir die besten  
Lernergebnisse aller spanischsprachigen  
Online-Universitäten der Welt.*

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft auszubilden. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten Online-Universität in Spanisch zu verbessern.





In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -Instrumente ausgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

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

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten neurokognitiven kontextabhängigen E-Learnings mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.



Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:



#### Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, 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 das audiovisuelle Format angewendet, um die TECH-Online-Arbeitsmethode zu schaffen. Und das alles mit den neuesten Techniken, die dem Studenten qualitativ hochwertige Stücke aus jedem einzelnen Material zur Verfügung stellen.



#### Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte Learning from an Expert baut Wissen und Gedächtnis auf und schafft Vertrauen für zukünftige schwierige Entscheidungen.



#### Fertigkeiten und Kompetenzen Praktiken

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



#### Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u.a. In der virtuellen Bibliothek von TECH haben die Studenten Zugang zu allem, was sie für ihre Ausbildung benötigen.





#### Fallstudien

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.



#### Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

Dieses einzigartige Bildungssystem für die Präsentation multimedialer Inhalte wurde von Microsoft als "europäische Erfolgsgeschichte" ausgezeichnet.



#### Prüfung und Nachprüfung

Die Kenntnisse der Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass die Studenten überprüfen können, wie sie ihre Ziele erreichen.





06

# Qualifizierung

Der Privater Masterstudiengang in *Software Engineering* und *Computersysteme* garantiert neben der strengsten und aktuellsten Ausbildung auch den Zugang zu einem von der TECH Technologischen Universität ausgestellten Diplom.



“

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



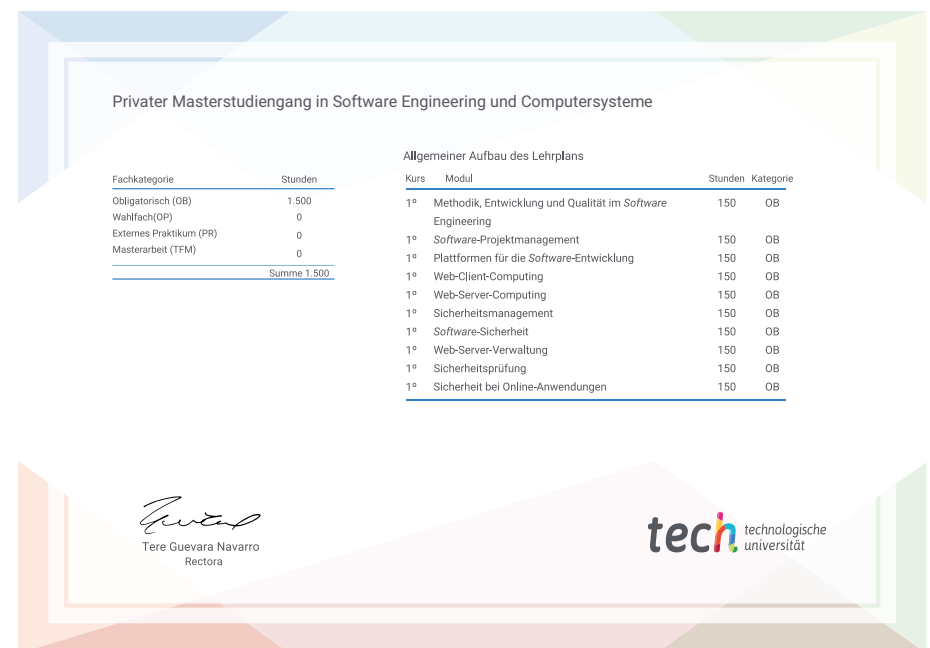
Dieser **Privater Masterstudiengang in Software Engineering und Computersysteme** enthält das vollständigste und aktuellste Programm auf dem Markt.

Sobald der Student die Prüfungen bestanden hat, erhält er/sie per Post\* mit Empfangsbestätigung das entsprechende Diplom, ausgestellt von der TECH Technologischen Universität.

Das von **TECH Technologische Universität** ausgestellte Diplom drückt die erworbene Qualifikation aus und entspricht den Anforderungen, die in der Regel von Stellenbörsen, Auswahlprüfungen und Berufsbildungsausschüssen verlangt werden.

Titel: **Privater Masterstudiengang in Software Engineering und Computersysteme**

Anzahl der offiziellen Arbeitsstunden: **1.500 Std.**



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

zukunft

gesundheit vertrauen menschen  
erziehung information tutoren  
garantie akkreditierung unterricht  
institutionen technologie lernen  
gemeinschaft verpflichtung  
persönliche betreuung innovation  
wissen gegenwart qualität  
online-Ausbildung  
entwicklung institut  
virtuelles Klassenzimmer

**tech** technologische  
universität

## Privater Masterstudiengang Software Engineering und Computersysteme

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

# Privater Masterstudiengang Software Engineering und Computersysteme

