

Kompetenzprofil, Business Outcomes und Lernziele

UXQCC Certified Professional in Software Usability with GenAI, Foundation Level (Deutsch)

Quelle: Syllabus UXQCC_GenAI_V1_013_DE (Build 1.013). Verbindlich ist der Syllabus; dieses Dokument fasst Kompetenzprofil, Business Outcomes und Lernziele als Referenz für Prüfungsfragen, Trainingsmaterialien und Kommunikation zusammen. Umfang: 6 generelle Business Outcomes, 37 kapitelbezogene Business Outcomes, 86 Lernziele (14 auf Stufe K1, 42 auf K2, 30 auf K3).

1. Was Absolventinnen und Absolventen können

Absolventinnen und Absolventen des Foundation Level können Usability-Arbeit mit Hilfe von GenAI früher, pragmatischer und systematischer in ihrem Softwareentwicklungsteam verankern, auch ohne dauerhaft verfügbare UX-Spezialrolle. Sie kennen die zentralen Begriffe und Qualitätskriterien der Usability und wenden sie auf ihre eigenen Entwicklungsartefakte an, von der User Story über UI-Entwurf, Prototyp und Code bis zu Test und Monitoring.

Das Kompetenzprofil richtet sich auch an UX-Spezialistinnen und UX-Spezialisten. Sie lernen, ihr vorhandenes UX-Wissen wirksam in den Softwareentwicklungsprozess einzubringen: Sie übersetzen Usability-Anliegen in die Arbeitsformen und Artefakte des Entwicklungsteams, etwa in User Stories, Akzeptanzkriterien, Definition of Ready und Definition of Done, Prüfungen und Review-Formate, und formulieren sie so, dass Teammitglieder ohne UX-Ausbildung, insbesondere Softwareentwicklerinnen und Softwareentwickler, Product Owner und Tester, sie verstehen, einordnen und in ihrer täglichen Arbeit anwenden können. Dazu gehört auch der gezielte Einsatz von GenAI, um Usability-Wissen teamgerecht aufzubereiten, sowie ein gemeinsames Begriffsverständnis, das die Zusammenarbeit zwischen UX-Expertise und Entwicklung trägt.

Im Einzelnen können Absolventinnen und Absolventen:

- GenAI entlang des gesamten agilen Entwicklungsprozesses als Werkzeug für Lernen, Strukturierung, Entwurf, Prüfung und Auswertung einsetzen: in Discovery, Backlog Refinement, Planung, Design und Prototyping, Implementierung, Evaluation und Monitoring.
- AI-gestützte Usability-Ergebnisse kritisch bewerten und dabei zwischen erkannten Mustern, plausiblen Hypothesen, AI-generierten Vorschlägen und tatsächlich validierten Erkenntnissen unterscheiden.
- grundlegende Usability-Kriterien wie Wahrnehmung, Verständlichkeit, kognitive Belastung, Accessibility und erwartbares Verhalten in Anforderungen, Backlog Items, UI-Entwürfen, Prototypen, Code, Reviews, Tests und Monitoring berücksichtigen.
- Nutzen und Grenzen von AI im Usability-Kontext einschätzen, insbesondere die Grundregel, dass AI bei bekannten Mustern stark und bei unbekanntem

Nutzungskontext schwach ist, und daraus ableiten, wann Domänenwissen, reale Nutzerbeobachtung oder zusätzliche Expertise notwendig sind.

- Verantwortung für Usability, Datenschutz, Accessibility, Code-Qualität und AI-Einsatz im Team verorten und geeignete Prüf- und Governance-Maßnahmen ableiten, einschließlich nachvollziehbarer Dokumentation des AI-Einsatzes.
- Erkenntnisse aus AI-gestützter Analyse, Evaluation, realer Nutzung, Feedback und Supportdaten in konkrete Verbesserungen, Backlog Items, UX-Debt-Maßnahmen und kontinuierliche Usability-Verbesserung überführen.

2. Generelle Business Outcomes

BO-G 1 Usability-Arbeit mit Hilfe von GenAI früher, pragmatischer und systematischer in Softwareentwicklungsteams verankern.

