Testing and Continous Integration

" We build software for people who build software "
<Codiak>

Powered by

Why write tests ?!

- **Tests tell you when your code is broken.**
- **Tests help later developers to undestand your code.**
- **Tests improve the design of your code.**

Types of tests ..

- **Unittests**
  - **Test « one » unit of code : fct/method**
  - **Must be isolated, small, fast and verbose**
  - **Require more refactoring when code changes**
- **Intergration tests**
  - **Test that application components talk to each other correctly.**
  - **Slower and exercice more code**
  - **Also called system tests | functional tests | acceptance tests.**

What is Unit testing ?

« Any repeatable activity that checks that the individual units of code within a module/application work as expected »

Some best practises

- **Bug triage unit testing :**

  - Don't fix the bug.

  - Write a test for the correct behaviour.

  - Change your code until it passes.

- **New feature unit testing :**

  - Write test(s) for the new feature.

  - Change your code until it passes.

- **Avoid premature optimisation :**

  - Write tests only for code that you are changing.

  - Don't need to test every possible scenario.

Some Python testing libs

- **Pytest, Nose :** Unittesting library (less boilerplate)
- **Mock :** easily create fakes for testing.
- **WebTest :** request/response testing for WSGI web apps.
- **Selenium :** browser automation (web).

Continous Integration .. What is it ?

(concepts)

- **The system must be able to be build and tested automatically.**
- **Everyone commits their changes frequently (every day or two).**
- **Upon commit the system is immediately and automatically « integrated ».**

Why using CI ?

- **Humans are not good in boring tests :**

  → Running tests manually is boring :(

  → Tests break and you don't know which change broke what.

- **How to ensure that the work done separately can be « integrated » successfully ?**

- **Levels of the integration problem :**

  → Merge conflics

  → Compile conflicts

  → Test conflicts

Golden rules

Continous Integration ---> Continous deployement

A CI system improves the productivity of a developement team by automating various things.

Use Jenkins

jenkins-ci.org

It is time to code a bit ..