

Campagne de test

Comprendre les étapes de la
campagne de test



Compétence demandée :
**Comprendre les étapes de la
campagne de test**

1. Campagne de test
2. La recette
3. La planification
4. La conception
5. Exécution
6. Bilan

LA CAMPAGNE DE TEST

La partie tests peut représenter 50% de la charge totale d'un projet

La stratégie de test doit être intégrée au processus de développement du logiciel.

La stratégie de test dépend :

- De la criticité du produit à réaliser
- Du coût de développement

Une campagne de test est un ensemble de cas de tests à exécuter sur une période donnée. Le résultat de ces exécutions est alors synthétisé dans un bilan.

Une campagne consiste à définir :

- Les ressources mises en œuvre (équipes, testeurs, outils...)
- Les processus de test

La **recette** est l'opération par laquelle le client reconnaît que le produit livré par le fournisseur est **conforme à la commande passée, qu'il est exploitable** dans le SI de l'entreprise et enfin qu'il est opportun de le mettre à **disposition des utilisateurs**.

Bâtir la stratégie de test : planifier les différentes activités sans négliger la préparation logistique nécessaire à une réalisation dans de bonnes conditions

Réalisation des tests : les defects sont remontés

Le bilan : améliorer la prochaine série de tests et validation du PV de recette

Le cahier de recettage et le **plan de tests** vous permettront de structurer les tests et leur résultat.

Vous le mettrez à jour à chaque phase de test importante et le garderez dans vos archives. Un plan bien conçu augmente vos chances de délivrer un produit solide et fiable.

- Phase de planification
- Phase de conception
- Phase d'exécution
- Le bilan

LA PLANIFICATION

Planification

Créer un plan de test

Les tests à exécuter

Créer les objectifs de test

Les exigences à vérifier

Prévoir le test

La campagne à associer

Vérifier la couverture de test

La complétude à vérifier

Le **plan de test** est utilisé par les responsables de l'assurance qualité pour présenter la **stratégie générale de test** pour un projet, ainsi que des informations de statut sur la **progression des tests** au niveau du projet.

Ils peuvent surveiller la couverture de test et le statut en cours du processus d'assurance qualité par rapport au plan de test défini et évaluer le niveau de maturité du logiciel via la gestion des défauts et d'autres mesures clés liées aux tests.

Les **exigences** produit qui décrivent le produit ou le système à un haut niveau.

Elles répondent aux **exigences métier** et sont couramment formulées comme les fonctionnalités que le système doit réaliser. On les appelle majoritairement **exigences fonctionnelles** ou **spécifications fonctionnelles**.

Cela dit, les exigences peuvent couvrir **des champs de conformité** (à ne pas mettre en opposition avec le fonctionnel) :

Exemple : **SQUARE**, ISO 25010

La couverture de test correspond au pourcentage de fonctionnalités avec au moins 1 cas de test effectué.

Concrètement qu'est-ce que cela veut dire ? Pour avoir une couverture des méthodes de 100%, il faut avoir au moins 1 test par fonctionnalité.

LA CONCEPTION

Conception

Test scénario

Histoire et contexte

Test case

La chose à tester

Test step

Quels sont les étapes du test ?

Résultat attendu

Que doit-on obtenir ?

Le **scénario de test** est une activité de test de logiciel qui utilise des scénarios : des **histoires hypothétiques pour aider le testeur à résoudre un problème** ou un système de test complexe.

Un **cas de test** a des composants qui décrivent une entrée, **une action et une réponse attendue**, afin de déterminer si une fonctionnalité d'une application fonctionne correctement.

Un **cas de test** est un ensemble d'instructions sur «**COMMENT**» pour valider un objectif / une cible de test particulier, qui, une fois suivi, nous dira si le comportement attendu du système est satisfait ou non.

L'EXECUTION

Execution

Test run

Qu'a-t-on testé ?

Defects

Quels sont les défauts à remonter ?

Un **test run** est une **exécution** d'un test soit par un utilisateur soit par une machine mais **dont le résultat obtenu a été comparé avec le résultat attendu**

La **détection des défauts** est la tâche la plus importante du testeur. Il est important de commencer les tests **le plus tôt possible** car des défauts peuvent être trouvés tout au long du processus de développement logiciel.

Plus nous les identifions tôt, plus les coûts de leur réparation sont faibles. Tous les defects doivent être enregistrés et suivis afin qu'ils puissent être correctement gérés et résolus.

LE BILAN

Bilan

```
graph LR; Bilan[Bilan] --- PV[Le PV de recette]; Bilan --- Retrospective[Retrospective]; PV --- FeuVert[Donne-t-on le feu vert ?]; Retrospective --- Difficultes[Quelles ont été les difficultés rencontrées ?];
```

Le PV de recette

Donne-t-on le feu vert ?

Retrospective

Quelles ont été les
difficultés rencontrées ?

La signature d'un procès-verbal de recette est une pratique qui s'est largement répandue auprès des prestataires proposant notamment des prestations dans le domaine informatique.

C'est une bonne chose en soi, mais cette démocratisation dans l'usage des PV de recette a également amené son lot de litiges.

Plusieurs jugements ont d'ailleurs permis de préciser la **réelle valeur juridique d'un PV de recette**, ainsi que la responsabilité du prestataire en cas de mécontentement du client après la signature d'un tel document.