Competency Profile, Business Outcomes, and Learning Objectives

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

Source: syllabus UXQCC_GenAI_V1_013_EN (Build 1.013). The syllabus is authoritative; this document summarizes the competency profile, business outcomes, and learning objectives as a reference for exam questions, training materials, and communication. Scope: 6 general business outcomes, 37 chapter-specific business outcomes, 86 learning objectives (14 at level K1, 42 at K2, 30 at K3).

1. What Certificate Holders Can Do

Holders of the Foundation Level certificate can embed usability work in their software development team earlier, more pragmatically, and more systematically with the help of GenAI, even without a permanently available dedicated UX role. They know the central terms and quality criteria of usability and apply them to their own development artifacts, from user story through UI draft, prototype, and code to testing and monitoring.

The competency profile also addresses UX specialists. They learn to bring their existing UX knowledge effectively into the software development process: They translate usability concerns into the development team's working formats and artifacts, such as user stories, acceptance criteria, Definition of Ready and Definition of Done, review questions, and review formats, and word them so that team members without UX training, in particular software developers, product owners, and testers, can understand them, put them into context, and apply them in their daily work. This also includes the targeted use of GenAI to prepare usability knowledge in a team-appropriate way, as well as a shared understanding of terms that supports the collaboration between UX expertise and development.

In detail, certificate holders can:

- use GenAI along the entire agile development process as a tool for learning, structuring, drafting, reviewing, and analysis: in discovery, backlog refinement, planning, design and prototyping, implementation, evaluation, and monitoring.
- critically assess AI-assisted usability results, distinguishing between recognized patterns, plausible hypotheses, AI-generated suggestions, and actually validated findings.
- consider fundamental usability criteria such as perception, understandability, cognitive load, accessibility, and expectable behavior in requirements, backlog items, UI drafts, prototypes, code, reviews, tests, and monitoring.
- assess the benefits and limits of AI in the usability context, in particular the basic rule that AI is strong with familiar patterns but weak with unfamiliar context, and derive from it when domain knowledge, real user observation, or additional expertise is required.

- locate responsibility for usability, data protection, accessibility, code quality, and AI use within the team and derive appropriate review and governance measures, including traceable documentation of AI use.
- turn insights from AI-assisted analysis, evaluation, real usage, feedback, and support data into concrete improvements, backlog items, UX debt measures, and continuous usability improvement.

2. General Business Outcomes

BO-G 1 Embed usability work in software development teams earlier, more pragmatically, and more systematically with the help of GenAI.

BO-G 2 Critically assess AI-assisted usability results and distinguish between recognized patterns, plausible hypotheses, AI-generated suggestions, and actually validated findings.

BO-G 3 Consider fundamental usability criteria, including perception, understandability, cognitive load, accessibility, and expectable behavior, in discovery, requirements, backlog items, UI drafts, prototypes, code, review formats, evaluation, testing, and monitoring.

BO-G 4 Assess the benefits and limits of AI in the usability context, in particular with regard to familiar patterns, unfamiliar context of use, domain knowledge, user contact, and validation needs.

BO-G 5 Locate responsibility for usability, data protection, accessibility, code quality, AI use, and professional decisions within the software development team, and derive appropriate review and governance measures.

BO-G 6 Turn insights from AI-assisted analysis, evaluation, real usage, user and customer feedback, and support data into concrete improvements, backlog items, and continuous usability improvement.

3. Business Outcomes by Chapter

Chapter 1: Usability with AI: Why This Topic Matters

BO 1.1 Consider and justify usability as part of professional software quality in development processes.

BO 1.2 Use GenAI as a low-threshold entry into concrete usability work, especially in teams without their own UX professionals, sufficient budget, time, or methodological background.

BO 1.3 From the principle “AI is strong with familiar patterns but weak with unfamiliar context”, derive where AI provides meaningful support in the usability process and where human judgment, domain knowledge, or real user observation remain necessary.

BO 1.4 Apply fundamental rules for responsible AI use in the usability context, in particular human oversight, data protection, transparency, traceability, and governance.

Chapter 2: Building Foundational Usability Knowledge with AI

BO 2.1 Use GenAI in a targeted way to learn, revise, and apply central usability foundations, including human perception, mental models, cognitive load, understandability, and accessibility, to one's own development artifacts.

BO 2.2 Build a shared basic understanding of usability in the software development team by using AI to explain and compare central terms, quality criteria, and typical usability problems and relate them to the team's own development artifacts.

