

Modulhandbuch

Bachelor »Code & Context«

TH Köln

11. Juni 2026

Version: 1024579 / Thu Jun 11 15:45:38 2026

Inhaltsverzeichnis

Studiengangbeschreibung

Mit »Code & Context« betreiben wir einen Bachelorstudiengang über die klassischen Fachgrenzen von Informatik, Design, und BWL hinweg. Die TH Köln bringt dafür als besondere Eigenschaft die Vielfalt ihres Fächerspektrums ein.

Die im Studiengang durchgeführte explorative und praxisnahe Informatikausbildung verfolgt einen ganzheitlichen Anspruch, der Programmieren ebenso einschließt wie integratives Design, das Denken in digitalen Geschäftsmodellen und den Blick auf gesellschaftliche und ökonomische Konsequenzen des eigenen Handelns. Zu den vermittelten Kompetenzen zählt ausdrücklich auch die unternehmerische Qualifikation der Studierenden durch Entrepreneurship Education (Technische Hochschule Köln, 2018).

Damit ist »Code & Context« ein interdisziplinärer, werkstatorientierter Informatik-Bachelorstudiengang, der Studierende in besonderer Weise auf die gesellschaftlichen, technischen und ökonomischen Anforderungen der Digitalisierung vorbereitet. Absolvent*innen werden dazu ausgebildet, den Herausforderungen einer sich rapide und nachhaltig wandelnden Gesellschaft und Wirtschaft zu begegnen.

Dieser Anspruch findet sich im Curriculum in vier Handlungsfeldern wider: Designing Futures, Creating Impact, Agile Coding und Developing Things.

Studienverlaufsplan

1. Semester	2. Semester	3. Semester	4. Semester	5. Semester	Optionales Semester	Finales Semester
Coding Software 1 Coding Essentials 1 Coding Essentials 2 6 CP	Coding Software 2 Agile Coding Practices Agile Team Practices 6 CP	Coding Software 3 Application Design 3 CP Designing Futures 3 Open Design 3 CP	Wahlpflicht 1 Zwei Wahlpflichtangebote aus den Handlungsfeldern 6 CP	Wahlpflicht 2 Zwei Wahlpflichtangebote aus den Handlungsfeldern 6 CP	Semester Abroad / Self-initiated projekt / Internship 30 CP	Project Launch 15 CP Bachelor Thesis 12 CP Bachelor Colloqu. 3 CP
Designing Futures 1 Prototyping Things Users and Situated Action 6 CP	Designing Futures 2 Generative Gestaltung Interaction Design 6 CP	Developing Things 2 Interactive Applications Experience Engineering 6 CP	Developing Things 3 Internet of Things Making of Things 6 CP	Creating Impact 2 Business Planning 1 Business Planning 2 6 CP		
Project Code Project Coding Essentials 3 12 CP	Developing Things 1 Things with Agency Designing Context 6 CP Project Design 6 CP	Creating Impact 1 Design Thinking Entrepreneurial Thinking 6 CP Project Impact 6 CP	Project Make 1 12 CP	Project Make 2 12 CP		
Community & Reflection 1 6 CP	Community & Reflection 2 6 CP	Community & Reflection 3 6 CP	Community & Reflection 4 6 CP	Community & Reflection 5 6 CP		

Alternativer Studienverlaufsplan

Modulname	Kürzel	CP	Modulverantwortlicher
1. Fachsemester			
Coding Software 1 enthält: Coding Essentials 1, Coding Essentials 2,	CS1	6	Prof. Dr. Frank Schimmel
Project Code enthält: Coding Essentials 3, Project Code: Project,	PC	12	Prof. Dr. Christian Faubel
Summe CP		18	
2. Fachsemester			
Community & Reflection 1	CR1	6	Prof. Dr. Frank Schimmel
Developing Things 1 enthält: Designing Context, Things with Agency,	DT1	6	Prof. Dr. Christian Faubel
Project Design	PD	6	Prof. Dr. Laura Popplow
Summe CP		18	
3. Fachsemester			
Designing Futures 1 enthält: Prototyping Things, Users and Situated Action,	DF1	6	Prof. Dr. Laura Popplow
Community & Reflection 2	CR2	6	Prof. Dr. Frank Schimmel
Creating Impact 1 enthält: Design Thinking, Entrepreneurial Thinking,	CI1	6	Prof. Dr. Markus Linden
Project Impact	PI	6	Prof. Dr. Jonas Schild
Summe CP		24	
4. Fachsemester			
Coding Software 2 enthält: Agile Coding Practices, Agile Team Practices,	CS2	6	Prof. Dr. Frank Schimmel
Designing Futures 2 enthält: Generative Gestaltung, Interaction Design,	DF2	6	Prof. Dr. Laura Popplow
Community & Reflection 3	CR3	6	Prof. Dr. Frank Schimmel
Summe CP		18	
5. Fachsemester			
Developing Things 2 enthält: Experience Engineering, Interactive Applications,	DT2	6	Prof. Dr. Jonas Schild
Coding Software 3 enthält: Application Design,	CS3	3	Prof. Dr. Stefan Bente
Designing Futures 3 enthält: Open Design,	DF3	3	Prof. Nina Juric
Community & Reflection 4	CR4	6	Prof. Dr. Frank Schimmel
Creating Impact 2 enthält: Business Planning 1, Business Planning 2,	CI2	6	Prof. Dr. Markus Linden
Summe CP		24	
6. Fachsemester			
Developing Things 3 enthält: Internet of Things, Making of Things,	DT3	6	Prof. Dr. Christian Faubel
Wahlpflichtmodul 1 enthält: Business Economics, Design As A Language, DevOps, Human-Computer Impact, Trend and Market Research,	WPF1	6	Prof. Dr. Jonas Schild
Project Make 1	PM1	12	Prof. Dr. Christian Faubel
Summe CP		24	
7. Fachsemester			
Wahlpflichtmodul 2 enthält: Advanced Coding, Blended International Project, Machine Learning Basics, Microservice Architectures, Open Ecosystems,	WPF2	6	Prof. Dr. Jonas Schild
Project Make 2	PM2	12	Prof. Dr. Jonas Schild
Community & Reflection 5	CR5	6	Prof. Dr. Frank Schimmel
Summe CP		24	
8. Fachsemester			
Project Launch	PL	15	alle Dozent*innen im Studiengang
Bachelorarbeit	BA	12	alle Dozent*innen im Studiengang
Bachelor Kolloquium	BK	3	alle Dozent*innen im Studiengang
Summe CP		30	

Absolvent*innenprofil

Der Studiengang bildet fachlich versierte, kreative Persönlichkeiten mit hohem Gestaltungswillen, technischem Verständnis und sozialem Verantwortungsbewusstsein aus. »Coding« bedeutet für die Studierenden eine Auseinandersetzung mit aktuellen Digitalisierungsthemen (wie beispielsweise Künstliche Intelligenz, Arbeitswelten der Zukunft, Coded Society) unter Anwendung von agilen Ansätzen, Design Thinking und Prototyping und mit dem Einsatz von Hardware und Software als Material.

Ein Leitbild für das Profil der Absolvent*innen lässt sich wie folgt formulieren:

»Code & Context« bildet neugierige Gestalter*innen und kritische Akteurinnen im Spannungsfeld zwischen Coding und Design aus. Als technisch handlungsfähige soziale Innovatorinnen begegnen sie den Herausforderungen der sich durch digitale Phänomene rapide wandelnden Gesellschaft, Kultur und Wirtschaft.

Durch ihre ganzheitliche Perspektive auf Hardware, Software und deren Wirkungsweisen können sie Potenziale über Fachgrenzen hinweg erkennen und bewerten, um somit zukunftsrelevante Szenarien, Produkte und Ökosysteme zu antizipieren und zu gestalten.

Als Absolvent*innen arbeiten sie in heterogenen Teams und erschließen sich selbstständig Methoden, Technologien und Tools, um neue Ideen vom Prototypen bis zum Minimum Viable Product zu entwickeln. Ihre kreative, experimentelle und agile Arbeitsweise ermöglicht es ihnen, in diversen soziokulturellen Kontexten unternehmerische Modelle zu realisieren.

- mit kritischem Denken, lösungsorientierten Ansätzen und ausgeprägter Persönlichkeit einen relevanten Beitrag zur zukünftigen Coded Society zu leisten,
- gesellschaftliche, ethische und wirtschaftliche Potenziale für digitale Entwicklungen zu erkennen, zu bewerten und zu antizipieren, fundiertes technisches Informatikwissen mit interdisziplinären, sozio-politischen, philosophischen und ästhetischen Schnittstellen zu verbinden und in konzeptstarke Entwurfsprozesse zu überführen,
- theoretische und methodische Kompetenz aus den drei Disziplinen Informatik, Design und BWL bei Problemdefinition einzusetzen und Lösungen auf der Basis theoretischer oder empirischer Grundlagen zu reflektieren, durch ein gutes Verständnis für soziokulturelle Phänomene, Zusammenhänge, Gefahren und Chancen über Fachgrenzen hinweg zu erkennen und zu bewerten,
- problemlos zukunftsrelevante Szenarien darzustellen, indem sie spekulativ die Bedürfnisse des Nutzers von morgen wahrnehmen, Produkte aus verschiedenen Blickwinkeln zu betrachten, zu definieren und neu zu verhandeln — ob Software, Hardware, Objekt, Plattform, Maschine, Service oder Prozess,
- mit Leichtigkeit sowohl im Netzwerk mit Spezialist*innen aus verschiedensten Disziplinen zu interagieren, als auch schnell und selbstständig die Kompetenzen, Technologien und Tools zu erlernen, um die prototypische Realisierung ihrer eigenen Ideen voranzutreiben und in frühen Phasen unternehmerisch zu betreiben,
- in beruflichen und lebensweltlichen Kontexten reflektiert zu handeln,
- selbständig und unternehmerisch zu agieren, Geschäftsmodelle bzw. Anwendungsfelder zu finden und die neuen Dimensionen von technologischer Produktgestaltung in einer ganzheitlichen Sicht, vom Prototypen bis zum Minimum Viable Product, zu entwickeln,
- es durch eine kreative Arbeitsweise zu ermöglichen, ihre Fähigkeiten auch abseits der herkömmlichen Wege zu nutzen und so in unterschiedlichsten Umgebungen zu sozialen Innovator*innen zu werden,
- mit kulturell unterschiedlichen Personengruppen zu kommunizieren, zusammenzuarbeiten und Konflikte zu lösen,

-
- wissenschaftlich, selbstständig, reflektiert und eigenverantwortlich zu denken und zu arbeiten.

Module

Übersicht über die Module

Alle Module können innerhalb eines Semesters abgeschlossen werden.

Modul	Coding Software	Developing Things	Designing Futures	Creating Impact	SDA	DRSS	IEE	PROD	AGILE	COMM	TSCE	CREA	ENSI	SOSE	Sem.	CP
Designing Futures 1			x				x	x				x			1	6
Coding Software 1	x				x										1	6
Project Code	x				x			x		x				x	1	12
Community & Reflection 1														x	1	6
Coding Software 2	x				x				x	x					2	6
Developing Things 1		x				x	x				x	x			2	6
Designing Futures 2			x		x		x					x			2	6
Project Design					x	x	x		x	x	x		x	x	2	6
Community & Reflection 2														x	2	6
Creating Impact 1				x					x				x		3	6
Developing Things 2		x			x	x	x	x							3	6
Coding Software 3	x				x				x						3	3
Designing Futures 3			x				x				x	x			3	3
Project Impact					x		x	x		x		x		x	3	6
Community & Reflection 3														x	3	6
Developing Things 3		x			x	x									4	6
Wahlpflichtmodul 1	x	x	x	x	x	x	x	x	x	x	x		x	x	4	6
Project Make 1					x	x	x	x		x		x		x	4	12
Community & Reflection 4														x	4	6
Creating Impact 2				x						x			x		5	6
Wahlpflichtmodul 2	x	x	x	x		x	x	x			x	x	x		5	6
Project Make 2							x	x		x	x	x	x	x	5	12
Community & Reflection 5														x	5	6
Optionales Praxissemester					x	x	x	x	x	x	x	x	x	x	0/6	30
Project Launch					x	x	x	x	x	x	x	x	x	x	6/7	15
Bachelorarbeit					x	x	x	x	x	x	x	x	x	x	6/7	12
Bachelor Kolloquium					x	x	x	x	x	x	x	x	x	x	6/7	3

Tabelle 0.1: Vernetzung der Module mit den Handlungsfeldern und Kompetenzclustern.

Modul »Designing Futures 1« (DF1)

Verantwortlich: Prof. Dr. Laura Popplow

Studiensemester: 1

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: IEE / PROD / CREA

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Absolvent*innen dieses Moduls beherrschen grundlegende Methoden des integrierten Designs im digitalen Kontext. Sie konzeptionieren radikal innovative Anwendungen, indem sie

- Szenarien für zukünftige Produkte und Dienstleistungen unter Berücksichtigung gesellschaftlicher, kultureller und ästhetischer Kontexte entwickeln und bewerten,
- in Prototyping und Modellbau Konzepten und Ideen materielle Form geben und zugleich neue Konzepte und Ideen im Dialog mit Werkzeug und Material entwickeln,
- sowie Methoden und Praktiken des User Research und Testing exemplarisch einsetzen,

um später in interdisziplinären Projekten gestalterische Experimente und spekulative Exploration mit Nutzer*innenbezug durchzuführen.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
DF11	Prototyping Things	40h / 50h	IEE / PROD	3	unbenotet - strukturiertes Feedback
DF12	Users and Situated Action	40h / 50h	IEE / CREA	3	unbenotet - strukturiertes Feedback

Modul »Coding Software 1« (CS1)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 1

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: SDA

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Studierende können in Einzelarbeit auf Basis gegebener Anforderungen eine kleine bis mittelgroße Applikation kodieren, indem sie algorithmisches Denken, Ausdrucksmittel von Programmiersprachen im Kleinen wie im Großen, sowie gängige Libraries und Frameworks anwenden, um später einfache bis mittel-komplexe Entwicklungsaufgaben übernehmen zu können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
CS11	Coding Essentials 1	40h / 50h	SDA	3	unbenotet - strukturiertes Feedback
CS12	Coding Essentials 2	40h / 50h	SDA	3	unbenotet - strukturiertes Feedback

Modul »Project Code« (PC)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 1

Häufigkeit: jährlich

Kreditpunkte: 12 ECTS

Arbeitsaufwand: 90h Kontaktzeit / 270h Selbstlernzeit

Empfohlene Voraussetzungen: Coding Software 1

Kompetenzcluster: SDA / PROD / COMM / SOSE

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Studierende können auf der Basis grundlegender Programmierkenntnisse in objektorientierter Programmierung und einem grundlegenden Verständnis von Client-Server Architekturen eine Anwendung mit Webkomponenten programmieren, indem sie das

- die in den Kursen erworbenen Kenntnisse durch praktische Anwendung und eigene Erfahrungen weiter vertiefen und verinnerlichen,
- erste Erfahrungen in der Teamdynamik bei größeren Aufgabenstellungen sammeln, um dies in spätere Module einbringen können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
PC1	Coding Essentials 3	40h / 50h	SDA	3	unbenotet - strukturiertes Feedback
PC2	Project Code: Project	50h / 220h	SDA / PROD / COMM / SOSE	9	unbenotet - strukturiertes Feedback

Modul »Community & Reflection 1« (CR1)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 1

Häufigkeit: fortwährend

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 20h Kontaktzeit / 160h Selbstlernzeit

Kompetenzcluster: SOSE

Learning Outcome / Qualifikationsziele des Moduls

Teilnehmer*innen der »Community & Reflection«-Module können an der Organisation, Selbstverwaltung, Kommunikation und Weiterentwicklung des Studiengangs teilhaben, in dem sie:

- organisatorische und institutionelle Strukturen, Abläufe und Zusammenhänge kennenlernen und ein kritisches Verständnis für selbige entwickeln
- Projekte und Events im Team planen und durchführen
- Aufgabenstellungen in heterogenen Teams eigenständig entwickeln
- Teamprozesse lösungsorientiert moderieren und Konflikte managen
- neuen Themen (Technologien, Methoden, Werkzeuge) erschließen und das neue Wissen bzw. neue Kompetenzen an Peers vermitteln
- Arbeitsgemeinschaften und Kommunikationsformate weiterführen und entwickeln
- Verantwortung für das eigene Studiumfeld übernehmen

um später im beruflichen und gesellschaftlichen Kontext professionell handeln zu können und das Arbeitsumfeld als auch die Disziplin als solche zu hinterfragen und mitgestalten zu können.

