

# Weiterbildender Masterstudiengang Softwaretechnik



## Weiterbildender Masterstudiengang Softwaretechnik

- » Modalität: online
- » Dauer: 2 Jahre
- » Qualifizierung: TECH Technische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

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

# Index

01

Präsentation

---

Seite 4

02

Ziele

---

Seite 8

03

Kompetenzen

---

Seite 14

04

Kursleitung

---

Seite 18

05

Struktur und Inhalt

---

Seite 22

06

Methodik

---

Seite 44

07

Qualifizierung

---

Seite 52

# 01

# Präsentation

Die Nachfrage nach Software ist in den letzten Jahren sprunghaft angestiegen. Mit dem Aufkommen neuer digitaler Plattformen, immer ausgefeilterer Hardware und der zunehmenden Virtualisierung alltäglicher Prozesse stehen *Software*-Ingenieure vor immer neuen Herausforderungen. Die Öffentlichkeit gewöhnt sich immer mehr an die neuen Technologien, und ihre Ansprüche werden immer höher. Daher müssen sich die Fachleute der *Softwareentwicklung* an diese Anforderungen anpassen und Produkte schaffen, die den Erwartungen des Marktes entsprechen. Dies erfordert ein hohes Maß an technischem Fachwissen in verschiedenen Bereichen der IT.



“

*Sie werden eine Schlüsselrolle für die technologische Zukunft vieler Unternehmen spielen. Spezialisieren Sie sich auf die Softwaretechnik und beginnen Sie mit der Entwicklung von Systemen, die einen Unterschied machen"*

Die Technologiebranche ist heute einer der wichtigsten Wirtschaftszweige, da fast jeder täglich mit irgendeinem digitalen Gerät interagiert. In diesem Zusammenhang stehen Software-Ingenieure an vorderster Front des gesamten technologischen Entwicklungsprozesses, da sie es sind, die ständig Systeme aktualisieren, neue entwickeln und intelligente Lösungen für auftretende Probleme anbieten müssen. So gesehen müssen Fachleute für Computertechnik sehr entscheidungsfreudige Persönlichkeiten sein, die über ein ausgeprägtes technisches Know-how und eine hervorragende Anpassungsfähigkeit an alle Arten von Entwicklungen und Umgebungen verfügen.

Mit diesem Ziel vor Augen hat TECH den Weiterbildenden Masterstudiengang in Softwaretechnik entwickelt, der allen Entwicklern, die ihre Karriere spezialisieren und auf die Entwicklung von Systemen ausrichten wollen, eine umfassende und hochqualifizierte Fortbildung bietet. Das Programm befasst sich zum einen mit den verschiedenen Methoden zur Erstellung und Verwaltung eines Softwareentwicklungsprojekts sowie mit allen Aspekten, die bei der Berechnung, den Anforderungen und den Plattformen zu berücksichtigen sind. Andererseits wird auch die Sicherheit sowohl der Software selbst als auch der Informationssysteme und der Arbeitsumgebung, die während des Prozesses genutzt werden, hervorgehoben. Nach Abschluss des Studiengangs wird der Student über alle notwendigen Kenntnisse verfügen, um ein effektiver und hochkompetenter Experte für die Softwaretechnik zu sein.

Einer der Hauptvorteile dieses Programms ist außerdem, dass es zu 100% online ist. Das bedeutet, dass der Student sich nicht an feste Stundenpläne anpassen muss und nicht verpflichtet ist, ein bestimmtes physisches Zentrum zu besuchen. Auf diese Weise hat der Student die Freiheit, das Studium des von ihm gewählten Faches in seinem eigenen Tempo und unter Berücksichtigung seiner Verpflichtungen zu gestalten und den Stundenplan so zu planen, wie es ihm am besten passt.

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

- ♦ Die Entwicklung von Fallstudien, die von Experten für Softwareentwicklung vorgestellt werden
- ♦ Der anschauliche, schematische und äußerst praxisnahe Inhalt soll wissenschaftliche und praktische Informationen zu den für die berufliche Praxis wesentlichen Disziplinen vermitteln
- ♦ Praktische Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens genutzt werden kann
- ♦ Sein besonderer Schwerpunkt liegt auf innovativen Methoden im Bereich der Softwaretechnik
- ♦ Theoretische Vorträge, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- ♦ Die Verfügbarkeit des Zugriffs auf die Inhalte von jedem, festen oder tragbaren Gerät, mit Internetanschluss



*Können Sie sich vorstellen, an der Entwicklung von Netflix beteiligt gewesen zu sein? Es ist an der Zeit, dass Sie aufhören, sich etwas vorzustellen, und Ihre Karriere auf die besten Softwareprojekte ausrichten“*

“ *Ihre Erfahrung und Ihr Fachwissen können bei großen Projekten mit vielen Anforderungen den Unterschied ausmachen. Verpassen Sie nicht die Gelegenheit, sich in Ihrer Karriere zu profilieren und schreiben Sie sich jetzt für diesen Weiterbildenden Masterstudiengang in Softwaretechnik ein*”

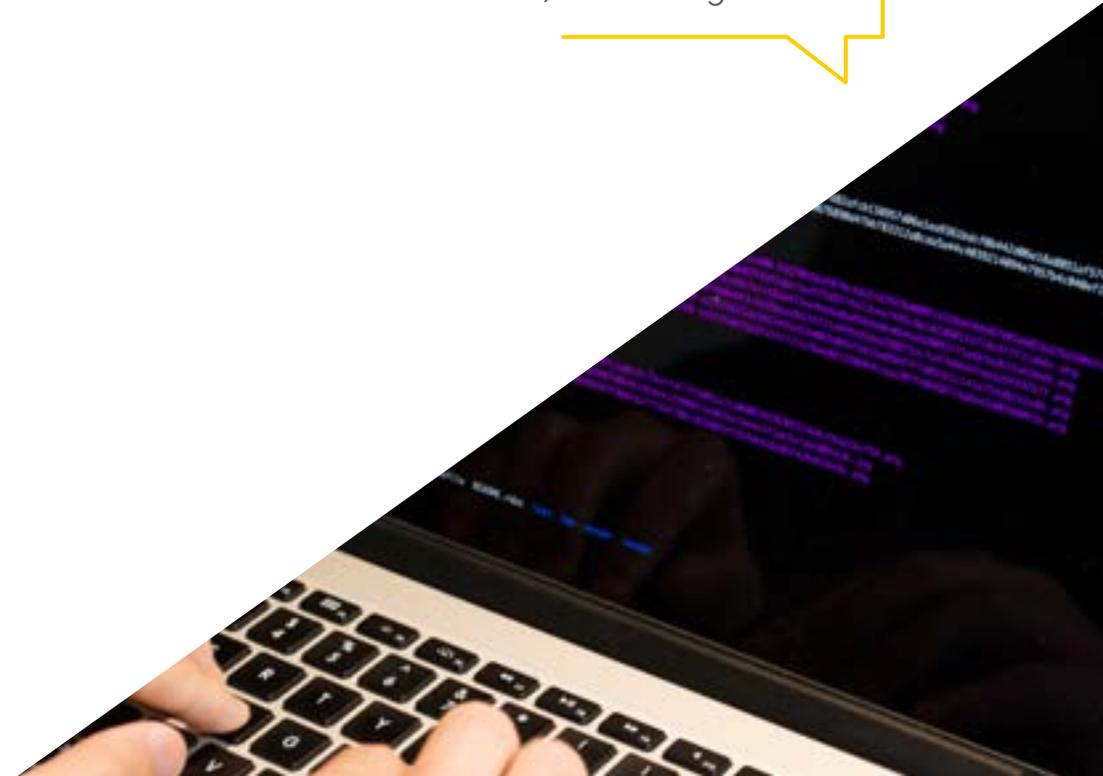
Das Dozententeam besteht aus Fachleuten aus dem Bereich der Softwaretechnik, die ihre Erfahrungen in dieses Programm einbringen, sowie aus anerkannten Fachleuten aus führenden Unternehmen und renommierten Universitäten.

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 Studium ermöglicht, das auf die Fortbildung in realen Situationen ausgerichtet ist.

Das Konzept dieses Programms basiert auf problemorientiertem Lernen, bei dem der Student versuchen muss, die verschiedenen Situationen der beruflichen Praxis zu lösen, die während des Programms auftreten. Dabei wird der Student durch ein innovatives interaktives Videosystem unterstützt, das von anerkannten Experten entwickelt wurde.

*Das Ziel von TECH ist es, Sie zu einem guten Computertechniker zu machen. Sie werden garantierten Zugang zum bestmöglichen Material und Unterricht haben.*

*Lernen Sie, wann, wo und wie Sie wollen. Das Programm ist zu 100% online und passt sich Ihren Bedürfnissen an, nicht umgekehrt.*



# 02 Ziele

Dieser Weiterbildende Masterstudiengang in Softwaretechnik wurde mit dem Ziel entwickelt, allen Fachleuten im IT-Bereich eine höhere Fortbildung zu bieten, die es ihnen ermöglicht, ihre Karriere auf die Entwicklung moderner Software zu konzentrieren, die an die neuen, sich verändernden Realitäten des Marktes angepasst ist. Mit dem hochtechnischen Wissen, das während des Studiengangs vermittelt wird, erhöhen sich die Chancen auf einen beruflichen Aufstieg und den Zugang zu Stellen in großen Unternehmen des Sektors erheblich.





“

*Ein weiterbildender Masterstudiengang, der Ihrer Karriere den größten positiven Schub in Richtung beruflicher Erfolg geben kann”*



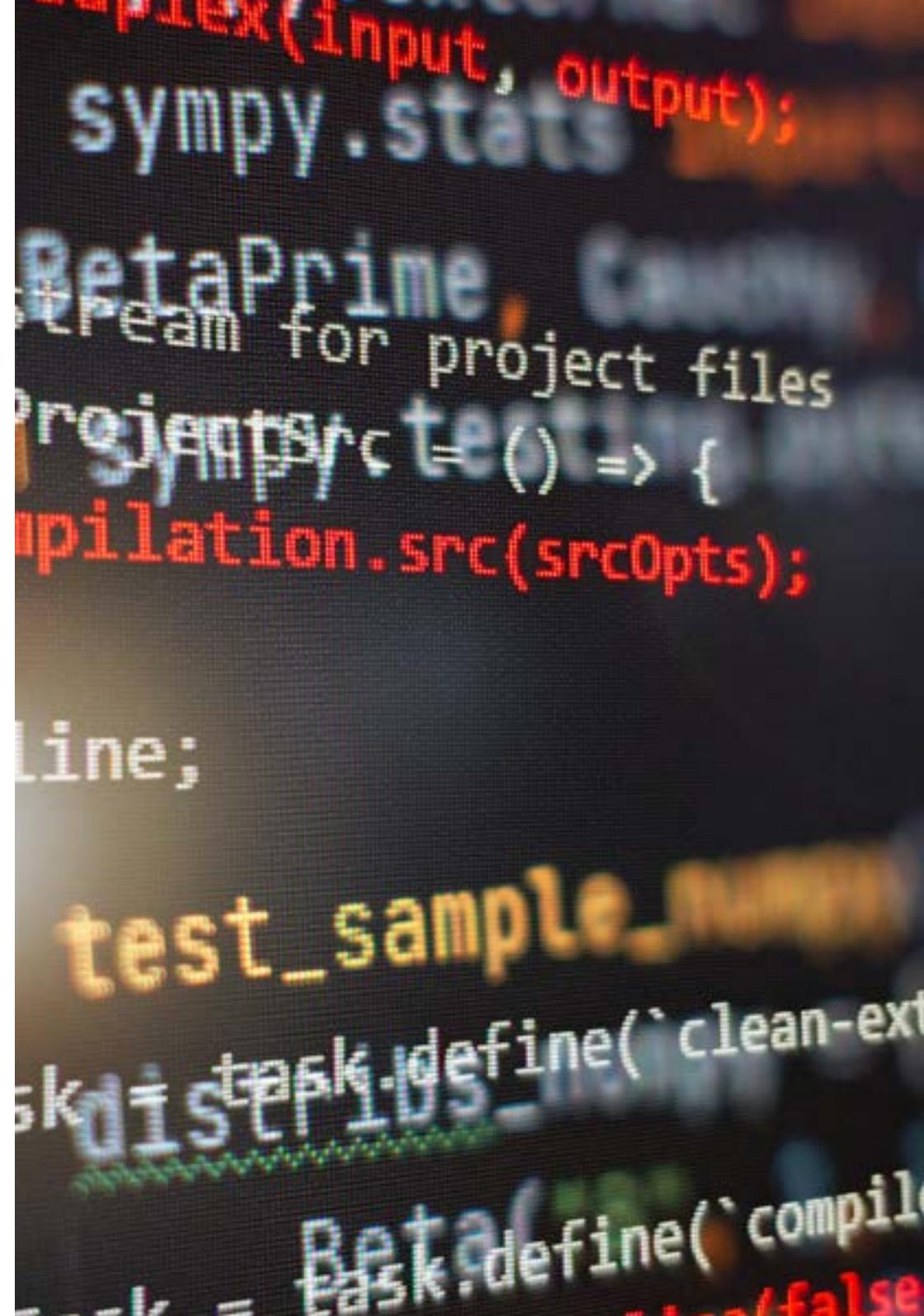
## Allgemeine Ziele

---

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



*Eine Spezialisierung, die Ihnen helfen wird, die Softwareentwicklung mit einem einzigartigen Satz von Fähigkeiten zu beherrschen, die von jedem führenden Unternehmen des Sektors verlangt werden"*





## Spezifische Ziele

---

- ◆ Kenntnis der Grundlagen der Softwaretechnik sowie des Regelwerks oder der Grundsätze der Ethik und der beruflichen Verantwortung während und nach der Entwicklung
- ◆ Verständnis des Softwareentwicklungsprozesses unter Berücksichtigung der verschiedenen Programmiermodelle und des objektorientierten Programmierparadigmas
- ◆ Verständnis der verschiedenen Arten der Anwendungsmodellierung und der Entwurfsmuster in der Unified Modelling Language (UML)
- ◆ Kenntnisse der grundlegenden Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
- ◆ Verständnis der Funktionsweise des Qualitätsmanagements in Projekten, einschließlich Planung, Sicherung, Kontrolle, statistischer Konzepte und verfügbarer Werkzeuge
- ◆ Aneignung der wesentlichen Kenntnisse im Zusammenhang mit der beruflichen Verantwortung im Projektmanagement
- ◆ Verständnis für die verschiedenen Softwareentwicklungsplattformen
- ◆ Erwerb der notwendigen Kenntnisse für die Entwicklung von Anwendungen und grafischen Oberflächen in den Sprachen *Java* und *.NET*
- ◆ Erlernen von Entwicklungsumgebungen für mobile *Android*-Anwendungen, Debugging- und Veröffentlichungsprozessen
- ◆ Verständnis der Entwicklung von Cloud-basierten Anwendungen und Ermittlung der richtigen Verfahren für ihre Implementierung
- ◆ Verständnis der Verfahren und Techniken zur Verbesserung des Erscheinungsbildes eines in HTML geschriebenen Dokuments
- ◆ Erwerb der notwendigen Kenntnisse für die Entwicklung von clientseitigen Webanwendungen
- ◆ Entwicklung von Anwendungen mit komplexen Strukturen unter Verwendung der verschiedenen Verfahren, Funktionen und Objekte, aus denen *JavaScript* besteht
- ◆ Erlernen der Verwendung der DOM-Programmierschnittstelle für HTML- und XML-Dokumente, um deren Struktur, Stil und Inhalt zu ändern
- ◆ Kennenlernen des Konzepts der Web-Usability, seiner Vorteile, Prinzipien, Methoden und Techniken, um eine Website für den Benutzer nutzbar zu machen
- ◆ Verständnis der Software-Architektur Model-View-Controller-View (MVC), 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
- ◆ Verständnis des Prozesses der Informationssicherheit, seiner Auswirkungen auf Vertraulichkeit, Integrität, Verfügbarkeit und wirtschaftliche Kosten
- ◆ Erlernen der Anwendung bewährter Sicherheitspraktiken bei der Verwaltung von Informationstechnologiediensten
- ◆ Erwerb von Kenntnissen für die korrekte Zertifizierung von Sicherheitsprozessen
- ◆ Verständnis der Authentifizierungsmechanismen und -methoden für die Zugangskontrolle sowie des Prozesses der Zugangsprüfung
- ◆ Verständnis von Sicherheitsmanagementprogrammen, Risikomanagement und Gestaltung von Sicherheitsrichtlinien
- ◆ Erlernen von Plänen zur Geschäftskontinuität, 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

- ◆ Verständnis von Software-Sicherheitsproblemen, Schwachstellen und deren Klassifizierung
- ◆ Analyse der verschiedenen Webserver, die auf dem heutigen Markt im Trend sind
- ◆ Verständnis des Prozesses der Nutzungsstatistik und des Lastausgleichs auf Webservern
- ◆ Aneignung der Kenntnisse, die für die korrekte Durchführung des internen IT-Prüfungs- und Kontrollprozesses erforderlich sind
- ◆ Verständnis der Konzepte und Prozesse des Softwaredesigns, Lernen der Designarchitektur und des komponentenbasierten und musterbasierten Designs
- ◆ Verständnis verschiedener Systemarchitekturen und Softwareentwurfsmuster sowie der Architektur von Cloud-Anwendungen
- ◆ Vertiefung der Verbesserung des Softwareentwicklungsprozesses und der Softwarequalität anhand von ISO/IEC-Normen
- ◆ Verständnis für die Bedeutung der Anforderungsanalyse im Softwareentwicklungsprozess
- ◆ Vertiefung des Verständnisses von Anforderungsquellen und Techniken zur Anforderungserhebung, da diese ein wesentlicher Bestandteil des Prozesses sind
- ◆ Verständnis und Anwendung von Prototypen als wesentlicher Bestandteil des Entwicklungsprozesses
- ◆ Grundlagen für die forensische Analyse in der Welt der Software- und IT-Prüfung legen
- ◆ Kenntnisse der grundlegenden Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
- ◆ 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 Werkzeuge





“

*Eine komplette Spezialisierung, die Ihnen das Wissen vermittelt, das Sie brauchen, um sich mit den Besten zu messen”*

# 03 Kompetenzen

Software-Ingenieure müssen ihr Wissen ständig aktualisieren, da die Werkzeuge und die Umgebung, in der sie arbeiten, täglich modernisiert werden. Dies erfordert einen regelmäßigen Lernprozess, für den eine Reihe allgemeiner Kompetenzen erforderlich sind, sowohl in der reinen Softwareentwicklung als auch in anderen Disziplinen wie dem Teammanagement. Aus diesem Grund hat TECH den Weiterbildenden Masterstudiengang in Softwaretechnik mit dem Ziel entwickelt, den Studenten alle möglichen und notwendigen Fähigkeiten zu vermitteln, um ihren eigenen kontinuierlichen Lernprozess zu vereinfachen und zu automatisieren.



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

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

```
ror_ob.select= 1
```

```
ifier_ob.select=1
```

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

```
nt("Selected" + str(modifier
```

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

```
e = bpy.context.selected
```

```
y.data.objects[mod
```

```
print(modifier
```

```
print(modifier
```

```
print(modifier
```

```
print(modifier
```

“

*Mit all den Fähigkeiten, die Sie durch das Studium des Weiterbildenden Masterstudiengangs in Softwaretechnik erwerben werden, werden Sie der beste Kandidat für jede Stelle sein, auf die Sie sich bewerben”*



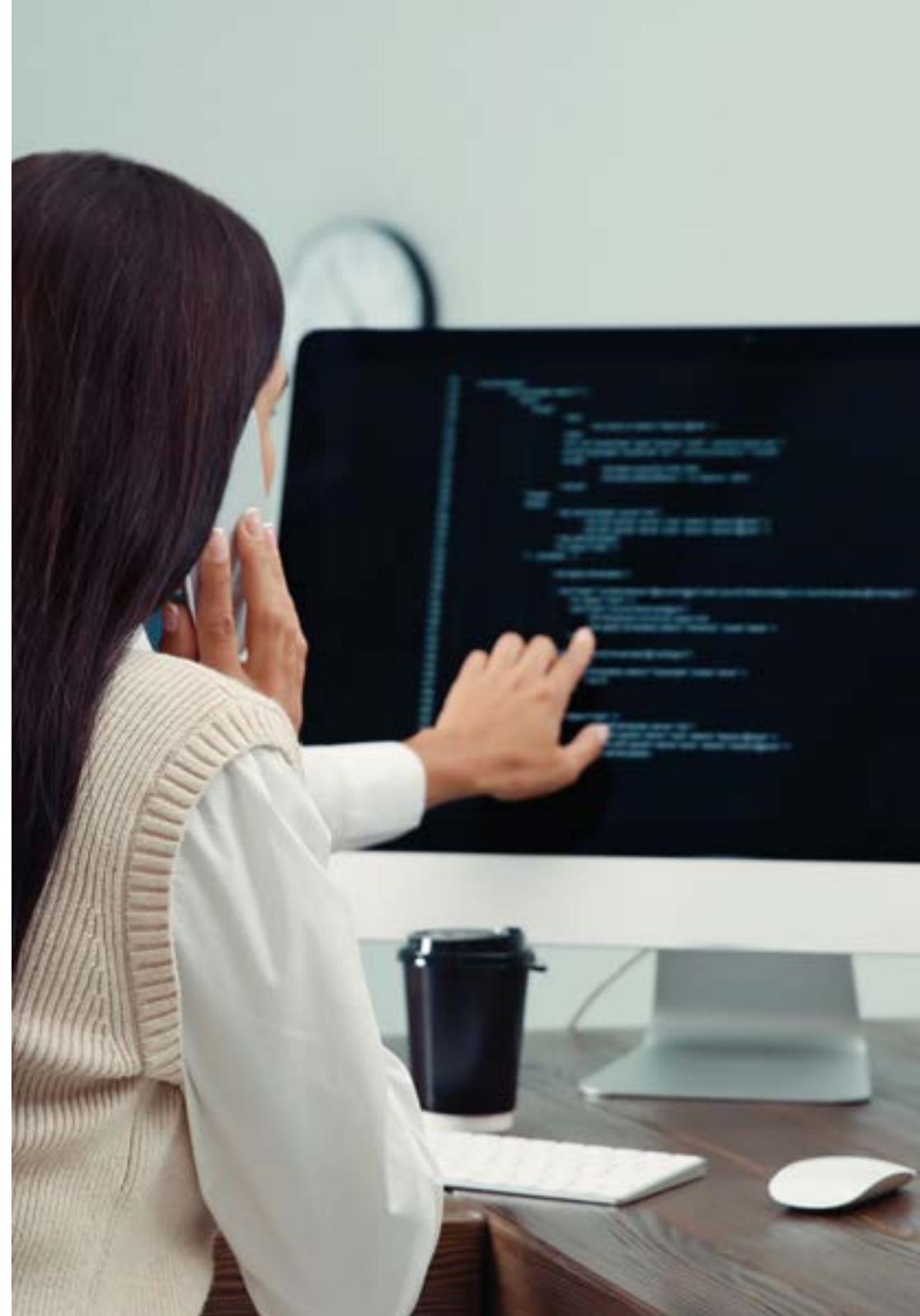
## Allgemeine Kompetenzen

---

- ◆ Entwicklung eines Softwaresystems unter Berücksichtigung aller Entwicklungsstufen, Sicherheitsplattformen und Sicherheitsfragen
- ◆ Korrekter und professioneller Umgang mit allen während der Entwicklung erzeugten Daten
- ◆ Anwendung der besten Arbeitsmethodik, die dem Projekt oder den am Projekt beteiligten Personen angemessen ist
- ◆ Kenntnis der gesamten Realität der Softwaretechnik und schnelle und effiziente Vorbeugung möglicher Risiken oder Probleme

“

*Sie können den Schritt in eine bessere berufliche Zukunft machen. Tun Sie es und schreiben Sie sich jetzt für diesen weiterbildenden Masterstudiengang ein, der Ihnen viele Türen für Ihre berufliche Laufbahn öffnen wird”*





## Spezifische Kompetenzen

---

- ◆ Verständnis der verschiedenen Arten der Anwendungsmodellierung und der Entwurfsmuster in der Unified Modelling Language (UML)
- ◆ Verstehen der Funktionsweise des Qualitätsmanagements in Projekten, einschließlich Planung, Sicherung, Kontrolle, statistischer Konzepte und verfügbarer Werkzeuge
- ◆ Erwerb der notwendigen Kenntnisse für die Entwicklung von Anwendungen und grafischen Oberflächen in den Sprachen *Java* und *.NET*
- ◆ Verständnis der Verfahren und Techniken zur Verbesserung des Erscheinungsbildes eines in HTML geschriebenen Dokuments
- ◆ Beherrschung des Prozesses der Kundeninteraktion durch den Einsatz von Formularen, *Cookies* und Sitzungsmanagement
- ◆ Verständnis der Authentifizierungsmechanismen und -methoden für die Zugangskontrolle sowie des Prozesses der Zugangsprüfung
- ◆ Verständnis für die Anwendung von Sicherheit in den verschiedenen Phasen des Software-Lebenszyklus
- ◆ Kenntnis des Konzepts, der Funktionsweise, der Architektur, der Ressourcen und des Inhalts eines Webservers
- ◆ Verständnis der verschiedenen Hilfsmittel, Methoden und anschließenden Analysen bei der Überprüfung der Sicherheit von Internet und mobilen Geräten
- ◆ Verständnis der Sicherheitsrichtlinien und -standards, die auf Online-Anwendungen anzuwenden sind
- ◆ In der Lage sein, Projekte im Bereich der technischen Informatik zu entwerfen, zu planen, zu entwickeln und zu unterzeichnen, die auf die Entwicklung oder den Betrieb von Informatiksystemen, -diensten und -anwendungen abzielen
- ◆ Leitung der Aktivitäten von IT-Projekten
- ◆ In der Lage sein, Hardware- und Softwareplattformen für die Entwicklung und Implementierung von IT-Systemen, -Dienstleistungen und -Anwendungen zu definieren, zu bewerten und auszuwählen
- ◆ Wissen, wie man mit *Scrum*, extremer Programmierung und wiederverwendungs-basierten Softwareentwicklungstechniken entwickelt
- ◆ Fähigkeit zur Konzeption, Entwicklung und Wartung von Computersystemen, -diensten und -anwendungen unter Verwendung von Software-Engineering-Methoden als Instrument der Qualitätssicherung
- ◆ Anwendung der Grundlagen der symmetrischen und asymmetrischen Kryptographie sowie ihrer wichtigsten Algorithmen
- ◆ Anwendung der wesentlichen Konzepte im Zusammenhang mit Informationssystemen im Unternehmen und Ermittlung der Möglichkeiten und des Bedarfs an Informationssystemen
- ◆ Wissen, wie ein Zeitplan für das Zeit-, Budget- und Risikoreaktionsmanagement zu erstellen ist
- ◆ Verständnis der Funktionsweise von IKT-Governance und -Management, der ISO/IEC-Normen, die dies regeln, und der bewährten Verfahren, die anzuwenden sind
- ◆ Planung des Sicherheitsmanagements und Verwaltung der wichtigsten Mechanismen für den Schutz von Informationswerten

# 04

# Kursleitung

Dieses akademische Programm verfügt über den spezialisiertesten Lehrkörper auf dem aktuellen Bildungsmarkt. Es handelt sich um Spezialisten, die von TECH ausgewählt wurden, um den gesamten Studiengang zu entwickeln. Auf diese Weise haben sie auf der Grundlage ihrer eigenen Erfahrung und der neuesten Erkenntnisse die aktuellsten Inhalte entworfen, die eine Qualitätsgarantie für ein so relevantes Thema bieten.



“

*TECH bietet Ihnen den spezialisiertesten  
Lehrkörper in diesem Fachgebiet. Schreiben  
Sie sich jetzt ein und genießen Sie die  
Qualität, die Sie verdienen”*

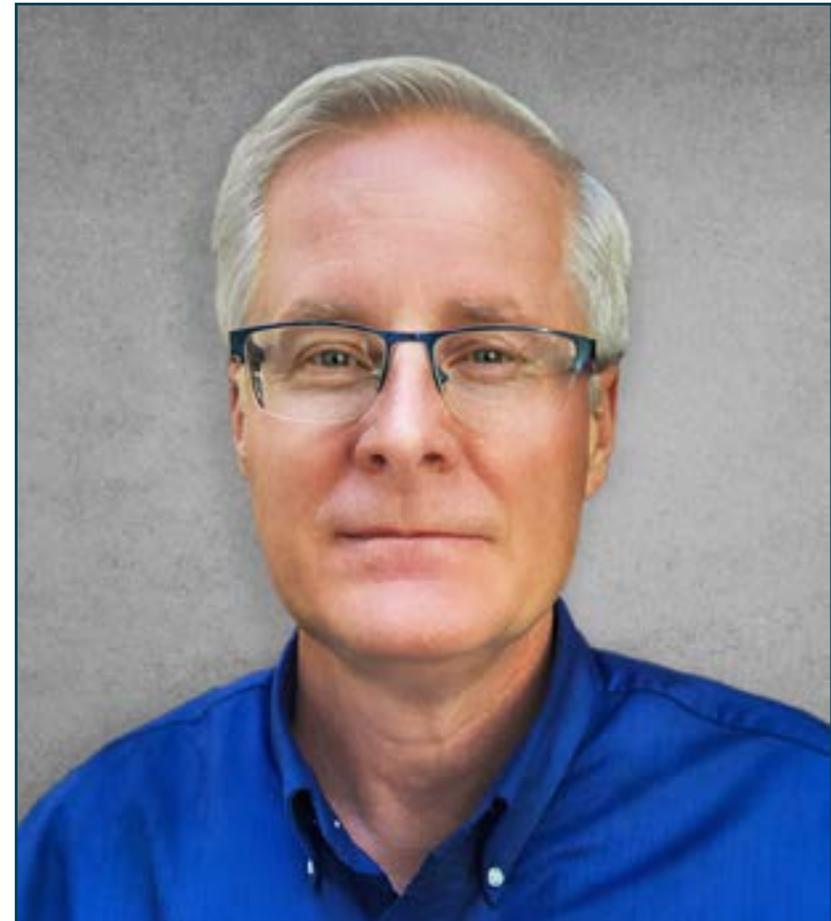
## Internationaler Gastdirektor

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

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

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

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



## Hr. Pulsipher, Darren

---

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



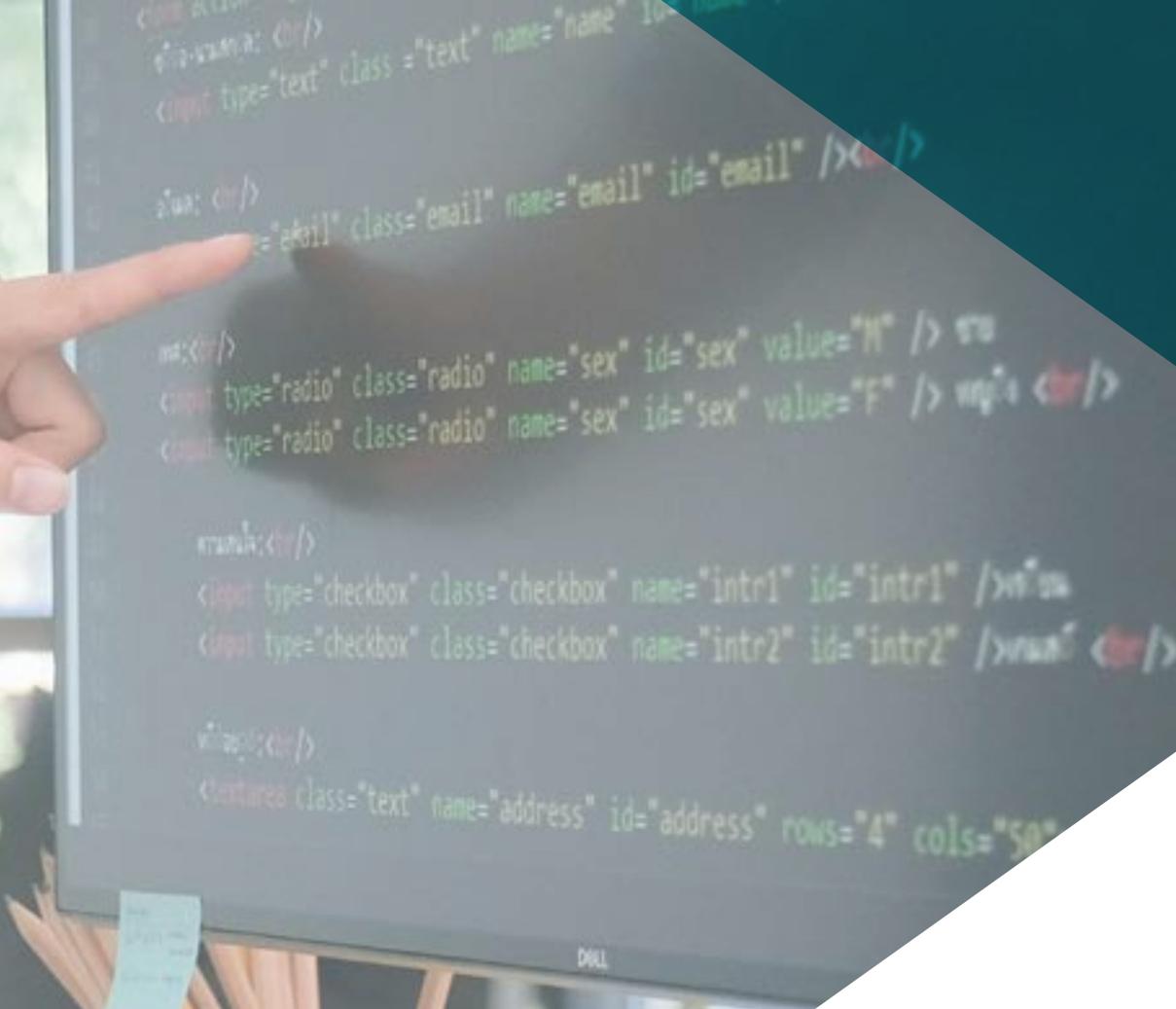
*Dank TECH werden Sie mit den besten Fachleuten der Welt lernen können*

# 05

## Struktur und Inhalt

Das didaktische Material dieses weiterbildenden Masterstudiengangs in Softwaretechnik ist so konzipiert, dass es den gesamten notwendigen und ergänzenden Unterricht zu diesem Thema abdeckt, mit den aktuellsten Methoden, Werkzeugen und Kenntnissen, die auf dem Markt verfügbar sind. Ein umfangreicher und erschöpfender Lehrplan, in dem die Verwendung von Programmiersprachen und fortgeschrittenen Umgebungen, die Verwaltung von Webservern und ihre Einbindung in den Entwicklungsprozess oder die eigentliche Verwaltung eines Projekts gelehrt werden. Dadurch wird sichergestellt, dass die Studenten die bestmöglichen Chancen haben, sich auf die Softwaretechnik zu spezialisieren und sich schnell in diesem Bereich auszuzeichnen.





“

*Zeigen Sie, dass Ihr Wissen mit Ihrer Einstellung, die beste Fachkraft zu sein, übereinstimmt und werten Sie Ihren Lebenslauf mit diesem weiterbildenden Masterstudiengang in Softwaretechnik auf“*

## Modul 1. Methodik, Entwicklung und Qualität in der Softwaretechnik

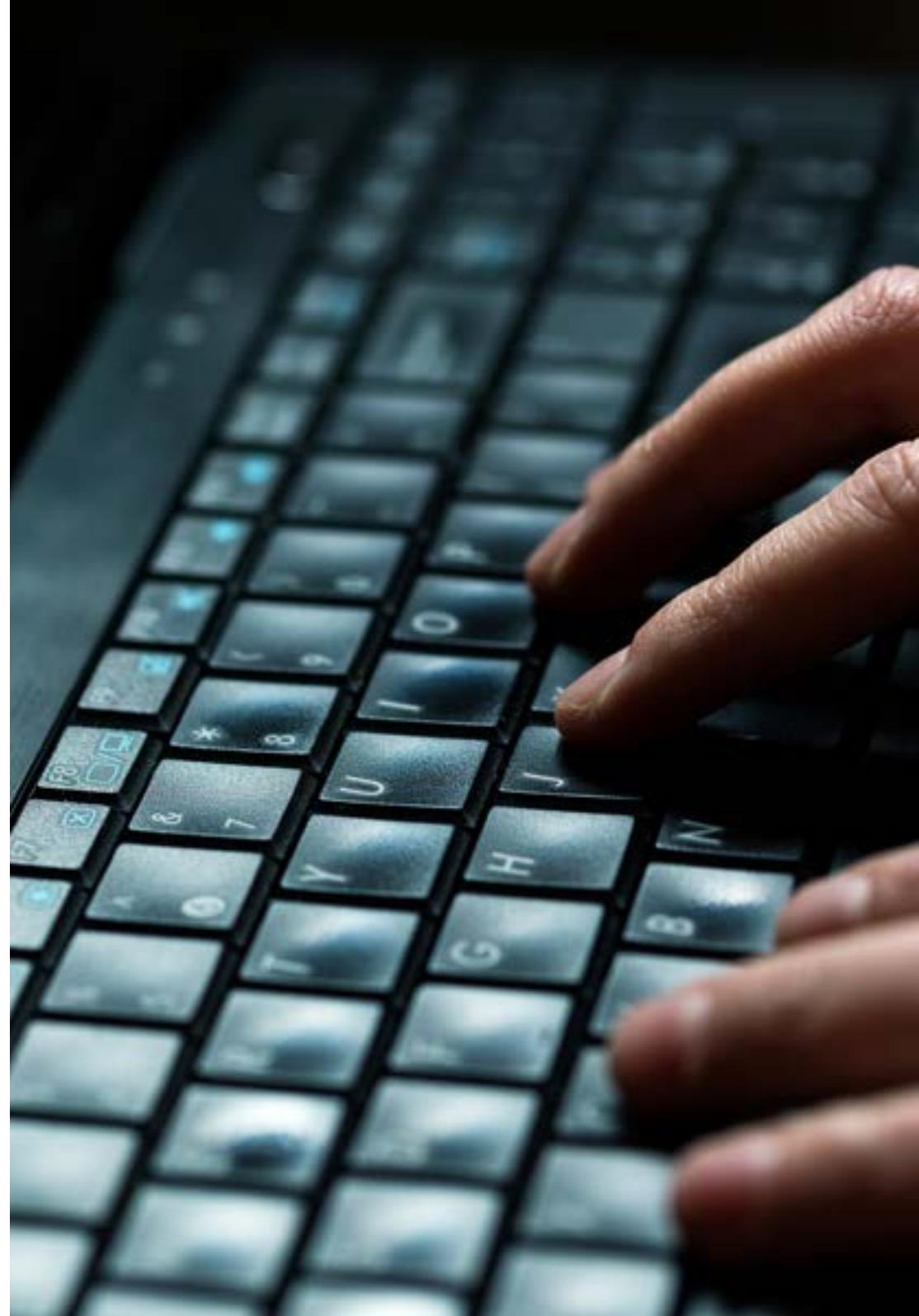
- 1.1. Einführung in die Softwaretechnik
  - 1.1.1. Einführung
  - 1.1.2. Die Software-Krise
  - 1.1.3. Unterschiede zwischen Softwaretechnik und Informatik
  - 1.1.4. Ethik und berufliche Verantwortung in der Softwaretechnik
  - 1.1.5. Software-Fabriken
- 1.2. Der Softwareentwicklungsprozess
  - 1.2.1. Definition
  - 1.2.2. Softwareprozessmodell
  - 1.2.3. Der einheitliche Softwareentwicklungsprozess
- 1.3. Objektorientierte Softwareentwicklung
  - 1.3.1. Einführung
  - 1.3.2. Grundsätze der Objektorientierung
  - 1.3.3. Definition von Objekt
  - 1.3.4. Definition der Klasse
  - 1.3.5. Objektorientierte Analyse vs. Objektorientiertes Design
- 1.4. Modellbasierte Softwareentwicklung
  - 1.4.1. Die Notwendigkeit eines Modells
  - 1.4.2. Modellierung von Softwaresystemen
  - 1.4.3. Modellierung von Objekten
  - 1.4.4. UML
  - 1.4.5. CASE-Werkzeuge
- 1.5. Anwendungsmodellierung und Designmuster mit UML
  - 1.5.1. Fortgeschrittene Anforderungsmodellierung
  - 1.5.2. Fortgeschrittene statische Modellierung
  - 1.5.3. Fortgeschrittene dynamische Modellierung
  - 1.5.4. Komponentenmodellierung
  - 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. Modellgesteuerte Technik
  - 1.6.1. Einführung
  - 1.6.2. Metamodellierung von Systemen
  - 1.6.3. MDA
  - 1.6.4. DSL
  - 1.6.5. Modellverfeinerungen mit OCL
  - 1.6.6. Modellumwandlungen
- 1.7. Ontologien in der Softwaretechnik
  - 1.7.1. Einführung
  - 1.7.2. Ontologietechnik
  - 1.7.3. Anwendung von Ontologien in der Softwaretechnik
- 1.8. Agile Methoden für die Softwareentwicklung, *Scrum*
  - 1.8.1. Was ist Software-Agilität?
  - 1.8.2. Das agile Manifest
  - 1.8.3. Der agile Projektfahrplan
  - 1.8.4. Der *Product Owner*
  - 1.8.5. Benutzerberichte
  - 1.8.6. Agile Planung und Schätzung
  - 1.8.7. Messung in der agilen Entwicklung
  - 1.8.8. Einführung in Scrum
  - 1.8.9. Die Rollen
  - 1.8.10. Der *Product Backlog*
  - 1.8.11. Der *Sprint*
  - 1.8.12. Die Sitzungen
- 1.9. Die Methodik der *Lean*-Softwareentwicklung
  - 1.9.1. Einführung
  - 1.9.2. *Kanban*

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

## Modul 2. Software-Projektmanagement

- 2.1. Grundlegende Konzepte des Projektmanagements und des Lebenszyklus des Projektmanagements
  - 2.1.1. Was ist ein Projekt?
  - 2.1.2. Einheitliche Methodik
  - 2.1.3. Was ist Projektmanagement?
  - 2.1.4. Was ist ein Projektplan?
  - 2.1.5. Vorteile
  - 2.1.6. Projektlebenszyklen
  - 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. Start 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 Reichweite
  - 2.3.1. Identifizierung von Interessengruppen
  - 2.3.2. Entwicklung eines Plans für das Management von Interessengruppen
  - 2.3.3. Management des Engagements von Interessengruppen
  - 2.3.4. Überwachung des Engagements der Interessengruppen
  - 2.3.5. Das Ziel des Projekts
  - 2.3.6. Management der Einsätze und ihr Plan
  - 2.3.7. Erfassen von Anforderungen
  - 2.3.8. Definition des Anwendungsbereichs
  - 2.3.9. Erstellung des Projektstrukturplans (PSP)
  - 2.3.10. Prüfen und Kontrolle des Umfangs
- 2.4. Entwicklung des Zeitplans
  - 2.4.1. Zeitmanagement und dessen Plan
  - 2.4.2. Definieren der Aktivitäten
  - 2.4.3. Festlegung der Abfolge der Aktivitäten
  - 2.4.4. Geschätzte Ressourcen für die Aktivitäten
  - 2.4.5. Geschätzte Dauer der Aktivitäten
  - 2.4.6. Entwicklung des Zeitplans und Berechnung des kritischen Pfads
  - 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. Die Konzepte des Risikos
  - 2.5.5. Wie wird eine Risikoanalyse durchgeführt?
  - 2.5.6. Die Entwicklung des Reaktionsplans
- 2.6. Qualitätsmanagement
  - 2.6.1. Planung der Qualität
  - 2.6.2. Qualitätssicherung
  - 2.6.3. Qualitätskontrolle
  - 2.6.4. Grundlegende statistische Konzepte
  - 2.6.5. Werkzeuge des Qualitätsmanagements

- 2.7. Kommunikation und Humanressourcen
  - 2.7.1. Planung des Kommunikationsmanagement
  - 2.7.2. Analyse der Kommunikationsanforderungen
  - 2.7.3. Technologie der Kommunikation
  - 2.7.4. Kommunikationsmodelle
  - 2.7.5. Kommunikationsmethoden
  - 2.7.6. Plan für das Kommunikationsmanagement
  - 2.7.7. Verwaltung der Kommunikation
  - 2.7.8. Verwaltung des Personalwesens
  - 2.7.9. Hauptakteure und ihre Rolle in den Projekten
  - 2.7.10. Organisationsformen
  - 2.7.11. Projektorganisation
  - 2.7.12. Das Projektteam
- 2.8. Die Beschaffung
  - 2.8.1. Der Beschaffungsprozess
  - 2.8.2. Planung
  - 2.8.3. Suche nach Lieferanten und Ausschreibungen
  - 2.8.4. Vergabe des Auftrags
  - 2.8.5. Vertragsverwaltung
  - 2.8.6. Verträge
  - 2.8.7. Arten von Verträgen
  - 2.8.8. Vertragsverhandlungen
- 2.9. Durchführung, Überwachung und Kontrolle sowie Abschluss
  - 2.9.1. Prozessgruppen
  - 2.9.2. Die Projektdurchführung
  - 2.9.3. Die 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 die Projektleitung
  - 2.10.4. Verantwortung vs. PMP®
  - 2.10.5. Beispiele für Verantwortung
  - 2.10.6. Vorteile der Professionalisierung



## Modul 3. Plattformen für die Softwareentwicklung

- 3.1. Einführung in die Anwendungsentwicklung
  - 3.1.1. Desktop-Anwendungen
  - 3.1.2. Programmiersprache
  - 3.1.3. Integrierte Entwicklungsumgebungen
  - 3.1.4. Web-Anwendungen
  - 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. Haupt 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 Oberfläche in *Eclipse*
  - 3.2.7. Entwerfen einer grafischen Oberfläche in *NetBeans*
- 3.3. Fehlerbereinigung und Testen in *Java*
  - 3.3.1. Testen und Fehlerbereinigung in *Java*-Programmen
  - 3.3.2. Fehlerbereinigung in *Eclipse*
  - 3.3.3. Fehlerbereinigung 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-Tools für GUI
  - 3.4.5. Die GUI mit *Windows Presentation Foundation*
  - 3.4.6. Fehlerbereinigung und Kompilierung einer WPF-Anwendung
- 3.5. Programmierung für .NET-Netzwerke
  - 3.5.1. Einführung in die .NET-Netzwerkprogrammierung
  - 3.5.2. Anfragen und Antworten in .NET
  - 3.5.3. Verwendung von Anwendungsprotokollen in .NET
  - 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. Anwendungsentwicklung in einer *Android Studio*-Umgebung
  - 3.7.1. Installation und Start von *Android Studio*
  - 3.7.2. Ausführen einer *Android*-Anwendung
  - 3.7.3. Entwicklung von grafischen Benutzeroberflächen in *Android Studio*
  - 3.7.4. Starten von Aktivitäten in *Android Studio*
- 3.8. Fehlerbereinigung und Veröffentlichung von *Android*-Anwendungen
  - 3.8.1. Fehlerbereinigung einer Anwendung in *Android Studio*
  - 3.8.2. Speicherung von Anwendungen in *Android Studio*
  - 3.8.3. Veröffentlichen einer Anwendung in *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. Dienste der *Google Cloud Platform*
  - 3.10.3. Werkzeuge der *Google Cloud Platform*

## Modul 4. Web-Client-Computing

- 4.1. Einführung in HTML
  - 4.1.1. Aufbau eines Dokuments
  - 4.1.2. Farbe
  - 4.1.3. Text
  - 4.1.4. Hypertext-Links
  - 4.1.5. Bilder
  - 4.1.6. Listen
  - 4.1.7. Tabellen
  - 4.1.8. Rahmen (*Frames*)
  - 4.1.9. Formulare
  - 4.1.10. Spezifische Elemente für mobile Technologien
  - 4.1.11. Ausgediente Elemente
- 4.2. Formatvorlagen für das Web (CSS)
  - 4.2.1. Elemente und Struktur einer Formatvorlage
    - 4.2.1.1. Erstellung von Formatvorlagen
    - 4.2.1.2. Anwendung von Stilen. Selektoren
    - 4.2.1.3. Formatvererbung und Kaskadierung
    - 4.2.1.4. Seitenformatierung mit Formatvorlagen
    - 4.2.1.5. Seitenstruktur mit Formatvorlagen Das Kastenmodell
  - 4.2.2. Designvorlagen für verschiedene Geräte
  - 4.2.3. Arten von Formatvorlagen: statisch und dynamisch. Pseudo-Klassen
  - 4.2.4. Bewährte Verfahren für die Verwendung von Formatvorlagen
- 4.3. Einführung und Geschichte von *JavaScript*
  - 4.3.1. Einführung
  - 4.3.2. Geschichte von *JavaScript*
  - 4.3.3. Zu verwendende Entwicklungsumgebung
- 4.4. Grundbegriffe der Webprogrammierung
  - 4.4.1. Grundlegende *JavaScript*-Syntax
  - 4.4.2. Primitive Datentypen und Operatoren
  - 4.4.3. Variablen und Domänen
  - 4.4.4. Textstrings und *Template Literals*
  - 4.4.5. Zahlen und Boolesche Werte
  - 4.4.6. Vergleiche
- 4.5. Komplexe *JavaScript*-Strukturen
  - 4.5.1. Vektoren oder *Arrays* und Objekte
  - 4.5.2. Sets
  - 4.5.3. Karten
  - 4.5.4. Disjunktionen
  - 4.5.5. Schleifen
- 4.6. Funktionen und Ziele
  - 4.6.1. Definition und Ausführung von Funktionen
  - 4.6.2. Argumente
  - 4.6.3. Pfeil-Funktionen
  - 4.6.4. Rückruf oder *Callback*-Funktionen
  - 4.6.5. Funktionen höherer Ordnung
  - 4.6.6. Literale Objekte
  - 4.6.7. Das Objekt *This*
  - 4.6.8. Objekte als Namensräume: das *Math*-Objekt und das *Date*-Objekt
- 4.7. Das Dokumentenobjektmodell (DOM)
  - 4.7.1. Was ist DOM?
  - 4.7.2. Ein bisschen Geschichte
  - 4.7.3. Navigieren und Abrufen von Elementen
  - 4.7.4. Ein virtuelles DOM mit JSDOM
  - 4.7.5. Abfrage-Selektoren oder *Query Selectors*
  - 4.7.6. Navigation über Einstellungen
  - 4.7.7. Zuweisung von Attributen zu Elementen
  - 4.7.8. Erstellen und Ändern von Knotenpunkten
  - 4.7.9. Aktualisierung 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 Versprechen und Funktionen
  - 4.8.6. *Closures*
  - 4.8.7. Funktionale Programmierung
  - 4.8.8. POO in *JavaScript*

- 4.9. Benutzerfreundlichkeit des Webs
  - 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 Zugänglichkeit und Benutzerfreundlichkeit
  - 4.9.5. Vorteile und Probleme bei der Kombination von Zugänglichkeit und Benutzerfreundlichkeit
  - 4.9.6. Vorteile und Schwierigkeiten bei der Umsetzung benutzerfreundlicher Websites
  - 4.9.7. Methoden zur Benutzerfreundlichkeit
  - 4.9.8. Analyse der Benutzeranforderungen
  - 4.9.9. Grundsätze der konzeptionellen Gestaltung. Benutzerorientiertes Prototyping
  - 4.9.10. Leitlinien für die Erstellung benutzerfreundlicher Websites
    - 4.9.10.1. Jakob Nielsens Richtlinien zur Benutzerfreundlichkeit
    - 4.9.10.2. Bruce Tognazzinis Richtlinien zur Benutzerfreundlichkeit
  - 4.9.11. Bewertung der Benutzbarkeit
- 4.10. Web-Zugänglichkeit
  - 4.10.1. Einführung
  - 4.10.2. Definition von Zugänglichkeit im Web
  - 4.10.3. Arten von Behinderungen
    - 4.10.3.1. Vorübergehende oder dauerhafte Behinderungen
    - 4.10.3.2. Sehschwäche
    - 4.10.3.3. Schwerhörigkeit
    - 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 Schwerhörige
    - 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 Zugänglichkeit im Internet
  - 4.10.6. Vorschriften und Normen für die Zugänglichkeit im Internet
  - 4.10.7. Regulierungsstellen für die Zugänglichkeit des Internets
  - 4.10.8. Vergleich von Normen und Standards
  - 4.10.9. Leitlinien für die Einhaltung von Normen und Standards
    - 4.10.9.1. Beschreibung der wichtigsten Leitlinien (Bilder, Videolinks, usw.)
    - 4.10.9.2. Leitlinien für eine zugängliche Navigation
      - 4.10.9.2.1. Wahrnehmbarkeit
      - 4.10.9.2.2. Operationalität
      - 4.10.9.2.3. Verständlichkeit
      - 4.10.9.2.4. Robustheit
  - 4.10.10. Beschreibung des Prozesses zur Einhaltung der Webzugänglichkeit
  - 4.10.11. Ebenen der Einhaltung
  - 4.10.12. Kriterien für die Einhaltung
  - 4.10.13. Anforderungen an die Einhaltung
  - 4.10.14. Methodik zur Bewertung der Zugänglichkeit von Websites

## Modul 5. Web-Server-Computing

- 5.1. Einführung in die serverseitige Programmierung: PHP
  - 5.1.1. Grundlagen der serverseitigen Programmierung
  - 5.1.2. Grundlegende PHP-Syntax
  - 5.1.3. Erzeugung von HTML-Inhalten mit PHP
  - 5.1.4. Entwicklungs- und Testumgebungen: XAMPP
- 5.2. Fortgeschrittenes PHP
  - 5.2.1. Kontrollstrukturen mit PHP
  - 5.2.2. Funktionen in PHP
  - 5.2.3. Array-Verarbeitung in PHP
  - 5.2.4. String-Verarbeitung mit PHP
  - 5.2.5. Objektorientiertes PHP
- 5.3. Datenmodelle
  - 5.3.1. Konzept der Daten. Lebenszyklus der Daten
  - 5.3.2. Datentypen
    - 5.3.2.1. Grundlagen
    - 5.3.2.2. Register
    - 5.3.2.3. Dynamische

- 5.4. Das relationale Modell
  - 5.4.1. Beschreibung
  - 5.4.2. Entitäten und Arten von Entitäten
  - 5.4.3. Datenelemente. Attribute
  - 5.4.4. Beziehungen: Typen, Untertypen, Kardinalität
  - 5.4.5. Schlüssel. Arten von Schlüssel
  - 5.4.6. Normalisierung. Normale Formen
- 5.5. Aufbau des logischen Datenmodells
  - 5.5.1. Spezifikation der Tabellen
  - 5.5.2. Definition von Säulen
  - 5.5.3. Wichtige Spezifikation
  - 5.5.4. Umwandlung in Normalformen. Abhängigkeit
- 5.6. Das physische Datenmodell. Datensätze
  - 5.6.1. Beschreibung der Datensätze
  - 5.6.2. Datentypen
  - 5.6.3. Zugriffsmodi
  - 5.6.4. Organisation von Datensätzen
- 5.7. Datenbankzugriff über PHP
  - 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 über PHP
  - 5.7.4. Einführung in *MySQL*
  - 5.7.5. Arbeiten mit einer *MySQL*-Datenbank: die SQL-Sprache
  - 5.7.6. Zugriff auf die *MySQL*-Datenbank über PHP
- 5.8. Kundeninteraktion über PHP
  - 5.8.1. PHP-Formulare
  - 5.8.2. *Cookies*
  - 5.8.3. Sitzungsmanagement
- 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. Dienstorientierte Architekturen (SOA): Webdienste
  - 5.10.3. Erstellung von SOAP- und REST-Webdiensten
  - 5.10.4. Das SOAP-Protokoll
  - 5.10.5. Das REST-Protokoll

## Modul 6. Sicherheitsmanagement

- 6.1. Die 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. Klassifizierung der Informationen
  - 6.1.6. Informationssicherheit ist Risikomanagement
  - 6.1.7. Sicherheit wird mit Sicherheitskontrollen verknüpft
  - 6.1.8. Sicherheit ist sowohl physisch als auch logisch
  - 6.1.9. Sicherheit betrifft Menschen
- 6.2. Die Fachkraft für Informationssicherheit
  - 6.2.1. Einführung
  - 6.2.2. Informationssicherheit als Beruf
  - 6.2.3. Zertifizierungen (ISC)2
  - 6.2.4. Die Norm ISO 27001
  - 6.2.5. Bewährte Sicherheitspraktiken im IT-Service-Management
  - 6.2.6. Reifegradmodelle für die Informationssicherheit
  - 6.2.7. Andere Zertifizierungen, Standards und professionelle Ressourcen
- 6.3. Zugangskontrolle
  - 6.3.1. Einführung
  - 6.3.2. Anforderungen an die Zugangskontrolle
  - 6.3.3. Authentifizierungsmechanismen
  - 6.3.4. Methoden der Autorisierung
  - 6.3.5. Buchhaltung und Revision des Zugangs
  - 6.3.6. «Triple A»-Technologien

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

## Modul 7. Sicherheit von Informationssystemen

- 7.1. Ein Überblick über Sicherheit, Kryptographie und klassische Kryptoanalyse
  - 7.1.1. Computersicherheit: Historische Perspektive
  - 7.1.2. Aber was genau ist mit Sicherheit gemeint?
  - 7.1.3. Geschichte der Kryptographie
  - 7.1.4. Substitutions-Chiffren
  - 7.1.5. Fallstudie: Die Enigma-Maschine
- 7.2. Symmetrische Kryptographie
  - 7.2.1. Einführung und grundlegende Terminologie
  - 7.2.2. Symmetrische Verschlüsselung
  - 7.2.3. Betriebsarten
  - 7.2.4. DES
  - 7.2.5. Der neue AES-Standard
  - 7.2.6. Stream-Verschlüsselung
  - 7.2.7. Kryptoanalyse
- 7.3. Asymmetrische Kryptographie
  - 7.3.1. Die Ursprünge der Public Key Kryptographie
  - 7.3.2. Grundlegende Konzepte und Bedienung
  - 7.3.3. Der RSA-Algorithmus
  - 7.3.4. Digitale Zertifikate
  - 7.3.5. Speicherung und Verwaltung von Schlüsseln
- 7.4. Netzwerk-Angriffe
  - 7.4.1. Bedrohungen und Angriffe aus dem Netzwerk
  - 7.4.2. Aufzählung
  - 7.4.3. Verkehrsüberwachung: *Sniffers*
  - 7.4.4. Denial-of-Service-Angriffe
  - 7.4.5. ARP-Poisoning-Angriffe
- 7.5. Sicherheitsarchitekturen
  - 7.5.1. Traditionelle Sicherheitsarchitekturen
  - 7.5.2. *Secure Socket Layer*: SSL
  - 7.5.3. SSH-Protokoll
  - 7.5.4. Virtuelle private Netzwerke (VPNs)
  - 7.5.5. Schutzmechanismen für externe Speicherlaufwerke
  - 7.5.6. Hardware-Schutzmechanismen

- 7.6. Systemschutztechniken und Entwicklung von sicherem Code
  - 7.6.1. Sicherheit im Betrieb
  - 7.6.2. Ressourcen und Kontrollen
  - 7.6.3. Überwachung
  - 7.6.4. Intrusion Detection Systeme
  - 7.6.5. Host IDS
  - 7.6.6. Netzwerk IDS
  - 7.6.7. Signatur-basiertes IDS
  - 7.6.8. Decoy Systeme
  - 7.6.9. Grundlegende Sicherheitsprinzipien bei der Code-Entwicklung
  - 7.6.10. Störungsmanagement
  - 7.6.11. Staatsfeind Nummer 1: Der Buffer Overflow
  - 7.6.12. Kryptographische Botschaften
- 7.7. Botnets und Spam
  - 7.7.1. Ursprung des Problems
  - 7.7.2. Prozess von Spam
  - 7.7.3. Spam verschicken
  - 7.7.4. Verfeinerung der Verteilerlisten
  - 7.7.5. Methoden zum Schutz
  - 7.7.6. Von Dritten angebotener Anti-Spam-Service
  - 7.7.7. Fallstudien
  - 7.7.8. Exotischer Spam
- 7.8. Web Auditing und Angriffe
  - 7.8.1. Sammeln von Informationen
  - 7.8.2. Angriffs-Techniken
  - 7.8.3. Instrumente
- 7.9. Malware und bössartiger Code
  - 7.9.1. Was ist Malware?
  - 7.9.2. Arten von Malware
  - 7.9.3. Virus
  - 7.9.4. Kryptoviren
  - 7.9.5. Würmer
  - 7.9.6. Adware

- 7.9.7. Spyware
- 7.9.8. Hoaxes
- 7.9.9. Phishing
- 7.9.10. Trojaner
- 7.9.11. Die Malware-Wirtschaft
- 7.9.12. Mögliche Lösungen
- 7.10. Forensische Analyse
  - 7.10.1. Sammeln von Beweisen
  - 7.10.2. Analyse der Beweise
  - 7.10.3. Anti-Forensische Techniken
  - 7.10.4. Praktische Fallstudie

## Modul 8. Software-Sicherheit

- 8.1. Software-Sicherheitsprobleme
  - 8.1.1. Einführung in die Problematik der Software-Sicherheit
  - 8.1.2. Schwachstellen und ihre Klassifizierung
  - 8.1.3. Sichere Software-Eigenschaften
  - 8.1.4. Referenzen
- 8.2. Grundsätze des Software-Sicherheitsdesigns
  - 8.2.1. Einführung
  - 8.2.2. Grundsätze des Software-Sicherheitsdesigns
  - 8.2.3. Arten von S-SDLC
  - 8.2.4. Software-Sicherheit in S-SDLC-Phasen
  - 8.2.5. Methodologien und Normen
  - 8.2.6. Referenzen
- 8.3. Sicherheit im Software-Lebenszyklus in der Anforderungs- und Entwurfsphase
  - 8.3.1. Einführung
  - 8.3.2. Angriffsmodellierung
  - 8.3.3. Fälle von Missbrauch
  - 8.3.4. Entwicklung von Sicherheitsanforderungen
  - 8.3.5. Risikoanalyse. Architektonisch
  - 8.3.6. Entwurfsmuster
  - 8.3.7. Referenzen

- 8.4. Sicherheit im Software-Lebenszyklus in den Phasen Kodierung, Test und Betrieb
  - 8.4.1. Einführung
  - 8.4.2. Risikobasierte Sicherheitstests
  - 8.4.3. Code-Überprüfung
  - 8.4.4. Penetrationstest
  - 8.4.5. Sicherheitsmaßnahmen
  - 8.4.6. Externe Überprüfung
  - 8.4.7. Referenzen
- 8.5. Sichere Kodierungsanwendungen I
  - 8.5.1. Einführung
  - 8.5.2. Sichere Kodierungspraktiken
  - 8.5.3. Bearbeitung und Validierung von Einträgen
  - 8.5.4. Speicherüberlauf
  - 8.5.5. Referenzen
- 8.6. Sichere Kodierungsanwendungen II
  - 8.6.1. Einführung
  - 8.6.2. *Integers Overflows*, Abbruchfehler und Probleme mit Typkonvertierungen zwischen Ganzzahlen
  - 8.6.3. Fehler und Ausnahmen
  - 8.6.4. Datenschutz und Vertraulichkeit
  - 8.6.5. Privilegierte Programme
  - 8.6.6. Referenzen
- 8.7. Sicherheit in der Entwicklung und in der Cloud
  - 8.7.1. Sicherheit in der Entwicklung; Methodik und Praxis
  - 8.7.2. PaaS, IaaS, CaaS und SaaS-Modelle
  - 8.7.3. Sicherheit in der Cloud und für Cloud-Dienste
- 8.8. Orchestrierung und Automatisierung der Sicherheit (SOAR)
  - 8.9.1. Komplexität der manuellen Verarbeitung; Notwendigkeit, Aufgaben zu automatisieren
  - 8.9.2. Produkte und Dienstleistungen
  - 8.9.3. SOAR-Architektur
- 8.9. Sicherheit bei der Telearbeit
  - 8.9.1. Bedarf und Szenarien
  - 8.9.2. Produkte und Dienstleistungen
  - 8.9.3. Sicherheit bei der Telearbeit

## Modul 9. Qualität und Prüfung von Informationssystemen

- 9.1. Einführung in Informationssicherheits-Managementsysteme
  - 9.1.1. Grundlegende Prinzipien des ISMS
  - 9.1.2. Goldene Regeln des ISMS
  - 9.1.3. Die Rolle der IT-Auditierung in ISMS
- 9.2. Planung im Sicherheitsmanagement
  - 9.2.1. Konzepte für das Sicherheitsmanagement
  - 9.2.2. Klassifizierung von Informationen: Ziele, Konzepte und Rollen
  - 9.2.3. Umsetzung der Sicherheitspolitik: Sicherheitspolitik, Standards und Verfahren
  - 9.2.4. Risikomanagement: Grundsätze und Analyse des Risikos von Informationsgütern
- 9.3. Wichtigste Mechanismen für den Schutz von Informationsgütern (I)
  - 9.3.1. Überblick über die wichtigsten kryptografischen Werkzeuge zum Schutz der CID-Trias
  - 9.3.2. Berücksichtigung der Erfordernisse des Datenschutzes, der Anonymität und der ordnungsgemäßen Verwaltung der Rückverfolgbarkeit der Nutzer
- 9.4. Wichtigste Mechanismen für den Schutz von Informationsgütern (II)
  - 9.4.1. Kommunikationssicherheit: Protokolle, Geräte und Sicherheitsarchitekturen
  - 9.4.2. Sicherheit der Betriebssysteme
- 9.5. Interne ISMS-Kontrollen
  - 9.5.1. Taxonomie der ISMS-Kontrollen: administrative, logische und physische Kontrollen
  - 9.5.2. Klassifizierung der Kontrollen nach der Art und Weise, wie sie die Bedrohung angehen: Kontrollen zur Vorbeugung, Aufdeckung und Korrektur von Bedrohungen
  - 9.5.3. Implementierung von internen Kontrollsystemen in ISMS
- 9.6. Arten von Audits
  - 9.6.1. Unterschied zwischen Audit und interner Prüfung
  - 9.6.2. Internes vs. externes Audit
  - 9.6.3. Klassifizierung des Audits nach dem Ziel und der Art der Analyse
- 9.7. Drehbuchautor und Drehbuch: Subjekt und Objekt geschützt durch das Recht des geistigen Eigentums
  - 9.7.1. Einführung in Penetrationstests und forensische Analyse
  - 9.7.2. Definition und Relevanz der Konzepte des *Fingerprinting* und *Footprinting*

- 9.8. Scannen auf Schwachstellen und Überwachung des Netzwerkverkehrs
  - 9.8.1. Werkzeuge für die Schwachstellenanalyse in Systemen
  - 9.8.2. Die wichtigsten Schwachstellen im Zusammenhang mit Web-Applikationen
  - 9.8.3. Analyse der Kommunikationsprotokollen
- 9.9. Der Prozess des IT-Audits
  - 9.9.1. Lebenszykluskonzept in der Systementwicklung
  - 9.9.2. Tätigkeits- und Prozessüberwachung: Sammlung und Verarbeitung von Beweisen
  - 9.9.3. Methodik des IT-Audits
  - 9.9.4. Prozess eines IT-Audits
  - 9.9.5. Identifizierung der wichtigsten Straftaten und Ordnungswidrigkeiten im Zusammenhang mit der Informationstechnologie
  - 9.9.6. Untersuchung von Internetkriminalität: eine Einführung in die forensische Analyse und ihre Beziehung zum IT-Audits
- 9.10. Planung von Betriebskontinuität und Wiederherstellung im Katastrophenfall
  - 9.10.1. Definition des Betriebskontinuitätsplans und des Konzepts der Betriebsunterbrechung
  - 9.10.2. NIST-Empfehlung zur Planung der Betriebskontinuität
  - 9.10.3. Plan zur Wiederherstellung im Katastrophenfall
  - 9.10.4. Prozess des Plan zur Wiederherstellung im Katastrophenfall
- 10.2. Umgang mit dem HTTP-Protokoll
  - 10.2.1. Funktionsweise und Struktur
  - 10.2.2. Beschreibung der Anträge oder *Request Methods*
  - 10.2.3. Status-Codes
  - 10.2.4. Kopfzeilen
  - 10.2.5. Kodierung des Inhalts. Code-Seiten
  - 10.2.6. Durchführung von HTTP-Anfragen im Internet über einen Proxy, *Livehttpheader* oder eine ähnliche Methode, Analyse des verwendeten Protokolls
- 10.3. Beschreibung von verteilten Architekturen auf mehreren Servern
  - 10.3.1. 3-Schichten-Modell
  - 10.3.2. Fehlertoleranz
  - 10.3.3. Lastverteilung
  - 10.3.4. Speicherung des Sitzungsstatus
  - 10.3.5. Cache-Speicher
- 10.4. *Internet Information Services (IIS)*
  - 10.4.1. Was ist IIS?
  - 10.4.2. Geschichte und Entwicklung des IIS
  - 10.4.3. Die wichtigsten Vorteile und Merkmale des IIS und darüber hinaus
  - 10.4.4. Architektur von IIS und späteren Versionen
- 10.5. IIS-Installation, -Verwaltung und -Konfiguration
  - 10.5.1. Präambel
  - 10.5.2. Installation von *Internet Information Services (IIS)*
  - 10.5.3. IIS-Verwaltungstools
  - 10.5.4. Erstellung, Konfiguration und Verwaltung von Websites
  - 10.5.5. Installation und Verwaltung von Erweiterungen im IIS
- 10.6. Erweiterte Sicherheit im IIS
  - 10.6.1. Präambel
  - 10.6.2. Authentifizierung, Autorisierung und Zugriffskontrolle in IIS
  - 10.6.3. Einrichten einer sicheren Website auf IIS mit SSL
  - 10.6.4. In IIS 10.x implementierte Sicherheitsrichtlinien
- 10.7. Einführung in Apache
  - 10.7.1. Was ist Apache?
  - 10.7.2. Die wichtigsten Vorteile von Apache
  - 10.7.3. Hauptmerkmale von Apache
  - 10.7.4. Architektur

## Modul 10. Verwaltung des Webservers

- 10.1. Einführung in Webserver
  - 10.1.1. Was ist ein Webserver?
  - 10.1.2. Architektur und Funktionsweise eines Webservers
  - 10.1.3. Ressourcen und Inhalte auf einem Webserver
  - 10.1.4. Anwendungsserver
  - 10.1.5. Proxy-Server
  - 10.1.6. Die wichtigsten Webserver auf dem Markt
  - 10.1.7. Statistiken zur Webserver-Nutzung
  - 10.1.8. Sicherheit des Webservers
  - 10.1.9. Lastausgleich auf Webservern
  - 10.1.10. Referenzen

- 10.8. Installation und Konfiguration von Apache
  - 10.8.1. Erstinstallation von Apache
  - 10.8.2. Konfiguration von Apache
- 10.9. Installation und Konfiguration der verschiedenen Module von Apache
  - 10.9.1. Installation von Apache-Modulen
  - 10.9.2. Arten von Modulen
  - 10.9.3. Sichere Konfiguration von Apache
- 10.10. Erweiterte Sicherheit
  - 10.10.1. Authentifizierung, Autorisierung und Zugangskontrolle
  - 10.10.2. Methoden der Authentifizierung
  - 10.10.3. Sichere Apache-Konfiguration mit SSL

## Modul 11. Sicherheit bei Online-Anwendungen

- 11.1. Schwachstellen und Sicherheitsprobleme in Online-Anwendungen
  - 11.1.1. Einführung in die Sicherheit von Online-Anwendungen
  - 11.1.2. Sicherheitsschwachstellen beim Entwurf von Webanwendungen
  - 11.1.3. Sicherheitsschwachstellen bei der Implementierung von Webanwendungen
  - 11.1.4. Sicherheitsschwachstellen bei der Bereitstellung von Webanwendungen
  - 11.1.5. Offizielle Listen von Sicherheitslücken
- 11.2. Richtlinien und Standards für die Sicherheit von Online-Anwendungen
  - 11.2.1. Säulen für die Sicherheit von Online-Anwendungen
  - 11.2.2. Sicherheitspolitik
  - 11.2.3. Managementsystem für die Informationssicherheit
  - 11.2.4. Sicherer Lebenszyklus der Softwareentwicklung
  - 11.2.5. Standards für die Anwendungssicherheit
- 11.3. Sicherheit beim Design von Webanwendungen
  - 11.3.1. Einführung in die Sicherheit von Webanwendungen
  - 11.3.2. Sicherheit beim Design von Webanwendungen
- 11.4. Prüfung der Online-Sicherheit von Webanwendungen
  - 11.4.1. Analyse und Prüfung der Sicherheit von Webanwendungen
  - 11.4.2. Sicherheit bei der Bereitstellung und Produktion von Webanwendungen
- 11.5. Sicherheit von Webdiensten
  - 11.5.1. Einführung in die Sicherheit von Webdiensten
  - 11.5.2. Sicherheitsfunktionen und -technologien für Webdienste

- 11.6. Prüfung der Online-Sicherheit von Webdiensten
  - 11.6.1. Sicherheitsbewertung von Webdiensten
  - 11.6.2. Online-Schutz. *Firewalls* und *Gateways* XML
- 11.7. Ethisches *Hacking*, *Malware* und *Forensic*
  - 11.7.1. Ethisches *Hacking*
  - 11.7.2. Malware Analyse
  - 11.7.3. Forensische Analyse
- 11.8. Störungsbehebung für Webdienste
  - 11.8.1. Überwachung
  - 11.8.2. Instrumente zur Leistungsmessung
  - 11.8.3. Eindämmungsmaßnahmen
  - 11.8.4. Analyse der Grundursache
  - 11.8.5. Proaktives Problemmanagement
- 11.9. Bewährte Verfahren zur Gewährleistung der Anwendungssicherheit
  - 11.9.1. Handbuch für bewährte Verfahren bei der Entwicklung von Online-Anwendungen
  - 11.9.2. Handbuch für bewährte Praktiken bei der Einführung von Online-Anwendungen
- 11.10. Häufige Fehler, die die Anwendungssicherheit untergraben
  - 11.10.1. Häufige Fehler bei der Entwicklung
  - 11.10.2. Häufige Fehler beim Hosting
  - 11.10.3. Häufige Fehler in der Produktion

## Modul 12. Software-Entwicklung

- 12.1. Einführung in die Softwareentwicklung und -modellierung
  - 12.1.1. Die Natur der Software
  - 12.1.2. Die Besonderheit von *Webapps*
  - 12.1.3. Softwareentwicklung
  - 12.1.4. Der Software-Prozess
  - 12.1.5. Die Praxis der Softwareentwicklung
  - 12.1.6. Software-Mythen
  - 12.1.7. Wie fängt das alles an?
  - 12.1.8. Objektorientierte Konzepte
  - 12.1.9. Einführung in UML

- 12.2. Der Software-Prozess
  - 12.2.1. Ein allgemeines Prozessmodell
  - 12.2.2. Vorgeschriebene Prozessmodelle
  - 12.2.3. Spezialisierte Prozessmodelle
  - 12.2.4. Der vereinheitlichte Prozess
  - 12.2.5. Personal- und Teamprozessmodelle
  - 12.2.6. Was ist Agilität?
  - 12.2.7. Was ist ein agiler Prozess?
  - 12.2.8. *Scrum*
  - 12.2.9. Werkzeugkasten für agile Prozesse
- 12.3. Grundsätze für die Praxis der Softwareentwicklung
  - 12.3.1. Leitprinzipien des Prozesses
  - 12.3.2. Prinzipien als Leitfaden für die Praxis
  - 12.3.3. Grundsätze der Kommunikation
  - 12.3.4. Grundsätze der Planung
  - 12.3.5. Grundsätze der Modellierung
  - 12.3.6. Konstruktionsprinzipien
  - 12.3.7. Grundsätze für die Einführung
- 12.4. Verständnis der Anforderungen
  - 12.4.1. Anforderungsmanagement
  - 12.4.2. Schaffung der Grundlagen
  - 12.4.3. Bedarfsermittlung
  - 12.4.4. Entwicklung von Anwendungsfällen
  - 12.4.5. Ausarbeitung des Anforderungsmodells
  - 12.4.6. Aushandeln von Anforderungen
  - 12.4.7. Validierung der Anforderungen
- 12.5. Anforderungsmodellierung: Szenarien, Informations- und Analyseklassen
  - 12.5.1. Analyse der Anforderungen
  - 12.5.2. Szenario-basiertes Modell
  - 12.5.3. UML-Modelle, die den Anwendungsfall liefern
  - 12.5.4. Konzepte der Datenmodellierung
  - 12.5.5. Klassen-basiertes Modell
  - 12.5.6. Klassendiagramme
- 12.6. Anforderungsmodellierung: Fluss, Verhalten und Muster
  - 12.6.1. Anforderungen die die Strategien gestalten
  - 12.6.2. Flussorientierte Modellierung
  - 12.6.3. Zustandsdiagramme
  - 12.6.4. Erstellung eines Verhaltensmodells
  - 12.6.5. Sequenzdiagramme
  - 12.6.6. Kommunikationsdiagramme
  - 12.6.7. Muster für die Modellierung von Anforderungen
- 12.7. Konzepte des Designs
  - 12.7.1. Design im Kontext der Softwareentwicklung
  - 12.7.2. Der Entwurfsprozess
  - 12.7.3. Konzepte des Designs
  - 12.7.4. Objektorientierte Konzepte des Designs
  - 12.7.5. Das Designmodell
- 12.8. Design der Architektur
  - 12.8.1. Software-Architektur
  - 12.8.2. Architektonische Gattungen
  - 12.8.3. Architektonische Stile
  - 12.8.4. Architektonischer Design
  - 12.8.5. Entwicklung von alternativen Designs für die Architektur
  - 12.8.6. Abbildung der Architektur mit Hilfe von Datenflüssen
- 12.9. Design auf Komponentenebene und musterbasierter Entwurf
  - 12.9.1. Was ist eine Komponente?
  - 12.9.2. Klassenbasiertes Komponentendesign
  - 12.9.3. Verwirklichung des Designs auf Komponentenebene
  - 12.9.4. Design der traditionellen Komponenten
  - 12.9.5. Komponentenbasierte Entwicklung
  - 12.9.6. Entwurfsmuster
  - 12.9.7. Musterbasiertes Softwaredesign
  - 12.9.8. Architektonische Muster
  - 12.9.9. Musterdesign auf Komponentenebene
  - 12.9.10. Musterdesign für Benutzeroberflächen

- 12.10. Softwarequalität und Projektmanagement
  - 12.10.1. Qualität
  - 12.10.2. Software Qualität
  - 12.10.3. Das Dilemma der Softwarequalität
  - 12.10.4. Erreichen von Softwarequalität
  - 12.10.5. Software-Qualitätssicherung
  - 12.10.6. Das administrative Spektrum
  - 12.10.7. Personal
  - 12.10.8. Das Produkt
  - 12.10.9. Der Prozess
  - 12.10.10. Das Projekt
  - 12.10.11. Grundsätze und Praktiken

## Modul 13. Fortgeschrittene Softwaretechnik

- 13.1. Einführung in agile Methodologien
  - 13.1.1. Prozessmodelle und Methodologien
  - 13.1.2. Agilität und agile Prozesse
  - 13.1.3. Das Agile Manifest
  - 13.1.4. Einige agile Methodologien
  - 13.1.5. Agil vs. Traditionell
- 13.2. Scrum
  - 13.2.1. Ursprünge und Philosophie von Scrum
  - 13.2.2. Scrum Werte
  - 13.2.3. Scrum Prozessablauf
  - 13.2.4. Scrum Rollen
  - 13.2.5. Scrum Artefakte
  - 13.2.6. Scrum Ereignisse
  - 13.2.7. Benutzerberichte
  - 13.2.8. Scrum Erweiterungen
  - 13.2.9. Agile Schätzungen
  - 13.2.10. Scrum Skalierung
- 13.3. Extreme Programmierung
  - 13.3.1. Grundprinzip und Überblick über XP
  - 13.3.2. Der Lebenszyklus von XP
  - 13.3.3. Die fünf Grundwerte
  - 13.3.4. Die zwölf Kernpraktiken von XP
  - 13.3.5. Rollen der Teilnehmer
  - 13.3.6. Industrie XP
  - 13.3.7. Kritische Beurteilung von XP
- 13.4. Softwareentwicklung auf der Grundlage von Wiederverwendung
  - 13.4.1. Wiederverwendung von Software
  - 13.4.2. Stufen der Wiederverwendung von Codes
  - 13.4.3. Spezifische Techniken der Wiederverwendung
  - 13.4.4. Komponentenbasierte Entwicklung
  - 13.4.5. Vorteile und Probleme der Wiederverwendung
  - 13.4.6. Planung der Wiederverwendung
- 13.5. Systemarchitektur und Software-Entwurfsmuster
  - 13.5.1. Architektonisches Design
  - 13.5.2. Allgemeine architektonische Muster
  - 13.5.3. Fehlertolerante Architekturen
  - 13.5.4. Architekturen verteilter Systeme
  - 13.5.5. Entwurfsmuster
  - 13.5.6. Gamma Muster
  - 13.5.7. Muster für die Interaktionsgestaltung
- 13.6. Architektur von Cloud-Anwendungen
  - 13.6.1. Grundlagen des *Cloud Computing*
  - 13.6.2. Qualität von Cloud-Anwendungen
  - 13.6.3. Architektonische Stile
  - 13.6.4. Entwurfsmuster
- 13.7. Software-Tests: TDD, ATDD und BDD
  - 13.7.1. Überprüfung und Validierung von Software
  - 13.7.2. Software-Tests
  - 13.7.3. *Test Driven Development* (TDD)
  - 13.7.4. *Acceptance Test Driven Development* (ATDD)
  - 13.7.5. *Behavior Driven Development* (BDD)
  - 13.7.6. BDD y Cucumber

- 13.8. Verbesserung von Software-Prozessen
  - 13.8.1. Verbesserung von Software-Prozessen
  - 13.8.2. Der Prozess der Prozessverbesserung
  - 13.8.3. Reifegradmodelle
  - 13.8.4. CMMI-Modell
  - 13.8.5. CMMI V13.0
  - 13.8.6. CMMI und Agilität
- 13.9. Qualität von Softwareprodukten: *SQuaRE*
  - 13.9.1. Qualität der Software
  - 13.9.2. Qualitätsmodelle für Softwareprodukte
  - 13.9.3. ISO/IEC 13000-Familie
  - 13.9.4. ISO/IEC 13010: Qualitätsmodell und Qualitätsmerkmale
  - 13.9.5. ISO/IEC 13.0113: Datenqualität
  - 13.9.6. ISO/IEC 13.013: Messung der Softwarequalität
  - 13.9.7. ISO/IEC 13.013., 13.013. und 13.013.: Software- und Datenqualitätsmetriken
  - 13.9.8. ISO/IEC 13040: Bewertung der Software
  - 13.9.9. Der Prozess der Zertifizierung
- 13.10. Einführung in *DevOps*
  - 13.10.1. DevOps-Konzept
  - 13.10.2. Grundlegende Praktiken

## Modul 14. Anforderungsmanagement

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

- 14.5. Prototyping Techniken
  - 14.5.1. Prototyping
  - 14.5.2. Prototypen nach ihrem Anwendungsbereich
  - 14.5.3. Prototypen nach ihrer Aktualität
  - 14.5.4. Die Treue eines Prototyps
  - 14.5.5. Prototypen von Benutzeroberflächen
  - 14.5.6. Bewertung von Prototypen
- 14.6. Analyse der Anforderungen
  - 14.6.1. Die Analyse der Anforderungen
  - 14.6.2. Bewährte Praktiken bei der Anforderungsanalyse
  - 14.6.3. Das Wörterbuch der Daten
  - 14.6.4. Priorisierung der Anforderungen
- 14.7. Dokumentation der Anforderungen
  - 14.7.1. Das Dokument zur Anforderungsspezifikation
  - 14.7.2. Aufbau und Inhalt eines SRS
  - 14.7.3. Dokumentation in natürlicher Sprache
  - 14.7.4. EARS: *Easy Approach to Requirements Syntax*
  - 14.7.5. Nichtfunktionale Anforderungen
  - 14.7.6. Attribute und Vorlagen in Tabellenform
  - 14.7.7. Bewährte Praktiken bei Spezifikationen
- 14.8. Validierung und Verhandlung von Anforderungen
  - 14.8.1. Validierung der Anforderungen
  - 14.8.2. Techniken zur Validierung von Anforderungen
  - 14.8.3. Verhandlung von Anforderungen
- 14.9. Modellierung und Anforderungsmanagement
  - 14.9.1. Die Modellierung der Anforderungen
  - 14.9.2. Die Nutzerperspektive
  - 14.9.3. Die Perspektive von Daten
  - 14.9.4. Die funktionale oder flussorientierte Perspektive
  - 14.9.5. Die Verhaltensperspektive
  - 14.9.6. Die Volatilität der Anforderungen
  - 14.9.7. Prozess des Anforderungsmanagements
  - 14.9.8. Werkzeuge für das Anforderungsmanagement
  - 14.9.9. Bewährte Praktiken im Anforderungsmanagement

- 14.10. Kritische Systeme und formale Spezifikation
  - 14.10.1. Kritische Systeme
  - 14.10.2. Risikoorientierte Spezifikation
  - 14.10.3. Formale Spezifikation

## Modul 15. Prozesse der Softwaretechnik

- 15.1. Rahmen der Softwaretechnik
  - 15.1.1. Eigenschaften der Software
  - 15.1.2. Die wichtigsten Prozesse der Softwaretechnik
  - 15.1.3. Prozessmodelle für die Softwareentwicklung
  - 15.1.4. Standard-Referenzrahmen für den Softwareentwicklungsprozess: die Norm ISO/IEC 12207
- 15.2. Einheitlicher Softwareentwicklungsprozess
  - 15.2.1. Einheitlicher Prozess
  - 15.2.2. Dimensionen des einheitlichen Prozesses
  - 15.2.3. Anwendungsfallorientierter Entwicklungsprozess
  - 15.2.4. Grundlegende Arbeitsabläufe einheitlicher Prozesse
- 15.3. Planung im Rahmen der agilen Softwareentwicklung
  - 15.3.1. Eigenschaften der agilen Softwareentwicklung
  - 15.3.2. Unterschiedliche Planungszeithorizonte in der agilen Entwicklung
  - 15.3.3. Rahmen für die agile Entwicklung nach Scrum und Planung von Zeithorizonten
  - 15.3.4. Anwenderberichte als Planungs- und Schätzungseinheit
  - 15.3.5. Übliche Techniken zur Ableitung einer Schätzung
  - 15.3.6. Skalen für die Interpretation von Schätzungen
  - 15.3.7. *Planning Poker*
  - 15.3.8. Gängige Planungsarten: Lieferplanung und Iterationsplanung
- 15.4. Designstile für verteilte Software und dienstorientierte Softwarearchitekturen
  - 15.4.1. Kommunikationsmodelle in verteilten Softwaresystemen
  - 15.4.2. Zwischenschicht oder *middleware*
  - 15.4.3. Architekturmuster für verteilte Systeme
  - 15.4.4. Allgemeiner Prozess zur Entwicklung von Softwarediensten
  - 15.4.5. Gestaltungsaspekte von Softwarediensten
  - 15.4.6. Zusammensetzung der Dienstleistungen
  - 15.4.7. Architektur der Webdienste
  - 15.4.8. Komponenten für Infrastruktur und SOA

- 15.5. Einführung in die modellgetriebene Softwareentwicklung
  - 15.5.1. Das Konzept des Modells
  - 15.5.2. Modellgestützte Softwareentwicklung
  - 15.5.3. MDA Rahmen für modellbasierte Entwicklung
  - 15.5.4. Elemente eines Transformationsmodells
- 15.6. Gestaltung der grafischen Benutzeroberfläche
  - 15.6.1. Grundlagen der Gestaltung der Benutzeroberfläche
  - 15.6.2. Architektonische Entwurfsmuster für interaktive Systeme: Model-View-Controller(MVC)
  - 15.6.3. Benutzererfahrung (UX *User Experience*)
  - 15.6.4. Benutzerzentriertes Design
  - 15.6.5. Prozess der Analyse und Gestaltung grafischer Benutzeroberflächen
  - 15.6.6. Benutzerfreundlichkeit der Benutzeroberfläche
  - 15.6.7. Zugänglichkeit der Benutzeroberflächen
- 15.7. Gestaltung von Web-Applikationen
  - 15.7.1. Eigenschaften von Web-Applikationen
  - 15.7.2. Benutzeroberfläche einer Web-Applikation
  - 15.7.3. Gestaltung der Navigation
  - 15.7.4. Grundlegendes Interaktionsprotokoll für Web-Applikationen
  - 15.7.5. Architekturstile für Web-Applikationen
- 15.8. Strategien und Techniken der Softwareprüfung und Faktoren der Softwarequalität
  - 15.8.1. Prüfstrategien
  - 15.8.2. Gestaltung von Testfällen
  - 15.8.3. Preis-Leistungs-Verhältnis
  - 15.8.4. Qualitätsmodelle
  - 15.8.5. ISO/IEC 25000-Normenfamilie (*SQuaRE*)
  - 15.8.6. Modell der Produktqualität (ISO 2501n)
  - 15.8.7. Modelle der Datenqualität (ISO 2501n)
  - 15.8.8. Software-Qualitätsmanagement
- 15.9. Einführung in Messgrößen in der Softwaretechnik
  - 15.9.1. Grundlegende Konzepte: Messgrößen, Metriken und Indikatoren
  - 15.9.2. Arten von Messgrößen in der Softwaretechnik
  - 15.9.3. Der Messprozess
  - 15.9.4. ISO 250215. Externe und verwendete Qualitätsmessgrößen
  - 15.9.5. Objektorientierte Messgrößen

- 15.10. Software-Wartung und Reengineering
  - 15.10.1. Wartungsprozess
  - 15.10.2. Standardrahmen für Wartungsprozesse. ISO/IEC 115,64
  - 15.10.3. Prozessmodell für das Software-Reengineering
  - 15.10.4. Reverse Engineering

## Modul 16. Systemintegration

- 16.1. Einführung in Wirtschaftsinformationssysteme
  - 16.1.1. Die Rolle der Informationssysteme
  - 16.1.2. Was ist ein Informationssystem?
  - 16.1.3. Dimensionen von Informationssystemen
  - 16.1.4. Geschäftsprozesse und Informationssysteme
  - 16.1.5. Die IS/IT-Abteilung
- 16.2. Möglichkeiten und Bedürfnisse für Informationssysteme im Unternehmen
  - 16.2.1. Organisationen und Informationssysteme
  - 16.2.2. Merkmale von Organisationen
  - 16.2.3. Auswirkungen von Informationssystemen auf das Unternehmen
  - 16.2.4. Informationssysteme als Wettbewerbsvorteil
  - 16.2.5. Einsatz von Systemen in der Unternehmensverwaltung und im Management
- 16.3. Informationssysteme und technologische Grundlagen
  - 16.3.1. Daten, Informationen und Wissen
  - 16.3.2. Informationssysteme und Technologie
  - 16.3.3. Technologie-Komponenten
  - 16.3.4. Klassifizierung und Arten von Informationssystemen
  - 16.3.5. Service- und geschäftsprozessbasierte Architekturen
  - 16.3.6. Formen der Systemintegration
- 16.4. Integrierte Systeme zur Verwaltung von Unternehmensressourcen
  - 16.4.1. Geschäftliche Anforderungen
  - 16.4.2. Ein integriertes Informationssystem für das Unternehmen
  - 16.4.3. Akquisition vs. Vorankommen
  - 16.4.4. ERP-Implementierung
  - 16.4.5. Auswirkungen auf das Management
  - 16.4.6. Wichtigste ERP-Anbieter

