



Certificate of Advanced Studies

# Agile Software Engineering

Kundenorientierte, zuverlässige und gut wartbare Software erfordert eine konsequente, methodische Vorgehensweise über den ganzen Lebenszyklus. Dieses CAS vermittelt alle Kompetenzen dazu – der ganze Produktentwicklungsprozess von der Bedarfserhebung über das Design und die Entwicklung bis zur erfolgreichen Lieferung werden in praxisorientierten Lernelementen vermittelt.

# Inhaltsverzeichnis

1	Umfeld	3
2	Zielpublikum	3
3	Ausbildungsziele	3
4	Voraussetzungen	3
5	Unterrichtssprache	3
6	Durchführungsort	3
7	Kompetenzprofil	4
8	Kursübersicht	4
9	Kursbeschreibungen	5
	9.1 Requirements Engineering	5
	9.2 Software-Analyse und -Design mit UML	6
	9.3 Analyse und Design Pattern	6
	9.4 Agile Produkteentwicklung	7
	9.5 Projektarbeit	7
10	Kompetenznachweis	8
11	Lehrmittel	8
12	Dozierende	8
13	Organisation	9

Stand: 07.01.2025

# 1 Umfeld

Digitale Produkte sind aus unserem Alltag nicht mehr wegzudenken. Die [Megatrends](#) wie Individualisierung, Digitalisierung und Globalisierung bewirken, dass sich die Bedürfnisse an digitale Produkte in raschem Tempo wandeln. Zusätzlich werden in den Organisationen die Geschäftsprozesse und die IT-Systemlandschaft immer komplexer. Diese erhöhte Komplexität bedingt ein radikales Umdenken, wie Software und damit digitale Produkte produziert werden. Der heutige Industriestandard ist die agile Produktentwicklung.

In der agilen Produktentwicklung werden in kleinen Schritten auslieferbare Produktteile erstellt. Diese müssen den etablierten Qualitätskriterien wie wartbar und erweiterbar entsprechen. Im CAS agile Software Engineering lernen Sie, wie dank fokussieren, validieren, lernen und exzellenten Software-Engineering-Praktiken in einem komplexen Umfeld erfolgreich Software-Produkte entstehen können.

## 2 Zielpublikum

Das CAS Software Engineering richtet sich an Fachkräfte mit IT-Background wie Softwareentwickler\*innen, Projektleiter\*innen, Business-Analyst\*innen und Tester\*innen, die sich fundierte Kenntnisse in diesem Bereich aneignen wollen.

## 3 Ausbildungsziele

- Sie sind in der Lage, anspruchsvolle Softwareprodukte gemäss den heutigen Industrie-Standards des Software Engineerings zu spezifizieren, analysieren, modellieren, planen, entwickeln, testen und führen.
- Sie kennen die Vorgehensweise in der objektorientierten Analyse und im Design nach UML.
- Sie können digitale Produktentwicklungen und das Entwicklungsteam adaptiv, effektiv und effizient führen.
- Dank der praxisorientierten Vermittlung eignen Sie sich bereits in der Ausbildung Best Practices an.

## 4 Voraussetzungen

- Sie haben Programmiererfahrung in einer objektorientierten Sprache.
- Sie haben bereits bei agilen oder klassischen Produktentwicklungen mitgearbeitet.

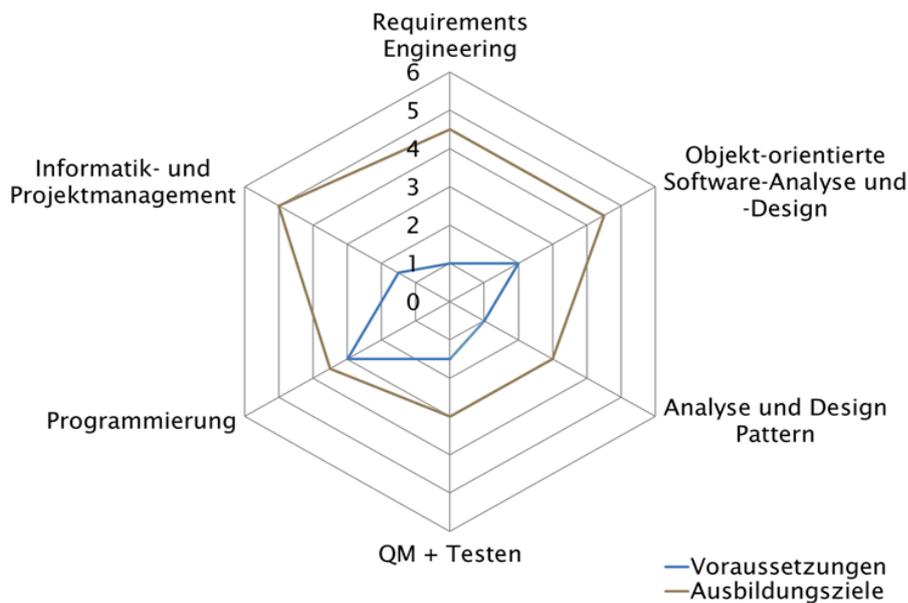
## 5 Unterrichtssprache

Die Unterrichtssprache ist Deutsch, die Unterlagen sind teilweise in Englisch.

