Modulidentifikation

ICT Berufsbildung
Formation professionnelle
Formazione professionale

Modulnummer	450	
Titel	Applikationen testen	
Kompetenz	Erstellt anhand einer Testbasis ein Testkonzept und leitet daraus Testfälle ab. Implementiert und dokumentiert diese, definiert Korrekturmassnahmen und überprüft Schnittstellen anhand von Sicherheitsvorgaben.	
Handlungsziele	1	Erstellt ein Testkonzept anhand von einem Praxisbeispiel inklusive Anforderungen (Testbasis).
	2	Beschreibt ein Testumfeld umfassend.
	3	Definiert innerhalb verschiedenen Teststufen/-arten (Unit Tests, Integrationstests, E2E-Tests, Systemtests, Last-/ Performancetests, Security-Tests, Benutzerakzeptanztests, Konformitätstests) Tests und deren Testmittel.
	4	Erarbeitet Vorschläge zur Verbesserung von Mängeln innerhalb von Code Reviews.
	5	Beschreibt aufgrund von Anforderungen wiederholbare Testfälle inklusive der erwarteten Resultate.
	6	Implementiert und führt automatisierte Testfälle aus und dokumentiert dessen Resultate nachvollziehbar.
	7	Definiert anhand von beobachteten/gegebenen Abweichungen Korrekturmassnahmen und setzt diese um (z.B. innerhalb Test Driven Development TDD).
	8	Testet gemäss Sicherheitskonzept Schnittstellen.
Kompetenzfeld	Testing	
Objekt	Applikation mit wenigen Features/Userstories für je eine kleine Anzahl Testfälle.	
Modulversion	1.0	
Erstellt am	11.03.2021	

Handlungsnotwendige Kenntnisse

Handlungsnotwendige Kenntnisse beschreiben Wissen, das die kompetente Ausführung der Handlungen eines Moduls unterstützt. Diese Kenntnisse dienen der Orientierung und sind nicht abschliessend definiert. Die daraus folgende Konkretisierung der Lernziele und das Festlegen des Lernwegs für den Kompetenzerwerb sind Sache der Bildungsanbieter.

Modulnummer	450
Titel	Applikationen testen
Kompetenz	Erstellt anhand einer Testbasis ein Testkonzept und leitet daraus Testfälle ab. Implementiert und dokumentiert diese, definiert Korrekturmassnahmen und überprüft Schnittstellen anhand von Sicherheitsvorgaben.

Handlungsziele und handlungsnotwendige Kenntnisse

1	1.1	Kennt die notwendigen Inhalte eines Testkonzepts wie Testziele, Testobjekte, Testarten, Testinfrastruktur, Testorganisation,Testplan.
	1.2	Kennt die Systemgrenzen zur Abgrenzung der im Testkonzept beschriebenen Umgebung.
	1.3	Kennt Strategien zur Klassifizierung von Mängeln.
2	2.1	Kennt die wesentlichen Komponenten einer Testumgebung.
	2.2	Kennt den Unterschied und die Gemeinsamkeiten einer Test- und Produktivumgebung.
3	3.1	Kennt Beispiele zu den verschiedenen Testarten.
	3.2	Kennt den Einsatz verschiedener Testarten.
	3.3	Kennt das V-Model und die Testpyramide beim Testing.
	3.4	Kennt verschiedene Testmittel.
	3.5	Kennt Möglichkeiten zum Umgang mit Abhängigkeiten (z.B. Interfaces, Injection, Mock, Spy, Stub, Dummy, Fake, Fixtures).
4	4.1	Kennt Ansätze zur statischen Überprüfung von Code und Anforderungen.
	4.2	Kennt Clean Code Prinzipien (z.B. DRY, KISS, YAGNI Namensgebung, Modularisierung, SRP) zur Erstellung von testbarem Code und zur Vermeidung von viel Testcode (z.B. auf Userstory/Verhalten testen anstelle auf Implementationsdetails).
	4.3	Kennt Codeverbesserungsmöglichkeiten, die innerhalb eines Code Reviews vorgeschlagen und umgesetzt werden können.
	4.4	Kennt den Unterschied zwischen Testing und Qualitätssicherung.
5	5.1	Kennt den Aufbau eines Testfalls (z.B. Vorbedingung, Eingabewerte, Resultat, Nachbedingung, positiv- und negativ/Fehler-Test).
	5.2	Kennt Kriterien, um einen Test wiederholbar zu machen (Retesting, Regression, Testdaten, Testbenutzer/Berechtigungen).
	5.3	Kennt den Unterschied zwischen manuellen und automatisierten Tests.
6	6.1	Kennt Testframeworks, um automatisierte Tests zu definieren (Schritte Given/Arrange, When/Act, Then/Assert) und durchzuführen.
	6.2	Kennt Methoden, um Testresultate nachvollziehbar zu dokumentieren (z.B. Traceability Tools).
	6.3	Kennt die wesentlichen Inhalte eines Testprotokolls: Tester, Testobjekt, Testzeitpunkt, Testumgebung (z.B. Browser), Testresultat.

Handlungsnotwendige Kenntnisse

7	7.1	Kennt Debugging und dessen Anwendung.
	7.2	Kennt den TDD Ansatz ausreichend, um ihn an einem einfachen Beispiel anzuwenden.
8	8.1	Kennt einfache Methoden zum Testen von Schnittstellen.

Modulversion	1.0
Erstellt am	11.03.2021