

Syllabus

Kursbeschreibung

Titel der Lehrveranstaltung	Software Engineering and Project
Code der Lehrveranstaltung	76273
Zusätzlicher Titel der Lehrveranstaltung	
Wissenschaftlich-disziplinärer Bereich	NN
Sprache	Deutsch
Studiengang	Bachelor in Informatik
Andere Studiengänge (gem. Lehrveranstaltung)	
Dozenten/Dozentinnen	Prof. Dr. Andrea Alexander Janes, Andrea.Janes@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/2237 Prof. Dr. Claus Pahl, Claus.Pahl@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/36376
Wissensch. Mitarbeiter/Mitarbeiterin	
Semester	Erstes Semester
Studienjahr/e	2
KP	12
Vorlesungsstunden	70
Laboratoriumsstunden	50
Stunden für individuelles Studium	180
Vorgesehene Sprechzeiten	36
Inhaltsangabe	Software Engineering: <ul style="list-style-type: none"> • Software-Lebenszyklus: Grundlagen und Methoden • Softwareprozesse und Softwareprojektmanagement

	<ul style="list-style-type: none"> • Anforderungsanalyse: Erfassung und Modellierung • Systemmodellierung und -konstruktion: UML, Entwurfsmuster • Softwaretests und -management: Grundlagen und Techniken • Aktuelle Themen der Softwareentwicklung: DevOps, Cloud, SE und KI <p>Coding Capstone:</p> <ul style="list-style-type: none"> • Projektinitiierung & Roadmap • Architektur & technisches Design • Entwicklung & Integration • Bereitstellung & Beobachtbarkeit • Abschließende Überprüfung & Retrospektive • Teamübergreifende Zusammenarbeit & Einbindung der Stakeholder
Themen der Lehrveranstaltung	<p>Modul M1 – Software Engineering – bietet eine Einführung in die verschiedenen Phasen eines Softwareentwicklungsprozesses. Es werden konkrete Tätigkeiten behandelt, die in die Bereiche Anforderungsanalyse, Systemmodellierung und Entwurf mit Schwerpunkt auf architektonischen Aspekten sowie Test und Wartung unterteilt sind. Zudem werden aktuelle relevante Themen erörtert. Die Studierenden erwerben die entsprechenden Kompetenzen in zwei separaten, gruppenorientierten und problemorientierten Projekten.</p> <p>Das Modul M2 – „Coding Capstone“ – folgt dem Software-Lebenszyklus in einem kollaborativen Umfeld, beginnend mit der Projektinitiierung, die in die Architektur- und Entwurfsphase mündet. In der Entwicklungs- und Integrationsphase werden diese Aspekte in die Programmieraktivitäten umgesetzt. Die Bereitstellung und Observability stellen sicher, dass eine Lösung veröffentlicht wird und die Leistungsüberwachung gewährleistet ist. Der Prozess endet mit einer Nachbesprechung und einer Retrospektive, um Erfolge zu reflektieren und Verbesserungsmöglichkeiten zu identifizieren.</p>
Stichwörter	Softwareentwicklung, Softwareprozesse, Softwarearchitektur, Softwareentwicklung, Bereitstellung.
Empfohlene Voraussetzungen	
Propädeutische	

