ICT Berufsbildung
Formation professionnelle
Formazione professionale

Identification du module

Numéro de module	**450**
Titre	Tester des applications
Compétence	Établir un concept de test à l'aide d'une base de test et en déduire des cas de test. Implémenter et documenter ces cas de test, définir des mesures correctives et vérifier les interfaces selon les directives de sécurité.

Objectifs opérationnels

1. Établir un concept de test basé sur un exemple pratique, exigences incluses (base de test).
2. Décrire un environnement de test de manière exhaustive
3. Définir les tests et les moyens de test en fonction des divers niveaux/types de test (tests unitaires, tests d'intégration, tests E2E, tests système, tests de charge/performance, tests de sécurité, tests d'acceptation des utilisateurs, tests de conformité).
4. Élaborer des propositions pour corriger les défauts/bugs dans les revues de code (code reviews).
5. Décrire en fonction des exigences des cas de test reproductibles, y compris les résultats escomptés.
6. Implémenter et exécuter des cas de test automatisés et en documenter les résultats de manière compréhensible
7. Définir sur la base des écarts constatés/donnés des mesures correctives et les appliquer (p. ex. dans le cadre du développement piloté par les tests (Test Driven Development [TDD]).
8. Tester des interfaces conformément au concept de sécurité.

Domaine de compétence	Application Engineering
Objet	Application avec peu de fonctionnalités/récits utilisateur pour un petit nombre de cas de test
Version du module	1.0
Crée le	11.03.2021

ICT Berufsbildung
Formation professionnelle
Formazione professionale

GRI

Connaissances opérationnelles nécessaires

Numéro de module **450**

Titre Tester des applications

Compétence Établir un concept de test à l'aide d'une base de test et en déduire des cas de test. Implémenter et documenter ces cas de test, définir des mesures correctives et vérifier les interfaces selon les directives de sécurité.

Objectifs opérationnels et connaissances opérationnelles nécessaires

1	1.1	Connaître les contenus nécessaires d'un concept de test tels que les objectifs de test, les objets à tester, les types de test, l'infrastructure de test, l'organisation de test et le plan de test.
	1.2	Connaître les limites du système pour délimiter l'environnement décrit dans le concept de test.
	1.3	Connaître des stratégies pour classifier les défauts/bugs.
2	2.1	Connaître les principaux composants d'un environnement de test.
	2.2	Connaître la différence et les points communs entre un environnement de test et un environnement productif.
3	3.1	Connaître des exemples pour les différents types de test.
	3.2	Connaître l'utilisation des différents types de test.
	3.3	Connaître le modèle en V et la pyramide de test lors du testing.

	3.4	Connaître différents moyens de test.
	3.5	Connaître des possibilités pour gérer les dépendances (p. ex. interface, injection, mock [simulacre], spy [espion], stub [bouchon], dummy [fantôme, bouffon], fake [substitut, simulateur], fixture).
4	4.1	Connaître des approches de contrôle statique de code et d'exigences.
	4.2	Connaître des principes de clean code (p. ex. DRY, KISS, YAGNI, dénomination, modularisation, SRP) pour écrire du code testable et éviter beaucoup de tests de code (p. ex. effectuer des tests sur les récits utilisateurs/comportement plutôt que sur les détails d'implémentation).
	4.3	Connaître des possibilités d'amélioration du code pouvant être proposées et mises en œuvre dans une revue de code.
	4.4	Connaître la différence entre testing et assurance qualité
5	5.1	Connaître la structure d'un cas de test (p. ex. précondition, données d'entrée, résultat, postcondition, positif et négatif/erreur de test).
	5.2	Connaître des critères de reproductibilité d'un test (retesting, régression, données de test, utilisateur de test/autorisations).
	5.3	Connaître la différence entre tests manuels et tests automatisés
6	6.1	Connaître des frameworks de test pour définir des tests automatisés (étapes given/arrange, when/act, then/assert) et les exécuter.
	6.2	Connaître des méthodes pour documenter les résultats de test de manière compréhensible (p. ex. traçabilité, outils).
	6.3	Connaître les principaux contenus d'un protocole de test: testeur, objet du test, date du test, environnement de test (p. ex. navigateur), résultat du test.

7	7.1	Connaître le débogage et son utilisation
	7.2	Connaître suffisamment bien l'approche TDD pour l'appliquer à un exemple simple.
8	8.1	Connaître des méthodes simples pour tester des interfaces.

Version du module	1.0
Crée le	11.03.2021

Niveau d'exigences	Niveau C	Description	Verbes typiques des activités
Savoir	**C1**	Restituer des informations et les retrouver dans des situations similaires.	Désigner, noter, énumérer, nommer, restituer.
Comprendre	**C2**	Non seulement restituer des informations, mais les comprendre.	Décrire, expliquer, commenter, reformuler, démontrer, caractériser
Appliquer	**C3**	Appliquer des informations circonstancielles dans différentes situations.	Appliquer, établir, exécuter, calculer, utiliser, traduire, transposer
Analyser	**C4**	Décomposer une situation en ses divers éléments, établir les relations entre ces éléments et en identifier les tenants et les aboutissants.	Interpréter, analyser, résoudre, différencier, décomposer, identifier, examiner, comparer, diviser, contrôler, mesurer
Synthétiser	**C5**	Combiner les éléments d'une situation pour former un tout, ou concevoir la solution d'un problème.	Justifier, noter, structurer, mettre en place, élaborer, projeter, développer, concevoir, combiner, construire, optimiser, planifier, rédiger, établir, élaborer
Evaluer	**C6**	Evaluer des informations et des situations déterminées selon des critères prédéfinis ou selon ses propres critères.	Apprécier, évaluer, qualifier

Niveaux d'exigences (taxonomie)
L'indication du niveau d'exigences des objectifs évaluateurs en reflète le degré de difficulté. On distingue six niveaux de compétences (C1 à C6) Le tableau ci-dessous les présente en détail.