*Intro to testing*

Il testing è la valutazione dei prodotti elaborati durante la produzione del software. È il prrocesso tramite cui si scoprono i difetti del sistema.

Fa in modo che il software faccia tutto ciò che ci si aspetta (oppure ad assicurarsi che faccia tuttociò che deve e niente di più.)

**Failure**: comportamento inaspettato del sistema

**Bug** == Difetto: qualcosa che “va storto” nel codice, e che causa una failure.

Quality assurance: mira a prevenire i difetti e a rimuovere quelli già esistenti, mentre il testing mira solo a scoprirli.

*Test early, often and the right art the right way.*

Test case: componente base del test, coppia = (input, expected result).

Input: info sullo stato del sistema o valori da processare

Expected Result: spesso include cambiamenti non percettibili per l’utente

Test suite: organizzano i test case, ad esempio li raggruppano per tipo

*COVERAGE*

Misura dell’adeguatezza del testing che osserva la quantità di codice(percentage of lines) coperto da casi di test oppure in termini di quanti requisiti sono valutati nelle specifiche e nei test.

Black box testing: basandosi sulle specifiche

White box testing: basandosi sull’implementazione

JUnit: framework opensource di testing per java

Code coverage

* Optimistic: coverage == 100% && failures == errors == 0 => è possible che ci siano bug
* Pessimistic: coverage == 100% && failures != 0 => è posibile che non ci siano bug (errore nel test)
* Impossbile: coverage < 100%

Anastasia:

I: giorni della settimana

O: media fatta male, ma controlli caso generico fortunato e edge case /0

Logaritmo, Condizione booleana che non viene verificata

P:

Giulio:

I: ora

O:

P: