

## Concevoir, déployer et automatiser des tests logiciels selon l'approche ISTQB

Date et durée	
Code formation : TLA-RS7450 Durée : 5 jours Nombre d'heures : 35 heures	
Description	





Cette formation permet d'acquérir une démarche structurée de conception, d'exécution et de pilotage des tests logiciels, conforme aux principes méthodologiques de l'approche ISTQB.

Les trois premiers jours sont consacrés à la maîtrise du processus complet de test : analyse des exigences, identification des risques, définition d'une stratégie adaptée, conception de scénarios traçables, exécution des tests, gestion des anomalies et production d'un rapport structuré. Des mises en situation professionnelles permettent la production de livrables concrets.

Les deux journées complémentaires introduisent l'automatisation en environnement Agile : rôle du testeur dans Scrum, rédaction de tests à partir de user stories (tableau, Gherkin), tests d'API avec Postman et principes d'intégration de l'automatisation dans le cycle de vie logiciel.

La formation alterne apports méthodologiques et travaux pratiques afin de garantir une acquisition opérationnelle des compétences, immédiatement mobilisables en contexte professionnel.

La formation alterne apports méthodologiques et travaux pratiques afin de garantir une acquisition opérationnelle des compétences, directement mobilisables en situation professionnelle.



# Oo2 Formations & Consulting

*Oo2 est un organisme de formation spécialisé dans les certifications professionnelles et les compétences techniques en management, cybersécurité, gestion de projet et qualité logicielle, accompagnant les entreprises et les professionnels dans le développement de leurs expertises et de leurs performances.*

## Objectifs

À l'issue de la formation, le participant sera en mesure de :

1. **Analyser le contexte d'un projet logiciel** afin d'identifier les enjeux qualité, les risques associés et les exigences de couverture nécessaires à la validation du produit.
2. **Structurer une stratégie de tests** en sélectionnant les niveaux et les types de tests appropriés au regard du cycle de développement, des contraintes projet et des risques identifiés.
3. **Formaliser une stratégie de tests dans un livrable structuré**, intégrant les objectifs, le périmètre, les responsabilités, les critères d'entrée et de sortie, ainsi que les ressources humaines et techniques mobilisées.
4. **Mobiliser les techniques de conception de tests adaptées** (partitions d'équivalence, valeurs limites, tables de décision, transitions d'états, couverture structurelle), en garantissant la cohérence méthodologique et la traçabilité des cas produits.
5. **Justifier le choix entre tests statiques et dynamiques**, en fonction de la nature du logiciel, du contexte d'analyse et des objectifs de validation recherchés.
6. **Élaborer des scénarios de test complets et reproductibles**, précisant les préconditions, les données d'entrée, les étapes d'exécution et les résultats attendus.
7. **Configurer et exploiter un environnement de test conforme aux prérequis définis**, dans le respect des procédures établies.
8. **Exécuter les cas de test et comparer systématiquement les résultats obtenus aux résultats attendus**, afin de statuer objectivement sur la conformité du comportement logiciel.
9. **Documenter les anomalies détectées dans un outil de suivi**, en décrivant précisément les conditions de reproduction, les écarts observés, la sévérité et la priorité associées.
10. **Organiser et piloter une campagne de test**, en planifiant les activités, en estimant la charge de travail, en priorisant les actions selon les risques et en construisant des outils de suivi adaptés.
11. **Analyser les indicateurs de test et évaluer les risques résiduels**, afin de produire un rapport structuré permettant d'éclairer la prise de décision.
12. **Contribuer à l'amélioration continue du processus de test**, en identifiant les anomalies récurrentes, en analysant les retours d'expérience et en formulant des recommandations correctives et préventives.  
----- **sensibilisation à l'automatisation des tests logiciels** -----
13. **Connaître les principes fondamentaux de l'automatisation des tests** et son positionnement dans le cycle de vie du logiciel.
14. **Démontrer les avantages et les limites de l'automatisation des tests** en fonction du contexte projet.
15. Énumérer les principales approches d'automatisation (scripts, frameworks, tests pilotés par les données, intégration continue, CI/CD, etc.).
16. **Identifier les stratégies d'automatisation adaptées** aux objectifs qualité, aux risques et aux contraintes techniques.
17. Comprendre les principes permettant d'**automatiser certaines étapes du cycle de vie logiciel** (exécution répétitive, intégration continue, suivi des résultats, maintenance des scripts).

