Coding Test:

Cuando estamos escribiendo scripts muy muy largos, o un programa entero, es muy fácil perderse y recibir errores, y aun mas fácil, no saber que es lo que esta causando el error. Para solucionar este problema, usaremos los tests, vamos a testear el script ates de correrlo.

Hay varias vías para hacer esto, tenemos la vía más básica, que es introducir nosotros mismos los valores que esperamos por parte del usuario, y lo que debe hacer el programa en caso de recibir otro. Pero esto no es realmente eficiente cuando tenemos mucho código para revisar y no sabemos donde esta el bug. La otra vía es usar los tests automáticos.

Unit Tests:

Unit tests es un termino que se refiere a que, en lugar de comprobar todo el código, o todo el script, lo que haremos será aislar cada una de las funciones para ver si estas están funcionando correctamente individualmente, una por una, así podremos saber con facilidad cual de estas es la que esta devolviendo el error.

White Box Test:

Este término se refiere a que conocemos el código que estamos testeando, posiblemente porque fuimos nosotros quienes lo creamos.

White Box Test:   
Este término se refiere a que conocemos el código que estamos testeando, posiblemente porque fuimos nosotros quienes lo creamos.

Integrated Test:

Este término se refiere a ejecutar los tests para diferentes tipos de instancias, o sea no un Unit test, que comprueba que un archivo o script funcione, sino que la interacción entre archivos o scripts funcione, un ejemplo de esto es el hecho de que la interacción entre el disco duro donde esta guardada la información que usamos en el curso, la computadora que lee la información, el internet, la Virtual Machine, y luego nuestra PC, funciones bien, o más bien, que le interacción entre todas las partes, o entre todos los scripts necesarios para que estos programas corran de manera correcta, no presente errores, o se comporte de manera diferente a la esperada. Dependiendo de los tipos de sistemas operativos involucrados en la prueba, es posible que tengamos que crear un test enviroment específicamente para esta prueba.

Smoke Test:

Este termino se refiere a hacer un test general, o sea, el script ni siquiera de inicia. Normalmente cuanto esta prueba no se pasa, el programa o script no pasara ninguna otra.

Load Test:

Este test se hace pasándole una gran cantidad de datos al programa y viendo como responde este cuando está bajo presión y se le pasan unos 10 000 argumentos por segundo o una barbaridad de esas.

Test Driven Development:

Este término significa no esperar a terminar de escribir el script para hacer la comprobación, sino irlo haciendo en conjunto, al tiempo que vamos escribiendo trozos del script, vamos realizando las pruebas pertinentes, en lugar de hacerlo al final de este.

Creando Test files:

Básicamente lo que hacemos acá es crear un nuevo archivo, convencionalmente, a este archivo se le llama por el mismo nombre que al archivo original añadiéndole \_test al final. Luego lo que hacemos es que en el archivo \_test, importamos la función que queremos testear. Luego comprobamos, usando una clase creada por nosotros, pero que tenga los mismos atributos que la clase TestCase del modulo unittest de Python, comprobamos que el output de nuestra función que importamos desde el archivo sea el esperado, de ser así, devolverá que la prueba fue correcta.

* Se importa desde el archivo que queremos importar la función, la función que queremos probar.
* Se importa el módulo unittest para poder usar sus métodos y funciones
* Se crea una clase especial para este test, que herede las características de la clase TestCase del modulo unittest, para que los métodos y funciones del unittest puedan ser efectivos.
* Después se define la función que será la que comprobara si los outputs del código y el esperado son idénticos, para esto usamos los métodos que heredamos desde el unittest módulo.
* Por último, pasamos el unittest.main() que es el metodo main() del módulo unittest, lo que significa que este script correrá como un test, o sea, el resultado final no será un print o un return, sino que será un output definido especialmente por el módulo unittest.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente