# Fundamental Test Process & Test Levels and Types

# Homework

1. Fill in the right term.

2. C A S E S

3. P L A N

4. O R A C L E

1. H A R N E S S

5. S T R A T E G Y

7. I M P L E M E N T A T I O N S

6. S U I T

1. Test [**harness**](https://en.wikipedia.org/wiki/Test_harness) is a collection of software and test data.
2. There are logical and concrete test **cases**.
3. The results from the planning activities should be documented in a test **plan**.
4. To predict the test results you use test **oracle.**
5. The main task of planning is to define test **strategy.**
6. You group the test cases in a test **suit.**
7. The quantity of tests and the test coverage determine the test **implementations**.
8. Write down the test levels in correspondence with their level. Start with the basic test level at the bottom of the pyramid.

|  |  |
| --- | --- |
| Letter | Test Level |
| A | Acceptance testing |
| B | System testing |
| C | Integration testing |
| D | Component testing |

1. Below we have some tasks for the testers in a department of a company. According to the activity described in the left column determine the type of the test they are going to perform.

|  |  |
| --- | --- |
| * Test Activity | * Test Type |
| * Open the main menu, go to Invoices and create new Invoice. Check if the invoice is visible in menu Invoice Journal. | **Functional test** - Verifies input-output behavior, testing is black-box. |
| * Open the application and try logging in with invalid credentials. This test is going to be performed after each daily build. | * **Risk-based test** – The application cannot function without logging and the impact will be crititcal.**/ Regression test** – To repeat the test to verify everything is till working. |
| * Create a test for logging in the application with legal credentials. Create a test to find how many users can log in simultaneously in the system. | * **Non-functional test** – connected with **load testing**. |
| * Make the same test that revealed a bug before the last build. The developer set the issue to “fixed”. | * **Re-testing** – to confirm that the bug is fixed. |
| * Create a unit test for the method AddItem of the class Order. | **Structural** **test** – uses information about the internal code structure or architecture – white-box testing, connected with **component/unit testing** |
| * Open the test used before the last release for exercising the online help option and change the link to the current location of the help info. | **Maintanance test** - Product is adapted to new operational conditions; Testing is needed even if only the environment is changed. In this case the link for help info is changed. |

1. Write test cases for a car. You need to cover the different levels and types. Try to achieve a good test coverage.

* Verify that all the components needed are present – chassis, engine, transmission, drive shaft, differential, and suspension and other important [parts](https://en.wikipedia.org/wiki/List_of_auto_parts).
* Check if the alternator is working properly.
* Check if the [automotive battery](https://en.wikipedia.org/wiki/Automotive_battery) is charged.
* Ensure that Gauges and meters are showing the real expected results.
* Check lighting and signaling system.
* Check sensors.
* Check if the starter is working smooth.
* Check wiring harnesses.
* Check all miscellaneous
  + Air bag control module
  + Alarm and siren
  + Central locking system
  + Chassis control computer
  + Cruise control computer
  + Door contact
  + Engine computer and management system
  + Engine control unit
  + Fuse box
  + Ground strap
  + Grab Handle
  + Performance chip
  + Performance monitor
  + Relay connector
  + Remote lock
  + Shift improver
  + Speed controller
  + Speedometer calibrator
  + Transmission computer
  + Wiring connector
* Check braking system components and perform a test drive.