BO-G 2 AI-gestützte Usability-Ergebnisse kritisch bewerten und zwischen erkannten Mustern, plausiblen Hypothesen, AI-generierten Vorschlägen und tatsächlich validierten Erkenntnissen unterscheiden.

BO-G 3 Grundlegende Kriterien für Usability einschließlich Wahrnehmung, Verständlichkeit, kognitiver Belastung, Accessibility und erwartbarem Verhalten in Discovery, Requirements, Backlog Items, UI-Entwürfen, Prototypen, Code, Review-Formaten, Evaluation, Testing und Monitoring berücksichtigen.

BO-G 4 Den Nutzen und die Grenzen von AI im Usability-Kontext einschätzen, insbesondere bei bekannten Mustern, unbekanntem Nutzungskontext, Domänenwissen, Nutzerkontakt und Validierungsbedarf.

BO-G 5 Verantwortung für Usability, Datenschutz, Accessibility, Code-Qualität, AI-Einsatz und fachliche Entscheidungen im Softwareentwicklungsteam verorten und geeignete Prüf- und Governance-Maßnahmen ableiten.

BO-G 6 Erkenntnisse aus AI-gestützter Analyse, Evaluation, realer Nutzung, Nutzer- und Kundenfeedback sowie Supportdaten in konkrete Verbesserungen, Backlog Items und kontinuierliche Usability-Verbesserung überführen.

3. Business Outcomes je Kapitel

Kapitel 1: Usability mit AI: Warum das Thema relevant ist

BO 1.1 Usability als Bestandteil professioneller Softwarequalität in Entwicklungsprozessen berücksichtigen und begründen.

BO 1.2 GenAI als niederschweligen Einstieg in konkrete Usability-Arbeit nutzen, insbesondere in Teams ohne eigene UX-Spezialisten, ausreichendes Budget, Zeit oder methodisches Vorwissen.

BO 1.3 Aus dem Prinzip „AI ist stark bei bekannten Mustern, aber schwach bei unbekanntem Kontext“ ableiten, wo AI im Usability-Prozess sinnvoll unterstützt und wo menschliche Bewertung, Domänenwissen oder reale Nutzerbeobachtung notwendig bleiben.

BO 1.4 Grundlegende Regeln für verantwortungsvollen AI-Einsatz im Usability-Kontext anwenden, insbesondere menschliche Kontrolle, Datenschutz, Transparenz, Nachvollziehbarkeit und Governance.

Kapitel 2: Mit AI Usability-Basiswissen aufbauen

BO 2.1 GenAI gezielt nutzen, um zentrale Usability-Grundlagen einschließlich menschlicher Wahrnehmung, mentaler Modelle, kognitiver Belastung, Verständlichkeit und Accessibility zu erlernen, zu wiederholen und auf eigene Entwicklungsartefakte anzuwenden.

BO 2.2 Ein gemeinsames Basisverständnis für Usability im Softwareentwicklungsteam aufbauen, indem zentrale Begriffe, Qualitätskriterien und typische Usability-Probleme mit Hilfe von AI erklärt, verglichen und auf eigene Entwicklungsartefakte bezogen werden.

BO 2.3 AI-Ausgaben kritisch prüfen und zwischen plausibel wirkenden Vorschlägen, brauchbaren Hypothesen und belastbaren Erkenntnissen unterscheiden.

BO 2.4 AI-Unterstützung bei bekannten Usability-Mustern nutzen und bei unbekanntem Kontext, impliziten Anforderungen und domänenspezifischen Fragestellungen gezielt überprüfen.

Kapitel 3: Discovery: Nutzer, Kontext, Probleme und Hypothesen verstehen

BO 3.1 Vorhandene Informationen wie Supporttickets, Kundenfeedback, Analytics, Interviews oder Fachwissen nutzen, um Usability-relevante Problemfelder zu identifizieren.

BO 3.2 GenAI einsetzen, um vorhandenes Nutzer- und Kundenfeedback, Supporthinweise, Beobachtungen und Daten zu strukturieren, zusammenzufassen und daraus überprüfbare Usability-Hypothesen abzuleiten.