## 6 Durchführungsort

Berner Fachhochschule, Weiterbildung, Aarbergstrasse 46 (Switzerland Innovation Park Biel/Bienne), 2503 Biel,  
Telefon +41 31 848 31 11, E-Mail [weiterbildung.ti@bfh.ch](mailto:weiterbildung.ti@bfh.ch).

## 7 Kompetenzprofil



### Kompetenzstufen

1. Kenntnisse/Wissen
2. Verstehen
3. Anwenden
4. Analyse
5. Synthese
6. Beurteilung

## 8 Kursübersicht

Kurs / Lehreinheit	Lektionen	Stunden	Dozierende
Requirements Engineering	24		Rudolf Gysi
Software-Analyse und -Design mit UML	44		Beatrice Amrhein
Analyse und Design Pattern	20		Beatrice Amrhein
Agile Produktentwicklung	48		Alexander Schley
Integrations-Projekt	20	~ 100	Beatrice Amrhein Alexander Schley
<b>Total</b>	<b>156</b>	<b>~ 100</b>	

Das CAS umfasst insgesamt 12 ECTS-Credits. Für die einzelnen Kurse ist entsprechend Zeit für Selbststudium, Prüfungsvorbereitung etc. einzurechnen.

## 9 Kursbeschreibungen

Nachfolgend sind die einzelnen Kurse dieses Studienganges beschrieben.

Der Begriff Kurs schliesst alle Veranstaltungstypen ein, es ist ein zusammenfassender Begriff für verschiedene Veranstaltungstypen wie Vorlesung, Lehrveranstaltung, Fallstudie, Living Case, Fach, Studienreise, Semesterarbeiten usw.

### 9.1 Requirements Engineering

Allgemein	Die Bedürfnisse des Kunden stehen bei jeder Produktentwicklung im Zentrum. Diese werden mit diversen Praktiken, die unter Requirements Engineering zusammengefasst werden, erhoben. Anschliessend werden sie ausformuliert und priorisiert. Diese Anforderungen dienen als Grundlage für die Analyse, das Design sowie für die Softwareentwicklung. In diesem Kurs werden sowohl die klassischen als auch die agilen, neuen Praktiken vermittelt.
Lernziele	Die Teilnehmenden: <ul style="list-style-type: none"> <li>- können die aktuellen Praktiken in der klassischen sowie in der agilen Anforderungserhebung anwenden.</li> <li>- werden handlungsfähig im Requirements Engineering.</li> <li>- sind in der Lage, Stakeholder zu identifizieren und ein adäquates Stakeholder-Management sowie Change Management durchzuführen.</li> <li>- Wissen, wie Anforderungen und technische Exzellenz zusammenhängen.</li> </ul>
Themen und Inhalte	<ul style="list-style-type: none"> <li>- Von der Idee zum Produkt aus der Sicht eines Product Owners</li> <li>- kundenzentrierte Produktentwicklung</li> <li>- Requirements-Engineering-Prozess</li> <li>- klassische Methoden und Modell in der Übersicht</li> <li>- agile Methoden des Anforderungsmanagements</li> <li>- Personas erstellen und validieren</li> <li>- Product Canvas, Start einer Produktentwicklung</li> <li>- Customer Journey anhand einer Story Map pflegen</li> <li>- Product Backlog erstellen und verwalten</li> <li>- Sprint Planning und Aufwandschätzung</li> <li>- Velocity-Messung und Forecasting</li> <li>- agiles Projektcontrolling und Monitoring</li> <li>- kontinuierliche Integration und Testautomatisierung</li> <li>- Pair Programming und Code Reviews</li> <li>- Test-driven Development (TDD) und Behavior-driven Development (BDD)</li> <li>- agile Architekturprinzipien und Design Patterns</li> <li>- agile Teamarbeit und Kommunikation: <ul style="list-style-type: none"> <li>- Teamrollen und Verantwortlichkeiten in agilen Teams</li> <li>- Abnahmekriterien und Acceptance Tests</li> <li>- Qualitätskriterien einer guten Software Requirements Spezifikation</li> <li>- Stakeholder-Management</li> <li>- effektive Kommunikation und Zusammenarbeit im agilen Umfeld</li> <li>- Produkt- und Release-Management in agilen Umgebungen</li> <li>- Kundenfeedback und Nutzerzentrierung</li> <li>- Produktvalidierung und kontinuierliche Verbesserung</li> <li>- agiles Change Management</li> </ul> </li> </ul>
Lehrmittel	Folien/Skript am Ende der Lektionen als PDF aus Miro

## 9.2 Software-Analyse und -Design mit UML

Allgemein	Analyse und Design sind Grundvoraussetzung für eine erfolgreiche Projekt-Realisierung. Diese Entwicklungsschritte werden anhand von UML (Unified Modeling Language) vermittelt und geübt. UML ist zum de facto Standard geworden und wird im Kurs erlernt.
Lernziele	Die Teilnehmenden: <ul style="list-style-type: none"> <li>– haben gute Kenntnisse des Software-Design-Vorgehens</li> <li>– sind in der Lage, kleinere Projekte selbstständig zu analysieren, zu entwerfen und mit UML zu modellieren und zu dokumentieren.</li> </ul>
Themen und Inhalte	<ul style="list-style-type: none"> <li>– Grobanalyse-Modell</li> <li>– OO-Analyse und OO-Design</li> <li>– Produktdatenblatt/Product Vision Box</li> <li>– Use-Case-Modell und -Beschreibung</li> <li>– Aktivitätsdiagramm</li> <li>– Zustandsdiagramm</li> <li>– Klassendiagramm</li> <li>– Generalisierung/Spezialisierung</li> <li>– Sequenzdiagramm</li> <li>– Paketdiagramm</li> <li>– Komponentendiagramm</li> </ul>
Lehrmittel	<ul style="list-style-type: none"> <li>– Folien/Skript</li> <li>– Literaturempfehlung Nr. 2</li> </ul>

## 9.3 Analyse und Design Pattern

Allgemein	<p>Design Pattern sind bewährte Lösungsschablonen für wiederkehrende Entwurfsprobleme in der Softwarearchitektur und -entwicklung. Sie stellen damit eine wiederverwendbare Vorlage zur Problemlösung dar, die in einem bestimmten Zusammenhang einsetzbar ist.</p> <p>Der primäre Nutzen eines Design Pattern liegt in der Beschreibung einer Lösung für ein Entwurfsproblem. Da jedes Pattern einen Namen hat, wird es damit möglich, abstrakt über verschiedene Lösungsansätze zu diskutieren.</p> <p>Über das Erlernen der klassischen OO-Patterns hinaus erhalten die Teilnehmenden einen Einblick in aktuelle Architekturpattern moderner Enterprise-Anwendungen.</p>
Lernziele	Die Teilnehmenden können in mittleren und grösseren Projekten als kompetente und sachkundige Diskussionspartner beim Finden eines sauberen Designs mithelfen.
Themen und Inhalte	<ul style="list-style-type: none"> <li>– Strukturmuster</li> <li>– Verhaltensmuster</li> <li>– Persistenzmuster</li> <li>– Erzeugungsmuster</li> <li>– Architekturmuster</li> </ul>
Lehrmittel	<ul style="list-style-type: none"> <li>– Folien/Skript</li> <li>– Literaturempfehlung Nr. 1</li> </ul>

## 9.4 Agile Produkteentwicklung

Lernziele	<p>Die Teilnehmenden:</p> <ul style="list-style-type: none"> <li>– erhalten tiefes Verständnis über Cynefin, Scrum, Kanban, SAFe, PDSA, Deming, TPS, Tuckman und Taylorismus in der Software-Entwicklung.</li> <li>– verfügen über einen umfassenden Einblick in die Wissens- und Erfahrungsbereiche des agilen Projektmanagements.</li> <li>– verstehen die Unterschiede zwischen agilen, klassischen und hybriden Ansätzen</li> <li>– können die wichtigsten Methoden und Techniken des agilen Projektmanagements im Integrationsprojekt einsetzen und anwenden.</li> <li>– können ein Projektteam zusammenstellen und mit agilen Methoden zum Projekterfolg führen.</li> </ul>
Themen und Inhalte	<ul style="list-style-type: none"> <li>– Rollen, Artefakte und Meetings in Scrum</li> <li>– Mit Kanban die Arbeitsprozesse in Teams effizient gestalten</li> <li>– Dank agilem Projektstart früher mit der Umsetzung beginnen</li> <li>– Risiken minimieren durch agile Planung</li> <li>– Aufwandsschätzung</li> <li>– Projektcontrolling</li> <li>– Mit agiler Produktentwicklung marktgerechte Lösungen bauen</li> <li>– Die richtige Teamkonstellation als Voraussetzung für Selbstorganisation</li> <li>– Retrospektiven als Grundlage des systemischen Regelkreises</li> </ul>
Lehrmittel	Folien/Skript