BO 2.3 Critically review AI output and distinguish between plausible-sounding suggestions, useful hypotheses, and reliable findings.

BO 2.4 Use AI support for familiar usability patterns and specifically verify it for unfamiliar context, implicit requirements, and domain-specific questions.

Chapter 3: Discovery: Understanding Users, Context, Problems, and Hypotheses

BO 3.1 Use existing information such as support tickets, customer feedback, analytics, interviews, or domain knowledge to identify usability-relevant problem areas.

BO 3.2 Use GenAI to structure and summarize existing user and customer feedback, support hints, observations, and data, and to derive verifiable usability hypotheses from them.

BO 3.3 From information structured with AI support, derive concrete user problems, usage hypotheses, and verifiable product assumptions that serve as a basis for backlog items, prototypes, or tests.

BO 3.4 Recognize when AI-assisted hypotheses are not sufficient and real users, domain experts, or additional research activities must be involved.

Chapter 4: Backlog Refinement and Usability

BO 4.1 Integrate usability requirements as part of professional requirements work into user stories, acceptance criteria, and Definition-of-Done criteria.

BO 4.2 Use GenAI to check backlog items for missing usability aspects, unclear wording, happy-path fixation, missing states, and possible edge cases.

BO 4.3 Integrate usability-relevant criteria into Definition of Ready and Definition of Done so that user goal, context of use, states, error cases, accessibility basics, and validation needs become verifiable.

BO 4.4 Professionally classify AI-generated requirements and acceptance criteria and decide which suggestions are relevant in the specific product and usage context.

Chapter 5: Planning and Team Organization

BO 5.1 Make usability tasks visible in planning and clarify basic responsibilities for UX within the software development team.

BO 5.2 Organizationally plan simple usability activities in the team, such as design critiques, heuristic reviews based on defined criteria, form tests, accessibility basic checks, and system state checks.

BO 5.3 Use GenAI for the preparation, structuring, and documentation of usability-related team activities without delegating responsibility and decision-making authority to AI.

BO 5.4 Document the use of AI in usability-relevant team activities in a traceable way, in particular purpose, inputs used, adopted results, and human review.

Chapter 6: UI, Information Architecture, States, and AI-Generated Drafts

BO 6.1 Assess AI-assisted UI drafts, wireframes, prototypes, and frontend suggestions based on fundamental usability and human-factors criteria, in particular perceivability, understandability, cognitive load, consistency, accessibility, and expectable behavior.

BO 6.2 Recognize typical weaknesses of AI-generated drafts, in particular missing states, edge cases, error paths, unclear navigation, insufficient accessibility, and excessive happy-path orientation.

BO 6.3 Use AI-generated prototypes as a basis for communication, learning, and validation without prematurely treating them as professionally correct or production-ready solutions.

BO 6.4 Compare AI-generated UI variants based on user goal, visual hierarchy, understandability, consistency, accessibility basics, and expectable behavior, and select among them with justification.

Chapter 7: Code Creation, Components, Accessibility, and Handoff

BO 7.1 Professionally classify AI-assisted coding tools for UI-related development tasks, components, validation, documentation, and tests, and use them in a targeted way.

BO 7.2 Consider fundamental usability quality aspects in implementation, in particular consistent components, states, semantic HTML, keyboard operability, focus order, labels, error messages, and accessibility basics.

BO 7.3 Check AI-generated UI implementations for their conformity with usability requirements, acceptance criteria, states, component rules, and tests.

BO 7.4 Recognize risks of AI-generated implementation and derive appropriate review measures such as code review, tests, security review, accessibility checks, and professional acceptance.

Chapter 8: Evaluation, Testing, and Validation

BO 8.1 Classify fundamental review and evaluation methods for usability, such as heuristic evaluation, cognitive walkthrough, and usability testing, within development processes and use them appropriately.

BO 8.2 Use GenAI for the preparation, structuring, and documentation of usability reviews and usability tests, in particular for test tasks, observation notes, findings, and report drafts.

BO 8.3 Recognize why AI simulations and heuristic AI analyses do not replace validation with real users and when real tests are necessary.

BO 8.4 Prioritize usability problems by severity, frequency, user impact, risk, implementation effort, and business relevance, and critically assess AI suggestions in this regard.

