

# Privater Masterstudiengang Fortgeschrittene Softwaretechnik

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

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

```
select =  
context.select(projects[0])
```



## Privater Masterstudiengang Fortgeschrittene Softwaretechnik

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

Internetzugang: [www.techtitude.com/de/informatik/masterstudiengang/masterstudiengang-fortgeschrittene-softwaretechnik](http://www.techtitude.com/de/informatik/masterstudiengang/masterstudiengang-fortgeschrittene-softwaretechnik)

# 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

Die Entwicklung von Informatiksystemen wäre ohne die Softwaretechnik nicht möglich. Dank des technologischen Fortschritts ist es heute möglich, immer komplexere und spezifischere Systeme und Strukturen zu finden, die auf der Grundlage von Programmierungsspezifikationen entwickelt wurden, um den Bedürfnissen und Anforderungen des Marktes gerecht zu werden. Es handelt sich um einen methodischen und geordneten Prozess, der spezielle Kenntnisse der Informatik und ihrer Werkzeuge erfordert, die der Absolvent im Laufe dieses sehr umfassenden Programms erwerben kann. Durch eine multidisziplinäre Erfahrung werden Sie in die wichtigsten IT-Prozesse und die Systemintegration eintauchen und die notwendigen Fähigkeiten erwerben, um Projekte auf höchstem Niveau auf der Grundlage der Anforderungen der *Stakeholder* durchzuführen. Und das alles in nur 12 Monaten 100%iger Online-Fortbildung.





“

*Wenn Sie auf der Suche nach einem Programm sind, das Sie dabei unterstützt, Ihr eigenes Softwareprojekt von Grund auf zu starten, ist dieser private Masterstudiengang genau das Richtige für Sie. Worauf warten Sie, um sich einzuschreiben?"*

Seit mehr als sechs Jahrzehnten steht die Softwaretechnik im Mittelpunkt der technologischen Revolution durch die Entwicklung immer komplexerer und spezialisierterer Programme und Anwendungen. Es handelt sich um einen Bereich, der vielen anderen als Stütze dient, um Fortschritte zu erzielen, und dessen Anwendung auf praktisch alle bestehenden Fachgebiete übertragen werden kann: Medizin, Landwirtschaft, Lehre, Administration, Industrie, usw. Man kann es drehen und wenden, wie man will: Selbst der einfachste Computerprozess, wie das Versenden einer E-Mail oder die Nutzung von Instant Messaging, etwas, das heutzutage ganz alltäglich ist, musste aufwändig konzipiert und programmiert werden, um seinen Zweck zu erfüllen: die Befriedigung der Bedürfnisse der Menschen.

Die vielfältigen Möglichkeiten, die sich aus dieser Wissenschaft und ihren zahlreichen Anwendungen ergeben, machen sie zu einem der gefragtesten Berufe auf dem Arbeitsmarkt, nicht nur für die Entwicklung neuer Projekte, sondern auch für die Betreuung, Wartung und Aktualisierung bestehender Projekte. Aus diesem Grund und gemäß der Maxime der TECH, allen ihren Absolventen die Möglichkeit zu bieten, sich in diesem Bereich zu spezialisieren, hat die Universität beschlossen, diesen sehr umfassenden Studiengang in fortgeschrittener Softwaretechnik einzuführen.

Es handelt sich um eine akademische Erfahrung, die 1.500 Stunden der besten theoretisch-praktischen und zusätzlichen Inhalte umfasst, die den gesamten Bereich abdecken, von den Ursprüngen bis zur Konzeption, Erstellung und Verwaltung innovativer und moderner Informationssysteme. Während der 12-monatigen Fortbildung kann sich der Informatiker mit den Besonderheiten dieses Fachgebiets vertraut machen: den technischen und strukturellen Anforderungen, den Schlüsselfaktoren für die Schaffung sicherer Architekturen, der Integration von IKT-gestützten Diensten, dem Management von Stakeholdern und ihrem Einflussbereich, der Entwicklung eines Projekts von der Planung bis zum Beginn, und vieles mehr!

Und das alles zu 100 % online, so dass der Absolvent jederzeit und überall Zugang zu diesem privaten Masterstudiengang hat, ohne Präsenzveranstaltungen oder eingeschränkte Stundenpläne. Außerdem können Sie von jedem Gerät mit Internetanschluss auf den Virtuellen Campus zugreifen, egal ob es sich um einen PC, ein Tablet oder ein Mobiltelefon handelt. Es ist daher eine einmalige Gelegenheit, sich durch einen Abschluss, der auf Ihre akademischen Bedürfnisse und die anspruchsvollsten Anforderungen der heutigen IT-Branche abgestimmt ist, auf die Softwaretechnik zu spezialisieren.

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

- ◆ Die anschaulichen, schematischen und äußerst praktischen Inhalte, mit denen es konzipiert ist, liefern praktische Informationen zu den Disziplinen, die für die berufliche Praxis unerlässlich sind
- ◆ Der anschauliche, schematische und äußerst praxisnahe Inhalt soll 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
- ◆ Ihr besonderer Schwerpunkt liegt auf innovativen Methoden
- ◆ 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



*Ein Abschluss, der die Grundlagen der Softwaretechnik vertieft: von ihren Ursprüngen, bis hin zu den heute durchgeführten Datenverarbeitungsprozessen"*

“

*Sie erhalten Zugang zu einem praktischen Leitfaden, der die Grundsätze der Softwaretechnik abdeckt, vom ersten Prozess bis hin zur Konstruktion und Bereitstellung"*

Zu den Lehrkräften des Programms gehören Fachleute aus der Branche, die ihre Berufserfahrung in diese Fortbildung einbringen, sowie renommierte Fachleute von Referenzgesellschaften und angesehenen 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 Fachkraft versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Zu diesem Zweck wird sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

*Ein Programm, das Sie in die Lage versetzt, von Anfang an die innovativsten Strategien der Anforderungsmodelle anzuwenden.*

*Möchten Sie die notwendigen Fähigkeiten erwerben, um komplexe und alternative Architekturen durch Datenfluss zu entwerfen? Mit diesem privaten Masterstudiengang erreichen Sie es in weniger als 12 Monaten.*



# 02 Ziele

Die Softwaretechnik ist zu einem der wichtigsten Instrumente der heutigen technologischen Entwicklung geworden. Ohne die Arbeit von Millionen von IT-Fachkräften wäre es nicht möglich gewesen, das sehr hohe Niveau der digitalen Innovation zu erreichen, das wir heute kennen. Ziel dieses privaten Masterstudiengangs ist es daher, Absolventen, die sich für diesen Bereich interessieren, alle Informationen zu vermitteln, die sie benötigen, um diesen Bereich im Detail zu kennen und ihre eigenen Programme und Anwendungen erfolgreich, effizient und mit höchster Qualität und Sorgfalt entwickeln zu können.





“

*Wenn eines Ihrer Ziele mit diesem privaten Masterstudiengang die Beherrschung von Scrum und anderen agilen Methodiken ist, bietet TECH Ihnen alles, was Sie brauchen, um diese Techniken garantiert zu beherrschen"*



## Allgemeine Ziele

---

- ♦ Wissenschaftliche und technologische Spezialisierung sowie Vorbereitung auf die Berufspraxis der Softwaretechnik mit einer transversalen und vielseitigen Weiterbildung, die an die neuen Technologien und Innovationen in diesem Bereich angepasst ist
- ♦ Erwerb umfassender Kenntnisse auf dem Gebiet der Softwaretechnik, aber auch auf dem Gebiet der Berechnung und der Computerstruktur, einschließlich der mathematischen, statistischen und physikalischen Grundlagen, die für das Ingenieurwesen wesentlich sind



*Sie werden sich intensiv mit dem Testen von Software mittels TDD, ATDD und BDD befassen, so dass Sie in der Lage sind, Software-Strukturen von höchster Qualität zu erstellen"*





## Spezifische Ziele

---

### Modul 1. Softwaretechnik

- ◆ Schaffung der Grundlagen der Softwareentwicklung und -modellierung, indem die wichtigsten Prozesse und Konzepte erlernt werden
- ◆ Verständnis des Softwareprozesses und der verschiedenen Modelle für die Softwareentwicklung einschließlich agiler Technologien
- ◆ Kenntnis der wichtigsten Normen in Bezug auf Softwarequalität und Projektmanagement