Inhalte / Aufbau des Moduls

- Formate für Arbeitsgemeinschaften und Kommunikation:
 - Studieninfo & Aufnahme
 - Raum
 - Code of Conduct
 - Internationalisierung
 - Öffentlichkeitsarbeit
 - Coding & Tool Support
 - Presentations & Events
 - Iteration
- Feedback und Teilnahme an der Teaching Analysis Poll (TAP)
- Präsentationen

Modul »Coding Software 2« (CS2)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 2

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Empfohlene Voraussetzungen: Coding Software 1 / Project Code

Kompetenzcluster: AGILE / SDA / COMM

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Studierende können im Team realitätsnahe digitale Produkte gestalten, deren Anforderungen sich dynamisch verändern und aus realen Quellen stammen, indem sie

- agile Coding-Praktiken begründet auswählen und einsetzen,
- Prinzipien für nachhaltige Code-Qualität anwenden,
- in Zusammenarbeit mit Benutzer*innen bzw. Kund*innen häufig funktionierenden Software ausliefern, sowie
- im Team über ihre Arbeitsweise reflektieren und diese anpassen,

um später Projekte in Ende-zu-Ende-Verantwortung im Team umsetzen zu können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
CS21	Agile Coding Practices	40h / 50h	AGILE / SDA	3	unbenotet - strukturiertes Feedback
CS22	Agile Team Practices	40h / 50h	AGILE / SDA / COMM	3	unbenotet - strukturiertes Feedback

Modul »Developing Things 1« (DT1)

Verantwortlich: Prof. Dr. Christian Faubel

Studiensemester: 2

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Empfohlene Voraussetzungen: Coding Software 1 / Coding Software 2 / Designing Futures 1

Kompetenzcluster: IEE / TSCE / DRSS / CREA

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Student*innen entwickeln dinglich, digitale Geräte und erkunden und entwickeln mit Methoden des Participatory Design soziale Kontexte, in dem sie

- lernen experimentell zu arbeiten um iterativ in einer Folge von Versuch, Irrtum und Erkenntnis testbare Prototypen zu bauen,
- lernen eigene Fehler zu machen, zu benennen und mit ihnen umzugehen,
- lernen das Prinzip des Prototyping von Testen, Versuch, Irrtum und Erkenntnis auf die Gestaltung sozialer Kontext anzuwenden.

um später spielerisch und risikobereit sinnhafte in sozialen Kontexten verankerte innovative Prototypen digital dinglicher Produkte entwickeln zu können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
DT11	Things with Agency	40h / 50h	DRSS / CREA / IEE	3	unbenotet - strukturiertes Feedback
DT12	Designing Context	40h / 50h	IEE / TSCE	3	unbenotet - strukturiertes Feedback

Modul »Designing Futures 2« (DF2)

Verantwortlich: Prof. Dr. Laura Popplow

Studiensemester: 2

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: CREA / SDA / IEE

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Absolvent*innen dieses Moduls können Interaktionen und digitale Handlungsfelder gestalten und reflektieren, indem sie

- Ideen und Szenarien spekulativ entwickeln, darstellen und kritisch hinterfragen,
- interaktive Systeme unter Berücksichtigung der Konventionen und Geschichte des Interaktionsdesigns und unter Einsatz experimenteller Designmethoden gestalten,
- sowie Programmierung als künstlerisch-gestalterisches Feld begreifen und Methoden des Creative Coding und der generativen Gestaltung im Spannungsfeld von Design und Kunst einsetzen,
- und dabei Methoden anwenden, die ihnen erlauben, die Wahrnehmung, Gewohnheiten und kontextuelle Situierung ihrer Nutzer*innen zu berücksichtigen und neu auszurichten,

um später interaktive Produkte und ihre Wirkung ganzheitlich gestalten zu können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
DF21	Generative Gestaltung	40h / 50h	CREA / SDA	3	unbenotet - strukturiertes Feedback
DF22	Interaction Design	40h / 50h	IEE / CREA	3	unbenotet - strukturiertes Feedback

Modul »Project Design« (PD)

Verantwortlich: Prof. Dr. Laura Popplow

Studiensemester: 2

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 30h Kontaktzeit / 150h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / AGILE / ENSI / COMM / TSCE / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Studierende können problem-basierte Szenarien in sozialen und ökonomischen Kontexten analysieren und explorieren. Durch eine agile und Realisierung realitätsnaher digitaler "Minimum Viable Products (MVP)" können die Studierenden Lösungen für dynamische Anforderungen finden, indem sie

- sich in selbstorganisierten Teams zusammenfinden,
- dabei agile Methoden für Teamorganisation, Qualitätssicherung und Aufgabenplanung anwenden,
- die Kompetenzen der begleitenden Kurse (insb. Entwicklung einer sinnvollen Produktidee sowie Coding und Architektur für bei der Implementierung) anwenden und in einen Zusammenhang bringen,
- eine Aufgabenstellung (die auf die begleitenden Kurse zugeschnitten ist) selbstständig bearbeiten, in geeignete Teilaufgaben zerlegen und diese im Team verteilen,

damit sie

- die in den Kursen erworbenen Kenntnisse durch praktische Anwendung und eigene Erfahrungen weiter vertiefen und verinnerlichen,
- Erfahrungen in der Teamdynamik bei der Lösung größeren Aufgabenstellungen mit agilen Methoden sammeln,
- und dies in spätere Module einbringen können.

Inhalt

Es werden jedes Jahr neue

Lehr- und Lernformen

Seminaristisch, Workshops, Projektarbeit

Material / Ressourcen

- Lehrmaterialien werden auf TH-Spaces bereitgestellt

Modul »Community & Reflection 2« (CR2)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 2

Häufigkeit: fortwährend

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 20h Kontaktzeit / 160h Selbstlernzeit

Kompetenzcluster: SOSE

Learning Outcome / Qualifikationsziele des Moduls

Teilnehmer*innen der »Community & Reflection«-Module können an der Organisation, Selbstverwaltung, Kommunikation und Weiterentwicklung des Studiengangs teilhaben, in dem sie:

- organisatorische und institutionelle Strukturen, Abläufe und Zusammenhänge kennenlernen und ein kritisches Verständnis für selbige entwickeln
- Projekte und Events im Team planen und durchführen
- Aufgabenstellungen in heterogenen Teams eigenständig entwickeln
- Teamprozesse lösungsorientiert moderieren und Konflikte managen
- neuen Themen (Technologien, Methoden, Werkzeuge) erschließen und das neue Wissen bzw. neue Kompetenzen an Peers vermitteln
- Arbeitsgemeinschaften und Kommunikationsformate weiterführen und entwickeln
- Verantwortung für das eigene Studiumfeld übernehmen

um später im beruflichen und gesellschaftlichen Kontext professionell handeln zu können und das Arbeitsumfeld als auch die Disziplin als solche zu hinterfragen und mitgestalten zu können.

Inhalte / Aufbau des Moduls

- Formate für Arbeitsgemeinschaften und Kommunikation:
 - Studieninfo & Aufnahme
 - Raum
 - Code of Conduct
 - Internationalisierung
 - Öffentlichkeitsarbeit
 - Coding & Tool Support
 - Presentations & Events
 - Iteration
- Feedback und Teilnahme an der Teaching Analysis Poll (TAP)
- Präsentationen

Modul »Creating Impact 1« (CI1)

Verantwortlich: Prof. Dr. Markus Linden

Studiensemester: 3

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: AGILE / ENSI

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Die Absolvent*innen des Moduls können soziale Kontexte und ökonomische Anforderungen für digitale Produkte und Applikationen evaluieren, indem sie

- Design Thinking als nutzerzentrierte Ansatz zur Entwicklung innovativer Produktideen anwenden,
- Zielgruppen und deren Bedürfnisse analysieren und anhand von Kreativmethoden erste Prototypen entwickeln,
- agile und inkrementell-iterative Arbeitsmethoden zur Exploration nutzen,
- grundlegende Konzepte des Entrepreneurship verstehen und anwenden und
- Geschäftsmodelle unter Anwendung etablierter und moderner Gründungskonzepte entwickeln,

um in der Lage zu sein, innovative, digitale und nutzerzentrierte Lösungen im unternehmerischen Kontext zu gestalten und umzusetzen.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
CI11	Entrepreneurial Thinking	40h / 50h	ENSI / AGILE	3	unbenotet - strukturiertes Feedback
CI12	Design Thinking	40h / 50h	AGILE	3	unbenotet - strukturiertes Feedback

Modul »Developing Things 2« (DT2)

Verantwortlich: Prof. Dr. Jonas Schild

Studiensemester: 3

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Empfohlene Voraussetzungen: Developing Things 1

Kompetenzcluster: SDA / DRSS / IEE / PROD

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Absolvent*innen dieses Moduls können interaktive Systeme und Experiences konzipieren, entwickeln und bewerten, indem sie

- Grundlagen der Interaktivität verstehen und in eigenen Konzepten anwenden,
- praktische Anwendungen entwickeln und Prototypen technisch umsetzen,
- gezielt Experiences für dinglich-digitale Interaktionen entwickeln,
- und den Impact für bestimmte Gruppen von Nutzenden oder Kontexte bewerten und optimieren,

um später interaktive Produkte ganzheitlich zu verstehen und zu entwickeln, welche leicht zu benutzen sind und besondere Erfahrungen bieten.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
DT21	Interactive Applications	40h / 50h	SDA / DRSS / IEE / PROD	3	unbenotet - strukturiertes Feedback
DT22	Experience Engineering	40h / 50h	SDA / DRSS / IEE / PROD	3	unbenotet - strukturiertes Feedback

Modul »Coding Software 3« (CS3)

Verantwortlich: Prof. Dr. Stefan Bente

Studiensemester: 3

Häufigkeit: jährlich

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Empfohlene Voraussetzungen: Coding Software 1 / Coding Software 2 / Project Code

Kompetenzcluster: AGILE / SDA

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Die Studierenden sind in der Lage, effizient und nachhaltig Softwareapplikationen zu erstellen, indem sie

- in einer bewussten, Kriterien-gestützten Entscheidung das beste Werkzeug (Sprache, Tool, Methode, Framework) für die aktuelle Aufgabe auswählen,
- bekannte Paradigmen, Technologien, Patterns und Architekturstile für den Entwurf und die Umsetzung eines Softwaresystems nutzen können, sowie
- in der Lage sind, diese problembezogen und pragmatisch anzupassen,

sodass sie digitale Produkte mit hohem Softwareanteil schnell und mit Blick auf spätere Weiterentwicklung herstellen können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
CS31	Application Design	40h / 50h	AGILE / SDA	3	unbenotet - strukturiertes Feedback

Modul »Designing Futures 3« (DF3)

Verantwortlich: Prof. Nina Juric

Studiensemester: 3

Häufigkeit: jährlich

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / TSCE / CREA

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Studierende kennen und verstehen die Entwicklung des Open Design und Open Web als Triebfeder digitaler Innovation und sind in der Lage, mit eigenen Beiträgen dessen Weiterentwicklung zu fördern. Sie basieren dies auf ihrem Bewusstsein einer Handlungsethik, die die Werte freiheitlicher, sozialer und geschlechtergerechter Weiterentwicklung von Gemeinwesen ins Zentrum setzt.

Hierzu führen sie

- Projekte mit Nutzung und zur Erweiterung der Open xx Plattformen
- Studien zu den Implikationen von Coded Artefacts auf die Gesellschaft und die Verantwortung des Entwicklers

durch, um

- zum Gemeineigentum im globalen Netz beizutragen und
- als Entwickler globaler Applikationen ethisch verantwortbar handeln zu können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
DF31	Open Design	40h / 50h	IEE / TSCE / CREA	3	unbenotet - strukturiertes Feedback

Modul »Project Impact« (PI)

Verantwortlich: Prof. Dr. Jonas Schild

Studiensemester: 3

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 30h Kontaktzeit / 150h Selbstlernzeit

Kompetenzcluster: SDA / IEE / PROD / COMM / CREA / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Studierende können die materiell-digitale Wirkung im gesellschaftlich-wirtschaftlichen und wissenschaftlichen Kontext erkunden, indem sie

- Öffentlichkeit auf Veranstaltungen, Workshops oder Tagungen einbinden,
- Aufmerksamkeit und Interesse für eigene Inhalte erregen,
- technisch/design-mäßig innovative Konzepte zur Mitwirkung erstellen,
- Zielgruppen definieren und adressieren,
- dafür begrenzte Ressourcen einwerben, planen und einsetzen,
- und sich eigenverantwortlich in Gruppen, Hierarchien und Strukturen organisieren

damit Sie den Aufwand und konkrete Ansätze von Wirksamkeitsmechanismen abschätzen und später planerisch für eigene Unternehmungen umsetzen können.

Inhalt

Die Studierenden organisieren sich in einem Projekt, um die Wirkung des der fachlichen Ansätze und Inhalte des Studiengangs nach außen zu explorieren, beispielsweise den Studiengang als ganzes nach außen in einer Jahresveranstaltung zu präsentieren. Dabei binden Sie andere Beteiligte wie Studierende anderer Jahrgänge, Angehörige der Hochschule oder externe kommerzielle wie wirtschaftliche Partner und Interessenten ein, organisieren eine Finanzierung und eine Öffentlichkeitsarbeit und veranstalten ein Event.

Modul »Community & Reflection 3« (CR3)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 3

Häufigkeit: fortwährend

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 20h Kontaktzeit / 160h Selbstlernzeit

Kompetenzcluster: SOSE

Learning Outcome / Qualifikationsziele des Moduls

Teilnehmer*innen der »Community & Reflection«-Module können an der Organisation, Selbstverwaltung, Kommunikation und Weiterentwicklung des Studiengangs teilhaben, in dem sie:

- organisatorische und institutionelle Strukturen, Abläufe und Zusammenhänge kennenlernen und ein kritisches Verständnis für selbige entwickeln
- Projekte und Events im Team planen und durchführen
- Aufgabenstellungen in heterogenen Teams eigenständig entwickeln
- Teamprozesse lösungsorientiert moderieren und Konflikte managen
- neuen Themen (Technologien, Methoden, Werkzeuge) erschließen und das neue Wissen bzw. neue Kompetenzen an Peers vermitteln
- Arbeitsgemeinschaften und Kommunikationsformate weiterführen und entwickeln
- Verantwortung für das eigene Studiumfeld übernehmen

um später im beruflichen und gesellschaftlichen Kontext professionell handeln zu können und das Arbeitsumfeld als auch die Disziplin als solche zu hinterfragen und mitgestalten zu können.

Inhalte / Aufbau des Moduls

- Formate für Arbeitsgemeinschaften und Kommunikation:
 - Studieninfo & Aufnahme
 - Raum
 - Code of Conduct
 - Internationalisierung
 - Öffentlichkeitsarbeit
 - Coding & Tool Support
 - Presentations & Events
 - Iteration
- Feedback und Teilnahme an der Teaching Analysis Poll (TAP)
- Präsentationen

Modul »Developing Things 3« (DT3)

Verantwortlich: Prof. Dr. Christian Faubel

Studiensemester: 4

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Empfohlene Voraussetzungen: Developing Things 1

Kompetenzcluster: DRSS / SDA

Modulprüfung: Teilmodulprüfungen, Integrierte Prüfung über Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Student*innen können integrierte, programmierte, dingliche Prototypen, die Menschen, Umwelt und mit digitaler Infrastruktur wechselwirken entwickeln und gestalten, indem sie

- die Grundprinzipien vernetzter digitaler Technologien und die Verknüpfung von Micro-Controllern mit Serverdienste erlernen.
- Softwarewerkzeuge entwickeln, um die Wechselwirkung zwischen Umwelt und Infrastruktur im laufenden Betrieb zu analysieren,
- Prinzipien digitaler Fertigung, computergestützten Entwurf und Computergestützte Fertigung experimentell erlernen

um später kreativ neue innovative, vernetzte, dinglich-digitale Geräte, bei denen sich Form und Funktion sinnvoll ergänzen, entwickeln zu können.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
DT31	Internet of Things	40h / 50h	DRSS / SDA	3	unbenotet - strukturiertes Feedback
DT32	Making of Things	40h / 50h	DRSS	3	unbenotet - strukturiertes Feedback

Modul »Wahlpflichtmodul 1« (WPF1)

Verantwortlich: Prof. Dr. Jonas Schild

Studiensemester: 4

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: AGILE / SDA / DRSS / IEE / TSCE / ENSI / SOSE / COMM / PROD

Modulprüfung: Teilmodulprüfungen, (benotet)

Learning Outcome / Qualifikationsziele des Moduls

Die Studierenden verbreitern bzw. vertiefen die in den vier Handlungsfeldern erworbenen Kenntnisse durch den Erwerb von Kompetenzen in einem überfachlichen Themengebiet. Es gibt ein variierendes Angebot der Wahlpflichtfächer. Es müssen zwei Kurse aus dem Katalog der wählbaren Teilmodule belegt werden oder in Absprache mit den jeweiligen Dozent*innen können auch Pflichtmodule aus anderen Studiengängen (insb. Informatik, Medieninformatik, IT Management, Wirtschaftsinformatik, Medientechnologie, Integrated Design) der TH Köln als Wahlpflichtkurs belegt werden, wenn sie dem Umfang von mindestens 3 ECTS entsprechen.

Prüfungsform

Die Prüfungsformen können je nach Teilmodul variieren und werden zu Beginn des Semesters mit dem Wahlangebot bekannt gegeben.

Angebot

Die Auswahl der Teilmodule wird zu Beginn jedes Semesters von den Lehrenden festgelegt und bekannt gegeben. Das jeweilige Angebot umfasst Teilmodule, die mindestens zwei der vier Handlungsfeldern Coding Software, Developing Things, Designing Futures und Creating Impact im Studiengang Code & Context zuzuordnen sind.

Teilmodule (Beispielhafte Auswahl):

- Business Economics (Creating Impact)
- Client Server Architectures (Coding Software)
- DevOps (Coding Software)
- Human-Computer Impact (Creating Impact, Developing Things)
- Trend & Market Research (Creating Impact)
- Advanced Coding (Coding Software)
- Blended International Project (Developing Things)
- Design as a Language (Designing Futures)
- Machine Learning Basics (Developing Things)
- Open Ecosystems (Designing Futures)

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.
WPF11	WPF z.B. Coding Software - DevOps, Microservice Architecture, Advanced Coding	40h / 50h
WPF12	WPF z.B. Creating Impact - Human-Computer Impact, Business Economics, Trend & Market Research	40h / 50h

Modul »Project Make 1« (PM1)

Verantwortlich: Prof. Dr. Christian Faubel

Studiensemester: 4

Häufigkeit: jährlich

Kreditpunkte: 12 ECTS

Arbeitsaufwand: 70h Kontaktzeit / 290h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD / COMM / CREA / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Studierende können (1) als Projektarbeit im Team ein digitales Produkt mit einem komplexen Software-Anteil erstellen, wobei sie sich die ökonomischen, sozialen, rechtlichen und kulturellen Aspekte einer gegebenen realistischen Anwendungsdomäne erschließen und die Produkthanforderungen in einem dynamischen Umfeld umsetzen, (2) ihre Projektarbeit in Hinblick auf alle vier Handlungsfelder des Studiengangs reflektieren, sowie (3) ihre Projektarbeit in Hinblick auf den aktuellen Wissensstand abgleichen und dokumentieren, indem sie

- in Projektteams mit agilen Organisationsformen arbeiten,
- der Komplexität und Dynamik von Domäne und technischen Lösungen mit erlernten Methoden und Konzepten begegnen,
- Prototypen, Prototypen und Minimum Viable Products umsetzen,
- Nutzer*innen und Kontexte analysieren und einbeziehen,
- Betriebs- oder Geschäftsmodelle für kommerzielle Produkte, Dienstleistungen, freie Software oder kollaborative Anstrengungen explorieren,
- und die Kompetenzen der begleitenden Kurse anwenden und in einen Zusammenhang bringen,
- ihre entwickelten Artefakte (Code, Dokumentation) so gestalten, dass eine nachhaltige Weiterentwicklung als Open-Source-Projekt (oder einer anderen Form eines digitalen Ökosystems) möglich ist, sowie
- ihr Produkt in angemessener Weise den Stakeholdern präsentieren,
- ihre Arbeit kompakt beschreiben und Lehrpersonen aus anderen Fachdisziplinen vorstellen, Feedback zu den Handlungsfeldern einfordern, diskutieren und verarbeiten sowie die Projektarbeit an die Resultate aus dieser Diskussion anpassen,
- Methoden zur wissenschaftlichen Recherche und Verarbeitung von Quellen erlernen und anwenden, Überblicke über vergleichbare Lösungsansätze oder Konkurrenzprodukte erarbeiten sowie seminaristische Ausarbeitungen anfertigen und präsentieren

damit Sie

- die in den Kursen erworbenen Kenntnisse durch praktische Anwendung und eigene Erfahrungen im Rahmen eines realistischen Projekts weiter vertiefen und verinnerlichen,
- und später innovative digitale Produkte erfolgreich an den Markt oder in die nachhaltige, gesellschaftliche Wirkung bringen.

Inhalte / Aufbau des Moduls

Das Project Make 1 besteht aus

-
- der Bearbeitung einer Projektaufgabe in Teams
 - einer Interdisciplinary Project Reflection zur interdisziplinären Reflexion und Optimierung der Projektbearbeitung mit den vier Handlungsfeldern des Studiengangs (Coding Software, Designing Futures, Developing Things und Creating Impact) mit fachlichem Input und Feedback-Sessions und
 - einem Research Colloquium zur Einbettung der Projektaufgabe in wissenschaftliche Arbeitsweise, Hintergrundrecherche, Informationsverarbeitungsmethoden und Dokumentation.

Die Bewertung erfolgt durch die das Projekt betreuende Lehrperson unter Einbeziehung der Leistung in allen drei Teilmodulen.

Lehr- und Lernformen

- Projektarbeit
- Research Colloquium
 - Seminar und Kolloquium
 - Ringvorlesungen
- Interdisciplinary Project Reflection
 - Panels und Feedback-Sessions
 - Rebuttals

Materialien/ Ressourcen

- Lehrmaterialien werden projektspezifisch bereitgestellt

Modul »Community & Reflection 4« (CR4)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 4

Häufigkeit: fortwährend

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 20h Kontaktzeit / 160h Selbstlernzeit

Kompetenzcluster: SOSE

Learning Outcome / Qualifikationsziele des Moduls

Teilnehmer*innen der »Community & Reflection«-Module können an der Organisation, Selbstverwaltung, Kommunikation und Weiterentwicklung des Studiengangs teilhaben, in dem sie:

- organisatorische und institutionelle Strukturen, Abläufe und Zusammenhänge kennenlernen und ein kritisches Verständnis für selbige entwickeln
- Projekte und Events im Team planen und durchführen
- Aufgabenstellungen in heterogenen Teams eigenständig entwickeln
- Teamprozesse lösungsorientiert moderieren und Konflikte managen
- neuen Themen (Technologien, Methoden, Werkzeuge) erschließen und das neue Wissen bzw. neue Kompetenzen an Peers vermitteln
- Arbeitsgemeinschaften und Kommunikationsformate weiterführen und entwickeln
- Verantwortung für das eigene Studiumfeld übernehmen

um später im beruflichen und gesellschaftlichen Kontext professionell handeln zu können und das Arbeitsumfeld als auch die Disziplin als solche zu hinterfragen und mitgestalten zu können.

Inhalte / Aufbau des Moduls

- Formate für Arbeitsgemeinschaften und Kommunikation:
 - Studieninfo & Aufnahme
 - Raum
 - Code of Conduct
 - Internationalisierung
 - Öffentlichkeitsarbeit
 - Coding & Tool Support
 - Presentations & Events
 - Iteration
- Feedback und Teilnahme an der Teaching Analysis Poll (TAP)
- Präsentationen

Modul »Creating Impact 2« (CI2)

Verantwortlich: Prof. Dr. Markus Linden

Studiensemester: 5

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: ENSI / COMM

Modulprüfung: Teilmodulprüfungen, siehe Kurse (unbenotet)

Learning Outcome / Qualifikationsziele des Moduls

Die Studierenden können die betriebswirtschaftlichen Konsequenzen des eigenen Handelns sowie die Wirkungen selbstentwickelter Geschäftsideen evaluieren, indem sie

- für innovative Produkte, Dienste und Geschäftsmodelle fundierte Trend- und Marktforschungen durchführen,
- die wichtigsten Entscheidungsbereiche wirtschaftlichen Handelns auf Unternehmensebene analysieren und
- konkrete Entwurfs- und Geschäftspläne für ihre unternehmerischen Vorhaben entwickeln,

um wirtschaftliche Potenziale für Produkte zu erkennen und diese in nachhaltigen Geschäfts- und Betreibermodellen umzusetzen.

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzcluster	ECTS	Prüfung
CI21	Business Planning 2	40h / 50h	ENSI	3	unbenotet - strukturiertes Feedback
CI22	Business Planning 1	40h / 50h	ENSI / COMM	3	benotet

Modul »Wahlpflichtmodul 2« (WPF2)

Verantwortlich: Prof. Dr. Jonas Schild

Studiensemester: 5

Häufigkeit: jährlich

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 80h Kontaktzeit / 100h Selbstlernzeit

Kompetenzcluster: IEE / PROD / ENSI / TSCE / DRSS / CREA

Modulprüfung: Teilmodulprüfungen, (benotet)

Learning Outcome / Qualifikationsziele des Moduls

Die Studierenden verbreitern bzw. vertiefen die in den vier Handlungsfeldern erworbenen Kenntnisse durch den Erwerb von Kompetenzen in einem überfachlichen Themengebiet. Es gibt ein variierendes Angebot der Wahlpflichtfächer. Es müssen zwei Kurse aus dem Katalog der wählbaren Teilmodule belegt werden oder in Absprache mit den jeweiligen Dozent*innen können auch Pflichtmodule aus anderen Studiengängen (insb. Informatik, Medieninformatik, IT Management, Wirtschaftsinformatik, Medientechnologie, Integrated Design) der TH Köln als Wahlpflichtkurs belegt werden, wenn sie dem Umfang von mindestens 3 ECTS entsprechen.

Prüfungsform

Die Prüfungsformen können je nach Teilmodul variieren und werden zu Beginn des Semesters mit dem Wahlangebot bekannt gegeben.

Angebot

Die Auswahl der Teilmodule wird zu Beginn jedes Semesters von den Lehrenden festgelegt und bekannt gegeben. Das jeweilige Angebot umfasst Teilmodule, die mindestens zwei der vier Handlungsfeldern Coding Software, Developing Things, Designing Futures und Creating Impact im Studiengang Code & Context zuzuordnen sind.

Teilmodule (Beispielhafte Auswahl):

- Business Economics (Creating Impact)
- Client Server Architectures (Coding Software)
- DevOps (Coding Software)
- Human-Computer Impact (Creating Impact, Developing Things)
- Trend & Market Research (Creating Impact)
- Advanced Coding (Coding Software)
- Blended International Project (Developing Things)
- Design as a Language (Designing Futures)
- Machine Learning Basics (Developing Things)
- Open Ecosystems (Designing Futures)

Teilmodule

Nr.	Kurs aus Katalog	Kontakt-/Selbstl.	Kompetenzclu
WPF21	WPF z.B. Designing Futures - Design As A Language, Open Ecosystems	40h / 50h	IEE / PROD / E
WPF22	WPF z.B. Developing Things - Machine Learning Basics, Blended International Project	40h / 50h	DRSS / CREA /

Modul »Project Make 2« (PM2)

Verantwortlich: Prof. Dr. Jonas Schild

Studiensemester: 5

Häufigkeit: jährlich

Kreditpunkte: 12 ECTS

Arbeitsaufwand: 70h Kontaktzeit / 290h Selbstlernzeit

Kompetenzcluster: IEE / PROD / ENSI / COMM / TSCE / CREA / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Studierende können (1) als Projektarbeit im Team hardware-basierte, smarte oder verteilte interaktive Produkte und ihre materiell-digitale Wirkung ganzheitlich gestalten, (2) ihre Projektarbeit in Hinblick auf alle vier Handlungsfelder des Studiengangs reflektieren, sowie (3) ihre Projektarbeit in Hinblick auf den aktuellen Wissensstand abgleichen und dokumentieren, indem sie

- software-/hardwarenahe, verteilte oder smarte Anwendungen realisieren und deren Interaktionen gestalten und diese reflektieren,
- in Projektteams mit agilen Organisationsformen arbeiten,
- der Komplexität und Dynamik mit Methoden und Konzepten begegnen,
- Prototypen, Prototypen und Minimum Viable Products umsetzen,
- Nutzer*innen und Kontexte analysieren und einbeziehen,
- Betriebs- oder Geschäftsmodelle für kommerzielle Produkte, Dienstleistungen, freie Software oder kollaborative Anstrengungen explorieren,
- Businesspläne entwickeln, Produktkosten abschätzen und Marktfähigkeit evaluieren,
- Beiträge zu Gesellschaftsqualität durch insb. Nachhaltigkeit, Global Citizenship, Persönlichkeitsentfaltung, Wohlstand motivieren,
- und die Kompetenzen der begleitenden Kurse anwenden und in einen Zusammenhang bringen,
- ihre Arbeit kompakt beschreiben und Lehrpersonen aus anderen Fachdisziplinen vorstellen, Feedback zu den Handlungsfeldern einfordern, diskutieren und verarbeiten sowie die Projektarbeit an die Resultate aus dieser Diskussion anpassen,
- Methoden zur wissenschaftlichen Recherche und Verarbeitung von Quellen erlernen und anwenden, Überblicke über vergleichbare Lösungsansätze oder Konkurrenzprodukte erarbeiten sowie seminaristische Ausarbeitungen anfertigen und präsentieren

damit Sie

- die in den Kursen erworbenen Kenntnisse durch praktische Anwendung und eigene Erfahrungen weiter vertiefen und verinnerlichen,
- und später innovative digitale Produkte erfolgreich an den Markt oder in die nachhaltige, gesellschaftliche Wirkung bringen.

Inhalte / Aufbau des Moduls

Das Project Make 2 besteht aus

-
- der Bearbeitung einer Projektaufgabe in Teams
 - einer Interdisciplinary Project Reflection zur interdisziplinären Reflexion und Optimierung der Projektbearbeitung mit den vier Handlungsfeldern des Studiengangs (Coding Software, Designing Futures, Developing Things und Creating Impact) mit fachlichem Input und Feedback-Sessions und
 - einem Research Colloquium zur Einbettung der Projektaufgabe in wissenschaftliche Arbeitsweise, Hintergrundrecherche, Informationsverarbeitungsmethoden und Dokumentation.

Die Bewertung erfolgt durch die das Projekt betreuende Lehrperson unter Einbeziehung der Leistung in allen drei Teilmodulen.

Lehr- und Lernformen

- Projektarbeit
- Research Colloquium
 - Seminar und Kolloquium
 - Ringvorlesungen
- Interdisciplinary Project Reflection
 - Panels und Feedback-Sessions
 - Rebuttals

Materialien/ Ressourcen

- Lehrmaterialien werden projektspezifisch bereitgestellt

Modul »Community & Reflection 5« (CR5)

Verantwortlich: Prof. Dr. Frank Schimmel

Studiensemester: 5

Häufigkeit: fortwährend

Kreditpunkte: 6 ECTS

Arbeitsaufwand: 20h Kontaktzeit / 160h Selbstlernzeit

Kompetenzcluster: SOSE

Learning Outcome / Qualifikationsziele des Moduls

Teilnehmer*innen der »Community & Reflection«-Module können an der Organisation, Selbstverwaltung, Kommunikation und Weiterentwicklung des Studiengangs teilhaben, in dem sie:

- organisatorische und institutionelle Strukturen, Abläufe und Zusammenhänge kennenlernen und ein kritisches Verständnis für selbige entwickeln
- Projekte und Events im Team planen und durchführen
- Aufgabenstellungen in heterogenen Teams eigenständig entwickeln
- Teamprozesse lösungsorientiert moderieren und Konflikte managen
- neuen Themen (Technologien, Methoden, Werkzeuge) erschließen und das neue Wissen bzw. neue Kompetenzen an Peers vermitteln
- Arbeitsgemeinschaften und Kommunikationsformate weiterführen und entwickeln
- Verantwortung für das eigene Studiumfeld übernehmen

um später im beruflichen und gesellschaftlichen Kontext professionell handeln zu können und das Arbeitsumfeld als auch die Disziplin als solche zu hinterfragen und mitgestalten zu können.

Inhalte / Aufbau des Moduls

- Formate für Arbeitsgemeinschaften und Kommunikation:
 - Studieninfo & Aufnahme
 - Raum
 - Code of Conduct
 - Internationalisierung
 - Öffentlichkeitsarbeit
 - Coding & Tool Support
 - Presentations & Events
 - Iteration
- Feedback und Teilnahme an der Teaching Analysis Poll (TAP)
- Präsentationen

Modul »Optionales Praxissemester« (OPT)

Verantwortlich: alle Dozent*innen im Studiengang

Studiensemester: 0 (6 bei optionalem Praxissemester)

Häufigkeit: fortwährend

Kreditpunkte: 30 ECTS

Arbeitsaufwand: 0h Kontaktzeit / 900h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD / AGILE / ENSI / COMM / TSCE / CREA / SOSE

Qualifikationsziele des Moduls

Im optionalen Praxissemester haben die Student*innen die Wahl zwischen drei Alternativen mit jeweils unterschiedlichen Qualifikationszielen zur Ausgestaltung des Moduls:

- Internship
- Self-Initiated Project
- Auslandssemester

Die Wahl lässt den Studierenden Freiraum zur Selbstgestaltung eines eigenen Profils. Das Praxissemester dient zudem als Mobilitätsfenster. Es ist unbenotet.

Modulvariante »Praktikum«

In der Modulvariante »Praktikum« haben die Studierenden die Möglichkeit, ihre bisher erlangten Kompetenzen in verschiedenen Bereichen praktisch zu erproben und zu vertiefen. Mögliche Bereiche sind beispielsweise Institutionen innerhalb von Communities, Vereine, NGOs, Unternehmen oder Startups. Die Studierenden erlangen ein vertieftes Verständnis ihrer berufspraktischen Tätigkeit und steigern ihre Kompetenzen hinsichtlich der Softskills und Querschnittsqualifikationen, bspw. Kommunikation, Organisation, Zeit- und Selbstmanagement.

Modulvariante »Self-Initiated Project«

In der Modulvariante »Self-Initiated Project« haben die Studierenden die Möglichkeit, ein eigenes Projekt mit einer hohen Workload von 900h in einem Team zu initialisieren und zu realisieren. Die Studierenden wenden die bisher gebildeten Kompetenzen an und können sich für ihre eigene neue Aufgabenstellung neue Kompetenzen und Technologien aneignen. Das Projekt soll das Potential haben, ein Minimum Viable Product zu realisieren, welches das Potenzial für eine weitere Verwertung hat (bspw. Pitch vor Investoren, Crowd-Funding, Bildung einer Community, Open Source Beitrag). Die Studierenden erlangen ein vertieftes Verständnis ihrer berufspraktischen Tätigkeit und steigern ihre Kompetenzen hinsichtlich der Softskills und Querschnittsqualifikationen, bspw. Kommunikation, Organisation, Projekt-, Konflikt, Zeit- und Selbstmanagement.

In der Modulvariante »Auslandssemester« haben die Studierenden die Möglichkeit, ihre Fachkompetenzen und wissenschaftlichen Fähigkeiten zu vertiefen und akademische Perspektive zu erweitern, bspw. durch den Besuch interdisziplinärer Lehrveranstaltungen im Ausland. Zudem können die Studierenden ihre inter-kulturellen Kompetenzen und Sprachkompetenzen stärken.

Prüfungsformen

- Anerkennung der erbrachten Leistungen beim Auslandssemester
- Projektdokumentation und Vortrag

Für alle Varianten des Praxissemesters gilt, dass für das erfolgreich abgeschlossene Praxissemester 30 ECTS durch die Mentor*innen vergeben werden.

Diese 30 ECTS setzen sich wie folgt zusammen: 25 ECTS sind nachzuweisen über ein Internship, ein Self-Initiated Project oder ein Auslandssemester. Weitere 3 ECTS entfallen auf die Präsentation und Dokumentation des Praxissemesters, welche im Rahmen der »Praxissemesterbörse« im folgenden Wintersemester präsentiert werden. Die »Praxissemesterbörse« wird von den Teilnehmer*innen des optionalen Praxissemesters selbst organisiert. Weitere 2 ECTS entfallen auf eine Leistung, die die Studierenden mit den jeweiligen Mentor*innen abstimmen (bspw. ein begleitendes Blog, ein Extraprojekt, ein Handbuch, Demonstration der Ergebnisse, Video zum Ergebnis).

Modul »Project Launch« (PL)

Verantwortlich: alle Dozent*innen im Studiengang

Studiensemester: 6 (7 bei optionalem Praxissemester)

Häufigkeit: fortwährend

Kreditpunkte: 15 ECTS

Arbeitsaufwand: 0h Kontaktzeit / 450h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD / AGILE / ENSI / COMM / TSCE / CREA / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Die Studierenden können in einem weitgehend selbstständigen Projekt,

- relevante, realitätsnahe Probleme identifizieren,
- mögliche Lösungsansätze context- sowie nutzungszentriert mit pragmatischen und wissenschaftlichen Methoden analysieren,
- geeignete Methoden und Techniken, die sie im Studium erlernt haben, auswählen und erproben,
- sich notwendige zusätzliche Fähigkeiten und Ressourcen erarbeiten und organisieren,
- eine iterative Umsetzung auf Basis von Prototypen, Prototypen oder Minimum Viable Products umsetzen,
- die Untersuchung bzgl. Innovation in Technologie, Design und/oder Markt steuern,
- die technik-ethische Einordnung zu Grundprinzipien von Gesellschaftsqualität wie insb. Nachhaltigkeit, Global Citizenship, Persönlichkeitsentfaltung, Wohlstand diskutieren,
- und somit erste Erfahrungen mit der Selbststeuerung und proaktiven Kommunikation in einem Projekt mittlerer Größe vorweisen,

um ggf. auf Basis dieser Ergebnisse eine Bachelorarbeit unter Einbeziehung der interdisziplinären Handlungsfelder des Studiengangs planen und bearbeiten zu können.

Inhalt

Modulinhalte des ersten bis fünften Semesters anhand von realen Anforderungen in einem praxisrelevanten Kontext anwenden und den Studierenden durch die Betreuung des Dozenten an eine selbstständige Projektdurchführung und Kommunikation heranführen. Das Project Launch kann auch beispielsweise in Communities, Vereinen, NGOs, Unternehmen, Startups oder in der Hochschule - dann eingebettet in wissenschaftliche Forschungsprojekte - erfolgen.

Lehr- und Lernformen

Angeleitetes, eigenverantwortliches Arbeiten

Materialien/ Ressourcen

Lehrmaterialien werden projektspezifisch empfohlen

Modul »Bachelorarbeit« (BA)

Verantwortlich: alle Dozent*innen im Studiengang

Studiensemester: 6 (7 bei optionalem Praxissemester)

Häufigkeit: fortwährend

Kreditpunkte: 12 ECTS

Arbeitsaufwand: 0h Kontaktzeit / 360h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD / AGILE / ENSI / COMM / TSCE / CREA / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Die Bachelorarbeit soll zeigen, dass der Prüfling befähigt ist, innerhalb einer vorgegebenen Frist eine praxisorientierte Aufgabe aus seinem Fachgebiet sowohl in ihren fachlichen Einzelheiten als auch in den fachübergreifenden Zusammenhängen nach wissenschaftlichen, fachpraktischen und gestalterischen Methoden selbstständig zu bearbeiten. Die Bachelorarbeit ist in der Regel eine eigenständige Untersuchung mit einer Aufgabenstellung aus der aktuellen Themen des Studiengangs und einer ausführlichen Beschreibung und Erläuterung ihrer Lösung. In fachlich geeigneten Fällen kann sie auch eine schriftliche Hausarbeit mit fachliterarischem Inhalt sein.

Inhalt

Selbstständiges wissenschaftliches, fachpraktisches und gestalterisches Bearbeiten einer Aufgabenstellung.

Modul »Bachelor Kolloquium« (BK)

Verantwortlich: alle Dozent*innen im Studiengang

Studiensemester: 6 (7 bei optionalem Praxissemester)

Häufigkeit: fortwährend

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 0h Kontaktzeit / 90h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD / AGILE / ENSI / COMM / TSCE / CREA / SOSE

Learning Outcome / Qualifikationsziele des Moduls

Das Kolloquium dient der Feststellung, ob der Prüfling befähigt ist, die Ergebnisse der Bachelorarbeit, ihre fachlichen Grundlagen, ihre fachübergreifenden Zusammenhänge und ihre außerfachlichen Bezüge mündlich darzustellen und selbständig zu begründen und ihre Bedeutung für die Praxis einzuschätzen. Dabei soll auch die Bearbeitung des Themas der Bachelorarbeit mit dem Prüfling erörtert werden.

Lehrform/SWS

- Angeleitetes, eigenverantwortliches Arbeiten

Inhalt

- Vortrag über das Thema der Bachelorarbeit
- Fachdiskussion und mündliche Verteidigung der Arbeit

Übersicht über Kurse und Katalog

Alle Kurse können innerhalb eines Semesters abgeschlossen werden.

Tabelle 0.1: Übersicht über die Teilmodule und Vernetzung der Kurse mit den einzelnen Kompetenzen

Teilmodul	Kurs / Lehrveranstaltung	SDA	DRSS	IEE	PROD	AGILE
DF11	Prototyping Things			x	x	
DF12	Users and Situated Action			x		
CS11	Coding Essentials 1	x				
CS12	Coding Essentials 2	x				
PC1	Coding Essentials 3	x				
PC2	Project Code: Project	x			x	
CS21	Agile Coding Practices	x				x
CS22	Agile Team Practices	x				x
DT12	Designing Context			x		
DT11	Things with Agency		x	x		
DF21	Generative Gestaltung	x				
DF22	Interaction Design			x		
CI12	Design Thinking					x
CI11	Entrepreneurial Thinking					x
DT22	Experience Engineering	x	x	x	x	
DT21	Interactive Applications	x	x	x	x	
CS31	Application Design	x				x
DF31	Open Design			x		
DT31	Internet of Things	x	x			
DT32	Making of Things		x			
WPF11	WPF z.B. Coding Software - DevOps, Microservice Architecture, Advanced Coding	x	x			x
WPF12	WPF z.B. Creating Impact - Human-Computer Impact, Business Economics, Trend & Market Research			x	x	
CI22	Business Planning 1					
CI21	Business Planning 2					
WPF21	WPF z.B. Designing Futures - Design As A Language, Open Ecosystems			x	x	
WPF22	WPF z.B. Developing Things - Machine Learning Basics, Blended International Project		x	x	x	

Kurs »Prototyping Things«

Verantwortlich: Prof. Dr. Laura Popplow

Sprache: deutsch,englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / PROD

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Block, Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Studierende lernen, was Design im Kontext eines Produktrahmens bedeutet, indem sie ...

- den Ansatz (> mindset > jackpot) kennenlernen
- Denkweisen - Herangehensweisen- Wirkungsweisen unterscheiden lernen,
- Szenarien zur Produktentwicklung in unterschiedlichen Kontexten entwickeln (Storylines / Storytelling / Text)
- Arten von Prototypes kennenlernen und verstehen (Material / Materialität)
- unterschiedliche Perspektiven einzunehmen lernen (Roleplay)
- unterschiedliche Ebenen eines Produktes multiperspektivisch interpretieren (Communication)
- Ideen, Bilder, Texte und Sounds, Filme bzw. ganze Kommunikationskampagnen dazu entwickeln,
- Sustainability, durability and recycling hinterfragen
- Methoden kennenlernen um social layer neu zu denken

um ...

- interaktive Umgebungen zu verstehen darin ein Produkt zu gestalten
- die Entwicklungsschritte zu kennen
- die Relevanz & Qualität eines Produktes einzuschätzen lernen,
- Pipelines zu verstehen und Konsequenzen abzusehen
- Prozesse innerhalb der Gestaltung zu steuern, zu planen
- "relational Environments: Connected products and ecosystems" kennenzulernen
- Konzeptionstärke für originellen Produktideen freizusetzen
- Gestaltung von Kommunikation und Erfahrungen zu bewerkstelligen

Inhalt

- What is a product? What is the context? What is a digital product within a certain context? What is virtual product?
- What means product design, when a product is not visible anymore? What are relational interactive environments? What is the difference between a feature and a bug?
- Genres & Formatfragen

Lehr- und Lernformen

- Vorlesung, Diskussion, Übungen, Gruppenprojekt

Materialien/ Ressourcen

- Prototyping-Materialien

Literatur

- Dunne, A., & Raby, F. ([2013]). *Speculative everything: design, fiction, and social dreaming*. The MIT Press.
- Bennett, J. ([2020]). *Vibrant Matter: A Political Ecology of Things*. Duke University Press. <https://doi.org/10.1515/9780822391623>

Kurs »Users and Situated Action«

Verantwortlich: Prof. Dr. Lasse Scherffig

Sprache: deutsch,englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / CREA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und in der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Absolvent*innen des Kurses können Interaktion für bestimmte Gruppen von Nutzer*innen gestalten, indem sie

- die Wahrnehmung von Nutzer*innen im Kontext verschiedener Interfaces und der damit zusammenhängenden Handlungszusammenhänge berücksichtigen – insbesondere im Hinblick auf Fragen von dis/ability, Inklusion und Barrierefreiheit –,
- Interfaces im Sinne ihrer kontextuellen Situierung verstehen und gestalten
- und Methoden der Erfassung von solchen Kontexten einsetzen

um Interaktion und die durch Interfaces erzeugten Handlungsräume ganzheitlich zu entwerfen und umzusetzen.

Inhalt

- Prototyping als Vermittlung zwischen (technischen, anwenderbezogenen, administrativen, ...) Kulturen und Kontexten
- Paradigmen des Umgangs mit Benutzer*innen: User Models, Mental Models, Embodiment, Situated Action
- Technology Probes

Lehr- und Lernformen

- Vorlesung
- Praktische Übung
- Kurzprojekt

Materialien/ Ressourcen

- Prototyping-Materialien
- Labor

Literatur

- Wong, A. (2020). Disability visibility : first-person stories from the twenty-first century.
- Suchman, L. A. (1987). Plans and situated actions: The problem of human-machine communication. Cambridge university press.

Kurs »Coding Essentials 1«

Verantwortlich: Prof. Dr. Christian Faubel

Sprache: deutsch,englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden können in Einzelarbeit und auf Basis exakt spezifizierter Aufgabenstellungen kleine, konsolenbasierte Programme kodieren, indem sie Programmiersprachenkonzepte für das "Programmieren im Kleinen" anwenden, um später große Systeme auch auf Detailebene kodieren zu können.

Inhalt

- Grundlegende Prozesse (Codierung, Kompilation/Interpretation)
- "Programmieren im Kleinen", z.B.
 - Expressions
 - Variablen, elementare Typen
 - Funktionen, Parameter
 - Verzweigungen, Schleifen
 - komplexe Typen (Arrays, Maps etc.)
- Grundlegende Tools
 - Compiler
 - IDE (z.B. Visual Studio Code)
 - * Projekte verwalten (Classpath etc.)
 - * Debugger bedienen

Lehr- und Lernformen

- Vorlesung
- Übungen zu einzelnen Inhalten
- begleitende Projektaufgabe:
 - Ausgangssituation: Exakt spezifizierte Anforderungen an ein sehr kleines System (Kommandozeilentool zu String-Manipulationen, Taschenrechner etc.)
 - Zielsituation: lauffähiges System

Materialien/ Ressourcen

- Liste ausgewählter Literatur und Web-Ressourcen

-
- Griffiths, D., Griffiths, D., & Lang, J. W. (2019). Kotlin von Kopf bis Fuß (1. Auflage). O'Reilly.
 - Videos und Tutorials
 - OOP mit Kotlin, Christian Kohls <https://www.youtube.com/playlist?list=PLYohfDwqINSE6jPghuVDKdA-9UlwhFxyQ>
 - Vorlesungsunterlagen
 - Übungsunterlagen inkl. Lösungen

Kurs »Coding Essentials 2«

Verantwortlich: Prof. Dr. Frank Schimmel

Sprache: deutsch,englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden können in Einzelarbeit und auf Basis gegebener Anforderungen größere Programme erstellen, indem sie Konzepte der Objekt-orientierten Programmierung sowie gängige Frameworks und Libraries verwenden, um später an der Entwicklung großer Software-Systeme mitwirken zu können.

Inhalt

- Objekt-Orientierte Programmierung, insbesondere
 - Abstraktion
 - Klassen
 - Interfaces
 - Vererbung
 - Polymorphie
 - Kapselung
- Basics UML zur Visualisierung
- Weitere Essentials, wie
 - Packages
 - Massendatentypen
 - Ausnahmebehandlung
 - ggfs. Lambdas
- Build Tools, Frameworks und Libraries

Lehr- und Lernformen

- Vorlesung (Videos)
- Übungen zu einzelnen Inhalten
- begleitende Projektaufgabe:
 - Ausgangssituation: Anforderungen an ein kleines System
 - Zielsituation: lauffähiges System

Materialien und Ressourcen

- Liste ausgewählter Literatur und Web-Ressourcen
 - Griffiths, D., Griffiths, D., & Lang, J. W. (2019). Kotlin von Kopf bis Fuß (1. Auflage). O'Reilly.
- Videos und Tutorials
 - OOP mit Kotlin, Christian Kohls <https://www.youtube.com/playlist?list=PLYohfDwqINSE6jPghuVDKdA-9UlwhFxyQ>
- Vorlesungsunterlagen
- Übungsunterlagen inkl. Lösungen
- Code Starter

Literatur

- Wirfs-Brock, R. McKean, A. (2003). Object Design - Roles, Responsibilities, and Collaborations. Addison-Wesley. ISBN 0-201-37943-0

Kurs »Coding Essentials 3«

Verantwortlich: Prof. Dr. René Wörzberger

Sprache: deutsch,englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Modulwoche nach dem Block (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden können in Einzelarbeit und auf Basis gegebener Anforderungen einfache Web-Applikationen erstellen, indem sie

- Client- und Server-Seitige Komponenten programmieren,
- Schnittstellen zu Betriebs- und File-System verwenden
- Features von Browsern sowie Frameworks und Libraries einsetzen,

um später bei der Entwicklung von Client-Server-Applikationen mitwirken zu können sowie alternative Plattformen, Frameworks und Libraries zur Applikationsentwicklung einordnen zu können.

Inhalt

- Client-Server-Architektur
- Browser als Client & Applikations-Plattform
 - HTML & CSS
 - Javascript
 - Netzwerk-Verbindungen, HTTP
- Betriebssysteme als Applikations-Plattform
 - Zugriff auf Dateien
 - Ein- und ausgehende Netzwerkverbindungen

Lehr- und Lernformen

- Vorlesung
- Übungen zu einzelnen Inhalten
- begleitende Aufgaben

Materialien/ Ressourcen

- Liste ausgewählter Literatur und Web-Ressourcen
 - Walls, C. (2022). Spring in action. Simon and Schuster.

– Aigner, S., Elizarov, R., Isakova, S., & Jemerov, D. (2024). Kotlin in Action (Second Edition). Manning.

- Vorlesungsunterlagen
- Übungsunterlagen inkl. Lösungen
- Quellcode-Beispiele
- vorkonfigurierte Entwicklungsumgebungen
- vorgefertigte Entwicklungsstände

Kurs »Project Code: Project«

Verantwortlich: Prof. Dr. Frank Schimmel

Sprache: deutsch, englisch

Kreditpunkte: 9 ECTS

Arbeitsaufwand: 50h Kontaktzeit / 220h Selbstlernzeit

Kompetenzcluster: SDA / PROD / COMM / SOSE

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag, Modulwoche nach dem Block (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Studierende können auf der Basis grundlegender Programmierkenntnisse in objektorientierter Programmierung ein problem-basiertes Szenario analysieren und mögliche Lösungen explorieren und umsetzen, indem sie

- eine umfangreiche Aufgabenstellung (die auf die begleitenden Kurse zugeschnitten ist) selbstständig bearbeiten,
- dabei die Kompetenzen der begleitenden Kurse anwenden und in einen Zusammenhang bringen,
- ein objektorientiertes Modell der gegebenen Domäne entwickeln und damit eine lauffähige Software erstellen,

damit sie

- die in den Kursen erworbenen Kenntnisse durch praktische Anwendung und eigene Erfahrungen weiter vertiefen und verinnerlichen,
- und dies in spätere Module einbringen können.

Inhalt

- Es wird eine komplexe Programmieraufgabe gestellt.

Lehr- und Lernformen

- Projektarbeit
- Code Kino (Feedback Sessions mit Code Reviews und Popcorn)

Materialien/ Ressourcen

- Grundlage sind die Ressourcen aus den Teilmodulen Coding Essentials 1-3.
- Das zu bearbeitenden Projektthema wechselt jährlich.

Literatur

Es gibt keine dedizierte Literatur für das Projekt. Die relevante Literatur ist in den Teilmodulen Coding Essentials 1-3 zu finden.

Kurs »Agile Coding Practices«

Verantwortlich: Prof. Dr. Frank Schimmel

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: AGILE / SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag, Modulwoche nach dem Block (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden können funktionierende, verständliche, wartbare und erweiterbare Software-Programme erstellen, bzw. vorhandene Software-Programme in eine besser verständliche, wartbare und erweiterbare Form bringen, indem sie

- Vorgehensweisen und Methoden des automatisierten Testens einsetzen und Software Test-getrieben entwickeln,
- für den Kontext relevante Prinzipien des Software-Designs erkennen und anwenden,
- Code Smells feststellen und beheben,
- und Refactorings begründet auswählen und durchführen,

um später neue und wechselnde Anforderungen von Kund*innen/Nutzer*innen dauerhaft zügig in ihrer Software umsetzen (programmieren) zu können.

Inhalt

- Clean-Code als elementares Leitmotiv beim Coding
 - Konzepte "Bad Smells" und Antipatterns
 - Gängige, oft auftretende Antipatterns / Bad Smells
- Refactoring als Methode und Haltung
 - Richtiges Vorgehen beim Refactoring
- Test-Driven Development als Paradigma
 - Zusammenspiel mit Clean Code Methoden
 - Praktische Hinweise zu einem pragmatischen Umgang
- Gemeinsames Programmieren mit Co-Creation Patterns
 - Pair Programming
 - Ensemble Programming

Lehr- und Lernformen

Mischungen aus Impulsen, Übungen, Übungen und noch mehr Übungen.

Literatur

- Beck, K. (2009). Test driven development: by example (14. print.). Addison-Wesley.
- Fowler, M. & Beck, K. (2019). Refactoring: improving the design of existing code (Second edition). Addison-Wesley.
- Thomas, D & Hunt, A. (2019). The Pragmatic Programmer, 20th Anniversary Edition, your journey to mastery. Pragmatic Bookshelf. ISBN 9780135957059

Kurs »Agile Team Practices«

Verantwortlich: Prof. Dr. Frank Schimmel

Sprache: deutsch,englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: AGILE / SDA / COMM

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Modulwoche nach dem Block (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden beherrschen die Grundlagen von Software-Entwicklung im Team, indem sie

- mittels agilen Prozessen und Werkzeugen neue Anforderungen selbstorganisiert und arbeitsteilig planen und umsetzen,
- agile Praktiken nutzen, um dabei effektiv vorzugehen,
- agile Prinzipien und Werte im Team umsetzen,
- Regeln zur Qualitätssicherung im Team formulieren und anwenden,
- teamorientierte Entwicklungs-, Test- und Auslieferungsprozesse durch Automatisierung beschleunigen,

damit sie später komplexe digitale Produkte in einem agilen Entwicklungsprozess mit kurzen Feedback-Zyklen und in hoher Qualität umsetzen können.

Inhalt

- Konzepte
 - Agiles Vorgehen im Projekt, agile Ansätze
 - agiles Anforderungsmanagement (User Story Mapping, Akzeptanzkriterien)
 - Reflexion der eigenen Arbeitsweise
- Agile Planung
 - Backlog-Management, Definieren von Anforderungen mit User Stories und Akzeptanzkriterien
 - Iterations- & Release-Planung
- Praktiken zur Entwicklung im Team
 - Elementare Continuous Integration
 - Tooling zur Automatisierung
 - Trunk-based Development
- Qualitätssicherung & Testen im Team
 - Testarten (Module, Integration, Acceptance)
 - Einführung in gängige Analyse- & Testing-Tools

Lehr- und Lernformen

- Impulse
- Teach-backs
- kontinuierliche Übung

Literatur

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifest für agile Softwareentwicklung. Abgerufen am, 20, 2024.
- Jeffries, R (2015). The Nature of Software Development: Keep It Simple, Make It Valuable, Build It Piece by Piece. Pragmatic Bookshelf. ISBN 9781680505085

Kurs »Designing Context«

Verantwortlich: Prof. Dr. Laura Popplow

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / TSCE

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Absolvent*innen des Kurses gestalten und nutzen digitale soziale Kontexte, indem sie

- unterschiedliche soziale Technologien, wie Wikis, Instant Messenger, Image Boards, oder Online-Spiele und die darin gängigen Kulturtechniken nutzen,
- neue soziale Organisationsformen wie Open Source, Crowd Sourcing, Remote Work, Hackathons, Hacker-spaces oder Co-working projektbezogen einsetzen,
- und beim Entwurf neuer Anwendungen die sozialen Gefüge berücksichtigen, die im Umfeld von Software und Computertechnologie entstehen,

um später den Community-Kontext digitaler Produkte und Anwendungen absehen, planen und gezielt aufbauen zu können.

Inhalt

- Social Computing
- Open Source, F(L)OSS, Creative Commons
- New work, remote work, collaboration
- Hackathons, Maker Spaces
- Digitale „Parallelwelten“ und ihre Kulturtechniken, von Online-Spielen bis Image Boards
- Geschichte sozialer Hubs: Startups, Networking
- Social Hubs als Gefüge von Personen, Unternehmungen, Netzwerken
- Persönlichkeitsbildung

Lehr- und Lernformen

- Kurzprojekt: Organisation eines sozialen Kontexts (Event, Open-Source-Projekt, ...)

Materialien/ Ressourcen

- Biella Coleman: Coding Freedom: The Ethics and Aesthetics of Hacking, University Press Group Ltd., 2013

Kurs »Things with Agency«

Verantwortlich: Prof. Dr. Christian Faubel

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: DRSS / CREA / IEE

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Absolvent*innen des Kurses beherrschen den Umgang mit materiellen und Code-basierten Entwurfsprozessen, indem sie

- Prototypen mit Methoden des Physical Computing entwickeln und dabei das Ineinandergreifen von Code und Material als gestalterisches Handlungsfeld nutzen,
- zur Realisierung dieser Prototypen sowohl klassischen Modellbau als auch digitale Fabrikation einsetzen und
- Gestaltung als iterativen Prozess betreiben, der den ständigen Dialog von Material, Code, Funktion und Ästhetik sucht

um später Prototypen unterschiedlichster Art (formal, funktional, narrativ) in den verschiedenen Stadien interdisziplinärer Projekte zu realisieren.

Inhalt

- Grundlagen des Physical Computing im Design-Kontext
- Modellbau
- digitale Fabrikation
- Formen von Prototypen
- Iterative Gestaltungsprozesse
- Code und Material

Lehr- und Lernformen

- Vorlesung
- Praktische Übung

Materialien/ Ressourcen

- Modellbauwerkstatt
- Werkzeuge digitaler Fabrikation
- 2D Entwurfssoftware
- Nicolas Collins: "Handmade Electronic Music"
- Patrick Schnabel: "Elektronik Fibel"

-
- Georg Trogemann: "Code und Material"

Kurs »Generative Gestaltung«

Verantwortlich: Prof. Christian Noss

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: CREA / SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Teilnehmer*innen des Kurses können eigene Konzepte und Code für regelbasierte und generierte Artefakte analysieren und entwickeln, indem sie

- einen Überblick über die gestalterischen Möglichkeiten kennen lernen
- Tools und Techniken kennen und einsetzen können
- die Grundprinzipien Wiederholung, Zufall und Logik verstehen und anwenden können

so dass sie generative Gestaltung im Entwurfsprozess nutzen zu können.

Inhalt

- Inspiration und Projekte
- Punkt, Linie, Fläche, Farbe & Form
- Zufall & Wiederholung
- Kontext

Lehr- und Lernformen

Workshop

Materialien/ Ressourcen

- Benedikt Groß, Hartmut Bohnacker, Julia Laub, Claudius Lazzaroni: Generative Gestaltung: Creative Coding im Web; Entwerfen, Programmieren und Visualisieren mit Javascript in p5.js; Verlag Hermann Schmidt, 2018
- p5.js

Weitere Informationen

Alles, aber auch alles zum Kurs gibt es auf der [Website zum Kurs Generative Gestaltung](#)

Kurs »Interaction Design«

Verantwortlich: Prof. Dr. Lasse Scherffig

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / CREA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Absolvent*innen des Kurses können die Interaktion mit digitalen Artefakten und Prozessen gestalten, indem sie

- Interaktionen mit GUIs und TUIs technisch realisieren,
- diese Interaktionen iterativ und experimentell gestalten (unter dem Einsatz von Prototypen und Methoden des Creative Coding),
- sie in das Feld der Interaktionsgestaltung (Geschichte und Theorie) einordnen,
- und ihre Benutzbarkeit als eine Frage der nachhaltigen Inklusion und Barrierefreiheit zu verstehen,

um später neuartige Interaktionsformen und User Experiences in interdisziplinären Projekten zur Anwendung zu bringen.

Inhalt

- Geschichte und Paradigmen des Interaction Design
- Experimentelle Interaktionsgestaltung

Lehr- und Lernformen

- Vorlesung
- Praktische Übung
- Kurzprojekt

Materialien/ Ressourcen

- Prototyping-Materialien
- Labor

Literatur

- Lialina, O. (2018). Rich user experience, UX and the desktopization of war. *Interface Critique*, (1), 176-193.

Kurs »Design Thinking«

Verantwortlich: Prof. Dr. Ivonne Preusser

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: AGILE

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Block, Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Studierende können Design Thinking als einen Ansatz zur Entwicklung von innovativen Ideen für Produkte anwenden, um später nach dieser Methode Ideen für eigene Projekte zu explorieren.

Absolvent*innen des Kurses lernen Design Thinking als agile, nutzerzentrierte Arbeitsweise zur Gestaltung innovativer Lösungen kennen und Methoden anwenden, indem sie

- die Design Thinking Grundlagen (z.B. „human-centered-Design“, Elemente, Mind-Set) kennenlernen,
- einen Design Thinking Prozess umsetzen und anhand realer Beispiele durchlaufen,
- Zielgruppen und ihre Bedürfnisse verstehen und dies in den Entwicklungsprozess einbeziehen,
- Methoden der Nutzer- und Kundenforschung anwenden,
- Kreativmethoden nutzen, um Ideenfindung voranzutreiben,
- Problem- und Lösungsräume miteinander verknüpfen,
- Ideen bis hin zu einem ersten Prototyp entwickeln,
- agile inkrementell-iterative Vorgehensmodelle anwenden,
- in selbstorganisierten Teams Lösungen entwickeln
- und Einsatzmöglichkeiten reflektieren

um Ideen für eigene Projekte zu explorieren und Lösungsideen evaluieren zu können.

Inhalt

Die Teilnehmer*innen des Kurses durchlaufen einen typischen Design Thinking Prozess mit den Phasen: Verständnis, Erkundung, Synthese, Ideenfindung, Prototypen und Test.

Lernziel ist es, Ideen unter Anwendung einer bestimmten Systematik (Design Thinking Prozess) reifen zu lassen und Prototypen zu entwickeln, bei dem der Mensch im Fokus der Betrachtung steht. Dabei unterstützen Kreativmethoden, iterative Testung und interviewgeleitetes Feedback.

- Design Thinking-Prozess
- Kreativmethoden
- Iterative Testung
- Prototypen

-
- Nutzer- und Kundenforschung (z.B. qualitative und quantitative Verfahren, Interviews, Befragung, Methoden der Marktforschung)
 - Bedürfnisse der Nutzer eruieren und prüfen
 - Persona erstellen, Methode um Kundenbedürfnisse zu identifizieren

Lehr- und Lernformen

- Workshop
- Impuls-Vorträge
- Interaktives Seminar
- Kollaborative Teamarbeit
- Praxis-Übungen
- Projektarbeit

Materialien/ Ressourcen

- Brenner, W.; Uebersnickel, F. (Hrsg.) (2016): Design Thinking for Innovation: Research and Practice. Cham: Springer International Publishing, 2016
- Brown, T. (2009): Change by design – How design thinking transforms organizations and inspires innovation. New York: Harper Business
- Plattner, H., Meinel, C., & Leifer, L. (Hrsg.). (2015). Understanding Innovation. Design thinking research: Building innovators. Cham, s.l.: Springer International Publishing. Retrieved from <http://dx.doi.org/10.1007/978-3-319-06823-7>

Kurs »Entrepreneurial Thinking«

Verantwortlich: Prof. Dr. Markus Linden

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: ENSI / AGILE

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierende verstehen die Grundlagen des Entrepreneurship und können diese anwenden, indem sie

- die Herkunft und wirtschaftliche Bedeutung von Gründungen kennen,
- die betriebswirtschaftliche Sichtweise von Gründer, Investoren und Kunden auf innovative Produkte verstehen, deren Sprache verstehen und sprechen können,
- bestehende Gründungen betriebswirtschaftlich analysieren und bewerten können
- etablierte und moderne Konzepte und Ansätze der Gründung beschreiben und anwenden können

um Ideen für innovative Produkte und Geschäftsmodelle in einem Gründungskontext fundiert zu diskutieren und selbst zu entwickeln.

