

Kompetenzband	HZ	Grundlagen	Fortgeschritten	Erweitert
Agile Methoden kennen	1	A1G: Ich kenne die linearen Vorgehensmodelle wie das Wasserfall-Modell, das V-Modell und RUP sowie die zyklischen Modelle wie das Spiralmodell und die unterschiedlichen Prototypenverfahren.	A1F: Ich kenne das „Agile Manifesto“ und kann diese Prinzipien erklären und weiss, was daran gut ist und wo Probleme auftauchen können.	A1E: Ich erkenne die linearen und die zyklischen Vorgehensmodelle und weiss, wie das Beste von allem kombiniert werden kann.
	1	A2G: Ich kenne das SCRUM-Modell mit dem zyklischen Ablauf, den Rollen, den Zeremonien (Meetings/Sitzungen) und die philosophie von SCRUM	A2F: Ich weiss, wie wir unser Vorhaben in eine „Vision“, in „Epics“ und in „User-Stories“ formulieren können.	A2E: Ich kann unser Vorhaben mit meinem Team in ein SCRUM-Ablauf einkleiden und die richtigen Schritte planen.
	1	A3G: Ich kenne die drei Aspekte, wie man eine Anforderung in einem Satz für eine User-Story formuliert.	A3F: Unter Einhaltung der DoR kann ich eine User-Story beschreiben, sie gemeinsam mit dem Team einordnen und deren Aufwand mit Story Points bewerten (Planning Pocker) sowie Abnahmekriterien dafür bestimmen (DoD).	A3E: Ich kann den „guten“ Schnitt zwischen User-Story und Aufgaben-Tasks zusammen mit dem Team bestimmen.
Agile Iteration realisieren	2	B1G: Ich kann bestehende Softwaremodule interpretieren und erklären, wo für die Umsetzung einer User Story die Software ergänzt werden muss.	B1F: Ich kann die Software für die Umsetzung einer User Story implementieren.	B1E: Ich kann bestehende Software für die Umsetzung einer User Story anpassen (Refactoring).
	2	B2G: Ich kann die Bedeutung einer lauffähigen Software im Rahmen der Agilen Softwareentwicklung erläutern.	B2F: Ich kann die von mir umgesetzte User-Story anhand der lauffähigen Software präsentieren und zeigen, dass die DoD eingehalten sind.	B2E: Ich kann eine Sprint-Demo vorbereiten (inkl. Burndown Chart) und durchführen.
Komponenten wiederverwenden	3	C2G: Ich kann das Prinzip von Softwarekomponenten (Wiederverwendbarkeit, Aufgaben Aufteilung, Schnittstellen) an einem Beispiel erläutern.	C2F: Ich kann für eine Applikation, unter Berücksichtigung der Wiederverwendbarkeit, Softwarekomponenten evaluieren und auswählen.	C2E: Ich kann in einem Entwurf Softwarekomponenten identifizieren und integrieren.

Vorgehen reflektieren und verbessern	4	D1G: Ich weiss wie ich mich auf das Daily Scrum vorbereite.	D1F: Ich kann meine Erfahrungen und Erkenntnisse an der Retrospektive einbringen und geeignete Verbesserungsmaßnahmen ableiten.	D1E: Ich kann eine Retrospektive vorbereiten und anleiten.
Versionsverwaltung anwenden	5	E1G: Ich kann ein Repository klonen, Änderungen committen und pushen, sowie Änderungen vom Sever übernehmen (fetch, pull).	E1F: Ich kann mit Merge Requests umgehen.	E1E: Ich kann Merge-Konflikte lösen.
	5	E2G: Ich kann das Anwenden von Branch erklären. (z.B. Git-Flow)	E2F: Ich kann Branch gezielt anwenden (z.B. Git-Flow)	E2E: Ich kann den Git-Tree kritisch hinterfragen und Verbesserungen umsetzen (z.B. rebase, cherry-pick)
Umgang mit Clean Code und Refactorings	6	F1G: Ich weiss, was zur guten Lesbarkeit von Code wichtig ist und welche Regeln es zu beachten gibt.	F1F: Ich kann einen Code durch die Anwendung der „Clean Code“-Empfehlungen besser lesbar machen und ein Refactoring durchführen.	F1E: Ich kann Funktionsbereiche zur besseren Verständlichkeit so umschreiben, dass die (zyklomatische) Komplexität entflochten wird.

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m426/kompetenzraster>Last update: **2024/03/28 14:07**