### Modul 2. Fortgeschrittene Softwaretechnik

- ◆ Vertiefte Kenntnisse der verschiedenen agilen Methoden, die in der Softwareentwicklung eingesetzt werden
- ◆ Lernen Sie, mit Scrum, extremer Programmierung und wiederverwendungs-basierten Softwareentwicklungstechniken zu entwickeln
- ◆ Verstehen der Konzepte und Prozesse des Softwaredesigns, Lernen der Designarchitektur und des komponentenbasierten und musterbasierten Designs
- ◆ Einführung in das Konzept von DevOps und seine wichtigsten Praktiken
- ◆ Erlernen der Durchführung von Softwaretests mit Methoden wie Test Driven Development, Acceptance Test Driven Development, Behavior Driven Development, BDD und Cucumber
- ◆ Verstehen verschiedener Systemarchitekturen und Softwareentwurfsmuster sowie der Architektur von Cloud-Anwendungen

### Modul 3. Anforderungsmanagement

- ◆ Verständnis des Anforderungsmanagements, seiner Entwicklung, Ausarbeitung, Verhandlung und Validierung
- ◆ Lernen Sie die Modellierung von Anforderungen und die verschiedenen Elemente wie Szenarien, Informationen, Analyseklassen, Fluss, Verhalten und Muster
- ◆ Verständnis der Bedeutung der Anforderungsanalyse im Softwareentwicklungsprozess
- ◆ Durchführung von Anforderungsanalysen und die richtige Dokumentation davon erlernen
- ◆ Vertiefung des Verständnisses von Anforderungsquellen und Techniken zur Anforderungserhebung, da diese ein wesentlicher Bestandteil des Prozesses sind
- ◆ Verständnis der Prozesse der Anforderungsvalidierung und -aushandlung sowie der Anforderungsmodellierung und -verwaltung
- ◆ Erwerb der notwendigen Kenntnisse für das Management kritischer Systeme und die formale Spezifikation von Anforderungen

### Modul 4. Prozesse der Softwaretechnik

- ◆ Vertiefung der Verbesserung des Softwareentwicklungsprozesses und der Softwarequalität anhand von ISO/IEC-Normen
- ◆ Verstehen und Anwenden von Prototypen als wesentlicher Bestandteil des Entwicklungsprozesses
- ◆ Kenntnis des Rahmens für die Softwaretechnik und der Norm ISO/IEC 12207
- ◆ Die Merkmale des einheitlichen Softwareentwicklungsprozesses und der Planung im Kontext der agilen Softwareentwicklung kennenlernen
- ◆ Die verschiedenen Arten von verteiltem Softwaredesign und dienstorientierten Softwarearchitekturen kennenlernen
- ◆ Die grundlegenden Konzepte für die Gestaltung grafischer Benutzeroberflächen lernen
- ◆ Verstehen der Grundlagen der Entwicklung von Web-Applikationen

### Modul 5. Qualität und Prüfung von Informationssystemen

- ◆ Vertiefung der Strategien und Techniken des Softwaretests, der Faktoren für die Softwarequalität und der verschiedenen verwendeten Messgrößen
- ◆ Aneignung von grundlegenden Kenntnissen über IT-Sicherheitsmanagementsysteme
- ◆ Einführung in die Konzepte des geistigen Eigentums in Informationsmanagementsystemen
- ◆ Vorbereitung der Studenten auf die Erstellung von Plänen für die Geschäftskontinuität und die Wiederherstellung im Notfall
- ◆ Die Planung des Sicherheitsmanagements und den Umgang mit den wichtigsten Mechanismen für den Schutz von Informationswerten erlernen
- ◆ Verständnis der verschiedenen Arten von Audits und den Ablauf eines IT-Audits

### Modul 6. Systemintegration

- ◆ Die wesentlichen Konzepte im Zusammenhang mit Informationssystemen im Unternehmen erwerben, sowie die Möglichkeiten und Bedürfnisse von Informationssystemen im Unternehmen identifizieren
- ◆ Kenntnis der Grundlagen von *Business Intelligence*, ihrer Strategien und Implementierung sowie der Gegenwart und Zukunft von BI
- ◆ Die Funktionsweise von Systemen für die integrierte Verwaltung von Unternehmensressourcen verstehen
- ◆ Die digitale Transformation aus dem Blickwinkel der Unternehmensinnovation, des Finanz- und Produktionsmanagements, des Marketings und des Personalmanagements verstehen

### Modul 7. Wiederverwendung von Software

- ◆ Kenntnisse über das Gesamtbild der Software-Wiederverwendungsstrategie
- ◆ Erlernen der verschiedenen Muster für die Wiederverwendung von Software, sowohl in Bezug auf Design, Erstellung, Struktur und Verhalten
- ◆ Einführung in das Konzept der *Frameworks*, sowie in die wichtigsten Typen, z. B. für die Gestaltung grafischer Benutzeroberflächen, die Entwicklung von Webanwendungen und die Verwaltung der Objektpersistenz in Datenbanken
- ◆ Die Funktionsweise des derzeit weit verbreiteten Model-View-Controller (MVC)-Musters verstehen

### Modul 8. Dienstleistungen der Informationstechnologie

- ◆ Fortbildung in den Bereichen IKT-Investitionsentscheidungen und Planung von Informationssystemen
- ◆ Kenntnis der Kontrollziele für Informations- und verwandte Technologien (COBIT)
- ◆ Erlernen der Funktionsweise der Information Technology Infrastructure Library (ITIL), der Strategien, des Service-Designs, der Übergänge und des Betriebs
- ◆ Vertiefung in das Service Management System, mit Kenntnis der Grundprinzipien von UNE-ISO/IEC 20000-1, der Struktur der ISO/IEC 20000 Normenreihe und der Anforderungen an das Service Management System (SMS)
- ◆ Verstehen der Funktionsweise von Informationssystemen und -technologien, ihrer Komponenten, Klassifizierungen, Architekturen und Formen der Systemintegration
- ◆ Erlernen der Norm ISO/IEC 12207, der Analyse, des Entwurfs, der Implementierung und der Abnahme von Informationssystemen

### Modul 9. Sicherheit von Informationssystemen

- ◆ Entwicklung des Zeitplans für Zeitmanagement, Budgetentwicklung und Risikobewältigung
- ◆ Analyse der Art von Netzwerkangriffen und der verschiedenen Arten von Sicherheitsarchitekturen
- ◆ Die verschiedenen Techniken zum Schutz von Systemen und zur Entwicklung von sicherem Code verstehen
- ◆ Die wesentlichen Komponenten von *Botnets* und Spam sowie von Malware und böartigem Code verstehen
- ◆ Grundlagen für die forensische Analyse in der Welt der Software- und IT-Prüfung legen
- ◆ Eine globale Perspektive auf Sicherheit, Kryptographie und klassische Kryptoanalysen gewinnen
- ◆ Die Grundlagen der symmetrischen und asymmetrischen Kryptographie sowie deren Hauptalgorithmen verstehen

### Modul 10. Projektmanagement

- ◆ Verstehen der Funktionsweise des Qualitätsmanagements in Projekten, einschließlich Planung, Sicherung, Kontrolle, statistischer Konzepte und verfügbarer Werkzeuge
- ◆ Verstehen der Funktionsweise der Prozesse der Beschaffung, Durchführung, Überwachung, Kontrolle und des Abschlusses eines Projekts
- ◆ Aneignung der wesentlichen Kenntnisse im Zusammenhang mit der beruflichen Verantwortung im Projektmanagement
- ◆ Kenntnisse der grundlegenden Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
- ◆ Die verschiedenen Phasen des Projektmanagements wie Initiierung, Planung, Verwaltung von *Stakeholdern* und Umfang zu verstehen

# 03

## Kompetenzen

Eines der wichtigsten Merkmale aller von TECH angebotenen Studiengänge ist, dass sie es den Absolventen ermöglichen, ihre Fähigkeiten im Laufe des Kurses zu perfektionieren. Dies ist möglich durch den Erwerb von Fachkenntnissen in einem bestimmten Bereich, in diesem Fall der Softwaretechnik. Der Schlüssel dazu liegt jedoch in der Lösung von Anwendungsbeispielen, die auf Situationen im aktuellen Kontext des IT-Panoramas basieren und dank derer die Studenten ihre Fähigkeiten in die Praxis umsetzen, die im Lehrplan beschriebenen Strategien anwenden und ihre Fähigkeiten bei der Konzeption, Analyse, Verwaltung und Einführung von Anwendungen und Programmen erweitern können.