BO 8.5 From AI-generated usability suggestions, derive testable assumptions and questions that can be verified through heuristic reviews, usability tests, or usage data.

Chapter 9: Release, Monitoring, and Improvement

BO 9.1 Monitor usability after release based on feedback, support tickets, usage data, and simple UX metrics, and identify improvement potential.

BO 9.2 Use GenAI to structure user and customer feedback, tickets, metrics, and UX debt, to reveal patterns, and to derive hypotheses for improvements.

BO 9.3 Recognize the limits of data- and AI-based analysis, in particular the difference between pattern, correlation, cause, priority, and a reliable product decision.

BO 9.4 Turn insights from feedback, metrics, support data, and AI-assisted analysis into prioritized backlog items, UX debt measures, or new discovery questions.

4. Learning Objectives by Chapter

Knowledge levels: K1 Remember (recall facts and terms), K2 Understand (explain and classify relationships), K3 Apply (apply knowledge to typical artifacts and situations).

Chapter 1: Usability with AI: Why This Topic Matters (approx. 90 minutes)

LO	K	Learning objective
LO 1.1	K1	Be able to reproduce the terms software usability, user experience, and UX quality in the context of software development.
LO 1.2	K2	Be able to explain why usability is part of professional software quality.
LO 1.3	K2	Be able to explain why agile processes do not automatically ensure good UX.
LO 1.4	K2	Be able to describe how GenAI enables a low-threshold entry into usability work.
LO 1.5	K2	Be able to explain the basic rule “AI is strong with familiar patterns but weak with unfamiliar context”.
LO 1.6	K1	Be able to name typical fields of application of GenAI along typical agile development activities.
LO 1.7	K3	Be able to classify possible uses of AI in agile development processes based on their benefits and limits.
LO 1.8	K2	Be able to explain fundamental rules for responsible AI use in the usability context.

Chapter 2: Building Foundational Usability Knowledge with AI (approx. 120 minutes)

LO	K	Learning objective
LO 2.1	K1	Be able to reproduce central basic usability terms such as user goal, context of use, mental model, conformity with user expectations, cognitive load, visual hierarchy, and accessibility.
LO 2.2	K1	Be able to name fundamental human factors for UX, in particular perception, attention, memory, cognitive load, motor abilities, and situational constraints.
LO 2.3	K2	Be able to explain why human perception is selective, limited, and context-dependent.
LO 2.4	K2	Be able to explain why mental models, conformity with user expectations, and cognitive load are central to assessing user interfaces.
LO 2.5	K2	Be able to explain why software development teams need a shared basic understanding of usability in order to assess AI output meaningfully.
LO 2.6	K2	Be able to describe how GenAI can support learning and revising usability foundations.
LO 2.7	K3	Be able to use the team's own development artifacts such as screens, user stories, error messages, prototypes, or code samples for usability reflection with AI support.
LO 2.8	K2	Be able to describe prompting techniques for usability learning, in particular explaining, comparing, generating counterexamples, checking assumptions, and having improvement suggestions justified.
LO 2.9	K2	Be able to classify AI output as hypotheses and not as reliable findings.
LO 2.10	K3	Be able to recognize and critically question plausible but potentially incorrect or incomplete AI answers.
LO 2.11	K3	Be able to check AI-assisted usability explanations and improvement suggestions against fundamental criteria of human perception, understandability, conformity with user expectations, and accessibility.

Chapter 3: Discovery: Understanding Users, Context, Problems, and Hypotheses (approx. 120 minutes)

LO	K	Learning objective
LO 3.1	K1	Be able to reproduce the terms user, user goal, task, context of use, problem hypothesis, and product assumption.
LO 3.2	K2	Be able to explain why usability work must begin before the solution idea with understanding user, task, and context.
LO 3.3	K1	Be able to name typical information sources for usability discovery, such as support tickets, customer feedback, analytics, interviews, domain knowledge, and existing documentation.
LO 3.4	K2	Be able to describe how GenAI can structure and summarize existing user and customer feedback, support hints, observations, and data.

LO	K	Learning objective
LO 3.5	K3	Be able to derive initial usability-relevant problem areas from existing information with AI support.
LO 3.6	K3	Be able to formulate AI-generated user assumptions as hypotheses and distinguish them from reliable findings.
LO 3.7	K2	Be able to explain why AI-generated personas and scenarios can be problematic without reliable data.
LO 3.8	K3	Be able to judge when real users, domain experts, or additional research activities are required.
LO 3.9	K3	Be able to derive verifiable product assumptions for backlog items, prototypes, or tests from discovery findings.

