



Institut National
Universitaire
Champollion

Chapitre 5

Tests



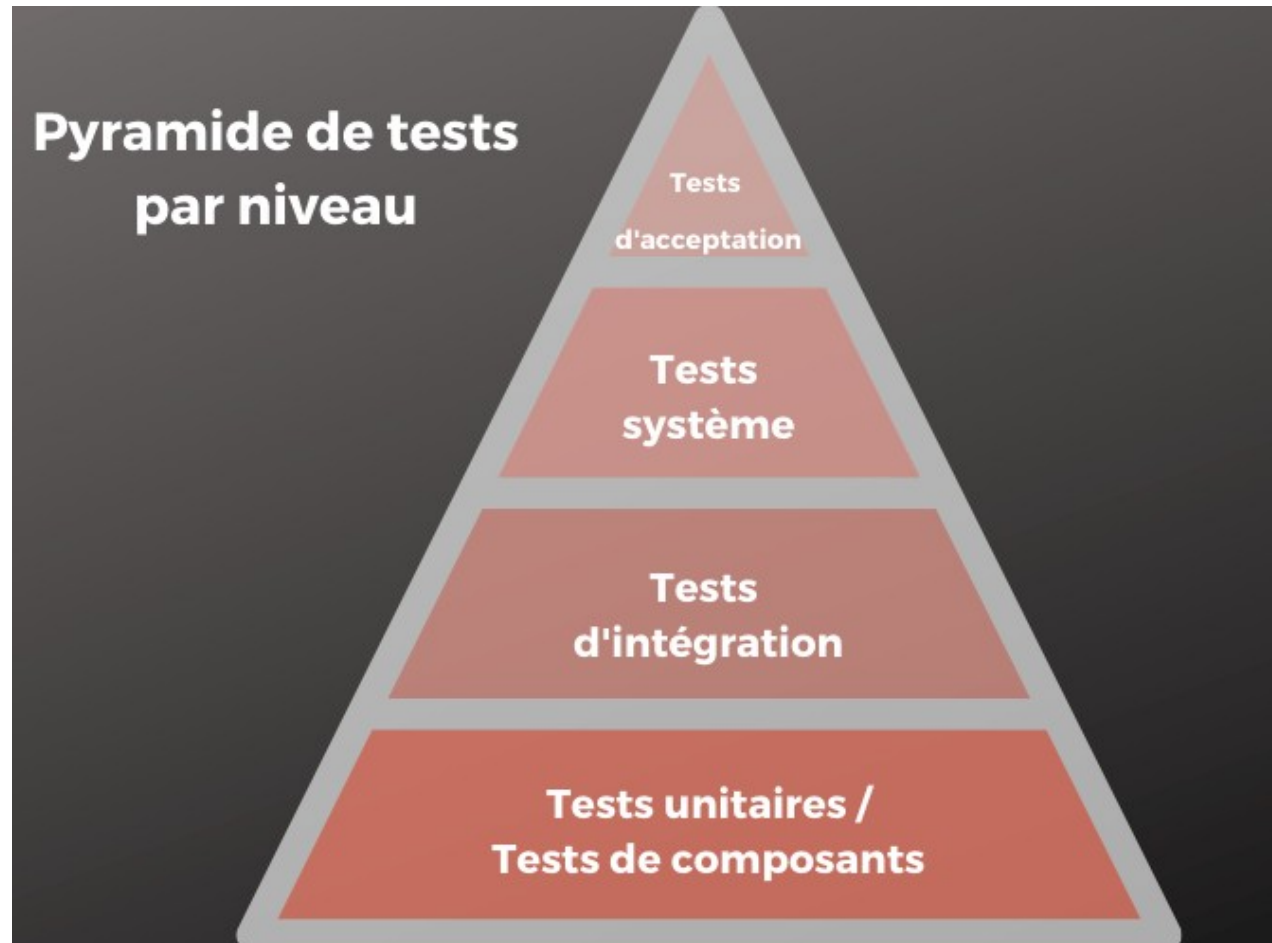
Plan du chapitre

1. introduction : notion de test	3
2. Les test unitaires	7

Notion de test

Notion de test

Plusieurs types de test :



Test d'acceptation

Ils permettent de valider le développement entre différentes équipes du projet puis auprès des utilisateurs.