“

*Das beste Programm auf dem aktuellen akademischen Markt, um die innovativsten und effektivsten Prototyping-Techniken im Softwareumfeld in Ihre IT-Praxis zu implementieren"*



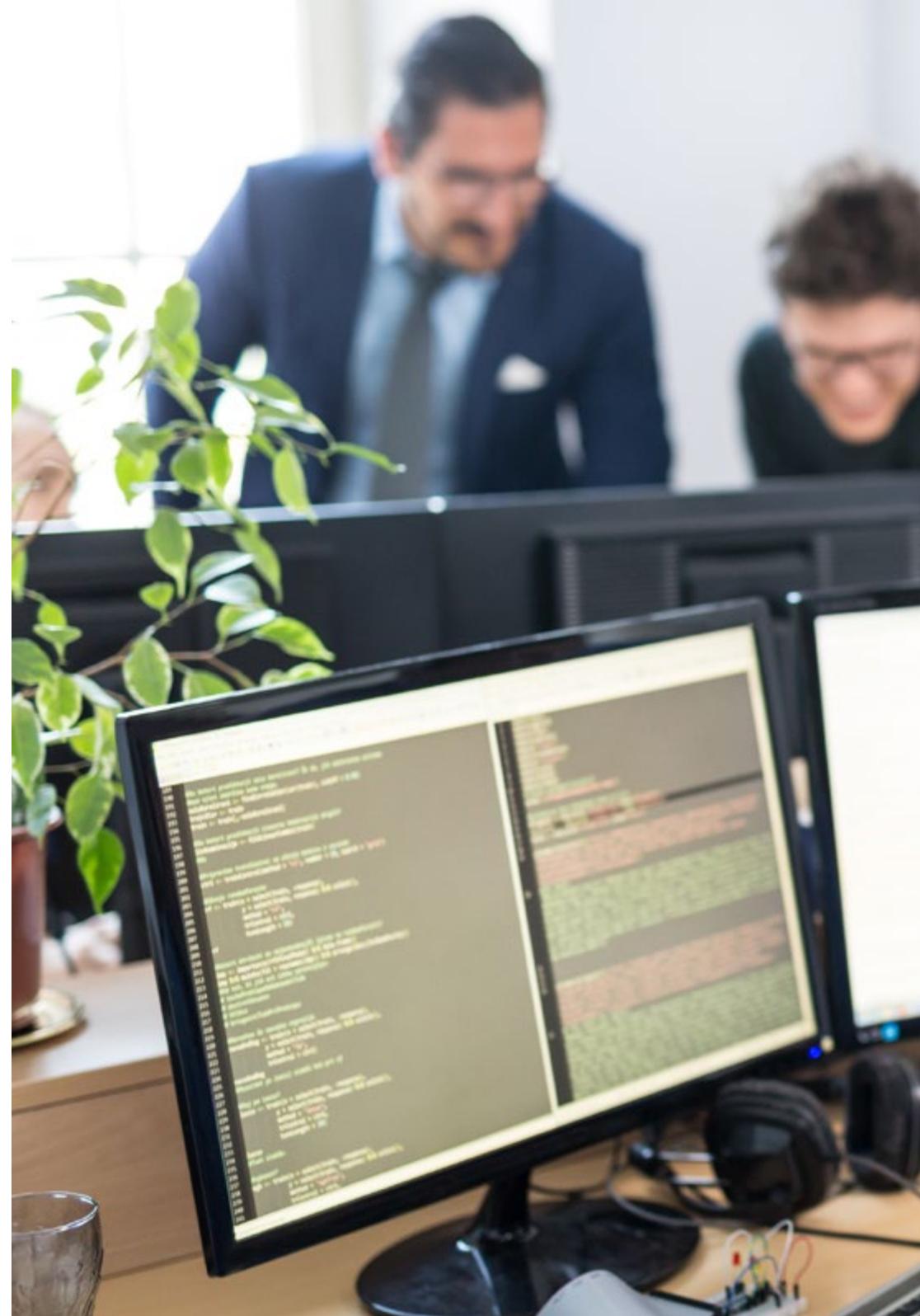
## Allgemeine Kompetenzen

---

- ◆ Antworten auf die aktuellen Bedürfnisse des Bereichs Fortgeschrittene Softwaretechnik
- ◆ Beherrschung der verschiedenen Arbeitssysteme der fortgeschrittenen Softwaretechnik
- ◆ Beschreibung und Nutzung von freier Software und öffentlichem Wissen, das im Internet verfügbar ist



*Dank der Kenntnis von Softwareteststrategien und -techniken werden Sie in der Lage sein, an der Verbesserung der Projekte anderer mitzuarbeiten, um sie intuitiver und spezialisierter zu machen"*





## Spezifische Kompetenzen

---

- ◆ Erlangung eines tiefgreifenden Verständnisses aller Facetten der Mensch-Computer-Interaktion und wie diese mit IT-Entwicklungen zusammenhängen
- ◆ Beherrschung der Nutzung von Datenbanken
- ◆ Entwicklung verschiedener Arten von Netzwerkanwendungen
- ◆ Arbeiten als Software-Ingenieur
- ◆ Kontrolle der Nutzung fortgeschrittener Datenbanken
- ◆ Durchführung einer fortgeschrittenen Programmierung
- ◆ Wissen, wie Software wiederverwendet wird
- ◆ Schaffung von Schnittstellen und Anwendungen im Netz

# 04

## Struktur und Inhalt

Der Lehrplan wurde auf der Grundlage der pädagogischen Effizienz entwickelt, wobei die Inhalte sorgfältig ausgewählt wurden, um einen vollständigen Kurs anzubieten, der alle für die Erlangung echter Kenntnisse des Themas wesentlichen Bereiche umfasst. Mit den neuesten Updates und Aspekten des Sektors.



“

*Alle Fächer und Wissensgebiete wurden in einem vollständigen und absolut aktuellen Lehrplan zusammengestellt, um den Studenten sowohl in der Theorie als auch in der Praxis auf das höchste Niveau zu bringen"*

## Modul 1. Softwaretechnik

- 1.1. Einführung in die Softwaretechnik und Modellierung
  - 1.1.1. Die Natur der Software
  - 1.1.2. Die Besonderheit von Webapps
  - 1.1.3. Softwaretechnik
  - 1.1.4. Der Software-Prozess
  - 1.1.5. Die Praxis der Softwaretechnik
  - 1.1.6. Software-Mythen
  - 1.1.7. Wie alles beginnt
  - 1.1.8. Objektorientierte Konzepte
  - 1.1.9. Einführung in UML
- 1.2. Der Software-Prozess
  - 1.2.1. Ein allgemeines Prozessmodell
  - 1.2.2. Vorgeschriebene Prozessmodelle
  - 1.2.3. Spezialisierte Prozessmodelle
  - 1.2.4. Der vereinheitlichte Prozess
  - 1.2.5. Personal- und Teamprozessmodelle
  - 1.2.6. Was ist Agilität?
  - 1.2.7. Was ist ein agiler Prozess?
  - 1.2.8. Scrum
  - 1.2.9. Werkzeugkasten für agile Prozesse
- 1.3. Prinzipien als Leitfaden für die Praxis der Softwareentwicklung
  - 1.3.1. Leitprinzipien des Prozesses
  - 1.3.2. Prinzipien als Leitfaden für die Praxis
  - 1.3.3. Grundsätze der Kommunikation
  - 1.3.4. Grundsätze der Planung
  - 1.3.5. Grundsätze der Modellierung
  - 1.3.6. Konstruktionsprinzipien
  - 1.3.7. Grundsätze für die Einführung
- 1.4. Verständnis der Anforderungen
  - 1.4.1. Anforderungsmanagement
  - 1.4.2. Schaffung der Grundlagen
  - 1.4.3. Bedarfsermittlung
  - 1.4.4. Entwicklung von Anwendungsfällen
  - 1.4.5. Ausarbeitung des Anforderungsmodells
  - 1.4.6. Aushandeln von Anforderungen
  - 1.4.7. Validierung der Anforderungen
- 1.5. Anforderungsmodellierung I: Szenarien, Informations- und Analyseklassen
  - 1.5.1. Analyse der Anforderungen
  - 1.5.2. Szenario-basiertes Modell
  - 1.5.3. UML-Modelle, die den Anwendungsfall liefern
  - 1.5.4. Konzepte der Datenmodellierung
  - 1.5.5. Klassen-basiertes Modell
  - 1.5.6. Klassendiagramme
- 1.6. Anforderungsmodellierung II: Fluss, Verhalten und Muster
  - 1.6.1. Anforderungen die die Strategien gestalten
  - 1.6.2. Flussorientierte Modellierung
  - 1.6.3. Zustandsdiagramme
  - 1.6.4. Erstellung eines Verhaltensmodells
  - 1.6.5. Sequenzdiagramme
  - 1.6.6. Kommunikationsdiagramme
  - 1.6.7. Muster für die Modellierung von Anforderungen
- 1.7. Konzepte des Designs
  - 1.7.1. Design im Kontext der Softwaretechnik
  - 1.7.2. Der Entwurfsprozess
  - 1.7.3. Konzepte des Designs
  - 1.7.4. Objektorientierte Konzepte des Designs
  - 1.7.5. Das Designmodell