BO 3.3 Aus AI-gestützt strukturierten Informationen konkrete Nutzerprobleme, Nutzungshypothesen und prüfbare Produktannahmen ableiten, die als Grundlage für Backlog Items, Prototypen oder Tests dienen.

BO 3.4 Erkennen, wann AI-gestützte Hypothesen nicht ausreichen und reale Nutzer, Domänenexperten oder zusätzliche Research-Aktivitäten einbezogen werden müssen.

Kapitel 4: Backlog Refinement und Usability

BO 4.1 Usability-Anforderungen als Bestandteil professioneller Requirements-Arbeit in User Stories, Akzeptanzkriterien und Definition-of-Done-Kriterien integrieren.

BO 4.2 GenAI nutzen, um Backlog Items auf fehlende Usability-Aspekte, unklare Formulierungen, Happy-Path-Fixierung, fehlende Zustände und mögliche Edge Cases zu prüfen.

BO 4.3 Usability-relevante Kriterien in Definition of Ready und Definition of Done integrieren, damit Nutzerziel, Nutzungskontext, Zustände, Fehlerfälle, Accessibility-Basics und Validierungsbedarf überprüfbar werden.

BO 4.4 AI-generierte Anforderungen und Akzeptanzkriterien fachlich einordnen und entscheiden, welche Vorschläge im konkreten Produkt- und Nutzungskontext relevant sind.

Kapitel 5: Planung und Teamorganisation

BO 5.1 Usability-Aufgaben in der Planung sichtbar machen und grundlegende Verantwortlichkeiten für UX im Softwareentwicklungsteam klären.

BO 5.2 Einfache Usability-Aktivitäten wie Design Critiques, heuristische Prüfungen anhand definierter Kriterien, Formulartests, Accessibility-Basic-Checks und Systemzustandsprüfungen im Team organisatorisch einplanen.

BO 5.3 GenAI zur Vorbereitung, Strukturierung und Dokumentation von Usability-bezogenen Teamaktivitäten einsetzen, ohne Verantwortung und Entscheidungsbefugnis an AI zu delegieren.

BO 5.4 Den Einsatz von AI in Usability-relevanten Teamaktivitäten nachvollziehbar dokumentieren, insbesondere Zweck, verwendete Eingaben, übernommene Ergebnisse und menschliche Prüfung.

Kapitel 6: UI, Informationsarchitektur, States und AI-generierte Entwürfe

BO 6.1 AI-gestützte UI-Entwürfe, Wireframes, Prototypen und Frontend-Vorschläge anhand grundlegender Usability- und Human-Factors-Kriterien beurteilen, insbesondere Wahrnehmbarkeit, Verständlichkeit, kognitive Belastung, Konsistenz, Accessibility und erwartbares Verhalten.

BO 6.2 Typische Schwächen AI-generierter Entwürfe erkennen, insbesondere fehlende States, Edge Cases, Fehlerpfade, unklare Navigation, mangelnde Accessibility und zu starke Happy-Path-Orientierung.

BO 6.3 AI-generierte Prototypen als Kommunikations-, Lern- und Validierungsgrundlage nutzen, ohne sie vorschnell als fachlich richtige oder produktionsreife Lösung zu behandeln.

BO 6.4 AI-generierte UI-Varianten anhand von Nutzerziel, visueller Hierarchie, Verständlichkeit, Konsistenz, Accessibility-Basics und erwartetbarem Verhalten vergleichen und begründet auswählen.

Kapitel 7: Code-Erstellung, Komponenten, Accessibility und Handoff

BO 7.1 AI-gestützte Coding-Werkzeuge für UI-nahe Entwicklungsaufgaben, Komponenten, Validierung, Dokumentation und Tests fachlich einordnen und gezielt einsetzen.

BO 7.2 Grundlegende Usability-Qualitätsaspekte in der Implementierung berücksichtigen, insbesondere konsistente Komponenten, Zustände, semantisches HTML, Tastaturbedienbarkeit, Fokusführung, Labels, Fehlermeldungen und Accessibility-Basics.

BO 7.3 AI-generierte UI-Implementierungen auf ihre Übereinstimmung mit Usability-Requirements, Akzeptanzkriterien, Zuständen, Komponentenregeln und Tests prüfen.

BO 7.4 Risiken AI-generierter Implementierung erkennen und geeignete Prüfmaßnahmen wie Code Review, Tests, Security Review, Accessibility-Prüfung und fachliche Abnahme ableiten.

Kapitel 8: Evaluation, Testing und Validierung

BO 8.1 Grundlegende Review- und Evaluationsmethoden für Usability wie heuristische Evaluation, Cognitive Walkthrough und Usability-Test in Entwicklungsprozessen einordnen und zweckmäßig einsetzen.

BO 8.2 GenAI zur Vorbereitung, Strukturierung und Dokumentation von Usability-Reviews und Usability-Tests nutzen, insbesondere für Testaufgaben, Beobachtungsnotizen, Findings und Berichtsentwürfe.

BO 8.3 Erkennen, warum AI-Simulationen und heuristische AI-Analysen keine Validierung mit realen Nutzern ersetzen und wann echte Tests notwendig sind.

BO 8.4 Usability-Probleme nach Schweregrad, Häufigkeit, Nutzerwirkung, Risiko, Umsetzungsaufwand und Business-Relevanz priorisieren und AI-Vorschläge dazu kritisch einordnen.

BO 8.5 Aus AI-generierten Usability-Vorschlägen testbare Annahmen und Fragestellungen ableiten, die durch heuristische Reviews, Usability-Tests oder Nutzungsdaten überprüft werden können.

Kapitel 9: Release, Monitoring und Verbesserungen

BO 9.1 Usability nach dem Release anhand von Feedback, Supporttickets, Nutzungsdaten und einfachen UX-Metriken beobachten und Verbesserungspotenziale identifizieren.

BO 9.2 GenAI einsetzen, um Nutzer- und Kundenfeedback, Tickets, Metriken und UX-Debt zu strukturieren, Muster sichtbar zu machen und Hypothesen für Verbesserungen abzuleiten.

BO 9.3 Grenzen daten- und AI-gestützter Auswertung erkennen, insbesondere den Unterschied zwischen Muster, Korrelation, Ursache, Priorität und belastbarer Produktentscheidung.

BO 9.4 Erkenntnisse aus Feedback, Metriken, Supportdaten und AI-gestützter Analyse in priorisierte Backlog Items, UX-Debt-Maßnahmen oder neue Discovery-Fragen überführen.

4. Lernziele je Kapitel

Wissensstufen: K1 Erinnern (Fakten und Begriffe wiedergeben), K2 Verstehen (Zusammenhänge erklären und einordnen), K3 Anwenden (Wissen auf typische Artefakte und Situationen anwenden).

Kapitel 1: Usability mit AI: Warum das Thema relevant ist (ca. 90 Minuten)

LO	K	Lernziel
LO 1.1	K1	Die Begriffe Software-Usability, User Experience und UX-Qualität im Kontext der Softwareentwicklung wiedergeben können.
LO 1.2	K2	Erklären können, warum Usability Bestandteil professioneller Softwarequalität ist.

LO	K	Lernziel
LO 1.3	K2	Erklären können, warum agile Prozesse gute UX nicht automatisch sicherstellen.
LO 1.4	K2	Beschreiben können, wie GenAI einen niederschweligen Einstieg in Usability-Arbeit ermöglicht.
LO 1.5	K2	Die Grundregel „AI ist stark bei bekannten Mustern, aber schwach bei unbekanntem Kontext“ erläutern können.
LO 1.6	K1	Typische Einsatzfelder von GenAI entlang typischer agiler Entwicklungsaktivitäten benennen können.
LO 1.7	K3	AI-Einsatzmöglichkeiten in agilen Entwicklungsprozessen anhand ihres Nutzens und ihrer Grenzen einordnen können.
LO 1.8	K2	Grundlegende Regeln für verantwortungsvollen AI-Einsatz im Usability-Kontext erklären können.

