

A tesztelés szintjei:

- Egységteszt
- Rendszereszt
- Integrációs teszt
- Átvételi teszt
- Komponens teszt

Egységteszt: A program egységeit külön (metódusok, osztályok) teszteljük, azok működését vizsgáljuk

Rendszereszt: A rendszert egyben, teljes egészében teszteljük

Integrációs teszt: Az egyes komponensek közti kapcsolatot, azok közös működését teszteljük.

Átvételi teszt: A program egészét a végfelhasználókat teszteljük.

Komponens teszt: A program komponenseit külön, modulokként teszteljük

A tesztelési tevékenységek:

A tesztelés nem csupán a tesztelés létrehozásából és futtatásából áll, különböző tevékenységek is eszükünkbe kell. Ezeket a lépéseket tesszük:

- teszterterv készítése
- tesztesetek létrehozása
- feltételezés a végfelhasználóknak
- tesztelés végrehajtása
- lépési feltételek vizsgálata
- eredmények értékelése
- jegyzőkönyv készítése

2, Tesztelési lehetőségek JUnit segítségével. Tesztelés részei, annotációs fajták, parametrizált tesztek létrehozása.

JUnit : - A JUnit segítségével az egyes modulokat tudjuk tesztelni (pl. metódusok, stb.)

- Tesztosztályokat használ.

- JUnit tesztelés Eclipse alatt Maven projektet használva létre.

- A tesztosztályokban teszt metódusokat használunk az egyes metódusok tesztelésére

- A teszt végrehajtása után lehetőséges eredmények

- Error

- Fail

- Pass

Annotation fajtái :

- Ignore

- Test public void method

- Ignore : Olyanban használjuk ha a tesztelendő metódus már megváltozott de a tesztmetódus még nem

- Test public void method : Azt jelöli hogy a teszteléshez egy tesztmetódust használunk

Parametrizált tesztek : Ugyanazt a tesztet felhajtja le többször de más más paraméterekkel

3. Nevezze meg azokat a modelleket ahol a tesztelésnek helye van a szoftver életciklusában

Létezik olyan szoftverfejlesztési modell, ahol a tesztelés már a szoftver fejlesztésének életciklusában elsődleges helyen van:

- V-modell
- Agilis szoftverfejlesztés

V-modell: - Nevét a formájáról kapta, a V-betűhöz hasonlóan.
 - A vízszintes modell, szitenkénti tesztelési szakaszokkal.
 - Egyik szára a fejlesztési szint másik szára a tesztelési szint.
 - A vízszintes modellhez hasonlóan megvalósítható mivel egyenesen, folyamatosan kell haladni a fejlesztési ciklusban végig, nem úgy mint modellek egy spirál modellnél.

KÖVETELMÉNY
SPECIFIKÁCIÓ

FUNKCIONÁLIS
SPECIFIKÁCIÓ

RENDSZERTERV

IMPLEMENTÁCIÓ

ATVETELI
TESZT

RENDSZER TESZT

INTEGRÁCIÓS
TESZT

MODULOKS
TESZT

Agilis fejlesztés:

Es egy "ismétlődő" szoftver-fejlesztési módszer mely során a tervek és a fejlesztés folyamatosan ~~reaktív~~ reagálnak a változásokra. Előnye hogy gyorsabb dokumentációval jár a régió modellekhez képest. Nem meglepő, mint a V-model, a követelmények akár a fejlesztés menete során is változtathatók. Ma a legelterjedtebb fejlesztési stratégiák. Folyamatos a kapcsolattartás mivel a megrendelő és fejlesztő cég között illetve szintén folyamatos a kapcsolat az egyes fejlesztői csoportok között.