Le métier de testeur

En quoi consiste ce métier ?

Informaticien, le testeur est le spécialiste de la chasse aux bugs, ces erreurs qui empêchent le bon fonctionnement d'un logiciel. À lui de les signaler au service développement.

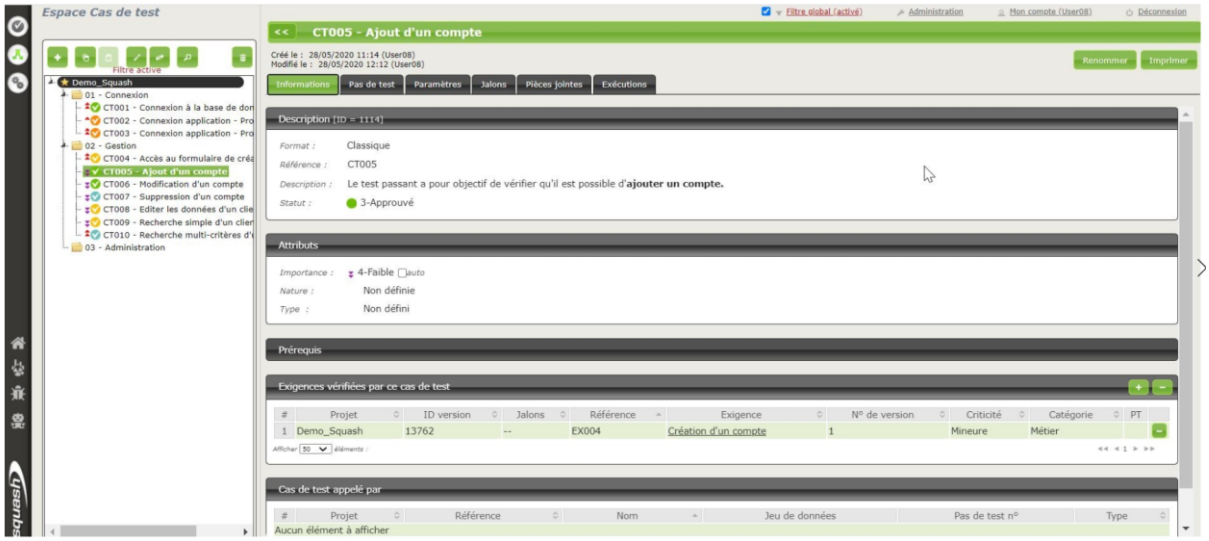
Il s'assure qu'une application correspond au cahier des charges, qu'elle ne détériore pas le système d'information, qu'elle est utilisable sur tous types de matériels informatiques et dans tous les cas d'utilisation. Il établit une stratégie, élabore les outils de tests, les exécute, analyse les résultats, rédige des rapports et transmet les anomalies détectées au développeur informatique chargé des corrections.

Le testeur peut travailler sur un logiciel médical, un logiciel culturel et pédagogique : il s'assure que l'accompagnement sonore et les dialogues sont synchronisés.