Lehrveranstaltungen	
Unterrichtsform	<p>Software Engineering: Der Kurs „Software Engineering“ verbindet Vorlesungen mit Team- und/oder Einzelprojekten. Vollzeitstudierende arbeiten in Teams, während Teilzeitstudierende, die nicht an den Vorlesungen teilnehmen, sich für Einzelprojekte entscheiden können. Die Projekte konzentrieren sich auf Anwendungsentwicklungen größeren Umfangs, die in einem team- und projektbasierten, kooperativen Arbeitsumfeld behandelt werden.</p> <p>Coding Capstone: Der Kurs „Coding Capstone“ ist als projektbasiertes Praktikum konzipiert, in dem die Studierenden einzeln oder in Teams an der Konzeption, Entwicklung, Implementierung und Bewertung eines Softwaresystems arbeiten. Der Kurs verbindet Vorlesungen und technische Einweisungen mit betreuten Entwicklungsaufgaben, technischen Überprüfungen, Meilensteinpräsentationen und Diskussionen mit den Projektbeteiligten und fördert so die Zusammenarbeit sowie die praktische Anwendung von Methoden des Software-Engineering in realistischen Projektsituationen.</p>
Anwesenheitspflicht	<p>Die Teilnahme ist nicht verpflichtend, wird jedoch dringend empfohlen. Studierende, die nicht am Unterricht teilnehmen, sollten sich zu Beginn des Kurses mit dem Dozenten in Verbindung setzen, um die Modalitäten des Selbststudiums zu vereinbaren.</p>
Spezifische Bildungsziele und erwartete Lernergebnisse	<p>Wissen und Verständnis:</p> <ul style="list-style-type: none"> - D1.2: Die grundlegenden Prinzipien der Programmierung im Detail kennen. - D1.7: Über fundierte Kenntnisse der wichtigsten grundlegenden Techniken und Methoden der Software-Konzeption, -Entwicklung und -Wartung verfügen <p>Anwendung von Wissen und Verständnis:</p> <ul style="list-style-type: none"> - D2.2: In der Lage sein, kleine und mittelgroße Programme unter Verwendung verschiedener Programmiersprachen und -paradigmen zu entwickeln. - D2.3: Probleme mithilfe von Programmiermethodiken lösen können. - D2.5: Das eigene Wissen auf die Analyse, den Entwurf, die Entwicklung und das Testen von Informationssystemen anwenden

können, die vorgegebene Anforderungen erfüllen.

- D2.8: Typische Probleme der Informatik auf der Grundlage von Software-Engineering-Methodiken lösen können, wie z. B. die Definition von Anforderungen, die Analyse möglicher Lösungsansätze, die Auswahl der am besten geeigneten Methoden und Werkzeuge sowie deren Anwendung.
- D2.9: Die Qualität von Informationssystemen bewerten und kritische Aspekte identifizieren können.
- D2.16: Kleine Projektteams koordinieren und mit den Mitgliedern der Gruppe interagieren können.

Urteilsvermögen:

- D3.1: Nützliche Daten sammeln und interpretieren sowie Informationssysteme und deren Anwendbarkeit beurteilen können.
- D3.2: In der Lage sein, entsprechend dem eigenen Wissens- und Verständnisstand selbstständig zu arbeiten.
- D3.3: In der Lage sein, Verantwortung für die Entwicklung von Projekten oder die IT-Beratung zu übernehmen.

Kommunikationsfähigkeiten:

- D4.1: In der Lage sein, eine der drei Sprachen Englisch, Italienisch und Deutsch zu verwenden und Fachbegriffe sowie die Kommunikation angemessen einzusetzen.
- D4.2: In der Lage sein, mit einem Kunden über die Definition der Voraussetzungen und Funktionen von Informationssystemen zu verhandeln.
- D4.3: In der Lage sein, technische Dokumentationen zu strukturieren und zu verfassen.
- D4.4: In der Lage sein, im Team an der Realisierung von IT-Systemen mitzuarbeiten.

Lernkompetenzen:

- D5.1: Lernfähigkeiten entwickelt haben, um ein weiterführendes Studium mit einem hohen Maß an Selbstständigkeit zu absolvieren.
- D5.2: Lernkompetenzen erworben haben, die es ermöglichen, Projektaktivitäten in Unternehmen, öffentlichen Einrichtungen oder in verteilten Entwicklungsgemeinschaften durchzuführen.
- D5.3: In der Lage sein, mit der raschen technologischen Entwicklung Schritt zu halten und sich in modernste IT-Technologien sowie innovative Aspekte von Informationssystemen