Inhalt

- Grundlagen Entrepreneurship – Herkunft, Entwicklung, Volkswirtschaftliche Bedeutung, Abgrenzung Corporate/Social/Intrapreneurship
- Etablierte Sichtweisen auf Gründung – Businessplan, Investoren Pitch
- Moderne Ansätze – Effectuation, Lean Start Up, Crowdfunding/funding
- Ansätze zur Opportunity Recognition

Lehr- und Lernformen

- Seminaristischer Unterricht zu Grundlagen
- Analyse eines bestehenden StartUps (Geschichte, Erfolgsfaktoren, Geschäftsmodell) z.B. Fond of Bags, True fruits, My Muesli, (weitere aus der IT Branche)

Materialien/ Ressourcen

- Engelen & van Gagern (2017): Opportunity Recognition: 15 Ansätze für mehr Unternehmenswachstum, Springer-Gabler
- Ries (2014): The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses
- Osterwalder & Pigneur (2010): Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers

-
- Sarasvathy (2009): Effectuation
 - Grichnik, Brettel et al. (2017): Entrepreneurship: Entscheiden und Handeln in innovativen und technologieorientierten Unternehmen
 - Engelen, Engelen, Bachmann (2015): Corporate Entrepreneurship: Unternehmerisches Management in etablierten Unternehmen, Springer-Gabler

Kurs »Experience Engineering«

Verantwortlich: Prof. Dr. Jonas Schild

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden sind in der Lage, die User Experience für Interaktive Systeme gezielt zu entwickeln und zu überprüfen, indem sie

- Grundlagen aus dem Games Engineering und Game Design anwenden, um gezielte Experiences in interaktiven Systemen zu erzeugen,
- Methoden zur Optimierung von Feedback Loops z.B. über Belohnungssysteme kennenlernen und anwenden können,
- Konzepte des Emotioneerings z.B. über interaktive Narrativen, Entscheidungsbäume, Information Balancing und Emotional Design verstehen und integrieren,
- Systeme auf Heuristiken, Style Guides und Grundprinzipien der Human-Computer Interaction für effiziente Bedienung überprüfen und daran anpassen,
- Verfahren zur wissenschaftlichen Analyse und Bewertung von User Experience und Usability anwenden, diskutieren und reflektieren können,

um später in der Lage zu sein, interaktive Produkte und dinglich-digitale Systeme hinsichtlich ihrer User Experience zu optimieren.

Lehr- und Lernformen

- Vorlesung
- Gastvortrag externer Referenten
- Seminar
- Praktische Übung
- Begleitende Projektaufgabe

Materialien/ Ressourcen

- Tynan Sylvester: "Designing Games"
- Steve Swink: "Game Feel: A Game Designer's Guide to Virtual Sensation"
- Jesse Schell: "The Art of Game Design: A Book of Lenses"
- Ernest Adams: "Fundamentals of Game Design"
- Katie Salen und Eric Zimmerman: "Rules of Play: Game Design Fundamentals"
- Raph Koster: "A Theory of Fun for Game Design"

-
- Jonathan Lazar et al., "Research Methods in Human-Computer-Interaction", Wiley, 2010

Kurs »Interactive Applications«

Verantwortlich: Prof. Dr. Jonas Schild

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: SDA / DRSS / IEE / PROD

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag (unbenotet - strukturiertes Feedback)

Der Kurs "Interactive Applications" vermittelt den Studierenden die theoretischen und praktischen Grundlagen der Konzeption und Implementierung interaktiver Systeme. Die Studierenden setzen sich intensiv mit der Entwicklung von eigenen Prototypen auseinander und lernen, wie sie Loops aus Action und Feedback in verschiedenen Systemen optimieren können. Dabei werden historische, aktuelle und zukünftige Technologien thematisiert.

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden sind in der Lage, Interaktive Systeme und Anwendungen zu entwickeln, indem sie

- Grundlagen der Interaktivität und der Konzeption interaktiver Systeme erlernen,
- wichtige Prinzipien z.B. technischer, kognitiver, mathematischer und gestalterischer Art, kennenlernen und anwenden können,
- Technologien historischer, aktueller und zukünftiger Hard- und Software Systeme analysieren und praktisch erproben,
- und diese in die Entwicklung und Implementierung von interaktiven Prototypen integrieren,
- unter Anwendung von aktuellen Umsetzungswerkzeugen wie z.B. Game Engines, Prototyping Tools, Generatoren und Programmiersprachen.

so dass sie später eigene, innovative Bedienkonzepte für ein digital-dingliches Produkt konzipieren und technisch umsetzen können.

Inhalte

- Einführung in die Grundlagen interaktiver Systeme und der Human-Computer Interaction
- Prinzipien innovativer Interaktionssysteme, z.B. Natural User Interfaces, Mixed Reality, Digital Games, Tangible Interaction
- Werkzeuge für die Entwicklung von Benutzeroberflächen und interaktiven Prototypen

Lehr- und Lernformen

- Vorlesung
- Gastvortrag externer Referenten
- Seminar
- Praktische Übung
- Begleitende Projektaufgabe

Materialien/ Ressourcen

- Moggridge, B., & Atkinson, B. (2007). Designing interactions (Vol. 17). Cambridge: MIT press.
- Butz, A., Krüger, A.: Mensch-Maschine-Interaktion, De Gruyter Oldenbourg, 2014
- Preim, B., Dachselt, R.: Interaktive Systeme, Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer 2010
- Preim, B., Dachselt, R.: Interaktive Systeme, Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces. Springer-Verlag, 2015
- Dörner R., Broll, W., Grimm, P, Jung, B.,, Virtual und Augmented Reality (VR/AR). Grundlagen und Methoden der Virtuellen und Augmentierten Realität, Springer Vieweg Berlin, Heidelberg, 2019, <https://link.springer.com/book/10.1007/978-3-662-58861-1>
- LaViola Joseph J. Jr., Kruijff Ernst, McMahan Ryan P., Bowman Doug, Poupyrev Ivan P., 3D User Interfaces 2nd ed., Theory and Practice, Pearson ITP, 2017

Kurs »Application Design«

Verantwortlich: Prof. Dr. Stefan Bente

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: AGILE / SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

- **Als** SW-Entwickler:in kann ich ein Softwaresystem für eine gegebene Aufgabenstellung iterativ konzipieren und mit gängigen Tools implementieren,
- **indem ich** ...
 - passende Prinzipien, Patterns und Architekturstile für mein IT-System auswähle, wie etwa die SOLID-Prinzipien und die Regeln des Clean Code, und dann eine gängige Schichtenarchitektur auswähle
 - ein Rich Domain Model nach dem Ansatz des Domain-Driven-Design (DDD) in Building Blocks konzipiere und mit Hilfe von Spring JPA umsetze
 - meine Domäne(n) in Aggregates strukturiere
- **damit ich** eine nachhaltig wartbare Software mit passender, langlebige und änderungsfähiger Architektur erstellt habe.

Inhalt

Application Design (AD) beschäftigt sich mit Patterns und Architekturstilen, um größere und verteilte Softwaresysteme bauen zu können. Besonderes Augenmerk liegt auf den Konzepten des Domain-Driven-Design (DDD). Damit wird eine nachhaltige Wartbarkeit sicher gestellt. Der Kurs wird im Block in Form von Workshops durchgeführt, bei denen wir inhaltlich diskutieren und Sie an einer durchgehenden Programmieraufgabe arbeiten. Jede:r Teilnehmer:in bekommt ein individuelles Repo, in dem die Aufgabenstellung beschrieben ist.

Lehr- und Lernformen

Der Kurs wird über ein personalisiertes und individuell für Sie generiertes Git-Repo durchgeführt. Jede:r Teilnehmer:in muss eine eigenständige Bearbeitung der Praktikumsaufgabe durchführen. Ich ermutige Sie ausdrücklich, Ihre Lösungen in Arbeitsgruppen zu diskutieren. Ein einfaches Copy-Paste von Lösungen (unter Anpassung der Personalisierung) werde ich aber als Täuschungsversuch. Studierende, bei denen ich dieses sehe (und wir haben automatisierte Möglichkeiten, dies zu sehen) werden ohne Vorwarnung vom Kurs ausgeschlossen. Das gilt sowohl für die Person, die kopiert hat, wie auch für die-/denjenigen, die/der die Lösung zur Verfügung gestellt hat.

Prüfungsform

Erfolgreich absolvierte Aufgaben im individuellen Git-Repo.

Materialien/ Ressourcen

Für diese Veranstaltung sind Lehrvideos auf Youtube verfügbar. Abonnieren Sie dafür am besten den Kanal [Archi-Lab](#). Für die Videos zu dieser Veranstaltung gibt es eine [Playlist](#).

Außerdem gibt eine Vielzahl von aktuellen Lehrmaterialien auf meiner Laborseite <https://www.archi-lab.io/>.

Kurs »Open Design«

Verantwortlich: Prof. Nina Juric

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / TSCE / CREA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , Block, Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Studierende können offene gestalterische Prozesse initiieren und auswerten, indem sie

- alte Elektronik und Elektronische Geräte hinterfragen
- DIY und Open-Source-Projekte untersuchen,
- Methoden der interaktiven Medienkunst und den darstellen (Interaktions-)Künsten, des Social und Participatory Design einsetzen
- die zur Entwicklung eigener Konzepte und sogenannter "Conversation Pieces" führen

um später Anwendungen wie Installationen entwickeln zu können, die relevant, sozialverträglich und innovativ sind.

Inhalt

- DIY / E-Waste / "Hyperlinked Originals"
- social design / partizipatorische prozesse
- durch Gesellschaft inspiriert im Design
- kritischer diskurs: Dialog Gesellschaft x Objekte x Code als Teilbereiche im Design, die mit Technologien zu tun haben
- Demo Szene x Produktdesign x Installationen
- Ästhetik / Materialität
- Co-Working
- Dramaturgie

Lehr- und Lernformen

- Schnittstellenperformances im Motion Experience Lab
- Live-Code x Body
- Seminaristische Inputs
- Case studies
- Subversive Intervention wie Flashmobs, Ausstellungen, Sketches
- Interviews
- spielerische Formate, etc.

Materialien/ Ressourcen

- Whose Time Is It? - Asocial Robots, Syncolonialism, and Artificial Chronological Intelligence; von Stamatia Portanova

How do the two technologies of blockchain and artificial intelligence actualize and, crucially, automatize the cognition of time? These kinds of machines are increasingly part of both our contemporary present and our prospective future, but how do we really define a present and a future? And more importantly, how do these machines themselves understand, know, and sense time? Can machines really think about the present and dream the future in an autonomous way? In order to unravel these questions, this book follows the “emerging life adventures and experiences” of Sophia, a robot animated by blockchain and AI, to present a study in temporal automation.

- Contemporaneity in Embodied Data Practices; Cornelia Sollfrank, Felix Stalder

What parallels are there between a human pranayama practitioner and a migratory bird in heavily datafied environments? And what can they tell us about the need to reorient our thinking towards the planetary?

In *Contemporaneity in Embodied Data Practices*, two artistic field studies provide the starting point for a dialogical reflection on the entanglement of diverse temporalities in body-related, datafied, and experiential practices. Shifting through lived, historical, evolutionary, and technological rhythms, Cornelia Sollfrank and Felix Stalder unfold their respective more-than-human frames of reference, and arrive at specific forms of agency in the contemporary moment.

- Und: Open Design Networks, Fablabs, www.org, [petter, uvm](http://petter.uvm).
- Design Noir - The secret Live of electronic objects

Kurs »Internet of Things«

Verantwortlich: Prof. Dr. Christian Faubel

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: DRSS / SDA

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden können hardwarenah vernetzte Geräte entwickeln, die ihre physikalische Umgebung wahrnehmen und verändern, indem sie

- Sensoren und Aktoren zur Messung und Veränderung der Umwelt auswählen,
- hardwarenahe Software für Mikrocontroller und Einplatinencomputer entwickeln,
- geeignete Protokolle zur Vernetzung im Internet of Things kennen und nutzen,
- sowie Prototyping als Entwicklungsansatz im IoT einsetzen,

um später Anwendungen und Produkte zu realisieren, bei denen digitale und dingliche Welten vernetzt sind und miteinander wechselwirken.

Inhalt

- Physical Computing
- Einfache Client-Server Communication
- Nachrichtenprotokolle
- Microcontroller-programmierung
- Microcontroller mit WIFI
- WIFI-Protokolle

Lehr und Lernformen

Projektarbeit mit seminaristischen Input.

Materialien / Ressourcen

auf TH-Spaces bereitgestellt:

- Lehrvideos,
- Kurtutorials,
- links zu Tutorials

Kurs »Making of Things«

Verantwortlich: Prof. Dr. Christian Faubel

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: DRSS

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Studierende lernen die Nutzung komplexer computergesteuerter Maschinen indem sie

- die Software zum computersteuerten Entwurf und zur computergesteuerten Fertigung anwenden,
- üben zu experimentieren, mit unerwarteten Ergebnissen umzugehen und sich auf den Prozess zu fokussieren,
- üben realistisch Gefahren abzuschätzen, die sich aus der Nutzung ergeben und eigene Gefahrenanalysen vornehmen,

um später mit diesen Kompetenzen zu emanzipierten und experimentierfreudigen Maschinenbediener*innen zu werden, welche die kreativen Potenziale in der Nutzung von Maschinen erkennen und nutzen.

Inhalt

Es wird semesterweise eine Fokussierung auf eine der bei Code & Context vorhandenen MASchinen geben:

- Lasercutter
- 3D Drucker
- CNC-Fräse
- Strickmaschine
- Stickmaschine

Unterricht zur Nutzung von CAD und CAM Software.

Lehr- und Lernformen

Praktikum, Seminar

Materialien / Ressourcen

auf TH-Spaces bereitgestellt

- Gefahrenunterweisungen
- Lehrvideos

Kurs »WPF z.B. Coding Software - DevOps, Microservice Architecture, Advanced Coding«

Verantwortlich: Prof. Dr. René Wörzberger, Prof. Dr. Stefan Bente, Prof. Dr. Frank Schimmel

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: AGILE / SDA / DRSS

Art: WPF

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag, kontinuierlich während der LV und in der Modulwoche (benotet)

Angestrebte Lernergebnisse / Learning Outcomes (Beispiel DevOps)

Die Studierenden können fortwährend entwickelte Software-Systeme automatisiert bauen und automatisiert in von ihnen aufgesetzten virtualisierten Cloud-Umgebungen installieren, indem sie

- Grundkenntnisse in Linux (als dem marktbeherrschenden Server-Betriebssystem) besitzen und zur Administration einsetzen können,
- Build, Test und Deployment der Software-Systeme über eine CI/CD-Pipeline automatisieren,
- grundlegende Administrationskenntnisse (Firewall-, Proxy-, Domain-Konfiguration, ...) nutzen,
- und Werkzeuge zur Virtualisierung und zum Cloud-Computing anwenden,

damit sie digitale Produkte agil über den gesamten Lebenszyklus und im Sinne einer Ende-zu-Ende-Verantwortung ("you build it - you run it") realisieren können.