Kapitel 2: Mit AI Usability-Basiswissen aufbauen (ca. 120 Minuten)

LO	K	Lernziel
LO 2.1	K1	Zentrale Usability-Basisbegriffe wie Nutzerziel, Nutzungskontext, mentales Modell, Erwartungskonformität, kognitive Last, visuelle Hierarchie und Accessibility wiedergeben können.
LO 2.2	K1	Grundlegende menschliche Faktoren für UX benennen können, insbesondere Wahrnehmung, Aufmerksamkeit, Gedächtnis, kognitive Belastung, motorische Fähigkeiten und situative Einschränkungen.
LO 2.3	K2	Erklären können, warum menschliche Wahrnehmung selektiv, begrenzt und kontextabhängig ist.
LO 2.4	K2	Erklären können, warum mentale Modelle, Erwartungskonformität und kognitive Last für die Beurteilung von Benutzeroberflächen zentral sind.
LO 2.5	K2	Erklären können, warum Softwareentwicklungsteams ein gemeinsames Basisverständnis für Usability benötigen, um AI-Ergebnisse sinnvoll beurteilen zu können.
LO 2.6	K2	Beschreiben können, wie GenAI beim Lernen und Wiederholen von Usability-Grundlagen unterstützen kann.
LO 2.7	K3	Eigene Entwicklungsartefakte wie Screens, User Stories, Fehlermeldungen, Prototypen oder Codebeispiele mit AI-Unterstützung zur Usability-Reflexion verwenden können.
LO 2.8	K2	Prompting-Techniken für Usability-Lernen beschreiben können, insbesondere Erklären, Vergleichen, Gegenbeispiele erzeugen, Annahmen prüfen und Verbesserungsvorschläge begründen lassen.
LO 2.9	K2	AI-Ausgaben als Hypothesen und nicht als gesicherte Erkenntnisse einordnen können.
LO 2.10	K3	Plausible, aber potenziell falsche oder unvollständige AI-Antworten erkennen und kritisch hinterfragen können.

LO	K	Lernziel
LO 2.11	K3	AI-gestützte Usability-Erklärungen und Verbesserungsvorschläge mit grundlegenden Kriterien menschlicher Wahrnehmung, Verständlichkeit, Erwartungskonformität und Accessibility abgleichen können.

Kapitel 3: Discovery: Nutzer, Kontext, Probleme und Hypothesen verstehen (ca. 120 Minuten)

LO	K	Lernziel
LO 3.1	K1	Die Begriffe Nutzer, Nutzerziel, Aufgabe, Nutzungskontext, Problemhypothese und Produktannahme wiedergeben können.
LO 3.2	K2	Erklären können, warum Usability-Arbeit vor der Lösungsidee mit dem Verständnis von Nutzer, Aufgabe und Kontext beginnen muss.
LO 3.3	K1	Typische Informationsquellen für Discovery für Usability benennen können, etwa Supporttickets, Kundenfeedback, Analytics, Interviews, Fachwissen und bestehende Dokumentation.
LO 3.4	K2	Beschreiben können, wie GenAI vorhandenes Nutzer- und Kundenfeedback, Supporthinweise, Beobachtungen und Daten strukturieren und zusammenfassen kann.
LO 3.5	K3	Aus vorhandenen Informationen mit AI-Unterstützung erste Usability-relevante Problemfelder ableiten können.
LO 3.6	K3	AI-generierte Nutzerannahmen als Hypothesen formulieren und von gesicherten Erkenntnissen unterscheiden können.
LO 3.7	K2	Erklären können, warum AI-generierte Personas und Szenarien ohne belastbare Daten problematisch sein können.
LO 3.8	K3	Beurteilen können, wann reale Nutzer, Domänenexperten oder zusätzliche Research-Aktivitäten erforderlich sind.
LO 3.9	K3	Aus Discovery-Erkenntnissen prüfbare Produktannahmen für Backlog Items, Prototypen oder Tests ableiten können.