	der neuesten Generation einzuarbeiten.
Spezifisches Bildungsziel und erwartete Lernergebnisse (zusätzliche Informationen)	
Art der Prüfung	<p>Die Bewertungsstruktur unterscheidet sich bei den beiden in diesem Modul enthaltenen Kursen. „Software Engineering“ wird anhand eines Projekts (40 %) und einer schriftlichen Prüfung (60 %) bewertet, während „Coding Capstone“ anhand eines Projekts (60 %) und einer mündlichen Prüfung (40 %) bewertet wird. Sowohl teilnehmende als auch nicht teilnehmende Studierende müssen ein Entwicklungsprojekt absolvieren, entweder einzeln oder in Teams. In der mündlichen Prüfung werden sowohl das theoretische Verständnis der Studierenden als auch ihre Fähigkeit, die Projektergebnisse zu erörtern und zu reflektieren, bewertet. Die Studierenden müssen die Projektkomponente mit einer ausreichenden Note abgeschlossen haben, bevor sie zur mündlichen Prüfung zugelassen werden, und beide Bewertungskomponenten müssen bestanden werden, um das Modul erfolgreich abzuschließen. Ein positiv bewertetes Projekt bleibt für drei Prüfungssessionen gültig.</p> <p>Der Kurs Software Engineering vermittelt den Studierenden fundierte Kenntnisse über die Prinzipien der Bereiche Software-Lebenszyklus, Softwareprozesse, Anforderungsanalyse, Systemmodellierung, Entwurfsmuster, Softwaretests sowie moderne Themen wie DevOps, Cloud Computing und KI-gestützte Softwareentwicklung. Diese Themen tragen dazu bei, die Fähigkeit zu erwerben, Softwaresysteme unter Verwendung geeigneter Methoden und Werkzeuge zu entwerfen, zu entwickeln, zu testen und zu bewerten (D2.2, D2.3, D2.5, D2.8, D2.9). Durch Modellierungsaufgaben, Dokumentation und Anforderungsanalysen stärken die Studierenden zudem ihre Kommunikations- und technischen Schreibfähigkeiten (D4.2, D4.3, D4.4), während die Auseinandersetzung mit neuen Technologien das selbstständige Lernen und die kontinuierliche berufliche Weiterentwicklung fördert (D5.1, D5.3).</p> <p>Der Kurs „Coding Capstone“ konzentriert sich auf die praktische</p>

	<p>Umsetzung von Softwaresystemen, von der Projektinitiierung über Architektur und technisches Design bis hin zu Entwicklung, Bereitstellung, Observability, Retrospektiven und der Zusammenarbeit mit Stakeholdern. Durch die Arbeit an realistischen Entwicklungsprojekten wenden die Studierenden Methoden des Software-Engineering in kollaborativen Umgebungen an und stärken ihre Fähigkeit, Softwarelösungen selbstständig zu entwerfen und zu implementieren (D2.2, D2.5, D2.8, D3.2, D3.3). Teamkoordination, teamübergreifende Zusammenarbeit und die Einbindung von Stakeholdern fördern Kommunikations- und Teamfähigkeiten (D2.16, D4.1, D4.2, D4.4), während Bereitstellung, Beobachtbarkeit und Retrospektiven die Fähigkeit entwickeln, die Systemqualität zu bewerten und Entwicklungsprozesse kritisch zu reflektieren (D2.9, D3.1). Die projektbasierte Struktur fördert zudem das selbstgesteuerte Lernen und bereitet die Studierenden darauf vor, in der industriellen und verteilten Softwareentwicklung effektiv zu arbeiten (D5.1, D5.2, D5.3). Beide Kurse tragen dazu bei, die Kenntnisse in den Bereichen Softwareentwicklung und Software-Engineering zu vertiefen (D1.2, D1.7).</p>
<p>Bewertungskriterien</p>	<p>Modul 1: Software Engineering Für teilnehmende Studierende setzt sich die Note aus (i) der Laborprüfung (Gewichtung: 40 %) und (ii) der schriftlichen Abschlussprüfung (Gewichtung: 60 %) zusammen. Nicht teilnehmende Studierende sollten den Abgabeterminen für die Laborprüfungen folgen; die Note wird auf die gleiche Weise berechnet.</p> <p>Modul 2: Coding Capstone Für teilnehmende Studierende setzt sich die Note aus (i) der Laborprüfung (Gewichtung: 60 %) und (ii) der schriftlichen Abschlussprüfung (Gewichtung: 40 %) zusammen. Nicht teilnehmende Studierende sollten den Zeitplan für die Laborprüfungen einhalten; die Note wird auf die gleiche Weise berechnet.</p> <p>Studierende müssen beide Module bestehen, um den Kurs zu bestehen. Die Abschlussnote ist der Durchschnittswert der Noten aus beiden Modulen.</p>

