|  |
| --- |
| PROCEDURE DE TEST |

PIQSOFT 

# SOMMAIRE :

[**I.** **Quelques généralités** 3](#_Toc124668031)

[***1.*** ***Pourquoi réaliser des tests*** 3](#_Toc124668032)

[***2.*** ***Qu’est-ce qu’un test ?*** 3](#_Toc124668033)

[***3.*** ***Les 7 Principes de base*** 3](#_Toc124668034)

[**II.** **Les niveaux de test** 4](#_Toc124668035)

[***1.*** ***Niveau test unitaire*** 4](#_Toc124668036)

[***2.*** ***Niveau test d’intégration*** 4](#_Toc124668037)

[***3.*** ***Niveau test système*** 4](#_Toc124668038)

[***4.*** ***Niveau de test d’acceptation*** 5](#_Toc124668039)

[**III.** **Les types de test** 5](#_Toc124668040)

[***1.*** ***Tests fonctionnels*** 5](#_Toc124668041)

[***2.*** ***Tests non fonctionnels*** 5](#_Toc124668042)

[***3.*** ***Tests de confirmation et de régression*** 5](#_Toc124668043)

[**IV.** **Déroulement de test** 6](#_Toc124668044)

[***1.*** ***Les attendus des différents tests*** 6](#_Toc124668045)

[***2.*** ***La procédure à suivre*** 6](#_Toc124668046)

## **Quelques généralités**

## ***Pourquoi réaliser des tests***

Les tests sont nécessaires au développement et à la réalisation correcte des systèmes d'informations. Les tests sont indispensables pour détecter rapidement les problèmes logiciels et de les corriger dans un bref délai.

Comme chaque chose, les logiciels présentent également des défauts qui peuvent avoir de conséquences importantes pour les utilisateurs mais aussi pour les « affaires ». De plus, afin d’améliorer la sureté et la connaissance du produit livré des tests sont réalisés.

## ***Qu’est-ce qu’un test ?***

Un test est une procédure de vérification partielle d’un système. En effet, c’est un processus consistant à évaluer et à vérifier qu'un produit ou une application logicielle fait ce qu'il ou elle est censé(e) faire. (Définition ibm)

L’objectif d’un test est donc dans un premier temps d’évaluer la qualité d’un produit pour ensuite l’améliorer si des problèmes sont présents.

On note que toutes vérifications et validations de tests doit être interprétés au sens de la norme NF525.

## ***Les 7 Principes de base***

Il existe 7 principes de base pour les tests qui sont des principes communs à tous les tests logiciels :

* **Principe 1** : **Les tests montrent la présence de défauts.**

Les tests montrent les défauts présents mais cela signifie que le logiciel ne vérifie pas complétement qu’il n’y a pas de défauts. Les tests permettent de réduire le nombre de défauts cachés dans le logiciel mais ne peuvent pas prouver que le logiciel est exempt de défauts.

* **Principe 2** : **Des tests exhaustifs sont impossibles.**

Il est impossible de tout tester dans le logiciel. Afin de ne pas réaliser des tests exhaustifs en termes de coût par exemple, il serait plus utile de tester les fonctions majeures du logiciel.

* **Principe 3**: **Des tests précoces.**

Afin de détecter les défauts de manière précoce, les activités de tests doivent être débutés assez rapidement. Réaliser les tests à un stade précoce peut aider à réduire le temps mais aussi le coût.

* **Principe 4**: **Regroupement des défauts.**

80% des défauts se trouvent dans 20% des modules, d’après la règle de Pareto de 80.20. Donc si l’on trouve un défaut dans un module cela signifierai que nous avons plus de chance de faire face à de nombreux défauts.

* **Principe 5** : **Le paradoxe des pesticides.**

Les tests ne présenteront plus de nouveaux défauts si le même test est réalisé à plusieurs reprises. Pour trouver de nouveaux défauts, il faut que les tests soient réévalués régulièrement. Pour cela, il est possible d’écrire de nouveaux tests, d’augmenter la durée du test par exemple.

* **Principe 6** **: Les tests dépendent du contexte.**

Il faut utiliser différentes approches, méthodologies, techniques pour effectuer des tests. Le contexte ne sera pas le même pour tous les logiciels.

* **Principe 7** : **Absence d’erreur**.

Il est important de se concentrer sur les besoins des clients. Si le système n’est pas utilisable ce n’est pas intéressant de trouver et de résoudre des défauts.