Kapitel 4: Backlog Refinement und Usability (ca. 120 Minuten)

LO	K	Lernziel
LO 4.1	K1	Die Begriffe Usability-Requirement, User Story, Akzeptanzkriterium, Definition of Ready und Definition of Done wiedergeben können.
LO 4.2	K2	Erklären können, warum Usability-Anforderungen Bestandteil professioneller Requirements-Arbeit sind.
LO 4.3	K2	Typische implizite Usability-Anforderungen beschreiben können, etwa Verständlichkeit, Fehlertoleranz, Systemfeedback, Systemzustände und Accessibility-Basics.
LO 4.4	K3	User Stories aus Nutzungsperspektive analysieren können, insbesondere im Hinblick auf Nutzerziel, Nutzungskontext und erwartetes Verhalten.
LO 4.5	K3	GenAI nutzen können, um Backlog Items auf unklare Formulierungen, fehlende Usability-Aspekte sowie Happy-Path-Fixierung zu prüfen.

LO	K	Lernziel
LO 4.6	K3	Akzeptanzkriterien mit Usability-Aspekten wie Fehlermeldungen, Eingabevalidierung, Systemzuständen, Tastaturbedienbarkeit oder verständlichem Systemfeedback ergänzen können.
LO 4.7	K2	Usability-relevante Kriterien für Definition of Ready und Definition of Done erklären können.
LO 4.8	K3	AI-generierte Vorschläge für Requirements und Akzeptanzkriterien fachlich beurteilen und entscheiden können, ob sie übernommen, angepasst oder begründet ausgeschlossen werden müssen.
LO 4.9	K2	Erklären können, warum AI keine verbindliche Priorisierung oder fachliche Notwendigkeit im konkreten Produktkontext bestimmen kann.

Kapitel 5: Planung und Teamorganisation (ca. 90 Minuten)

LO	K	Lernziel
LO 5.1	K1	Typische Usability-Aufgaben in der Umsetzung benennen können, etwa Usability-Check, Design Critique, Systemzustandsprüfung, Formulartest, Accessibility-Basic-Check und Vorbereitung von Review-Formaten oder Prüffragen.
LO 5.2	K2	Erklären können, warum Usability-Aufgaben in der Planung sichtbar gemacht und nicht nur implizit erwartet werden sollten.
LO 5.3	K2	Die Beiträge typischer Rollen in agilen Softwareentwicklungsteams zur UX-Qualität beschreiben können, etwa Product Owner, Entwicklung, Testerinnen und Tester, Qualitätssicherung, Scrum Master, agile Coaches und gegebenenfalls UX-Spezialistinnen und UX-Spezialisten.
LO 5.4	K3	Einfache Usability-Aktivitäten in der Planung der Umsetzung organisatorisch einplanen können.
LO 5.5	K2	Erklären können, wo agile Softwareentwicklungsteams einfache Usability-Aufgaben selbst übernehmen können und wo zusätzliche Expertise erforderlich ist.
LO 5.6	K2	Design Critiques als strukturiertes Format zur Bewertung von Entwürfen beschreiben können.
LO 5.7	K3	GenAI zur Vorbereitung von Usability-Meetings, Prüffragen, Checklisten und Entscheidungsoptionen einsetzen können.
LO 5.8	K3	Den Einsatz von AI in Usability-relevanten Teamaktivitäten nachvollziehbar dokumentieren können, insbesondere Zweck, Eingaben, übernommene Ergebnisse und menschliche Prüfung.

Kapitel 6: UI, Informationsarchitektur, States und AI-generierte Entwürfe (ca. 135 Minuten)

LO	K	Lernziel
LO 6.1	K1	Grundlegende Begriffe des UI-Designs wie visuelle Hierarchie, Layout, Spacing, Typografie, Konsistenz, Interaktion, State und Control wiedergeben können.