- 1.8. Design der Architektur
  - 1.8.1. Software-Architektur
  - 1.8.2. Architektonische Gattungen
  - 1.8.3. Architektonische Stile
  - 1.8.4. Architektonischer Design
  - 1.8.5. Entwicklung von alternativen Designs für die Architektur
  - 1.8.6. Abbildung der Architektur mit Hilfe von Datenflüssen
- 1.9. Design auf Komponentenebene und musterbasierter Entwurf
  - 1.9.1. Was ist eine Komponente?
  - 1.9.2. Klassenbasiertes Komponentendesign
  - 1.9.3. Verwirklichung des Designs auf Komponentenebene
  - 1.9.4. Design der traditionellen Komponenten
  - 1.9.5. Komponentenbasierte Entwicklung
  - 1.9.6. Entwurfsmuster
  - 1.9.7. Musterbasiertes Softwaredesign
  - 1.9.8. Architektonische Muster
  - 1.9.9. Musterdesign auf Komponentenebene
  - 1.9.10. Musterdesign für Benutzeroberflächen
- 1.10. Softwarequalität und Projektmanagement
  - 1.10.1. Qualität
  - 1.10.2. Software Qualität
  - 1.10.3. Das Dilemma der Softwarequalität
  - 1.10.4. Erreichen von Softwarequalität
  - 1.10.5. Software-Qualitätssicherung
  - 1.10.6. Das administrative Spektrum
  - 1.10.7. Personal
  - 1.10.8. Das Produkt
  - 1.10.9. Der Prozess
  - 1.10.10. Das Projekt
  - 1.10.11. Grundsätze und Praktiken

## Modul 2. Fortgeschrittene Softwaretechnik

- 2.1. Einführung in agile Methodologien
  - 2.1.1. Prozessmodelle und Methodologien
  - 2.1.2. Agilität und agile Prozesse
  - 2.1.3. Das Agile Manifest
  - 2.1.4. Einige agile Methodologien
  - 2.1.5. Agil vs. Traditionell
- 2.2. Scrum
  - 2.2.1. Ursprünge und Philosophie von Scrum
  - 2.2.2. Scrum Werte
  - 2.2.3. Scrum Prozessablauf
  - 2.2.4. Scrum Rollen
  - 2.2.5. Scrum Artefakte
  - 2.2.6. Scrum Ereignisse
  - 2.2.7. Benutzerberichte
  - 2.2.8. Scrum Erweiterungen
  - 2.2.9. Agile Schätzungen
  - 2.2.10. Scrum Skalierung
- 2.3. Extreme Programmierung
  - 2.3.1. Grundprinzip und Überblick über XP
  - 2.3.2. Der Lebenszyklus von XP
  - 2.3.3. Die fünf Grundwerte
  - 2.3.4. Die zwölf Kernpraktiken von XP
  - 2.3.5. Rollen der Teilnehmer
  - 2.3.6. Industrie XP
  - 2.3.7. Kritische Beurteilung von XP
- 2.4. Softwareentwicklung auf der Grundlage von Wiederverwendung
  - 2.4.1. Wiederverwendung von Software
  - 2.4.2. Stufen der Wiederverwendung von Codes
  - 2.4.3. Spezifische Techniken der Wiederverwendung
  - 2.4.4. Komponentenbasierte Entwicklung
  - 2.4.5. Vorteile und Probleme der Wiederverwendung
  - 2.4.6. Planung der Wiederverwendung

- 2.5. Systemarchitektur und Software-Entwurfsmuster
  - 2.5.1. Architektonisches Design
  - 2.5.2. Allgemeine architektonische Muster
  - 2.5.3. Fehlertolerante Architekturen
  - 2.5.4. Architekturen verteilter Systeme
  - 2.5.5. Entwurfsmuster
  - 2.5.6. Gamma Muster
  - 2.5.7. Muster für die Interaktionsgestaltung
- 2.6. Architektur von Cloud-Anwendungen
  - 2.6.1. Grundlagen des Cloud Computing
  - 2.6.2. Qualität von Cloud-Anwendungen
  - 2.6.3. Architektonische Stile
  - 2.6.4. Entwurfsmuster
- 2.7. Software-Tests: TDD, ATDD und BDD
  - 2.7.1. Überprüfung und Validierung von Software
  - 2.7.2. Software-Tests
  - 2.7.3. Test Driven Development (TDD)
  - 2.7.4. Acceptance Test Driven Development (ATDD)
  - 2.7.5. Behavior Driven Development (BDD)
  - 2.7.6. BDD y Cucumber
- 2.8. Verbesserung von Software-Prozessen
  - 2.8.1. Verbesserung von Software-Prozessen
  - 2.8.2. Der Prozess der Prozessverbesserung
  - 2.8.3. Reifegradmodelle
  - 2.8.4. CMMI-Modell
  - 2.8.5. CMMI V2.0
  - 2.8.6. CMMI und Agilität

- 2.9. Qualität von Softwareprodukten: SQuaRE
  - 2.9.1. Qualität der Software
  - 2.9.2. Qualitätsmodelle für Softwareprodukte
  - 2.9.3. ISO/IEC 25000-Familie
  - 2.9.4. ISO/IEC 25010: Qualitätsmodell und Qualitätsmerkmale
  - 2.9.5. ISO/IEC 25012: Datenqualität
  - 2.9.6. ISO/IEC 25020: Messung der Softwarequalität
  - 2.9.7. ISO/IEC 25022, 25023 und 25024: Messgrößen für die Software- und Datenqualität
  - 2.9.8. ISO/IEC 25040: Bewertung der Software
  - 2.9.9. Der Prozess der Zertifizierung
- 2.10. Einführung in DevOps
  - 2.10.1. DevOps-Konzept
  - 2.10.2. Grundlegende Praktiken

### Modul 3. Anforderungsmanagement

- 3.1. Einführung in das Anforderungsmanagement
  - 3.1.1. Die Bedeutung der Anforderungen
  - 3.1.2. Konzept der Anforderung
  - 3.1.3. Dimensionen der Anforderungen
  - 3.1.4. Stufen und Arten von Anforderungen
  - 3.1.5. Eigenschaften der Anforderungen
  - 3.1.6. Das Anforderungsmanagement
  - 3.1.7. Der Prozess des Anforderungsmanagement
  - 3.1.8. *Frameworks* für Anforderungsmanagement
  - 3.1.9. Bewährte Praktiken im Anforderungsmanagement
  - 3.1.10. Der Business-Analyst
- 3.2. Die Quellen der Anforderungen
  - 3.2.1. Das Netzwerk der Anforderungen
  - 3.2.2. Die *Stakeholder*
  - 3.2.3. Geschäftliche Anforderungen
  - 3.2.4. Dokument zur Vision und zum Anwendungsbereich

- 3.3. Techniken zur Erhebung von Anforderungen
  - 3.3.1. Die Erhebung von Anforderungen
  - 3.3.2. Probleme der Erhebung von Anforderungen
  - 3.3.3. Kontexte der Entdeckung
  - 3.3.4. Interviews
  - 3.3.5. Beobachtung und „Lernen“
  - 3.3.6. Ethnographie
  - 3.3.7. Workshops
  - 3.3.8. *Focus groups*
  - 3.3.9. Umfragen
  - 3.3.10. *Brainstorming* und kreative Techniken
  - 3.3.11. Gruppenmedien
  - 3.3.12. Analyse der Systemschnittstellen
  - 3.3.13. Dokumentenanalyse und "Archäologie"
  - 3.3.14. Anwendungsbeispiele und Szenarien
  - 3.3.15. Prototypen
  - 3.3.16. Reverse Engineering
  - 3.3.17. Wiederverwendung von Anforderungen
  - 3.3.18. Bewährte Erhebungsmethoden
- 3.4. Benutzeranforderungen
  - 3.4.1. Personen
  - 3.4.2. Anwendungsfälle und Anwenderberichte
  - 3.4.3. Szenarien
  - 3.4.5. Arten von Szenarien
  - 3.4.6. Wie man Szenarien entdeckt
- 3.5. Prototyping Techniken
  - 3.5.1. Prototyping
  - 3.5.2. Prototypen nach ihrem Anwendungsbereich
  - 3.5.3. Prototypen nach ihrer Aktualität
  - 3.5.4. Die Treue eines Prototyps
  - 3.5.5. Prototypen von Benutzeroberflächen
  - 3.5.6. Bewertung von Prototypen
- 3.6. Analyse der Anforderungen
  - 3.6.1. Die Analyse der Anforderungen
  - 3.6.2. Bewährte Praktiken bei der Anforderungsanalyse
  - 3.6.3. Das Wörterbuch der Daten
  - 3.6.4. Priorisierung der Anforderungen
- 3.7. Dokumentation der Anforderungen
  - 3.7.1. Das Dokument zur Anforderungsspezifikation
  - 3.7.2. Aufbau und Inhalt eines SRS
  - 3.7.3. Dokumentation in natürlicher Sprache
  - 3.7.4. EARS: Easy Approach to Requirements Syntax
  - 3.7.5. Nichtfunktionale Anforderungen
  - 3.7.6. Attribute und Vorlagen in Tabellenform
  - 3.7.7. Bewährte Praktiken bei Spezifikationen
- 3.8. Validierung und Verhandlung von Anforderungen
  - 3.8.1. Validierung der Anforderungen
  - 3.8.2. Techniken zur Validierung von Anforderungen
  - 3.8.3. Verhandlung von Anforderungen
- 3.9. Modellierung und Anforderungsmanagement
  - 3.9.1. Die Modellierung der Anforderungen
  - 3.9.2. Die Nutzerperspektive
  - 3.9.3. Die Perspektive von Daten
  - 3.9.4. Die funktionale oder flussorientierte Perspektive
  - 3.9.5. Die Verhaltensperspektive
  - 3.9.6. Die Volatilität der Anforderungen
  - 3.9.7. Prozess des Anforderungsmanagements
  - 3.9.8. Werkzeuge für das Anforderungsmanagement
  - 3.9.9. Bewährte Praktiken im Anforderungsmanagement
- 3.10. Kritische Systeme und formale Spezifikation
  - 3.10.1. Kritische Systeme
  - 3.10.2. Risikoorientierte Spezifikation
  - 3.10.3. Formale Spezifikation