## Points forts

- Formation **certifiante**, avec examen inclus
- **Tarif tout compris** : formation, supports, certification
- **Retake - session de rattrapage gratuite** en cas d'échec
- Approche **pratique et concrète**, centrée sur des cas réels
- Encadrement par des **professionnels du test logiciel**

## Certification

La certification est délivrée à l'issue d'un processus d'évaluation structuré en **deux étapes complémentaires**, fondé sur une **mise en situation professionnelle contextualisée**. L'objectif est de valider la maîtrise

opérationnelle des différentes composantes d'une campagne de test logiciel.

### Étape 1 - Dossier professionnel (50 % de la note finale)

Le candidat rédige en toute autonomie (à la maison, sans télésurveillance) un dossier à partir d'un cas fictif fourni par Oo2. Le dossier devra comprendre obligatoirement les éléments suivants :

- Un document argumenté de stratégie de test (10 à 15 pages max)
  - Des scénarios de test rédigés
  - Un tableau de planification des tests
  - Un modèle de ticket d'anomalie
  - Un tableau de bord de suivi des résultats de tests
- > **Objectif** : évaluer la capacité du candidat à structurer, planifier, concevoir, exécuter et documenter une campagne de test cohérente avec les exigences du projet.

### □ Étape 2 - Soutenance orale (50 % de la note finale) - 30 min

Le candidat présente oralement son travail devant un jury :

- 20 minutes de présentation orale à partir d'un support synthétisant le dossier
  - 10 minutes de questions-réponses avec le jury
- > **Objectif** : apprécier la posture professionnelle, la capacité à justifier ses choix, la maîtrise des outils, ainsi que la clarté de la restitution.

### Conditions de réussite

- Le candidat doit valider **au moins 75 % des critères d'évaluation** pour chaque compétence.
- La certification est obtenue uniquement si **toutes les compétences** sont validées.

### Modalités d'évaluation

Quiz / QCM  
Travaux Pratiques  
Etude de cas  
Examen blanc

### Pré-requis

Suivre cette formation nécessite les prérequis suivants :

- Justifier d'au moins 1 an d'expérience professionnelle dans le domaine de la production logicielle

### Public

Cette formation s'adresse aux publics suivants :

- **Les professionnels de la production logicielle** (testeurs, développeurs, analystes, chefs de projet, product owners, architectes logiciels...)
- **Les managers ou responsables d'équipes techniques** souhaitant renforcer leurs compétences en matière de tests pour mieux piloter la qualité des livrables
- Toute personne impliquée dans la **conception, la validation ou la mise en qualité** d'un produit logiciel

### Programme

### JOUR 1 - Fondamentaux et stratégie de test

## 1. Fondamentaux du test logiciel selon l'approche ISTQB

- Pourquoi tester ? Objectifs et limites du test
- Les 7 principes du test
- Terminologie : erreur, défaut, défaillance
- Indépendance des tests et rôles des parties prenantes
- Processus de test structuré (planification, analyse, conception, implémentation, exécution, clôture)
- Place du test dans le cycle de vie logiciel (cycle en V, Agile)

## 2. Structurer une stratégie de test

- Identification des risques et zones critiques
- Définition et justification du niveau de couverture
- Sélection des niveaux de test (unitaire, intégration, système, acceptation)
- Sélection des types de test (fonctionnel, non fonctionnel, régression...)
- Formalisation d'une stratégie structurée :
  - Objectifs
  - Responsabilités
  - Critères d'entrée et de sortie
  - Moyens humains et techniques mobilisés