Chapter 4: Backlog Refinement and Usability (approx. 120 minutes)

LO	K	Learning objective
LO 4.1	K1	Be able to reproduce the terms usability requirement, user story, acceptance criterion, Definition of Ready, and Definition of Done.
LO 4.2	K2	Be able to explain why usability requirements are part of professional requirements work.
LO 4.3	K2	Be able to describe typical implicit usability requirements, such as understandability, error tolerance, system feedback, system states, and accessibility basics.
LO 4.4	K3	Be able to analyze user stories from a usage perspective, in particular with regard to user goal, context of use, and expected behavior.
LO 4.5	K3	Be able to use GenAI to check backlog items for unclear wording, missing usability aspects, and happy-path fixation.
LO 4.6	K3	Be able to complement acceptance criteria with usability aspects such as error messages, input validation, system states, keyboard operability, or understandable system feedback.
LO 4.7	K2	Be able to explain usability-relevant criteria for Definition of Ready and Definition of Done.
LO 4.8	K3	Be able to professionally assess AI-generated suggestions for requirements and acceptance criteria and decide whether they must be adopted, adapted, or excluded with justification.
LO 4.9	K2	Be able to explain why AI cannot determine binding prioritization or professional necessity in the specific product context.

Chapter 5: Planning and Team Organization (approx. 90 minutes)

LO	K	Learning objective
LO 5.1	K1	Be able to name typical usability tasks in implementation, such as usability check, design critique, system state check, form test, accessibility basic check, and preparation of review formats or review questions.

LO	K	Learning objective
LO 5.2	K2	Be able to explain why usability tasks should be made visible in planning and not merely expected implicitly.
LO 5.3	K2	Be able to describe the contributions of typical roles in agile software development teams to UX quality, such as product owners, development, testers, quality assurance, Scrum Masters, agile coaches, and, where available, UX professionals.
LO 5.4	K3	Be able to organizationally plan simple usability activities in implementation planning.
LO 5.5	K2	Be able to explain where agile software development teams can take on simple usability tasks themselves and where additional expertise is required.
LO 5.6	K2	Be able to describe design critiques as a structured format for assessing drafts.
LO 5.7	K3	Be able to use GenAI to prepare usability meetings, review questions, checklists, and decision options.
LO 5.8	K3	Be able to document the use of AI in usability-relevant team activities in a traceable way, in particular purpose, inputs, adopted results, and human review.

Chapter 6: UI, Information Architecture, States, and AI-Generated Drafts (approx. 135 minutes)

LO	K	Learning objective
LO 6.1	K1	Be able to reproduce fundamental UI design terms such as visual hierarchy, layout, spacing, typography, consistency, interaction, state, and control.
LO 6.2	K2	Be able to explain why UI design foundations and human-factors criteria are necessary to assess AI-generated drafts.
LO 6.3	K2	Be able to explain the importance of information architecture, navigation, naming, mental models, and findability for usability.
LO 6.4	K1	Be able to name typical prototype types and their purpose, in particular wireframe, mockup, low-fidelity prototype, high-fidelity prototype, clickable prototype, functional prototype, horizontal prototype, vertical prototype, and scenario prototype.
LO 6.5	K2	Be able to describe how AI-assisted rapid prototyping can be used in agile development processes.
LO 6.6	K3	Be able to assess AI-generated UI drafts based on user goal, context of use, human perception, mental models, visual hierarchy, understandability, cognitive load, consistency, accessibility, and expectable behavior.
LO 6.7	K2	Be able to explain typical strengths of AI with familiar UI patterns, such as login, form, dashboard, navigation, search, filter, or onboarding.
LO 6.8	K3	Be able to recognize typical weaknesses of AI-generated drafts, in particular missing states, edge cases, error paths, permissions, alternative usage situations, and incomplete accessibility consideration.

LO	K	Learning objective
LO 6.9	K3	Be able to compare AI-generated UI variants based on defined usability criteria and select among them with justification
LO 6.10	K3	Be able to use prototypes as a basis for communication, learning, evaluation, and, where applicable, validation within the team.