LO	K	Lernziel
LO 6.2	K2	Erklären können, warum UI-Design-Grundlagen und Human-Factors-Kriterien notwendig sind, um AI-generierte Entwürfe beurteilen zu können.
LO 6.3	K2	Die Bedeutung von Informationsarchitektur, Navigation, Benennungen, mentalen Modellen und Auffindbarkeit für Usability erklären können.
LO 6.4	K1	Typische Prototyp-Arten und deren Zweck benennen können, insbesondere Wireframe, Mockup, Low-Fidelity-Prototyp, High-Fidelity-Prototyp, klickbarer Prototyp, funktionsnaher Prototyp, horizontaler Prototyp, vertikaler Prototyp und Szenario-Prototyp.
LO 6.5	K2	Beschreiben können, wie AI-gestütztes Rapid Prototyping in agilen Entwicklungsprozessen eingesetzt werden kann.
LO 6.6	K3	AI-generierte UI-Entwürfe anhand von Nutzerziel, Nutzungskontext, menschlicher Wahrnehmung, mentalen Modellen, visueller Hierarchie, Verständlichkeit, kognitiver Last, Konsistenz, Accessibility und erwartbarem Verhalten beurteilen können.
LO 6.7	K2	Typische Stärken von AI bei bekannten UI-Mustern erklären können, etwa Login, Formular, Dashboard, Navigation, Suche, Filter oder Onboarding.
LO 6.8	K3	Typische Schwächen AI-generierter Entwürfe erkennen können, insbesondere fehlende States, Edge Cases, Fehlerpfade, Berechtigungen, alternative Nutzungssituationen und unvollständige Accessibility-Berücksichtigung.
LO 6.9	K3	AI-generierte UI-Varianten anhand definierter Usability-Kriterien vergleichen und begründet auswählen können
LO 6.10	K3	Prototypen als Grundlage für Kommunikation, Lernen, Evaluation und gegebenenfalls Validierung im Team einsetzen können.

Kapitel 7: Code-Erstellung, Komponenten, Accessibility und Handoff (ca. 135 Minuten)

LO	K	Lernziel
LO 7.1	K1	Typische AI-gestützte Coding-Werkzeuge und Einsatzbereiche für UI-nahe Entwicklung benennen können.
LO 7.2	K2	Erklären können, wie AI-gestützte Code-Erstellung die Schnittstelle zwischen Usability-Anforderungen, UI-Design und Implementierung verändert.
LO 7.3	K2	Grundlegende Usability-Qualitätsaspekte in der Implementierung beschreiben können, insbesondere Systemzustände, Komponentenlogik, Fehlermeldungen, Labels, Fokusführung, Tastaturbedienbarkeit und erwartbares Verhalten.
LO 7.4	K2	Die Bedeutung konsistenter, wiederverwendbarer Komponenten für Usability, Wartbarkeit und Accessibility erklären können.
LO 7.5	K2	Die Funktion von Designsystemen und Design Tokens als Verbindung zwischen Design, UI-Implementierung und Code beschreiben können.
LO 7.6	K3	AI-generierte UI-Implementierungen mit Usability-Requirements, Akzeptanzkriterien, Systemzuständen, Komponentenregeln und Designsystem-Vorgaben abgleichen können.

LO	K	Lernziel
LO 7.7	K3	AI nutzen können, um Testfälle, Dokumentation, Stories für einen Component Workshop, Prüffragen oder Accessibility-Hinweise vorzubereiten.
LO 7.8	K1	Grundlegende Accessibility-Basics in der Umsetzung benennen können, insbesondere semantisches HTML, Kontrast, Labels, Fokusführung, Tastaturbedienbarkeit und ARIA nur wo nötig.
LO 7.9	K2	Risiken AI-generierter Implementierung erklären können, insbesondere Sicherheitslücken, Accessibility-Probleme, Inkonsistenzen, Wartbarkeitsprobleme, Performance-Probleme und fachlich falsche Logik.
LO 7.10	K3	Geeignete Prüfmaßnahmen für AI-generierten Code ableiten können, insbesondere Code Review, funktionale Tests, UI-Tests, Security Review, Accessibility-Prüfung und fachliche Abnahme.
LO 7.11	K2	Erklären können, wie AI genutzt werden kann, um aus einer bestehenden Anwendung wiederkehrende UI-Muster, Inkonsistenzen und erste Designsystem-Bausteine abzuleiten.