## 9.5 Projektarbeit

Allgemein	<p>Eine Aufgabenstellung aus dem Umfeld der Studierenden wird zur Festigung des Gelernten als Gruppenarbeit spezifiziert, analysiert und entworfen sowie mittels der Realisierung eines Prototyps oder eines Minimum Viable Products verifiziert. Für die Implementation werden Programmiersprache und Entwicklungsumgebung innerhalb der Arbeitsgruppen definiert.</p> <p>Zur Beurteilung der Arbeit werden Software Requirement, Analyse und Design, DB- und GUI-Design, sowie das Projektmanagement hinzugezogen.</p>
Lernziele	Die Teilnehmenden führen alle Stufen des Software Engineerings im Kontext eines anspruchsvollen Projektes aus und festigen damit das Gelernte in den Kursen des CAS.
Themen und Inhalte	<p>Anwendung der verschiedenen Themen des Software Engineering, im Besonderen:</p> <ul style="list-style-type: none"> <li>– Requirements Engineering</li> <li>– objektorientierte Analyse und Design nach UML</li> <li>– agile Produktentwicklung</li> <li>– Management und Testen von Informatik-Projekten</li> <li>– Validierung und Verifikation</li> <li>– Präsentation</li> </ul>
Lehrmittel	Folien/Skript

## 10 Kompetenznachweis

Für die Anrechnung der 12 ECTS-Credits ist das erfolgreiche Bestehen der Qualifikationsnachweise (Prüfungen, Projektarbeiten) erforderlich, gemäss folgender Aufstellung:

Kompetenznachweis	Gewicht	Art der Qualifikation	Erfolgsquote Studierende
Requirements Engineering	2.0	Prüfung	0 - 100 %
Software-Analyse und -Design mit UML	2.5	Prüfung	0 - 100 %
Analyse und Design Pattern	1.0	Gruppenarbeit	0 - 100 %
Agile Produktentwicklung	2.5	Prüfung	0 - 100 %
Integrationsprojekt	2.0	Gruppenarbeit	0 - 100 %
Gesamtgewicht/Erfolgsquote	10		0 - 100 %

Der gewichtete Mittelwert der Erfolgsquoten der einzelnen Kompetenznachweise wird in eine Note zwischen 3 und 6 umgerechnet. Die Note 3 (gemittelte Erfolgsquote weniger als 50%) ist ungenügend. Die Noten 4, 4.5, 5, 5.5 und 6 (gemittelte Erfolgsquote zwischen 50% und 100%) sind genügend.

## 11 Lehrmittel

Die nachfolgend aufgeführten Lehrmittel sind wesentlich für das Lernen während des geführten Unterrichtes. Sie sind durch die Studierenden zu beschaffen.

Nr.	Titel	Autoren	Verlag	Jahr	ISBN-Nr.
1.	Patterns kompakt	Karl Eilbrecht, Gernot Starke	Springer	2018	978-3-662-57936-7

Für das Einlesen und als Begleitmaterial werden nachfolgend aufgeführte, ergänzende Bücher empfohlen. Die Beschaffung liegt im Ermessen der Studierenden.

Nr.	Titel	Autoren	Verlag	Jahr	ISBN-Nr.
2.	UML 2.5, Das umfassende Handbuch	Christoph Kecher	Rheinwerk Verlag	2021	978-3-8362-8447-9

## 12 Dozierende

Vorname Name	Firma	E-Mail
Beatrice Amrhein	BFH	beatrice.amrhein@bfh.ch
Alexander Schley	SBB	alexander.schley@bfh.ch
Rudolf Gysi	Wertwandler	rudolf.gysi@bfh.ch

## 13 Organisation

### **CAS-Leitung:**

Prof. Dr. Beatrice Amrhein

Tel: +41 31 848 32 76

E-Mail: [beatrice.amrhein@bfh.ch](mailto:beatrice.amrhein@bfh.ch)

### **CAS-Administration:**

Andrea Moser

Tel: +41 31 848 32 11

E-Mail: [andrea.moser@bfh.ch](mailto:andrea.moser@bfh.ch)

Während der Durchführung des CAS können sich Anpassungen bezüglich Inhalten, Lernzielen, Dozierenden und Kompetenznachweisen ergeben. Es liegt in der Kompetenz der Dozierenden und der Studienleitung, aufgrund der aktuellen Entwicklungen in einem Fachgebiet, der konkreten Vorkenntnisse und Interessenslage der Teilnehmenden, sowie aus didaktischen und organisatorischen Gründen Anpassungen im Ablauf eines CAS vorzunehmen.

### **Berner Fachhochschule**

Weiterbildung

Aarbergstrasse 46 (Switzerland Innovation Park Biel/Bienne)

2503 Biel

Telefon +41 31 848 31 11

E-Mail: [weiterbildung.ti@bfh.ch](mailto:weiterbildung.ti@bfh.ch)

[bfh.ch/ti/weiterbildung](http://bfh.ch/ti/weiterbildung)

[bfh.ch/cas-aswe](http://bfh.ch/cas-aswe)