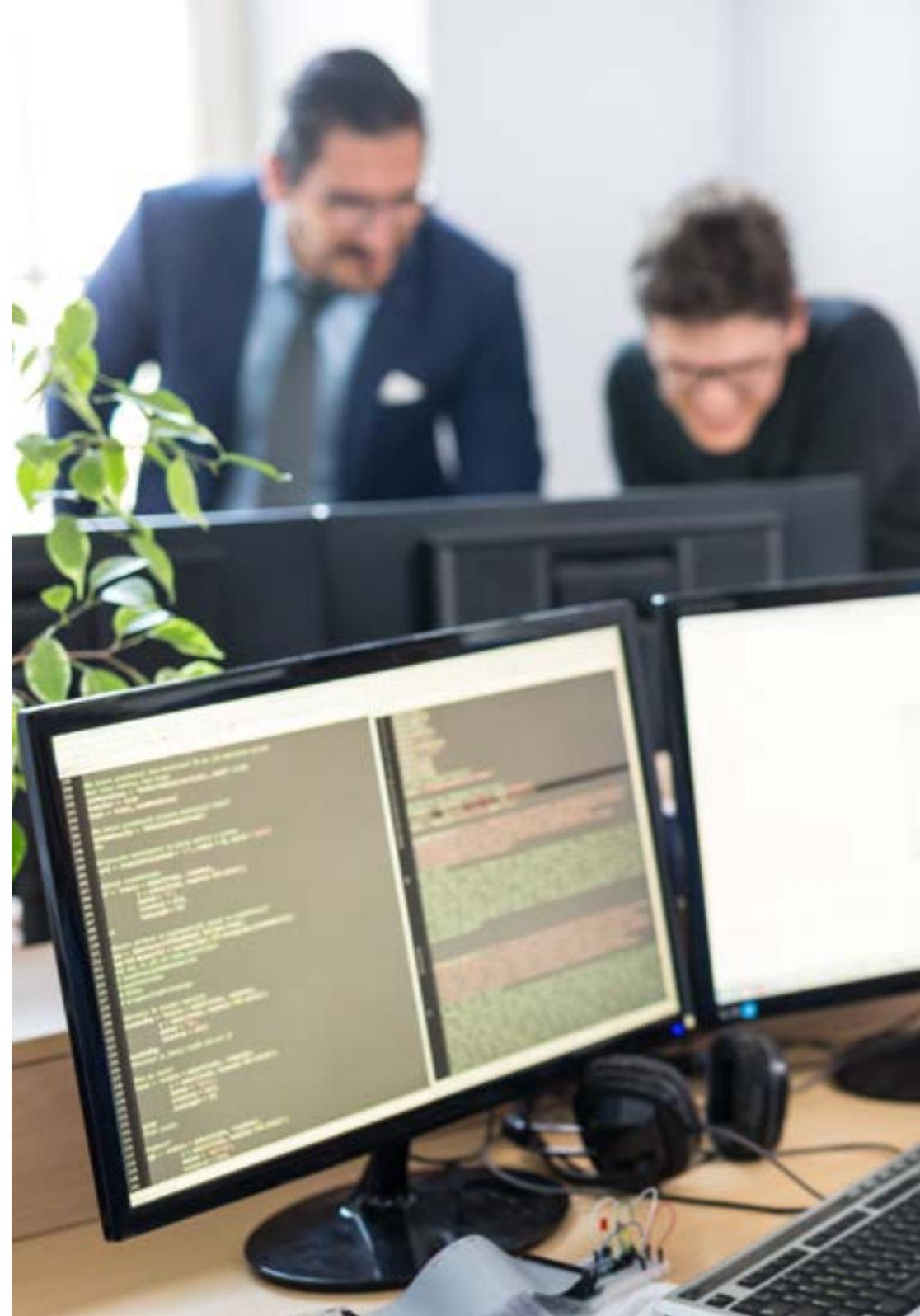
- 16.5. Informationssysteme für die Verwaltung von Lieferketten und Kundenbeziehungen
  - 16.5.1. Definition der Lieferkette
  - 16.5.2. Effektives Management der Lieferkette
  - 16.5.3. Die Rolle der Informationssysteme
  - 16.5.4. Lösungen für das Lieferkettenmanagement
  - 16.5.5. Verwaltung von Kundenbeziehungen
  - 16.5.6. Die Rolle der Informationssysteme
  - 16.5.7. Einführung eines CRM-Systems
  - 16.5.8. Kritische Erfolgsfaktoren bei der CRM-Implementierung
  - 16.5.9. CRM, e-CRM und andere Trends
- 16.6. Entscheidungsfindung für IKT-Investitionen und Planung von Informationssystemen
  - 16.6.1. Kriterien für IKT-Investitionsentscheidungen
  - 16.6.2. Verknüpfung des Projekts mit dem Management- und Geschäftsplan
  - 16.6.3. Auswirkungen auf das Management
  - 16.6.4. Neugestaltung von Geschäftsprozessen
  - 16.6.5. Entscheidung des Managements über Implementierungsmethoden
  - 16.6.6. Notwendigkeit der Planung von Informationssystemen
  - 16.6.7. Zielsetzung, Teilnehmer und Zeitplan
  - 16.6.8. Aufbau und Entwicklung des Systemplans
  - 16.6.9. Überwachung und Aktualisierung
- 16.7. Sicherheitserwägungen bei der Nutzung von IKTs
  - 16.7.1. Risikoanalyse
  - 16.7.2. Sicherheit in Informationssystemen
  - 16.7.3. Thomas-Schiene
- 16.8. Durchführbarkeit von IKT-Projekten und finanzielle Aspekte von Informationssystemprojekten
  - 16.8.1. Beschreibung und Ziele
  - 16.8.2. Teilnehmer an der Machbarkeitsstudie des Systems
  - 16.8.3. Techniken und Praktiken
  - 16.8.4. Kostenstruktur
  - 16.8.5. Finanzielle Projektion
  - 16.8.6. Budgets

- 16.9. Business Intelligence
  - 16.9.1. Was ist Business Intelligence?
  - 16.9.2. BI-Strategie und -Implementierung
  - 16.9.3. Gegenwart und Zukunft von BI
- 16.10. ISO/IEC 12207
  - 16.10.1. Was bedeutet "ISO/IEC 12207"?
  - 16.10.2. Analyse von Informationssystemen
  - 16.10.3. Entwurf eines Informationssystems
  - 16.10.4. Implementierung und Akzeptanz des Informationssystems

## Modul 17. Wiederverwendung von Software

- 17.1. Überblick über die Wiederverwendung von Software
  - 17.1.1. Was ist die Wiederverwendung von Software?
  - 17.1.2. Vor- und Nachteile der Wiederverwendung von Software
  - 17.1.3. Wichtigste Techniken der Wiederverwendung von Software
- 17.2. Einführung in die Entwurfsmuster
  - 17.2.1. Was ist ein Entwurfsmuster?
  - 17.2.2. Katalog der wichtigsten Entwurfsmuster
  - 17.2.3. Wie man Muster zur Lösung von Designproblemen einsetzt
  - 17.2.4. Wie man das beste Entwurfsmuster auswählt
- 17.3. Erzeugungsmuster
  - 17.3.1. Erzeugungsmuster
  - 17.3.2. *Abstract Factory* Muster
  - 17.3.3. Beispielimplementierung des *Abstract Factory* Musters
  - 17.3.4. *Builder* Muster
  - 17.3.5. Beispielimplementierung des *Builder* Muster
  - 17.3.6. *Abstract Factory* Muster vs. *Builder*
- 17.4. Erzeugungsmuster (II)
  - 17.4.1. *Factory Method* Muster
  - 17.4.2. *Factory Method* vs. *Abstract Factory*
  - 17.4.3. *Singleton*-Muster

- 17.5. Strukturmuster
  - 17.5.1. Strukturmuster
  - 17.5.2. Adapter Muster
  - 17.5.3. Bridge Muster
- 17.6. Strukturmuster (II)
  - 17.6.1. Composite-Muster
  - 17.6.2. Decorator Muster
- 17.7. Strukturmuster (III)
  - 17.7.1. Facade Muster
  - 17.7.2. Proxy Muster
- 17.8. Verhaltensmuster
  - 17.8.1. Konzept der Verhaltensmuster
  - 17.8.2. Verhaltensmuster: Kette der Verantwortung
  - 17.8.3. Verhaltensmuster Ordnung
- 17.9. Verhaltensmuster (II)
  - 17.9.1. Interpreter Muster oder *Interpreter*
  - 17.9.2. Iterator Muster
  - 17.9.3. Beobachter Muster
  - 17.9.4. Strategie Muster
- 17.10. Frameworks
  - 17.10.1. Konzepte des Frameworks
  - 17.10.2. Entwicklung mit Frameworks
  - 17.10.3. Model View Controller Muster
  - 17.10.4. Framework für die Gestaltung grafischer Benutzeroberflächen
  - 17.10.5. Frameworks für die Entwicklung Web-Applikationen
  - 17.10.6. Frameworks für die Verwaltung der Objektpersistenz in Datenbanken



**Modul 18. Dienstleistungen der Informationstechnologie**

- 18.1. Digitale Transformation (I)
  - 18.1.1. Business Innovation
  - 18.1.2. Management der Produktion
  - 18.1.3. Finanzielle Verwaltung
- 18.2. Digitale Transformation (II)
  - 18.2.1. Marketing
  - 18.2.2. Management des Personalwesens
  - 18.2.3. Ein integriertes Informationssystem
- 18.3. Fallstudie
  - 18.3.1. Präsentation des Unternehmens
  - 18.3.2. Methoden zur Analyse der IT-Beschaffung
  - 18.3.3. Bestimmung von Kosten, Nutzen und Risiken
  - 18.3.4. Wirtschaftliche Bewertung der Investition
- 18.4. IKT-Governance und -Verwaltung
  - 18.4.1. Definition der Governance von Informationstechnologien und -systemen
  - 18.4.2. Unterschied zwischen ITS-Governance und Management
  - 18.4.3. ITS-Governance und Managementrahmen
  - 18.4.4. Standards und die Leitung und Verwaltung von ITSs
- 18.5. IKT-Unternehmensführung
  - 18.5.1. Was ist gute Unternehmensführung?
  - 18.5.2. Hintergrund zur IKT-Governance
  - 18.5.3. ISO/IEC 318,00:2008-Norm
  - 18.5.4. Umsetzung einer guten IKT-Governance
  - 18.5.5. IKT-Governance und bewährte Praktiken
  - 18.5.6. Unternehmensführung. Überblick und Trends
- 18.6. Kontrollziele für Informations- und verwandte Technologien (COBIT)
  - 18.6.1. Rahmen für die Umsetzung
  - 18.6.2. Bereich: Planung und Organisation
  - 18.6.3. Bereich: Akquisition und Implementierung
  - 18.6.4. Bereich: Lieferung und Unterstützung
  - 18.6.5. Bereich: Überwachung und Bewertung
  - 18.6.6. Anwendung des COBIT-Leitfadens
- 18.7. Die Informationstechnologie-Infrastruktur-Bibliothek (ITIL)
  - 18.7.1. Einführung in ITIL
  - 18.7.2. Service-Strategie
  - 18.7.3. Service Design
  - 18.7.4. Dienstübergang
  - 18.7.5. Dienstbetrieb
  - 18.7.6. Serviceverbesserung
- 18.8. Das Service Management System
  - 18.8.1. Grundlegende Prinzipien von UNE-ISO/IEC 20000-1
  - 18.8.2. Die Struktur der Normenreihe ISO/IEC 20000
  - 18.8.3. Anforderungen an das Service Management System (SMS)
  - 18.8.4. Gestaltung und Umstellung neuer oder geänderter Dienste
  - 18.8.5. Prozesse der Dienstleistungserbringung
  - 18.8.6. Prozessgruppen
- 18.9. Das Software Asset Management System
  - 18.9.1. Rechtfertigung des Bedarfs
  - 18.9.2. Hintergrund
  - 18.9.3. Präsentation der Norm 19770
  - 18.9.4. Umsetzung der Verwaltung
- 18.10. Management der Geschäftskontinuität
  - 18.10.1. Plan zur Aufrechterhaltung des Geschäftsbetriebs
  - 18.10.2. Implementierung eines BCM



*Ein umfassendes Programm, das für Ihre berufliche Entwicklung von grundlegender Bedeutung sein wird*

# 06 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.

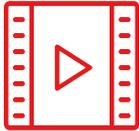
*Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihr Fachgebiet einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.*

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

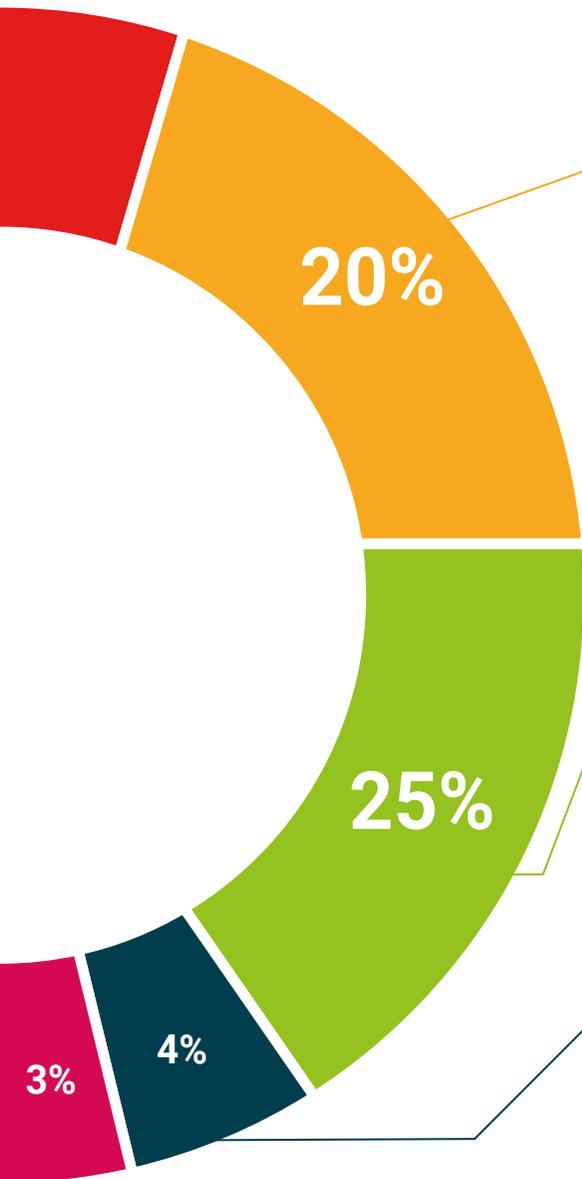
Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Praktiken und Dynamiken zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



### 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.



07

# Qualifizierung

Der Weiterbildender Masterstudiengang in 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 **Weiterbildender Masterstudiengang in 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: **Weiterbildender Masterstudiengang in Softwaretechnik**

Anzahl der offiziellen Arbeitsstunden: **3.000 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

Weiterbildender  
Masterstudiengang  
Softwaretechnik

- » Modalität: online
- » Dauer: 2 Jahre
- » Qualifizierung: TECH Technologische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

# Weiterbildender Masterstudiengang Softwaretechnik