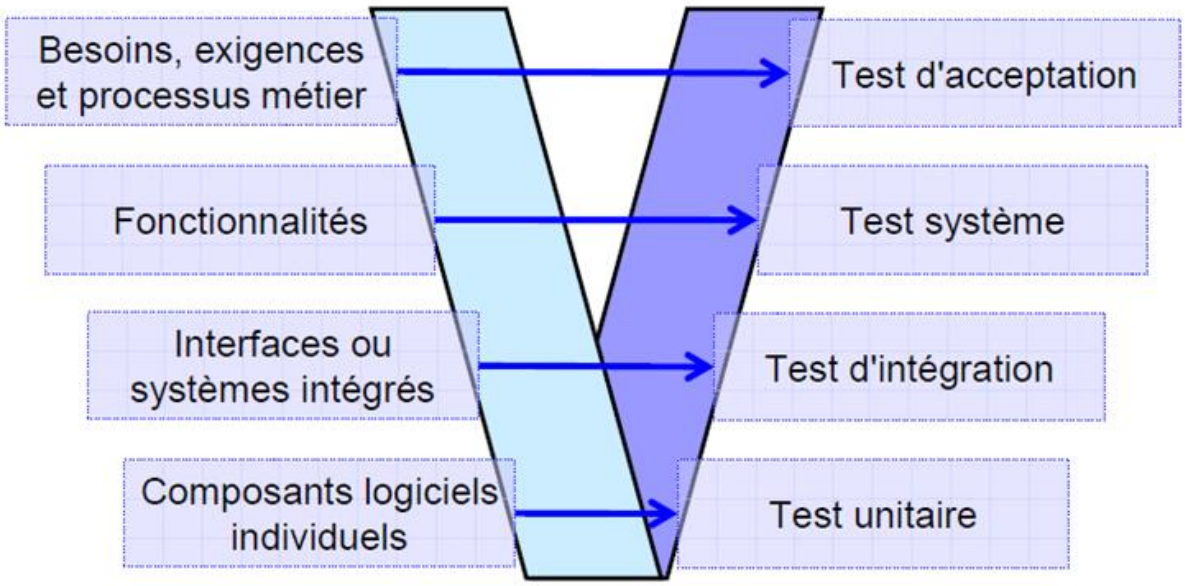
Excel cahier de test

NURSERIE		Nom du projet			
		Description:		Redacteur :	Testeur :
		Module :			
N°	Action	Résultat Attendu	Cas de tests	Résultat obtenus	Correction à effectué
	Cliquer sur le bouton ajouter une formation	Redirection sur le formulaire d'ajout de formation			
	Cliquer sur le bouton ajouter une expérience	Redirection sur le formulaire d'ajout d'expérience			

Logiciel pour les tests (Squash)



LES NIVEAUX DE TESTS



1.Test de composants, ou test unitaire

Tester unitairement les programmes avec l'accès au code

Objectif : trouver des défauts et réduire les risques liés aux composants individuels du système en cours de test avant son intégration

Base de test : Code (compilateur, débogueur, revues de codes), base de données (formatage des données), normes de codage (% commentaires, taille des fichiers, règles de nommage, indentation du code…), spécifications détaillées (nouvelle classe à créer par exemple)

Types de tests : fonctionnels (boîte noire), utilisation des ressources (gestion de la mémoire), performance (temps de calculs, échanges de data), structurels (boîte blanche), basés sur les erreurs

Objet de test : composants, petits programmes, utilitaires de conversion de données, de migration et requêtes de base de données

Outils : niveau API (pilotes et bouchons), open source et commercial)

Responsables : les développeurs

2.Test d'intégration

Tester les interfaces et interactions entre composants ou systèmes

Règles :

- Plus la portée de l'intégration est vaste, plus il est difficile d'isoler le composant responsable des défaillances
- Intégration incrémentale préférable au « Big Bang »
- Utilisation de bouchons ou simulateurs

- **Objectif** : s'assurer du bon fonctionnement de chaque assemblage de composants, des interactions avec le matériel, des interfaces entre les modules
- **Base de test** : Conception, architecture, schémas structurels, flux de données, workflows de changements d'état, cas d'utilisation
- **Types de tests** : Fonctionnels, utilisation des ressources, performances, liens internet
- **Harnais et outils** : niveau API (Application Program Interface) et CLI (Command Line Interface)
- **Responsable** : idéalement à la fois les testeurs et les programmeurs, mais souvent personne

3. Tester fonctionnellement le comportement système complet

Objectif : trouver des défauts, obtenir de la confiance et réduire les risques dans les comportements généraux et particuliers, les fonctions, et les réponses du système

Base de test : exigences, conception de haut niveau, cas d'utilisation, user expérience, environnements

Types de tests : fonctionnels, sécurité, performances, fiabilité, utilisabilité, portabilité, tests de bout en bout (end-to-end)

Objets de test : le système complet, dans un environnement le plus réaliste possible, incluant les manuels utilisateurs et opérationnels, et les données de configuration

Responsable : testeurs indépendants

4. Tests d'acceptation :

Valider que le produit atteint les exigences et niveau de confiance demandé

Les différentes formes de tests d'acceptation :

Tests d'acceptation utilisateur

- Utilisabilité du système par les utilisateurs finaux

Tests opérationnels

- Test d'exploitabilité (backup/restauration, reprise, maintenance)

Tests d'acceptation contractuelle et réglementaire

- Réponds aux exigences contractuelles et réglementaires

Tests alpha et beta

- Validation par des utilisateurs finaux avant la livraison

Tests Alpha :

les tests Alpha sont réalisés « en usine », sur invitation, par un petit groupe de key-users, qui va réaliser des tests d'acceptation sur le site de développement de l'application, mais sans lien aucun avec l'équipe de développement.

Tests Bêta :

appelés tests sur le terrain, ils sont proposés à tout utilisateur final de l'application (configuration « closed-beta») ; ils sont réalisés en dehors du site de développement ; lorsqu'ils sont proposés au public le plus large possible (configuration «open-beta») le but est de récolter des avis de la part de personnes décorrélées du contexte, qui peuvent, de surcroît, proposer l'ajout de nouvelles fonctionnalités pour une prochaine version