## Modul 4. Prozesse der Softwaretechnik

- 4.1. Rahmen der Softwaretechnik
  - 4.1.1. Eigenschaften der Software
  - 4.1.2. Die wichtigsten Prozesse der Softwaretechnik
  - 4.1.3. Prozessmodelle für die Softwareentwicklung
  - 4.1.4. Standard-Referenzrahmen für den Softwareentwicklungsprozess: die Norm ISO/IEC 12207
- 4.2. Einheitlicher Softwareentwicklungsprozess
  - 4.2.1. Einheitlicher Prozess
  - 4.2.2. Dimensionen des einheitlichen Prozesses
  - 4.2.3. Anwendungsfallorientierter Entwicklungsprozess
  - 4.2.4. Grundlegende Unified Process Workflows
- 4.3. Planung im Rahmen der agilen Softwareentwicklung
  - 4.3.1. Eigenschaften der agilen Softwareentwicklung
  - 4.3.2. Unterschiedliche Planungszeithorizonte in der agilen Entwicklung
  - 4.3.3. Rahmen für die agile Entwicklung nach Scrum und Planung von Zeithorizonten
  - 4.3.4. Anwenderberichte als Planungs- und Schätzungseinheit
  - 4.3.5. Übliche Techniken zur Ableitung einer Schätzung
  - 4.3.6. Skalen für die Interpretation von Schätzungen
  - 4.3.7. *Planning poker*
  - 4.3.8. Gängige Planungsarten: Lieferplanung und Iterationsplanung
- 4.4. Designstile für verteilte Software und dienstorientierte Softwarearchitekturen
  - 4.4.1. Kommunikationsmodelle in verteilten Softwaresystemen
  - 4.4.2. Zwischenschicht oder middleware
  - 4.4.3. Architekturmuster für verteilte Systeme
  - 4.4.4. Allgemeiner Prozess zur Entwicklung von Softwarediensten
  - 4.4.5. Gestaltungsaspekte von Softwarediensten
  - 4.4.6. Zusammensetzung der Dienstleistungen
  - 4.4.7. Architektur der Webdienste
  - 4.4.8. Komponenten für Infrastruktur und SOA
- 4.5. Einführung in die modellgetriebene Softwareentwicklung
  - 4.5.1. Das Konzept des Modells
  - 4.5.2. Modellgestützte Softwareentwicklung
  - 4.5.3. MDA Rahmen für modellbasierte Entwicklung
  - 4.5.4. Elemente eines Transformationsmodells
- 4.6. Gestaltung der grafischen Benutzeroberfläche
  - 4.6.1. Grundlagen der Gestaltung der Benutzeroberfläche
  - 4.6.2. Architektonische Entwurfsmuster für interaktive Systeme: Model-View-Controller(MVC)
  - 4.6.3. Benutzererfahrung (UX User Experience)
  - 4.6.4. Benutzerzentriertes Design
  - 4.6.5. Prozess der Analyse und Gestaltung grafischer Benutzeroberflächen
  - 4.6.6. Benutzerfreundlichkeit der Benutzeroberfläche
  - 4.6.7. Zugänglichkeit der Benutzeroberflächen
- 4.7. Gestaltung von Web-Applikationen
  - 4.7.1. Eigenschaften von Web-Applikationen
  - 4.7.2. Benutzeroberfläche einer Web-Applikation
  - 4.7.3. Gestaltung der Navigation
  - 4.7.4. Grundlegendes Interaktionsprotokoll für Web-Applikationen
  - 4.7.5. Architekturstile für Web-Applikationen
- 4.8. Strategien und Techniken der Softwareprüfung und Faktoren der Softwarequalität
  - 4.8.1. Prüfstrategien
  - 4.8.2. Gestaltung von Testfällen
  - 4.8.3. Preis-Leistungs-Verhältnis
  - 4.8.4. Qualitätsmodelle
  - 4.8.5. ISO/IEC 25000-Normenfamilie (SQuaRE)
  - 4.8.6. Modell der Produktqualität (ISO 2501n)
  - 4.8.7. Modelle der Datenqualität (ISO 2501n)
  - 4.8.8. Software-Qualitätsmanagement

- 4.9. Einführung in Messgrößen in der Softwaretechnik
  - 4.9.1. Grundlegende Konzepte: Messgrößen, Metriken und Indikatoren
  - 4.9.2. Arten von Messgrößen in der Softwaretechnik
  - 4.9.3. Der Messprozess
  - 4.9.4. ISO 25024. Externe und verwendete Qualitätsmessgrößen
  - 4.9.5. Objektorientierte Messgrößen
- 4.10. Software-Wartung und Reengineering
  - 4.10.1. Wartungsprozess
  - 4.10.2. Standardrahmen für Wartungsprozesse. ISO/IEC 14764
  - 4.10.3. Prozessmodell für das Software-Reengineering
  - 4.10.4. Reverse Engineering

## Modul 5. Qualität und Prüfung von Informationssystemen

- 5.1. Einführung in Informationssicherheits-Managementsysteme
  - 5.1.1. Grundlegende Prinzipien des ISMS
  - 5.1.2. Goldene Regeln des ISMS
  - 5.1.3. Die Rolle der IT-Auditierung in ISMS
- 5.2. Planung im Sicherheitsmanagement
  - 5.2.1. Konzepte für das Sicherheitsmanagement
  - 5.2.2. Klassifizierung von Informationen: Ziele, Konzepte und Rollen
  - 5.2.3. Umsetzung der Sicherheitspolitik: Sicherheitspolitik, Standards und Verfahren
  - 5.2.4. Risikomanagement: Grundsätze und Analyse des Risikos von Informationsgütern
- 5.3. Wichtigste Mechanismen für den Schutz von Informationsgütern I
  - 5.3.1. Überblick über die wichtigsten kryptografischen Werkzeuge zum Schutz der CID-Trias
  - 5.3.2. Berücksichtigung der Erfordernisse des Datenschutzes, der Anonymität und der ordnungsgemäßen Verwaltung der Rückverfolgbarkeit der Nutzer
- 5.4. Wichtigste Mechanismen für den Schutz von Informationsgütern II
  - 5.4.1. Kommunikationssicherheit: Protokolle, Geräte und Sicherheitsarchitekturen
  - 5.4.2. Sicherheit der Betriebssysteme
- 5.5. Interne ISMS-Kontrollen
  - 5.5.1. Taxonomie der ISMS-Kontrollen: administrative, logische und physische Kontrollen
  - 5.5.2. Klassifizierung der Kontrollen nach der Art und Weise, wie sie die Bedrohung angehen: Kontrollen zur Vorbeugung, Aufdeckung und Korrektur von Bedrohungen
  - 5.5.3. Implementierung von internen Kontrollsystemen in ISMS
- 5.6. Arten von Audits
  - 5.6.1. Unterschied zwischen Audit und interner Prüfung
  - 5.6.2. Internes vs. externes Audit
  - 5.6.3. Klassifizierung des Audits nach dem Ziel und der Art der Analyse
- 5.7. Drehbuchautor und Drehbuch: Subjekt und Objekt geschützt durch das Recht des geistigen Eigentums
  - 5.7.1. Einführung in Penetrationstests und forensische Analyse
  - 5.7.2. Definition und Relevanz der Konzepte des *fingerprinting* und *footprinting*
- 5.8. Scannen auf Schwachstellen und Überwachung des Netzwerkverkehrs
  - 5.8.1. Werkzeuge für die Schwachstellenanalyse in Systemen
  - 5.8.2. Die wichtigsten Schwachstellen im Zusammenhang mit Web-Applikationen
  - 5.8.3. Analyse der Kommunikationsprotokollen
- 5.9. Der Prozess des IT-Audits
  - 5.9.1. Lebenszykluskonzept in der Systementwicklung
  - 5.9.2. Tätigkeits- und Prozessüberwachung: Sammlung und Verarbeitung von Beweisen
  - 5.9.3. Methodik des IT-Audits
  - 5.9.4. Prozess eines IT-Audits
  - 5.9.5. Identifizierung der wichtigsten Straftaten und Ordnungswidrigkeiten im Zusammenhang mit der Informationstechnologie
  - 5.9.6. Untersuchung von Internetkriminalität: eine Einführung in die forensische Analyse und ihre Beziehung zum IT-Audits
- 5.10. Planung von Betriebskontinuität und Wiederherstellung im Katastrophenfall
  - 5.10.1. Definition des Betriebskontinuitätsplans und des Konzepts der Betriebsunterbrechung
  - 5.10.2. NIST-Empfehlung zur Planung der Betriebskontinuität
  - 5.10.3. Plan zur Wiederherstellung im Katastrophenfall
  - 5.10.4. Prozess des Plan zur Wiederherstellung im Katastrophenfall