Chapter 7: Code Creation, Components, Accessibility, and Handoff (approx. 135 minutes)

LO	K	Learning objective
LO 7.1	K1	Be able to name typical AI-assisted coding tools and fields of application for UI-related development.
LO 7.2	K2	Be able to explain how AI-assisted code creation changes the interface between usability requirements, UI design, and implementation.
LO 7.3	K2	Be able to describe fundamental usability quality aspects in implementation, in particular system states, component logic, error messages, labels, focus order, keyboard operability, and expectable behavior.
LO 7.4	K2	Be able to explain the importance of consistent, reusable components for usability, maintainability, and accessibility.
LO 7.5	K2	Be able to describe the function of design systems and design tokens as the connection between design, UI implementation, and code.
LO 7.6	K3	Be able to check AI-generated UI implementations against usability requirements, acceptance criteria, system states, component rules, and design system rules.
LO 7.7	K3	Be able to use AI to prepare test cases, documentation, stories for a component workshop, review questions, or accessibility guidance.
LO 7.8	K1	Be able to name fundamental accessibility basics in implementation, in particular semantic HTML, contrast, labels, focus order, keyboard operability, and ARIA only where necessary.
LO 7.9	K2	Be able to explain risks of AI-generated implementation, in particular security vulnerabilities, accessibility problems, inconsistencies, maintainability problems, performance problems, and professionally wrong logic.
LO 7.10	K3	Be able to derive appropriate review measures for AI-generated code, in particular code review, functional tests, UI tests, security review, accessibility checks, and professional acceptance.
LO 7.11	K2	Be able to explain how AI can be used to derive recurring UI patterns, inconsistencies, and initial design system building blocks from an existing application.

Chapter 8: Evaluation, Testing, and Validation (approx. 150 minutes)

LO	K	Learning objective
LO 8.1	K1	Be able to name fundamental evaluation methods for usability as well as expert-based procedures, in particular heuristic evaluation, cognitive walkthrough, and usability test.

LO	K	Learning objective
LO 8.2	K2	Be able to explain the purpose of evaluation and validation in software development processes.
LO 8.3	K2	Be able to explain why review formats, such as the sprint review, should consider not only functionality, but also user goal, understandability, system states, and error tolerance.
LO 8.4	K3	Be able to prepare a heuristic evaluation with AI support and critically classify its results.
LO 8.5	K2	Be able to explain the basic idea of the cognitive walkthrough as an expert-based procedure.
LO 8.6	K2	Be able to describe the central components of a simple usability test, in particular research question, test participants, test object, test tasks, facilitation, observation, and analysis.
LO 8.7	K3	Be able to formulate test tasks in a goal-oriented way without prescribing the solution path.
LO 8.8	K2	Be able to explain the purpose and limits of think-aloud and structured observation.
LO 8.9	K3	Be able to use GenAI to structure observation notes, findings, and report drafts.
LO 8.10	K3	Be able to prioritize usability problems by severity, frequency, user impact, risk, implementation effort, and business relevance.
LO 8.11	K3	Be able to derive testable assumptions and research questions from AI-generated usability suggestions.
LO 8.12	K2	Be able to explain why AI simulations, AI-assisted heuristic pre-checks, and synthetic users do not replace validation with real users.

Chapter 9: Release, Monitoring, and Improvement (approx. 90 minutes)

LO	K	Learning objective
LO 9.1	K1	Be able to name typical sources of usability insights after release, in particular user feedback, customer feedback, support tickets, usage data, technical signals, and UX metrics.
LO 9.2	K2	Be able to explain why usability must be further observed and improved after release.
LO 9.3	K2	Be able to describe fundamental usability metrics such as task success, error rate, abandonment rate, time on task, return rate, use of central functions, and support volume.
LO 9.4	K2	Be able to explain the basic idea of the HEART framework and classify the dimensions happiness, engagement, adoption, retention, and task success as a structuring aid for UX goals and UX metrics.
LO 9.5	K3	Be able to use GenAI to structure feedback, tickets, metrics, and indications of usability problems or UX debt and to reveal patterns.

LO	K	Learning objective
LO 9.6	K2	Be able to explain the difference between pattern, correlation, cause, priority, and a reliable product decision.
LO 9.7	K3	Be able to recognize UX debt based on typical usability-related examples, such as incomplete system states, inconsistent components, unclear texts, missing accessibility, or unresolved usability problems.
LO 9.8	K3	Be able to turn insights from feedback, metrics, support data, and AI-assisted analysis into backlog items, UX debt measures, usability improvements, or new discovery questions.