Kapitel 8: Evaluation, Testing und Validierung (ca. 150 Minuten)

LO	K	Lernziel
LO 8.1	K1	Grundlegende Evaluationsmethoden für Usability sowie expertenbasierte Verfahren benennen können, insbesondere heuristische Evaluation, Cognitive Walkthrough und Usability-Test.
LO 8.2	K2	Den Zweck von Evaluation und Validierung in Softwareentwicklungsprozessen erklären können.
LO 8.3	K2	Erklären können, warum in Review-Formaten, etwa im Sprint Review, nicht nur Funktionalität, sondern auch Nutzerziel, Verständlichkeit, Systemzustände und Fehlertoleranz betrachtet werden sollten.
LO 8.4	K3	Eine heuristische Evaluation mit AI-Unterstützung vorbereiten und deren Ergebnisse kritisch einordnen können.
LO 8.5	K2	Den Grundgedanken des Cognitive Walkthrough als expertenbasiertes Verfahren erklären können.
LO 8.6	K2	Die zentralen Bestandteile eines einfachen Usability-Tests beschreiben können, insbesondere Untersuchungsfrage, Testpersonen, Testgegenstand, Testaufgaben, Moderation, Beobachtung und Auswertung.
LO 8.7	K3	Testaufgaben zielorientiert formulieren können, ohne den Lösungsweg vorzugeben.
LO 8.8	K2	Den Zweck und die Grenzen von Think-Aloud und strukturierter Beobachtung erklären können.
LO 8.9	K3	GenAI zur Strukturierung von Beobachtungsnotizen, Findings und Berichtsentwürfen einsetzen können.
LO 8.10	K3	Probleme der Usability nach Schweregrad, Häufigkeit, Nutzerwirkung, Risiko, Umsetzungsaufwand und Business-Relevanz priorisieren können.

LO	K	Lernziel
LO 8.11	K3	Aus AI-generierten Vorschlägen zu Usability testbare Annahmen und Untersuchungsfragen ableiten können.
LO 8.12	K2	Erklären können, warum AI-Simulationen, AI-gestützte heuristische Vorprüfungen und synthetische Nutzer keine Validierung mit realen Nutzerinnen und Nutzern ersetzen.

Kapitel 9: Release, Monitoring und Verbesserungen (ca. 90 Minuten)

LO	K	Lernziel
LO 9.1	K1	Typische Quellen für Erkenntnisse zu Usability nach dem Release benennen können, insbesondere Nutzerfeedback, Kundenfeedback, Supporttickets, Nutzungsdaten, technische Signale und UX-Metriken.
LO 9.2	K2	Erklären können, warum Usability nach dem Release weiter beobachtet und verbessert werden muss.
LO 9.3	K2	Grundlegende Metriken für Usability wie Aufgabenerfolg, Fehlerrate, Abbruchrate, Bearbeitungszeit, Wiederkehrrate, Nutzung zentraler Funktionen und Supportaufkommen beschreiben können.
LO 9.4	K2	Den Grundgedanken des HEART-Frameworks erklären und die Dimensionen Happiness, Engagement, Adoption, Retention und Task Success als Strukturierungshilfe für UX-Ziele und UX-Metriken einordnen können.
LO 9.5	K3	GenAI einsetzen können, um Feedback, Tickets, Metriken und Hinweise auf Usability-Probleme oder UX-Debt zu strukturieren und Muster sichtbar zu machen.
LO 9.6	K2	Den Unterschied zwischen Muster, Korrelation, Ursache, Priorität und belastbarer Produktentscheidung erläutern können.
LO 9.7	K3	UX-Debt anhand typischer Beispiele mit Bezug zu Usability erkennen können, etwa unvollständige Systemzustände, inkonsistente Komponenten, unklare Texte, fehlende Accessibility oder ungelöste Usability-Probleme.
LO 9.8	K3	Erkenntnisse aus Feedback, Metriken, Supportdaten und AI-gestützter Analyse in Backlog Items, UX-Debt-Maßnahmen, Verbesserungen der Usability oder neue Discovery-Fragen überführen können.