

1_Introduction

Ajouter de la complexité à un projet implique une augmentation du risque de bugs. C'est dans ce genre de cas que les tests démontrent toute leur utilité.



En quoi faire des tests permettrait-il de réduire le risque de bugs ?

Tout simplement, cela permet de savoir, au travers de chaque test, quelle partie de notre application a un problème. De plus, les tests vont permettre de vérifier que l'implémentation d'une nouvelle fonctionnalité n'a pas cassé le fonctionnement de notre application à un endroit différent. Cela représente un gain considérable de temps si un problème survient.

Les tests en informatique

Lorsqu'on crée un nouveau projet, on vérifie progressivement que **le code fonctionne** en lançant le programme. Puis on considère que le programme est terminé lorsqu'il intègre bien toutes les fonctionnalités à développer.

Nous, développeurs, sommes assez **paresseux**. Nous aimons créer des programmes pour **automatiser** ce que nous n'aimons pas faire, et ainsi passer plus de temps sur ce qui nous plaît. Or, quoi de plus lassant que de tester manuellement le fonctionnement d'un programme ? De plus, cela est assez **limité**, car les vérifications manuelles ne nous permettent pas de vérifier que la tuyauterie interne de notre programme fonctionne bien.

C'est pourquoi de nombreux langages intègrent la possibilité de **tester automatiquement** un programme. Concrètement, nous allons écrire un

script en parallèle de notre programme, qui aura pour but unique d'**effectuer les tests que nous ne voulons pas faire manuellement**.

Les **avantages** sont nombreux ! Nous avons déjà parlé de l'**économie de temps** réalisée. Au-delà de ce gain important, un bon ensemble de tests améliore la **collaboration** entre développeurs et l'ajout de fonctionnalités. En effet, on a pas à craindre de "casser" les fonctionnalités déjà développées : les tests nous diront tout de suite si un problème est arrivé.

Dans ce cours, nous verrons ensemble comment concevoir un plan de tests et comment les coder en Python.

Il existe deux grandes formes de tests :

- les tests **automatisés** : ils sont codés par des informaticiens et exécutés par un ordinateur. Leur but est de confirmer que les différentes parties du programme correspondent aux spécifications techniques ;
- les tests **manuels** : ils sont réalisés par des humains, généralement une personne responsable de la qualité d'un projet, ou le client lui-même. Leur but est de vérifier que les différentes fonctionnalités développées correspondent aux spécifications fonctionnelles.



Afin de bien comprendre les différents tests, imaginons le logiciel "Crocoplus". Il s'agit d'un système de gestion destiné **aux zoos**. Une des fonctionnalités est de compter le nombre de crocodiles et le nombre d'humains, afin de s'assurer que l'un n'a pas mangé l'autre.