Tests systèmes

ils permettent de s'assurer de la fonctionnalité globale du logiciel et de son comportement sur les terminaux d'utilisation.

Tests d'intégration

Ils permettent de s'assurer que plusieurs composantes de votre logiciel interagissent conformément aux cahiers des charges et délivrent les résultats attendus. (interactions entre modules, classes ...) équipes du projet puis auprès des utilisateurs.

Tests unitaires ou de composants

Initiés par le développeur lui-même dans l'optique de vérifier son code au niveau du composant qu'il doit réaliser.

Ces tests doivent être automatisés rapidement pour permettre de valider la non régression du fonctionnement du composant lors des multiples livraisons des différentes version du logiciel.

Ce sont les tests que nous mettrons en oeuvre (sans les automatiser).

Nos composants seront nos procédures/fonctions)

Les tests unitaires

Test en boîte Blanche

Test qui se fait en connaissance du code qu'il faut tester.

Sensible aux défauts fins de programmation, mais aveugle aux fonctionnalités absentes ...

Les erreurs de conception risquent de ne pas être détectées ...

Test en boîte Noire

Test qui se fait sur les spécifications du problème (énoncé), sans la connaissance du code.

Permet d'assurer la conformance spécification - code, mais aveugle aux défauts fins de programmation.

Méthodes de test BN :

- Test des domaines d'entrées
 - partition des domaines
 - test combinatoire
 - + test aux limites
- Couverture de la spécification

Pour nous :

L'idée principale est de tester chaque procédure ou fonction en **automatisant** les tests.

- Tester les paramètres d'entrée
- tester les sorties (résultats de calculs par exemple)
- notion de cas limite

Exemple 1

Tester une fonction qui, prenant en entrée une liste d'entiers, doit ressortir cette liste décalée vers la gauche (le premier elt devient le dernier).

Prendre un exemple concret :

$L = [1, 2, 3, 4, 5]$ sortie : $L = [2, 3, 4, 5, 1]$ à tester

cas L a deux éléments

cas L a un élément

Cas L est vide

Exemple 1

```
1 def testDecale():
2     # cas liste vide
3     L = []
4     L1 = decale(L)
5     if L1 == L:
6         print("test liste vide : OK")
7     else:
8         print("test liste vide : pas OK")
9     # cas liste à un élément
10    L = [5]
11    L1 = decale(L)
12    if (L == L1):
13        print("test liste 1 elt : OK")
14    else:
15        print("test liste 1 elt : pas OK")
16    # cas liste deux éléments
17    L = [5,4]
18    L1 = decale(L)
19    if (L1 == [4,5]):
20        print("test liste 2 elts : OK")
21    else:
22        print("test liste 2 elt : pas OK")
23    # cas liste plus de deux éléments
24    L = [5,4,6]
25    L1 = decale(L)
26    if (L1 == [4,6,5]):
27        print("test liste nb elt > 2 : OK")
28    else:
29        print("test liste nb elt > 2 : pas OK")
```

Exemple 1

```
31 def decale(L):
32     retour = L
33     if (len(L) >= 2):
34         aux = L[0]
35         for i in range(len(L)-1):
36             L[i] = L[i+1]
37             L[len(L)-1] = aux
38     return retour
39
40 # programme principal
41 testDecale()
```

```
test liste vide : OK
test liste 1 elt : OK
test liste 2 elts : OK
test liste nb elt > 2 : OK
```

- La fonction test ne prend pas de paramètre.
- la fonction test peut s'écrire avant la fonction testée
- les impressions ne détaillent pas le jeu de données

Exemple 2

Écrivez un programme de test d'une fonction réalisant la fusion de deux listes d'entiers L1 et L2, en alternant un élément de L1 avec un élément de L2. On commence avec un élément de L1. Les deux listes sont de même taille.

Quels cas à tester ?

Écrire le programme de test.

Exemple 3

Un programme prend en entrée trois entiers rangés dans une liste. Ces entiers sont interprétés comme les longueurs des côtés d'un triangle.

Le programme imprimera selon les cas :

triangle isocèle
triangle équilatéral
triangle quelconque
pas triangle

On considère qu'un triangle plat n'est pas un triangle.

Quels cas à tester ?

Écrire un programme de test.