Pflichtliteratur	<p>Unterrichtsmaterialien, Vorlesungsfolien und weitere Begleitmaterialien werden im Rahmen der Vorlesungen zur Verfügung gestellt.</p> <p>Software Engineering: Sommerville, I. Software Engineering. 10th Edition. Pearson</p>
Weiterführende Literatur	
Weitere Informationen	
Ziele für nachhaltige Entwicklung (SDGs)	Industrie, Innovation und Infrastruktur, Hochwertige Bildung

Kursmodul

Titel des Bestandteils der Lehrveranstaltung	Software Engineering and Project M1: Software Engineering
Code der Lehrveranstaltung	76273A
Wissenschaftlich-disziplinärer Bereich	INFO-01/A
Sprache	Deutsch
Dozenten/Dozentinnen	<p>Prof. Dr. Claus Pahl, Claus.Pahl@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/36376</p>
Wissensch. Mitarbeiter/Mitarbeiterin	
Semester	Erstes Semester
KP	6
Verantwortliche/r Dozent/in	
Vorlesungsstunden	40
Laboratoriumsstunden	20
Stunden für individuelles Studium	90
Vorgesehene Sprechzeiten	18
Inhaltsangabe	<ul style="list-style-type: none"> • Software-Lebenszyklus: Grundlagen und Methoden • Softwareprozesse und Softwareprojektmanagement

	<ul style="list-style-type: none"> • Anforderungsanalyse: Erfassung und Modellierung • Systemmodellierung und -konstruktion: UML, Entwurfsmuster • Softwaretests und -management: Grundlagen und Techniken • Aktuelle Themen der Softwareentwicklung: DevOps, Cloud, SE und KI
Themen der Lehrveranstaltung	
Unterrichtsform	Vorlesungen, Übungen; Gruppen- und/oder Einzelprojekte.
Pfichtliteratur	<ul style="list-style-type: none"> • Sommerville, I. Software Engineering. 10th Edition. Pearson.
Weiterführende Literatur	

Kursmodul

Titel des Bestandteils der Lehrveranstaltung	Software Engineering and Project M2: Coding Capstone
Code der Lehrveranstaltung	76273B
Wissenschaftlich-disziplinärer Bereich	INFO-01/A
Sprache	Deutsch
Dozenten/Dozentinnen	Prof. Dr. Andrea Alexander Janes, Andrea.Janes@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/2237
Wissensch. Mitarbeiter/Mitarbeiterin	
Semester	Erstes Semester
KP	6
Verantwortliche/r Dozent/in	
Vorlesungsstunden	30
Laboratoriumsstunden	30
Stunden für individuelles Studium	90
Vorgesehene Sprechzeiten	18
Inhaltsangabe	<ul style="list-style-type: none"> • Projektstart und Roadmap • Architektur und technisches Design

	<ul style="list-style-type: none"> • Entwicklung und Integration • Bereitstellung und Überwachbarkeit • Abschließende Überprüfung und Retrospektive • Teamübergreifende Zusammenarbeit und Einbindung der Stakeholder
Themen der Lehrveranstaltung	
Unterrichtsform	<p>Der Coding-Capstone-Kurs ist als projektbasiertes Labor organisiert, in dem die Studierenden einzeln oder in Teams an der Konzeption, Entwicklung, Bereitstellung und Evaluierung eines Softwaresystems arbeiten. Der Kurs kombiniert Frontalunterricht und technische Briefings mit betreuten Entwicklungsaktivitäten, technischen Reviews, Meilensteinpräsentationen und Diskussionen mit Stakeholdern und fördert dadurch die Zusammenarbeit sowie die praktische Anwendung von Software-Engineering-Methoden in realistischen Projektszenarien.</p>
Pfichtliteratur	<p>Lehrmaterialien, Vorlesungsfolien und zusätzliche unterstützende Unterlagen werden während der Lehrveranstaltungen zur Verfügung gestellt.</p>
Weiterführende Literatur	