## Modul 6. Systemintegration

- 6.1. Einführung in Wirtschaftsinformationssysteme
  - 6.1.1. Die Rolle der Informationssysteme
  - 6.1.2. Was ist ein Informationssystem?
  - 6.1.3. Dimensionen von Informationssystemen
  - 6.1.4. Geschäftsprozesse und Informationssysteme
  - 6.1.5. Die IS/IT-Abteilung
- 6.2. Möglichkeiten und Bedürfnisse für Informationssysteme im Unternehmen
  - 6.2.1. Organisationen und Informationssysteme
  - 6.2.2. Merkmale von Organisationen
  - 6.2.3. Auswirkungen von Informationssystemen auf das Unternehmen
  - 6.2.4. Informationssysteme als Wettbewerbsvorteil
  - 6.2.5. Einsatz von Systemen in der Unternehmensverwaltung und im Management
- 6.3. Informationssysteme und technologische Grundlagen
  - 6.3.1. Daten, Informationen und Wissen
  - 6.3.2. Informationssysteme und Technologie
  - 6.3.3. Technologie-Komponenten
  - 6.3.4. Klassifizierung und Arten von Informationssystemen
  - 6.3.5. Service- und geschäftsprozessbasierte Architekturen
  - 6.3.6. Formen der Systemintegration
- 6.4. Integrierte Systeme zur Verwaltung von Unternehmensressourcen
  - 6.4.1. Geschäftliche Anforderungen
  - 6.4.2. Ein integriertes Informationssystem für das Unternehmen
  - 6.4.3. Akquisition vs. Vorankommen
  - 6.4.4. ERP-Implementierung
  - 6.4.5. Auswirkungen auf das Management
  - 6.4.6. Wichtigste ERP-Anbieter
- 6.5. Informationssysteme für die Verwaltung von Lieferketten und Kundenbeziehungen
  - 6.5.1. Definition der Lieferkette
  - 6.5.2. Effektives Management der Lieferkette
  - 6.5.3. Die Rolle der Informationssysteme
  - 6.5.4. Lösungen für das Lieferkettenmanagement
  - 6.5.5. Verwaltung von Kundenbeziehungen
  - 6.5.6. Die Rolle der Informationssysteme
  - 6.5.7. Einführung eines CRM-Systems
  - 6.5.8. Kritische Erfolgsfaktoren bei der CRM-Implementierung
  - 6.5.9. CRM, e-CRM und andere Trends
- 6.6. Entscheidungsfindung für IKT-Investitionen und Planung von Informationssystemen
  - 6.6.1. Kriterien für IKT-Investitionsentscheidungen
  - 6.6.2. Verknüpfung des Projekts mit dem Management- und Geschäftsplan
  - 6.6.3. Auswirkungen auf das Management
  - 6.6.4. Neugestaltung von Geschäftsprozessen
  - 6.6.5. Entscheidung des Managements über Implementierungsmethoden
  - 6.6.6. Notwendigkeit der Planung von Informationssystemen
  - 6.6.7. Zielsetzung, Teilnehmer und Zeitplan
  - 6.6.8. Aufbau und Entwicklung des Systemplans
  - 6.6.9. Überwachung und Aktualisierung
- 6.7. Sicherheitserwägungen bei der Nutzung von IKTs
  - 6.7.1. Risikoanalyse
  - 6.7.2. Sicherheit in Informationssystemen
  - 6.7.3. Thomas-Schiene
- 6.8. Durchführbarkeit von IKT-Projekten und finanzielle Aspekte von Informationssystemprojekten
  - 6.8.1. Beschreibung und Ziele
  - 6.8.2. Teilnehmer an der Machbarkeitsstudie des Systems
  - 6.8.3. Techniken und Praktiken
  - 6.8.4. Kostenstruktur
  - 6.8.5. Finanzielle Projektion
  - 6.8.6. Budgets

- 6.9. *Business Intelligence*
  - 6.9.1. Was ist Business Intelligence?
  - 6.9.2. BI-Strategie und -Implementierung
  - 6.9.3. Gegenwart und Zukunft von BI
- 6.10. ISO/IEC 12207
  - 6.10.1. Was bedeutet "ISO/IEC 12207"?
  - 6.10.2. Analyse von Informationssystemen
  - 6.10.3. Entwurf eines Informationssystems
  - 6.10.4. Implementierung und Akzeptanz des Informationssystems

## Modul 7. Wiederverwendung von Software

- 7.1. Überblick über die Wiederverwendung von Software
  - 7.1.1. Was ist die Wiederverwendung von Software?
  - 7.1.2. Vor- und Nachteile der Wiederverwendung von Software
  - 7.1.3. Wichtigste Techniken der Wiederverwendung von Software
- 7.2. Einführung in die Entwurfsmuster
  - 7.2.1. Was ist ein Entwurfsmuster?
  - 7.2.2. Katalog der wichtigsten Entwurfsmuster
  - 7.2.3. Wie man Muster zur Lösung von Designproblemen einsetzt
  - 7.2.4. Wie man das beste Entwurfsmuster auswählt
- 7.3. Erzeugungsmuster I
  - 7.3.1. Erzeugungsmuster
  - 7.3.2. Abstract Factory Muster
  - 7.3.3. Beispielimplementierung des Abstract Factory Musters
  - 7.3.4. Builder Muster
  - 7.3.5. Beispielimplementierung des Builder Muster
  - 7.3.6. Abstract Factory Muster vs. Builder
- 7.4. Erzeugungsmuster II
  - 7.4.1. Factory Method Muster
  - 7.4.2. Factory Method vs. Abstract Factory
  - 7.4.3. Singleton Muster
- 7.5. Strukturmuster I
  - 7.5.1. Strukturmuster
  - 7.5.2. Adapter Muster
  - 7.5.3. Bridge Muster
- 7.6. Strukturmuster II
  - 7.6.1. Composite Muster
  - 7.6.2. Decorator Muster
- 7.7. Strukturmuster III
  - 7.7.1. Facade Muster
  - 7.7.2. Proxy Muster
- 7.8. Verhaltensmuster I
  - 7.8.1. Konzept der Verhaltensmuster
  - 7.8.2. Verhaltensmuster: Kette der Verantwortung
  - 7.8.3. Verhaltensmuster Ordnung
- 7.9. Verhaltensmuster II
  - 7.9.1. Interpreter Muster oder *Interpreter*
  - 7.9.2. Iterator Muster
  - 7.9.3. Beobachter Muster
  - 7.9.4. Strategie Muster
- 7.10. *Frameworks*
  - 7.10.1. Konzepte des *framework*
  - 7.10.2. Entwicklung mit *frameworks*
  - 7.10.3. Patrón Model View Controller
  - 7.10.4. *Framework* für die Gestaltung grafischer Benutzeroberflächen
  - 7.10.5. *Frameworks* für die Entwicklung Web-Applikationen
  - 7.10.6. *Frameworks* für die Verwaltung der Objektpersistenz in Datenbanken