**Travaux pratiques :** Rédaction d'une stratégie de test argumentée à partir d'un cas projet.

## JOUR 2 - Conception et exécution des tests

### 3. Tests statiques et revues

- Objectifs des tests statiques
- Types de revues : informelle, walkthrough, inspection
- Rôles et responsabilités dans une revue
- Apports des revues dans la prévention des défauts

### 4. Techniques de conception des tests

- Techniques boîte noire :
  - Partitions d'équivalence
  - Valeurs limites
  - Tables de décision
  - Tests basés sur états
- Techniques boîte blanche :
  - Couverture des instructions (statement coverage)
  - Couverture des décisions (decision coverage)
- Justification du choix des techniques selon le contexte
- Traçabilité des cas de test

**Travaux pratiques :** Conception de scénarios complets (Préconditions, Données d'entrée, Étapes, Résultats attendus)

### 5. Exécution des tests et gestion des défauts

- Configuration de l'environnement de test
- Exécution des cas selon les procédures définies
- Comparaison systématique résultat attendu / obtenu
- Statut succès / échec
- Cycle de vie d'un défaut
- Rédaction structurée d'un ticket :

- Description
- Conditions de reproduction
- Environnement
- Gravité / priorité

**Travaux pratiques :** Exécution et documentation complète d'anomalies.

### **JOUR 3 - Pilotage, outils et amélioration continue**

#### **6. Organisation et pilotage des activités de test**

- Planification des activités
- Estimation de charge
- Priorisation selon les risques
- Recensement des ressources mobilisées
- Construction d'un tableau de bord
- Indicateurs :
  - Taux de réussite
  - Couverture
  - Densité de défauts

#### **7. Outils de test et gestion**

- Outils de gestion des tests
- Outils de gestion des anomalies
- Principes de l'automatisation (concepts)
- Bénéfices et limites des outils

#### **8. Restitution et amélioration continue**

- Rédaction d'un rapport de test structuré
- Évaluation des risques résiduels
- Restitution claire et priorisée des résultats
- Exploitation des indicateurs
- Identification des anomalies récurrentes
- Proposition d'actions correctives et préventives
- Intégration des retours d'expérience

**Travaux pratiques :** Analyse complète d'une campagne simulée et production d'un rapport final.

### **☐ JOUR 4 - Tests en environnement Agile et API**

#### **Module 9 - Le rôle du testeur en environnement Agile**

- Rappel des principes Agile et Scrum
- Rôle du testeur dans une équipe Scrum
- Collaboration avec Product Owner et développeurs
- Backlog, user stories et critères d'acceptation
- Rédaction de tests :
  - Méthode tableau
  - Gherkin (Given / When / Then)
- Intégration des tests dans les sprints

**Travaux pratiques :** Rédaction de cas de test à partir d'une user story et formalisation en Gherkin.

#### **Module 10 - Tests API et environnements**



- Notions fondamentales API
- Format JSON
- Méthodes HTTP (GET, POST, PUT, DELETE)
- Tests API avec Postman
- Analyse des réponses (statut, payload)
- Environnements :
  - Développement
  - Staging
  - Production
  - Canary / Blue-Green

**Travaux pratiques :** Création et exécution de requêtes API avec Postman et analyse des résultats.

## 📅 JOUR 5 - Introduction à l'automatisation et gestion des anomalies

### Module 11 - Sensibilisation à l'automatisation des tests

- Différence entre test manuel et automatisé
- Quand automatiser ?
- Tests répétitifs et régression
- Introduction aux outils d'automatisation
- Démonstration Playwright (principe de fonctionnement)
- Place de l'automatisation dans un projet Agile

**Travaux dirigés :** Analyse d'un cas de test et identification des éléments automatisables.

### Module 12 - Gestion des anomalies et qualité continue

- Structure d'un rapport d'anomalie
- Bonnes pratiques de rédaction
- Utilisation d'un outil de suivi (exemple Jira)
- Cycle de vie d'un bug
- Suivi et communication dans l'équipe Agile
- Contribution à l'amélioration continue

**Travaux pratiques :** Rédaction d'un ticket d'anomalie complet à partir d'un cas simulé.