Inhalt (Beispiel DevOps)

- Basiswissen Linux und Netzwerke
 - Grundlegende Verfahren wie z.B. Public-Key-Authentifizierung
 - Grundlegende Werkzeuge wie z.B. bash und ssh
- Systematische Kollaboration in Versionierungs-Workflows (Git)
- Build- und Test-Automatisierung
- Betrieb von Applikationen in der Cloud
- Virtualisierung mit Containern

Lehr- und Lernformen (Beispiel DevOps)

- Impuls-Vorträge
- ggf. Gastvorträge externer Referenten
- Übungen
- Projekt
 - Entwicklung oder Erweiterung einer kleinen Backend-Applikation (entsprechende Kenntnisse aus Client Server Basics werden vorausgesetzt)

-
- Umsetzung des Build- und Continuous-Delivery-Prozesses inkl. Qualitätssicherung für die Backend-Applikation

Materialien / Ressourcen (Beispiel DevOps)

- Vorlesungsunterlagen und -videos
- Übungsunterlagen inkl. Lösungen
- Projekt-Meilensteinbeschreibungen
- Zugänge, Lizenzen und Kontingente für
 - gemeinsame Code-Repositories (z.B. GitLab)
 - Cloud-Computing-Provider (z.B. Google Cloud)
- Literatur
 - The DevOps Handbook, Gene Kim, Jez Humble, Patrick Debois, John Willis (2016)
 - Accelerate, Nicole Forsgren, Jez Humble, Gene Kim (2018)
 - Continuous Delivery, Jez Humble, David Farley (2010)
 - Site Reliability Engineering, Betsy Beyer, Chris Jones, Jennifer Petoff, Niall R. Murphy (2016)

Kurs »WPF z.B. Creating Impact - Human-Computer Impact, Business Economics, Trend & Market Research«

Verantwortlich: Prof. Dr. Jonas Schild, Prof. Dr. Markus Linden

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / TSCE / ENSI / SOSE / COMM / PROD

Art: WPF

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag, kontinuierlich während der LV und in der Modulwoche (benotet)

Angestrebte Lernergebnisse / Learning Outcomes (Beispiel Human Computer Impact)

Die Studierenden können nutzungsorientierte als auch ethische Aspekte von innovativen Technologiesystemen analysieren, evaluieren und erörtern, indem Sie

- Nutzungsstudien zu Experiences und Effizienz von interaktiven Systemen kennenlernen, konzipieren, planen, durchführen und resultierende Daten mit wissenschaftlichen Methoden analysieren
- Grundlagen der Technikethik kennenlernen, in Diskussionen anwenden, mögliche Technikfolgen analysieren und unter Betrachtung von Wertekonflikten reflektieren, um die Bewertung von Nutzungsvor- und Nachteilen eigenständig vornehmen zu können und dabei die Vulnerabilität von Gesellschaften im Global Citizenship nachzuvollziehen und ethische Verantwortung zu übernehmen.

Inhalt (Beispiel Human Computer Impact)

- Grundlagen: Begriffe aus der Technikethik, Technikfolgenabschätzung, VDI 3780, einfache Beispiele
- Durchführung einer Projektstudie, z.B. Nutzungsstudie, Technikfolgenabschätzung
- Veranstaltung eines pluralistischen Diskussionspanels

Lehr- und Lernformen (Beispiel Human Computer Impact)

- Vortrag
- Seminar
- Projektarbeit

Materialien/ Ressourcen (Beispiel Human Computer Impact)

- Vorlesungsunterlagen
- Jonathan Lazar et al., "Research Methods in Human-Computer-Interaction", Wiley, 2010
- Grunwald, A., Hillerbrand, R., Handbuch Technikethik, 2-. Auflage, 2021, J.B. Metzler, Springer Verlag
- Böschen, S., Grunwald, A., Krings, B.-J., & Rösch, C. (Eds.). (2021). Technikfolgenabschätzung

Kurs »Business Planning 1«

Verantwortlich: Prof. Dr. Monika Engelen

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: ENSI / COMM

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (benotet)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierende können den Marketingmix und die Markteintrittsstrategie für ein innovatives Produkt entwerfen, indem sie die 4 Ps des Marketingmixes (Product, Price, Promotion, Place) für ein spezifisches Produkt und Zielgruppe analysieren und gestalten können und den Markteintritt konkret planen insb.

- Produktgestaltung, -komponenten sowie Marke basierend auf Kundenprofilen und -wertversprechen (Value Proposition Map) ableiten
- Preissetzung und -strategie sowie Erlösmodell orientiert an den Nachfragern, Wettbewerb und den eigenen Preisen definieren
- Basierend auf den Zielgruppen und Zielsetzungen, geeignete Kommunikationsmedien, -träger und -botschaften definieren und eine Kommunikationskampagne zum Markteintritt planen
- die geeigneten Vertriebsstrategie und -kanäle auswählen, die Stufen des Sales Processes zu planen und geeignete CRM Systeme einsetzen
- sowie Ihre Ideen zielkundengerecht (Investoren oder Kunden) überzeugend zu präsentieren (Pitch)

um Ideen für innovative Produkte und Geschäftsmodelle erfolgreich am Markt zu platzieren und zu etablieren.

Inhalt

- Grundlagen Entrepreneurial Marketing
- Marketingstrategie, Zielkundenanalyse (Value Proposition Maps)
- Produkt-, Preis-, Kommunikation- und Vertriebspolitik für innovative Produkte/StartUps
- Erfolgreich Präsentieren – Storytelling und Pitchtraining

Lehr- und Lernformen

- Seminaristischer Unterricht zu Grundlagen
- Eigene Analyse der Markteintrittsstrategie eines (erfolgreichen) StartUps (z.B. Tesla, fruitcore, Jimdo, true fruits, etc.)
- Erstellung eines Markteintrittskonzept entlang der 4 Ps für das Projekt
- Workshop Storytelling und Pitchtraining

Materialien/ Ressourcen

- Marketing Basics

Homburg (2020): Grundlagen des Marketingmanagements, 6. Auflage (German) <https://link.springer.com/book/10.1007%2F978-3-658-29638-4> Kotler et. al. (2019): Principles of marketing, international edition <https://ebookcentral.proquest.com/lib/koln/detail.action>

- Entrepreneurial Marketing

Hisrich & Ramadani (2018): Entrepreneurial Marketing – A practical managerial approach <https://www.elgaronline.com/view/978178811>

- „Fun“

Osterwalder & Pigneur (2010): Business Model Generation <https://ebookcentral.proquest.com/lib/koln/detail.action?docID=581476>

Osterwalder & Pigneur (2013): Value Proposition Design, <https://ebookcentral.proquest.com/lib/koln/detail.action?docID=1887760>

Guestimation, <https://www.degruyter.com/document/doi/10.1515/9781400824441/html>

Kurs »Business Planning 2«

Verantwortlich: Prof. Dr. Markus Linden

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: ENSI

Art: Basis

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (unbenotet - strukturiertes Feedback)

Angestrebte Lernergebnisse / Learning Outcomes

Die Studierenden können für eine innovative Produktidee ein fundiertes betriebswirtschaftliches Konzept entwickeln, indem sie

- die Rahmenwerke unterschiedlicher Geschäftsmodelle kennenlernen,
- die finanzorientierten Instrumente eines Unternehmens (Bilanz, GuV und Liquiditätsrechnung) anwenden und
- die personellen sowie rechtlichen Rahmenbedingungen evaluieren,

um eine eigene Geschäftsidee für die Markteintrittsphase wirtschaftlich fundiert vorzubereiten.

Inhalt

- Strategische Geschäftsmodellierung und -steuerung
- Externes und internes Rechnungswesen
- Personal und Organisation
- Grundlegende Rechtsformen

Lehr- und Lernformen

- Seminaristischer Unterricht zu Grundlagen
- Projektbezogene Gruppenarbeit zur Erstellung eines Business Plans

Materialien/ Ressourcen

- NUK (2018): Handbuch zum Businessplan
- Osterwalder/Pigneur (2010): Business Model Generation
- Wöhe (2023): Einführung in die Allgemeine Betriebswirtschaftslehre

Kurs »WPF z.B. Designing Futures - Design As A Language, Open Ecosystems«

Verantwortlich: Prof. Nina Juric, Prof. Dr. Laura Popplow

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: IEE / PROD / ENSI / TSCE

Art: WPF

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag, kontinuierlich während der LV und in der Modulwoche (benotet)

Angestrebte Lernergebnisse / Learning Outcomes (Beispiel Design As A Language)

“Design is more that just a few tricks to the eye - It’s a few tricks to the brain.” Neville Brody

Absolvent*innen des Kurses können Design-Systeme als “Sprache” ganzheitlich gestalten, indem sie

- das Ziel einer gestalterisch-kommunikativen Aufgabe definieren,
- in Hinblick auf dieses Ziel Frameworks konzeptionieren, die ein Regelwerk und wiederverwendbare Komponenten, Standards und Dokumentation vereinen,
- dabei der Philosophie eines kontextualisierten Prototypings folgen, das alle Dimensionen von Gestaltung berücksichtigt um die einzelnen Komponenten eines solchen Frameworks und deren Zusammenspiel zu verstehen, mit dem Ziel, eine Idee zu Realisierung und Anwendbarkeit zu führen,
- um ästhetische Kohärenz und die Kraft des eigenständigen Ausdrucks auf einer produktionstechnischen, technologischen und handwerklichen Ebene zu gewährleisten, um später in interdisziplinären Projekten ein holistisches Design-Verständnis zum Einsatz bringen zu können.

Inhalt (Beispiel Design As A Language)

- Einführung in Design-Theorien und Sprachprinzipien
- Entwicklung von Design-Systemen und Frameworks
- Konzeption von wiederverwendbaren Design-Komponenten
- Anwendung von kontextualisiertem Prototyping
- Analyse und Synthese verschiedener Designelemente zur Gewährleistung ästhetischer Kohärenz
- Interdisziplinäre Anwendung von Designlösungen

Lehr- und Lernformen (Beispiel Design As A Language)

- Impulse zu den Grundlagen und Theorien des Designs
- Praktische Übungen zur Erstellung von Design-Systemen
- Workshops zum Prototyping und Testing
- Gastvorträge von Experten aus der Industrie
- Projekte, bei denen Studierende ein eigenes Design-Konzept entwickeln und präsentieren

Materialien / Ressourcen (Beispiel Design As A Language)

- Impulsfolien und Skripte
- Zugang zu Design-Software (z.B. Adobe Creative Cloud)
- Beispiel-Frameworks und Dokumentationen
- Online-Ressourcen und Bibliotheken für Design-Inspiration
- Plattformen für die Präsentation und den Austausch von Entwürfen (z.B. Behance, Dribbble)

Kurs »WPF z.B. Developing Things - Machine Learning Basics, Blended International Project«

Verantwortlich: Prof. Dr. Christian Faubel

Sprache: deutsch, englisch

Kreditpunkte: 3 ECTS

Arbeitsaufwand: 40h Kontaktzeit / 50h Selbstlernzeit

Kompetenzcluster: DRSS / CREA / IEE / PROD / ENSI / TSCE

Art: WPF

Teilmodulprüfung: Lernportfolio / Mündlicher Beitrag , kontinuierlich während der LV und der Modulwoche (benotet)

Angestrebte Lernergebnisse / Learning Outcomes (Beispiel Machine Learning Basics)

Studierende lernen die Grundlagen des maschinellen Lernens, indem sie

- sich intensiv mit neuronalen Netzen und dem maschinellen Lernprozess auseinandersetzen,
- die erforderlichen Programmierertools zur Realisierung solcher Systeme nutzen,
- und theoretische Konzepte durch das Umsetzen in interaktive Anwendungen praktische Erfahrung sammeln,

um später eigenständig interaktive Anwendungen auf Basis neuronaler Netzwerke zu entwickeln und die Funktionsweise solcher Systeme kritisch zu reflektieren.

Studierende lernen die Grundlagen neuronaler Netze,

- indem sie sich intensiv mit dem maschinellen Lernprozess auseinandersetzen und* die erforderlichen Programmierertools zur Realisierung solcher Systeme erwerben,

um später in der Lage zu sein, interaktive Anwendungen auf Basis neuronaler Netzwerke zu entwickeln.

Inhalt (Beispiel Machine Learning Basics)

- Grundlagen neuronaler Netzwerke und ihre Bedeutung für Künstliche Intelligenz kennenlernen.
- Studium des Perzeptron-Modells als einfaches neuronales Netzwerk.
- Entwicklung einer interaktiven Webanwendung zur Erkennung handgeschriebener Zeichen.
- Anwendung ermöglicht interaktive Lernmethoden und Fehlerkorrekturen.
- Visualisierung der Anpassungen im Netzwerk während des Lernprozesses.
- Parallele Erkundung von Lernprozessen im maschinellen Lernen und persönlicher Lernerfahrungen.
- Kritische Betrachtung der aktuellen KI-Trends und Verbesserung des eigenen Lernverständnisses.
- Analyse wissenschaftlicher Arbeiten zum Thema Lernen, um ein tieferes Verständnis zu entwickeln.

Lehr- und Lernformen (Beispiel Machine Learning Basics)

- Seminaristisch

-
- Inverted Classroom
 - Projektarbeit
 - **Materialien / Ressourcen (Beispiel Machine Learning Basics)**
 - Lehrvideos und Unterrichtsmaterialien auf TH-Spaces
 - Literatur
 - Rashid, T. (2024). Neuronale Netze selbst programmieren: ein verständlicher Einstieg mit Python. O'Reilly.
 - Haykin, S. (1994). Neural networks: a comprehensive foundation. Prentice hall PTR.

Kompetenzorientierung

Handlungsfelder

Handlungsfeld »Coding Software« (CS)

Im Bereich der Software Entwicklung modellieren und abstrahieren Absolvent*innen die fachlichen Aspekte von Anwendungssystemen. Sie dekomponieren Problemstellungen in Teilprobleme, die in dedizierten Software-Komponenten durch eine Implementierung gelöst werden. Das Handlungsfeld »Coding Software« fasst die späteren beruflichen Tätigkeiten in diesem Bereich zusammen. Weitere Aktivitäten der Absolvent*innen in diesem Handlungsfeld werden beispielsweise sein:

- das Schreiben von Source Code,
- das Implementieren von Software in Teams,
- das Anwenden agiler Vorgehensmodelle,
- das Entwerfen von Software Architekturen,
- das Entwickeln von Services im Backend und von Interfaces,
- das Testen von Source Code und Komponenten,
- das Absichern von Software Systemen,
- das Bereitstellen und in Betrieb nehmen von Software-Systemen.

Studierende des Studiengangs bilden in diesem Bereich Kompetenzen, um Software von einem ersten low-fidelity Prototypen bis zum Minimal Viable Product entwickeln zu können. Diese Kompetenzen werden insbesondere den Kompetenzclustern »Software Development and Architecture« und »Agile Methoden« zugeordnet.

Handlungsfeld »Developing Things« (DT)

Im Bereich der Entwicklung digitaler Produkte realisieren die Absolvent*innen smarte Objekte, die nicht nur ein digitales Abbild, sondern auch eine dingliche Repräsentanz besitzen, bzw. mit der realen Welt korrespondieren. Zum Handlungsfeld »Developing Things« gehören des Weiteren folgende exemplarische Tätigkeiten:

- das Prototyping verschiedener Alternativen im Entwicklungsprozess,
- die Verwendung von Sensoren, um Input der dinglichen Realität zu messen,
- die Nutzung von Aktoren, um eine physikalische Wirkung zu erzeugen,
- das Implementieren von Software für Mikrocontroller, Kleinstcomputer und mobile Devices,
- die gestalterische Arbeit mit den Wechselbeziehungen programmierter und materieller Eigenschaften smarter Objekte,
- das integrative Entwerfen von Code und Material,
- das Reflektieren von Funktionalität und Ästhetik,
- das Implementieren von smartem Systemverhalten auf Basis künstlicher Intelligenz,
- die Vernetzung physikalisch verteilter Komponenten,
- die Realisierung von Clients auf Basis einer natürlichen Mensch-Computer-Interaktion.

Für das berufliche Handlungsfeld »Developing Things« bilden die Studierenden im Studiengang Kompetenzen, um dingliche und realweltliche Artefakte mit digitalen Systemen vernetzen zu können. Diese Kompetenzen werden insbesondere den Kompetenzclustern »Dual Reality and Smart Spaces« und »Creative Coding zugeordnet«.

Handlungsfeld »Designing Futures« (DF)

Absolvent*innen erfassen die ganzheitliche Wirkung von interdisziplinären Ideen, Phänomenen, Szenarien, Visionen, Hypothesen, Produkten und deren Wirkungsweisen. Sie stellen sich den Herausforderungen einer sich rapide und nachhaltig wandelnden Gesellschaft, Kultur, Politik und Wirtschaft. Ein ganzheitliches Designverständnis und Ansätze wie Integrated Design, Critical Design, Speculative Design und Design Research befähigen sie, Erfahrungen in mehreren Dimensionen - auch jenseits von UI-, UX- oder HCI-Themen - zu gestalten und die kritische Auseinandersetzung mit relevanten Themen zu suchen. Zu den Tätigkeiten in diesem Handlungsfeld gehören exemplarisch:

- das Erfassen von Kontexten,
- das kritische Hinterfragen und spekulative Entwickeln neuer Produktideen,
- der Einsatz künstlerischer Ansätze zur kreativen Gestaltung,
- die Stimulation kreativen Denkens zur Ideenbildung und Förderung von Kreativprozessen,
- das Beobachten und Verstehen sozio-kultureller Phänomene,
- die Entwicklung und Darstellung zukunftsrelevanter Szenarien, basierend auf der spekulativen Wahrnehmung der Bedürfnisse der Nutzer*innen von morgen,
- Design im Dialog mit verschiedenen Stakeholdern, durch den Einsatz von Participatory-Design- und Co-Design-Methoden,
- das Prototyping in verschiedenen Dimensionen unter Verwendung verschiedener Prototyping-Methoden,
- das Experimentieren mit Material, Modell, Methoden, Wissenschaft, Ethik und Philosophie.

Für das berufliche Handlungsfeld »Designing Futures« entwickeln die Studierenden im Studiengang Kompetenzen, um innovative Produkte und ihre Wirkung ganzheitlich gestalten zu können. Diese Kompetenzen lassen sich den Clustern »Interaction, Empathy and Emotion«, »Technical Society, Culture and Ethics«, »Communication« und »Creative Coding« zuordnen.

Handlungsfeld »Creating Impact« (CI)

Absolvent*innen entwerfen, entwickeln und evaluieren digitale Produkte und Applikationen, die auf soziale, kulturelle und ökonomische Phänomene und Entwicklungen reagieren und einwirken. Sie erkennen gesellschaftliche und wirtschaftliche Strömungen, können diese bewerten und daraus Potenziale für digitale Produkte und soziale Innovationen ableiten. Zu den Tätigkeiten in diesem Gebiet gehören:

- das Entwickeln und Bewerten von Ideen für neuartige Produkte und Prozesse,
- das Reflektieren dieser Produkte und Prozesse in ihrer gesellschaftlichen Rolle (Technikfolgenabschätzung, STS: Science Technology and Society),
- das Erproben von Zwischenständen mit realen Nutzern (Prototyping),
- das Entwickeln von Lösungen unter Einbeziehung dieser Nutzer (partizipative Strategien),
- die Nutzbarmachung dieser Lösungen als kommerzielles Produkt, Dienstleistung, freie Software oder kollaborative Anstrengung einer Community,
- das Entwickeln von Geschäfts- und Betreibermodellen, also betriebswirtschaftlichen Konzepten zum Aufbau von Unternehmen, Communities und anderer Strukturen für ein Produkt oder eine Dienstleistung,

-
- Konzeption, Design und Entwicklung von Kommunikation und Produkten aus den Perspektiven relevanter Stakeholder (wie bspw. User*innen, Kund*innen, Designer*innen, Coder*innen, Bürger*innen, Staat, Börse),
 - das Hervorbringen sozialer Innovationen auf Basis vernetzter digitaler Applikationen,
 - das Gründen von Start-ups und alternativer Strukturen (non-profit, NGO) zum Erproben von Geschäftsideen,

Für das berufliche Handlungsfeld »Creating Impact« entwickeln die Studierenden im Studiengang Kompetenzen, um relevante Produkte und Applikation zu entwickeln, ihre Ergebnisse nachhaltig Dritten zur Verfügung zu stellen und diese in ökonomischen und kulturellen Kontexten wirksam werden zu lassen. Daher steht dieses Handlungsfeld mit den Kompetenzclustern »Product Dimensions«, »Technical Society, Culture and Ethics« und »Entrepreneurship and Social Innovation« in Beziehung.

Kompetenzcluster

Kompetenzcluster »Software Development and Architecture« (SDA)

Die Absolvent*innen können große Software-Systeme professionell designen, kontinuierlich weiterentwickeln und neue Versionen zügig produktiv setzen, indem Sie

- Grundprinzipien des »Computational Thinking« kennen und anwenden,
- Systeme mit fortschrittlichen Mitteln gängiger Programmiersprachen, Libraries, Frameworks und Tools und Beachtung akzeptierter Richtlinien und Best Practices codieren,
- Techniken zur Realisierung webbasierter, verteilter und mobiler Systeme anwenden, die dem Stand der Technik bzgl. Usability, Sicherheit, Robustheit, Skalierbarkeit etc. entsprechen,
- Tools zur Automatisierung von Software-Builds, -Qualitätssicherung und -Deployments einsetzen,

um später qualitativ hochwertige digitale Produkte realisieren und diese ständig verbessern zu können.

Kompetenzcluster »Dual Reality and Smart Spaces« (DRSS)

Die Absolvent*innen können dinglich-digitale und smarte Produkte entwickeln, indem sie

- die dingliche Welt durch Sensoren erfassen und durch Aktoren beeinflussen,
- das Eigenverhalten und die »Programmierbarkeit« von Materialien ausnutzen,
- Sensoren und Aktoren mit Microcomputern steuern und der Cloud vernetzen,
- die Daten der Sensoren durch Methoden, Verfahren und Algorithmen der künstlichen Intelligenz verarbeiten und Aktoren entsprechend steuern,
- informationsverarbeitende Computer physikalisch und kognitiv in Produkte einbetten,
- digitale Produktionstechniken für die Herstellung von Prototypen anwenden,

um später smarte innovative Produkte mit einer ganzheitlichen Perspektive auf Software und Hardware zu entwickeln und technisch handlungsfähig zu sein.

Kompetenzcluster »Interaction, Empathy and Emotion« (IEE)

Die Absolvent*innen können Geschichten erzählen und diese emotionalen Werte aktiv gestalten, indem sie

- Theorien, Thesen, Mechanismen, Muster, Materialien und Positionen, die das Themenfeld Interaction, Empathy and Emotion ausmachen, kennenlernen,
- inter- & transdisziplinäre Konzepte zu Verhaltensmechanismen im Feld des Interaction Designs und der Interactive Environments untersuchen (Case Studies) und sich kritisch mit diesen auseinandersetzen,
- Methoden wie Design Fiction, Fiction as a Method, Storytelling und Telling in Time kennen und für Fiktionen und Narrationen anwenden
- durch Kenntnisse im Bereich Body and Embodiment eigene emotionale Interaktion durch körperliche Bewegung, darstellende Künste (Tanz, Schauspiel), Form und Materialstudien (Soft & Hard) explorieren,

- dabei Raum- & Zeitkonzepte kennenlernen und in die Gestaltung mit einfließen lassen (Animationen / Conversational Interfaces / Audio-Visualität etc.),
- Sensationen und Phänomene verbalisieren und in Materialität und Interaktionen übersetzen,
- dabei das komplexe Zusammenspiel von Wahrnehmung, Handlung und Kontext in der Interaktion und dessen Wirkung auf Ästhetik und Bedeutung digitaler Produkte berücksichtigen,
- Charakterstudien / Kriterien und formästhetische Aspekte kennenlernen,
- Prototyping als Skizzenbuch praktizieren (in Soft- & Hardware),
- Experimente mit Material hinterfragen,
- die Klaviatur der Dimensionen von Wahrnehmung, Vorstellung und Darstellung beherrschen,
- gezielte Planung und Gestaltung von Verhaltensweisen konzipieren lernen,
- in der Lage sind, zukünftige Nutzende als gleichberechtigte Partner in den Designprozess einzubeziehen und
- sich mit sozio-affektiven und sozio-kognitive Prozessen befassen / sozio-affektive als auch sozio-kognitive Funktionen als Bedingung für prosoziales Verhalten erkennen und anwenden / adaptives Sozialverhalten als ein Ergebnis des dynamischen Zusammenspiels von sozio-affektiven und sozio-kognitiven Prozessen verstehen

um Dinge, Ideen, Produkte und Geschäftsmodelle zu beseelen und soziale Innovation im Rahmen von gesellschaftlichen Kontexten mit neuen Kommunikations- und Interaktionsformen nachhaltig zu verbessern.

Kompetenzcluster »Product Dimensions« (PROD)

Die Absolventen*innen können einem Produkt und ihrem Ecosystem auf den Grund gehen, indem sie

- Denkweisen, Herangehensweisen, Wirkungsweisen eines Produktes oder einer Produktwelt verstehen und verändern,
- neue Dimensionen des Produktbegriffs kennen und deklinieren,
- den sozio-kulturellen Kontext der Produktentwicklung in Betracht ziehen,
- den Product-Launch in eine Zukunft abstrahieren können,
- unterschiedliche Produktebenen multiperspektivisch interpretieren, planen und gestalten,
- Sustainability, Durability und Recycling verstehen und planen,
- soziale Verantwortung als Kern jeder Produktentwicklung ansehen,
- und die Wichtigkeit dieses Umdenkens als Grundstein für soziale Innovation verstehen,

um einen Unterschied in der Zukunft von Geschäftsmodellen, unserer Gesellschaft und unserer Umwelt zu machen.

Kompetenzcluster »Agile Methoden« (AGILE)

Die Absolvent*innen können Prozesse zur Entwicklung innovativer digitaler Produkte in heterogenen Teams agil managen, indem sie

- agile inkrementell-iterative Vorgehensmodelle anwenden,
- Kunden in den Entwicklungsprozess durch Methoden des Design Thinking, Co-Creation und Participatory Design einbeziehen,
- Kreativtechniken zur Moderation von Gruppen zur Ideenfindung und Variantenbildung nutzen,
- in selbstorganisierten Teams mitarbeiten,
- und verschiedene Fragestellungen in Projekten durch Prototypen beantworten,

um Projekte mit visionären Zielen aber unklaren und vagen Anforderungen zu managen und dabei Erkenntnisse, die während des Projekts gewonnen werden, in den Entwicklungsprozess zu integrieren.

Kompetenzcluster »Communication« (COMM)

Die Absolventen*innen können miteinander und über ihr Tun versiert sprechen und schreiben. Sie haben ein tieferes Verständnis des Themenkomplexes der Kommunikation im Rahmen von Mensch-Maschine Interaktion. Sie können soziale Innovationen vermittelt, indem sie

- Komplexes einfach machen,
- mit entsprechendem Vokabular ausgestattet werden und dieses erlernen und anwenden,
- den Paradigmenwechsel der Terminologien und Infrastrukturen innerhalb der Handlungsfelder kritisch hinterfragen, reflektieren und mitgestalten,
- Marketingprozesse und Mechanismen verstehen, und optimieren,
- Designprozesse in Form von Kommunikation durch verschiedene (neue) Medienformate befördern
- ein sinnloses von einem sinnvollem Produkt / Prozess / Geschäftsmodell unterscheiden,
- Interdisziplinarität praktisch entdecken, ausleben und im Anschluss dokumentieren,
- soziale Kompetenzen im Sinne einer verbesserten Umwelt und Kultur ausbauen
- und Lernen das Richtige zu pitchen

um für die Notwendigkeit einer verstärkten Zusammenarbeit in der nachhaltigen Welt von morgen und am »Arbeitsplatz der Zukunft« insbesondere zwischen Kreativ-, Wirtschafts- und Technologieabteilungen gewappnet zu sein und diese Kommunikationsprozesse und -terminologien mitzuerschaffen und auszugestalten.

Kompetenzcluster »Technical Society, Culture and Ethics« (TSCE)

Die Absolvent*innen können ihr Handeln im Kontext einer digitalisierten, vernetzten und demokratischen Gesellschaft, Kultur und Wirtschaft nachhaltig gestalten und reflektieren, indem sie

- den Wert offener Software-Entwicklungen (Open Source) kennenlernen und durch eigene offene Projekte neue Werte schaffen,
- die Bedeutung des Open Web für demokratische Gesellschaften einschätzen und seine Bedrohung durch proprietäre Entwicklungen exemplarisch analysieren,
- soziale und kulturelle Entwicklungen im Kontext der Digitalisierung untersuchen und insbesondere auch Gefährdungen, wie den Wegfall sicherer Arbeitsverhältnisse, analysieren,
- Übersichtswissen zu wichtigen juristischen Fragestellungen im Zusammenhang mit der Erstellung digitaler Systeme (z.B. Einbindung fremder Bestandteile unter GNU-Lizenz) und ihrem Betrieb (z.B. Haftungsfragen bei Störungen) erwerben und exemplarisch auf eigene Projekte anwenden,
- ethische Fragestellungen im Zusammenhang mit neuen digitalen Entwicklungen und Produkten aufwerfen, im Lichte von bekannten Problemfällen diskutieren und Alternativen abwägen,
- durch ihr Verständnis für soziokulturelle Phänomene Zusammenhänge, Gefahren und Chancen über Fachgrenzen hinweg erkennen und bewerten,

um als soziale Innovatoren gesellschaftliche Potenziale für digitale Produkte und Services zu erkennen und diese als kritische Akteure nachhaltig zu gestalten. Sie werden damit ihrer Verantwortung als »Global Citizen« in diversen sozio-kulturellen Kontexten gerecht werden.

Kompetenzcluster »Creative Coding« (CREA)

Die Absolvent*innen haben eine kreative Arbeitsweise und können auch durch unkonventionellem Prozesse abseits der herkömmlichen Wege zu Lösungen gelangen. Sie verstehen Coding als kreative Disziplin um Klänge, Bilder, Animationen und synästhetische, oftmals echtzeitgenerierte Interaktionen zu erzeugen. Absolvent*innen können sich kreativ mit Code ausdrücken indem sie

- künstlerische Methoden, Künstler*innen und ihre Werke / Positionen kennenlernen,
- die Historie, Einflüsse und Strömungen von Open Source und Maker-Bewegungen kennen und daran anknüpfen,
- an offenen und kreativen Communities partizipieren und eigene Beiträge leisten,
- die Grenzen des technisch Machbaren challengen und durch unkonventionelle Herangehensweisen lösen oder umgehen,
- bestehende Tools und Werkzeuge kennen und als Baukasten nutzen und einem »Do it Yourself«-Ansatz folgend sich eigene Werkzeuge entwickeln wo nötig,
- bewusst unkonventionelle Herangehensweisen und Artefakte in geänderten Kontexten einsetzen,
- Code als Mittel zur Realisierung von Funktionalität aber auch zum Ausdruck von Ästhetik nutzen,
- ästhetische Gesichtspunkte analysieren und einschätzen,
- die Kopie vom Original unterscheiden,
- Medien flexibel ineinander verschachteln,
- alle Sinne miteinbeziehen,
- Werke audio-visueller Kunst und Design erstellen, wie z.B. Live-Visuals, Sound Art, Maschinen-Installationen, Multiscreen-Projektionen, Synthesizer oder Roboter,
- kommerzielle Medieninszenierungen und Produktinstallationen entwickeln,
- und Produktprototypen simulieren,

um eine eigene Formsprache zu entwickeln und als neugierige Gestalter*innen kreative Lösung für herausfordernde Fragestellungen zu finden.

Kompetenzcluster »Entrepreneurship and Social Innovation« (ENSI)

Die Absolvent*innen können Geschäftsmodelle für digitale Produkte entwickeln, bewerten und in frühen Phasen unternehmerisch umsetzen, indem sie

- zukunftsrelevante Szenarien darstellen und spekulativ die Bedürfnisse der Nutzer*innen von morgen wahrnehmen,
- wirtschaftliche und gesellschaftliche Potenziale für Produkte über Fachgrenzen hinweg erkennen und bewerten,
- Betriebs- und Geschäftsmodelle konzipieren und bewerten,
- Produkte oder Dienstleistungen als Minimum Viable Product früh an den Markt bringen und dort evaluieren,
- betriebswirtschaftliche Grundlagen zur Gründung und für den Betrieb von StartUps anwenden,
- ökonomisch überzeugend und informationsbelegt präsentieren,

um ihre eigenen Innovationen nachhaltig in die Welt zu bringen und als soziale Innovatoren zu wirken.

Kompetenzcluster »Soziale Kompetenz und Selbstkompetenz« (SOSE)

Die Absolvent*innen sind in der Lage sich selbstständig neue Methoden und Wissen anzueignen und zu erkennen, welches Wissen für sie relevant ist, indem sie wissenschaftlich, analytisch und reflektiert arbeiten. Sie kennen ihre berufliche Rolle in zivilgesellschaftlicher, politischer und kultureller Hinsicht, sowie die damit verbundenen Erwartungen und ggf. vorhandene Rollenkonflikte. Sie können zur Konfliktlösung beitragen, indem sie Kenntnisse zum Konfliktmanagement wirksam machen, in kontroversen Diskussionen zielorientiert argumentieren und mit Kritik sachlich umzugehen.