## Modul 8. Dienstleistungen der Informationstechnologie

- 8.1. Digitale Transformation I
  - 8.1.1. Business Innovation
  - 8.1.2. Management der Produktion
  - 8.1.3. Finanzielle Verwaltung
- 8.2. Digitale Transformation II
  - 8.2.1. Marketing
  - 8.2.2. Management des Personalwesens
  - 8.2.3. Ein integriertes Informationssystem
- 8.3. Fallstudie
  - 8.3.1. Präsentation des Unternehmens
  - 8.3.2. Methoden zur Analyse der IT-Beschaffung
  - 8.3.3. Bestimmung von Kosten, Nutzen und Risiken
  - 8.3.4. Wirtschaftliche Bewertung der Investition
- 8.4. IKT-Governance und -Verwaltung
  - 8.4.1. Definition der Governance von Informationstechnologien und -systemen
  - 8.4.2. Unterschied zwischen ITS-Governance und Management
  - 8.4.3. ITS-Governance und Managementrahmen
  - 8.4.4. Standards und die Leitung und Verwaltung von ITSs
- 8.5. IKT-Unternehmensführung
  - 8.5.1. Was ist gute Unternehmensführung?
  - 8.5.2. Hintergrund zur IKT-Governance
  - 8.5.3. ISO/IEC 38500:2008-Norm
  - 8.5.4. Umsetzung einer guten IKT-Governance
  - 8.5.5. IKT-Governance und bewährte Praktiken
  - 8.5.6. Unternehmensführung. Überblick und Trends
- 8.6. Kontrollziele für Informations- und verwandte Technologien (COBIT)
  - 8.6.1. Rahmen für die Umsetzung
  - 8.6.2. Bereich: Planung und Organisation
  - 8.6.3. Bereich: Akquisition und Implementierung
  - 8.6.4. Bereich: Lieferung und Unterstützung
  - 8.6.5. Bereich: Überwachung und Bewertung
  - 8.6.6. Anwendung des COBIT-Leitfadens

- 8.7. Die Informationstechnologie-Infrastruktur-Bibliothek (ITIL)
  - 8.7.1. Einführung in ITIL
  - 8.7.2. Service-Strategie
  - 8.7.3. Service Design
  - 8.7.4. Dienstübergang
  - 8.7.5. Dienstbetrieb
  - 8.7.6. Serviceverbesserung
- 8.8. Das Service Management System
  - 8.8.1. Grundlegende Prinzipien von UNE-ISO/IEC 20000-1
  - 8.8.2. Die Struktur der Normenreihe ISO/IEC 20000
  - 8.8.3. Anforderungen an das Service Management System (SMS)
  - 8.8.4. Gestaltung und Umstellung neuer oder geänderter Dienste
  - 8.8.5. Prozesse der Dienstleistungserbringung
  - 8.8.6. Prozessgruppen
- 8.9. Das Software Asset Management System
  - 8.9.1. Rechtfertigung des Bedarfs
  - 8.9.2. Hintergrund
  - 8.9.3. Präsentation der Norm 19770
  - 8.9.4. Umsetzung der Verwaltung
- 8.10. Management der Geschäftskontinuität
  - 8.10.1. Plan zur Aufrechterhaltung des Geschäftsbetriebs
  - 8.10.2. Implementierung eines BCM

## Modul 9. Sicherheit von Informationssystemen

- 9.1. Ein Überblick über Sicherheit, Kryptographie und klassische Kryptoanalyse
  - 9.1.1. Computersicherheit: Historische Perspektive
  - 9.1.2. Aber was genau ist mit Sicherheit gemeint?
  - 9.1.3. Geschichte der Kryptographie
  - 9.1.4. Substitutions-Chiffren
  - 9.1.5. Fallstudie: Die Enigma-Maschine

- 9.2. Symmetrische Kryptographie
  - 9.2.1. Einführung und grundlegende Terminologie
  - 9.2.2. Symmetrische Verschlüsselung
  - 9.2.3. Betriebsarten
  - 9.2.4. DES
  - 9.2.5. Der neue AES-Standard
  - 9.2.6. Stream-Verschlüsselung
  - 9.2.7. Kryptoanalyse
- 9.3. Asymmetrische Kryptographie
  - 9.3.1. Die Ursprünge der Public Key Kryptographie
  - 9.3.2. Grundlegende Konzepte und Bedienung
  - 9.3.3. Der RSA-Algorithmus
  - 9.3.4. Digitale Zertifikate
  - 9.3.5. Speicherung und Verwaltung von Schlüsseln
- 9.4. Netzwerk-Angriffe
  - 9.4.1. Bedrohungen und Angriffe aus dem Netzwerk
  - 9.4.2. Aufzählung
  - 9.4.3. Verkehrsüberwachung: *sniffers*
  - 9.4.4. Denial-of-Service-Angriffe
  - 9.4.5. ARP-Poisoning-Angriffe
- 9.5. Sicherheitsarchitekturen
  - 9.5.1. Traditionelle Sicherheitsarchitekturen
  - 9.5.2. Secure Socket Layer: SSL
  - 9.5.3. SSH-Protokoll
  - 9.5.4. Virtuelle private Netzwerke (VPNs)
  - 9.5.5. Schutzmechanismen für externe Speicherlaufwerke
  - 9.5.6. Hardware-Schutzmechanismen
- 9.6. Systemschutztechniken und Entwicklung von sicherem Code
  - 9.6.1. Sicherheit im Betrieb
  - 9.6.2. Ressourcen und Kontrollen
  - 9.6.3. Überwachung
  - 9.6.4. Intrusion Detection Systeme
  - 9.6.5. Host IDS
  - 9.6.6. Netzwerk IDS
  - 9.6.7. Signatur-basiertes IDS
  - 9.6.8. Decoy Systeme
  - 9.6.9. Grundlegende Sicherheitsprinzipien bei der Code-Entwicklung
  - 9.6.10. Störungsmanagement
  - 9.6.11. Staatsfeind Nummer 1: Der Buffer Overflow
  - 9.6.12. Kryptographische Botschaften
- 9.7. *Botnets* und Spam
  - 9.7.1. Ursprung des Problems
  - 9.7.2. Prozess von Spam
  - 9.7.3. Spam verschicken
  - 9.7.4. Verfeinerung der Verteilerlisten
  - 9.7.5. Methoden zum Schutz
  - 9.7.6. Von Dritten angebotener Anti-Spam-Service
  - 9.7.7. Fallstudien
  - 9.7.8. Exotischer Spam
- 9.8. Web Auditing und Angriffe
  - 9.8.1. Sammeln von Informationen
  - 9.8.2. Angriffs-Techniken
  - 9.8.3. Instrumente
- 9.9. Malware und bössartiger Code
  - 9.9.1. Was ist Malware?
  - 9.9.2. Arten von Malware
  - 9.9.3. Virus
  - 9.9.4. Kryptoviren
  - 9.9.5. Würmer
  - 9.9.6. Adware
  - 9.9.7. Spyware
  - 9.9.8. Hoaxes
  - 9.9.9. Pishing
  - 9.9.10. Trojaner
  - 9.9.11. Die Malware-Wirtschaft
  - 9.9.12. Mögliche Lösungen

- 9.10. Forensische Analyse
  - 9.10.1. Sammeln von Beweisen
  - 9.10.2. Analyse der Beweise
  - 9.10.3. Anti-Forensische Techniken
  - 9.10.4. Praktische Fallstudie