# **Les niveaux de test**

## ***Niveau test unitaire***

Ces tests permettent d’assurer le bon fonctionnement d'une partie spécifique du logiciel ou d'une partie du programme. Il s’agit d’isoler une partie d’un code et de vérifier qu’il fonctionne correctement.

## 

## ***Niveau test d’intégration***

Les tests d’intégration font suite aux tests unitaires. Le but de ce type de test est d’assembler tous les modules indépendants et de vérifier que toutes les parties fonctionnent ensemble.

## ***Niveau test système***

Les tests système permettent de tester le comportement entier du système afin d’évaluer sa conformité aux exigences spécifiées. Pour les tests système, aucune connaissance de la conception interne ou de la logique du système est requise, il s’agit des techniques de test que l’on nomme boîte noire.

Ce sont les tests qui sont le plus utilisés.

## ***Niveau de test d’acceptation***

Les tests d’acceptation qui sont effectués par des utilisateurs finaux (par exemple avec une bêta test) ont pour but de confirmer que le produit obtenu à la fin est bien en correspondance avec les attentes des utilisateurs.

# **Les types de test**

Les tests sont classés en trois catégories :

## ***Tests fonctionnels***

Les tests fonctionnels sont tous les types de tests qui permettent de vérifier si la conformité de l’application testée est bien celle attendue. En effet, il permet de tester toutes les fonctionnalités du logiciel. On peut citer comme test fonctionnel les tests unitaires ou encore les tests d’intégration.

## ***Tests non fonctionnels***

Les tests non fonctionnels sont des tests de performances, de charges, de stress, d’utilisation mais aussi de fiabilité. Ils se chargent de tester « comment » le système va fonctionner. Ils sont décris comme des tests permettant de vérifier les éléments non fonctionnels du logiciel.

## ***Tests de confirmation et de régression***

Une fois qu’un défaut est trouvé et résolu, il faut que le logiciel soit à nouveau testé afin de s’assurer que le défaut original a été résolu. C'est ce que l'on appelle les tests de confirmation. Ils sont utilisés pour vérifier si un logiciel fonctionne correctement en répondant aux exigences spécifiées.

Les tests de régression sont utilisés pour vérifier si les modifications apportées au logiciel n’ont pas induit d’erreurs ou de bugs dans d’autres parties du système.

Ensemble, ils sont utilisés pour s’assurer que les modifications apportées au système ne causent pas de problème pour les utilisateurs.

# **Déroulement de test**

## ***Les attendus des différents tests***

Pour pouvoir réaliser des tests, nous exécutons différentes tâches pour conclure sur des résultats. Pour cela il faut déjà que :

L'exactitude du résultat du test effectué et l'expérience utilisateur sont vérifiées afin de s'assurer que la fonctionnalité ajoutée ou le bug corrigé pendant le processus de test est correct et utile.

## ***La procédure à suivre***

La procédure du test est composée de différentes étapes. Afin de contrôler la qualité et la convivialité de notre test et d'identifier et de prévenir les problèmes pouvant survenir lors de la phase d'utilisation, nous passons le processus de test avec les étapes suivantes.

* La fiche ouverte dans Trello pour la fonctionnalité ajoutée au programme ou le bug corrigé par le programmeur est transféré dans l'onglet du processus de test.
* Le programmeur indique le but et le résultat attendu du code écrit à l'utilisateur qui testera cette carte.
* Un contexte de test de démonstration est créé avec les données nécessaires pour effectuer le test correct.
* L'utilisateur de test teste la fonction spécifiée sur la carte en utilisant différentes variantes du programme.
* Si dans la partie précédente, il y a des erreurs ou des notes qui posent problèmes, ceux-ci sont alors notifiés et inscrits dans la carte qui a été créée sur Trello.
* S'il y a des choses à corriger à partir du résultat du test, cette carte est à nouveau transférée dans la section des tâches.
* Une fois les dispositions nécessaires prises, il est renvoyé au processus de test par l'équipe du logiciel.
* Lorsque le résultat du test est réussi, le test effectué est ajouté à la version de mise à jour du projet.

Une image contenant texte, intérieur, capture d’écran

Description générée automatiquement

Voici un exemple de la procédure expliquée précédemment. On peut apercevoir ici les attendus que l’on a entré mais aussi les résultats du test après avoir été exécutée.