## Modul 10. Projektmanagement

- 10.1. Grundlegende Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
  - 10.1.1. Was ist ein Projekt?
  - 10.1.2. Einheitliche Methodik
  - 10.1.3. Was ist Projektmanagement?
  - 10.1.4. Was ist ein Projektplan?
  - 10.1.5. Vorteile
  - 10.1.6. Projektlebenszyklen
  - 10.1.7. Prozessgruppen oder Lebenszyklus des Projektmanagements
  - 10.1.8. Die Beziehung zwischen Prozessgruppen und Wissensgebieten
  - 10.1.9. Beziehung zwischen Produkt- und Projektlebenszyklus
- 10.2. Start und Planung
  - 10.2.1. Von der Idee zum Projekt
  - 10.2.2. Entwicklung der Projektcharta
  - 10.2.3. Projekt-Kick-off-Meeting
  - 10.2.4. Aufgaben, Kenntnisse und Fähigkeiten im Gründungsprozess
  - 10.2.5. Der Projektplan
  - 10.2.6. Entwicklung des Basisplans. Schritte
  - 10.2.7. Aufgaben, Kenntnisse und Fähigkeiten im Planungsprozess
- 10.3. Management von *Stakeholdern* und Reichweite
  - 10.3.1. Identifizierung von Interessengruppen
  - 10.3.2. Entwicklung eines Plans für das Management von Interessengruppen
  - 10.3.3. Management des Engagements von Interessengruppen
  - 10.3.4. Überwachung des Engagements der Interessengruppen
  - 10.3.5. Das Ziel des Projekts
  - 10.3.6. Management der Einsätze und ihr Plan
  - 10.3.7. Erfassen von Anforderungen
  - 10.3.8. Definition des Anwendungsbereichs
  - 10.3.9. Erstellung des Projektstrukturplans (PSP)
  - 10.3.10. Prüfen und Kontrolle des Umfangs
- 10.4. Entwicklung des Zeitplans
  - 10.4.1. Zeitmanagement und dessen Plan
  - 10.4.2. Definieren der Aktivitäten
  - 10.4.3. Festlegung der Abfolge der Aktivitäten
  - 10.4.4. Geschätzte Ressourcen für die Aktivitäten
  - 10.4.5. Geschätzte Dauer der Aktivitäten
  - 10.4.6. Entwicklung des Zeitplans und Berechnung des kritischen Pfads
  - 10.4.7. Zeitplan-Kontrolle
- 10.5. Budgetentwicklung und Risikobewältigung
  - 10.5.1. Schätzung der Kosten
  - 10.5.2. Entwicklung des Budgets und der S-Kurve
  - 10.5.3. Kostenkontrolle und Earned-Value-Methode
  - 10.5.4. Die Konzepte des Risikos
  - 10.5.5. Wie man eine Risikoanalyse durchführt
  - 10.5.6. Die Entwicklung des Reaktionsplans
- 10.6. Qualitätsmanagement
  - 10.6.1. Planung der Qualität
  - 10.6.2. Qualitätssicherung
  - 10.6.3. Qualitätskontrolle
  - 10.6.4. Grundlegende statistische Konzepte
  - 10.6.5. Werkzeuge des Qualitätsmanagements



- 10.7. Kommunikation und Humanressourcen
  - 10.7.1. Planung des Kommunikationsmanagement
  - 10.7.2. Analyse der Kommunikationsanforderungen
  - 10.7.3. Technologie der Kommunikation
  - 10.7.4. Kommunikationsmodelle
  - 10.7.5. Kommunikationsmethoden
  - 10.7.6. Plan für das Kommunikationsmanagement
  - 10.7.7. Verwaltung der Kommunikation
  - 10.7.8. Verwaltung des Personalwesens
  - 10.7.9. Hauptakteure und ihre Rolle in den Projekten
  - 10.7.10. Organisationsformen
  - 10.7.11. Projektorganisation
  - 10.7.12. Das Projektteam
- 10.8. Die Beschaffung
  - 10.8.1. Der Beschaffungsprozess
  - 10.8.2. Planung
  - 10.8.3. Suche nach Lieferanten und Ausschreibungen
  - 10.8.4. Vergabe des Auftrags
  - 10.8.5. Vertragsverwaltung
  - 10.8.6. Verträge
  - 10.8.7. Arten von Verträgen
  - 10.8.8. Vertragsverhandlungen
- 10.9. Durchführung, Überwachung und Kontrolle sowie Abschluss
  - 10.9.1. Prozessgruppen
  - 10.9.2. Die Projektdurchführung
  - 10.9.3. Die Projektüberwachung und -kontrolle
  - 10.9.4. Abschluss des Projekts
- 10.10. Berufliche Verantwortung
  - 10.10.1. Berufliche Verantwortung
  - 10.10.2. Merkmale der sozialen und beruflichen Verantwortung
  - 10.10.4. Verantwortung vs. PMP®
  - 10.10.5. Beispiele für Verantwortung
  - 10.10.6. Vorteile der Professionalisierung

# 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.



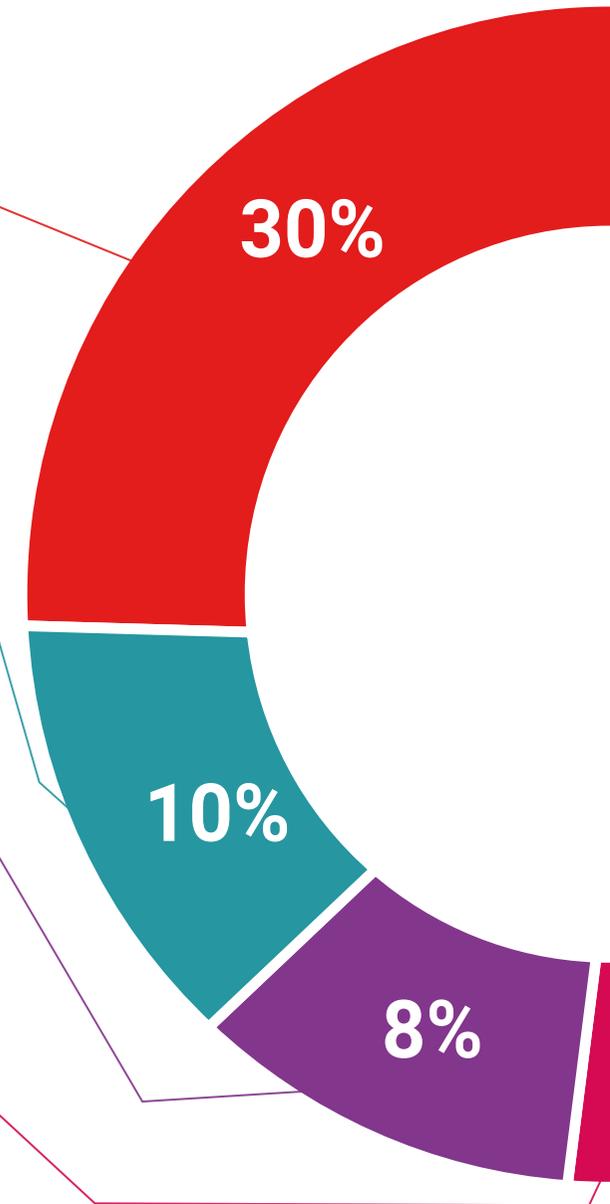
#### Fertigkeiten und Kompetenzen Praktiken

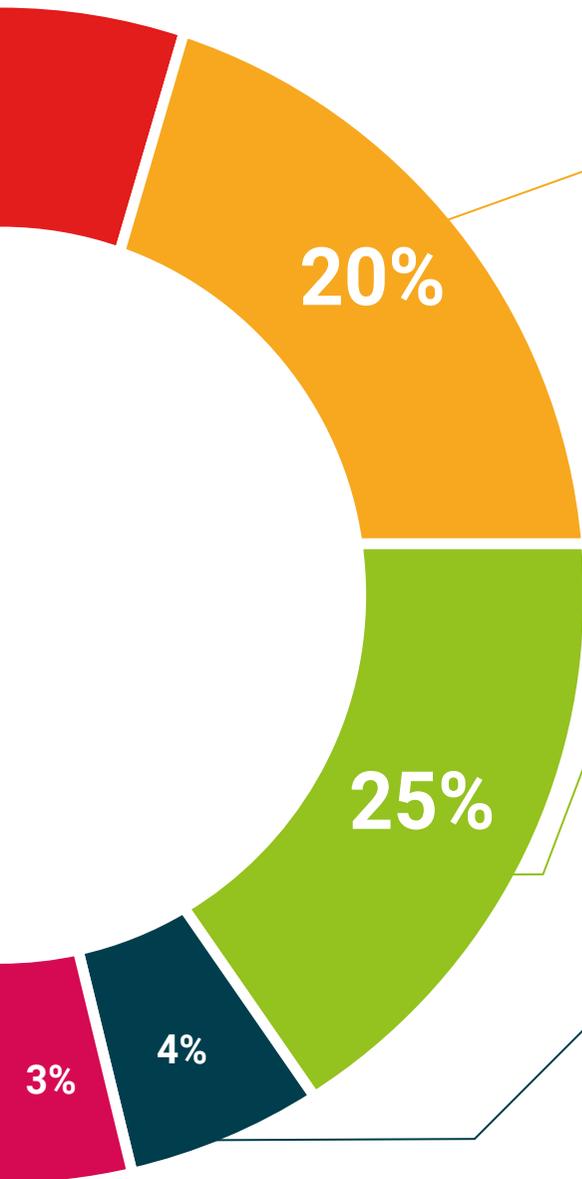
Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Praktiken und Dynamiken zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



#### 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 Fortgeschrittene Softwaretechnik 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 Fortgeschrittene Softwaretechnik** 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 Fortgeschrittene Softwaretechnik**

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 tutoeren  
garantie akkreditierung unterricht  
institutionen technologie lernen  
gemeinschaft verpflichtung  
persönliche betreuung innovation  
wissen gegenwart qualität  
online-Ausbildung  
entwicklung instituten  
virtuelles Klassenzimmer

**tech** technologische  
universität

Privater Masterstudiengang  
Fortgeschrittene Softwaretechnik

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

