2024.01.08

Burka Emil
BFATJO

1.

Tesztelési tevékenység:

Nemcsak a tesztek készítéséből és futtatásából áll, hanem tartalmazza a tesztek végrehajtását, a belépési feltételek vizsgálatát ~~az~~ az eredmény értékelését, és a jelentéskészítést.

Tesztelési szintek:

~~Fejlesztési~~ Fejlesztői teszt:

- Komponens teszt: - A rendszer egy komponensét tesztelő önmagában.
  - Általában fehérdobozos teszt
  - pl.: unit teszt
- Integrációs teszt: - Interfészek tesztelése
  - komponens-komponens közötti kölcsönhatások tesztelése
  - rendszer-rendszer közötti tesztelés
- Rendszer teszt: - az egész rendszert együtt teszteli
  - Megfelel-e a követelményeknek, specifikációknak, ~~rendszer~~ tervnek

Átvételi teszt: felhasználói szempont alapján az egész rendszer tervezése

- alfa teszt: - kész termék tesztje a fejlesztő céggel
- béta teszt: - végfelhasználók egy szűk csoportja végzi
- felhasználói átvételi teszt: - majdnem összes felhasználó meghatározza tesztelésre
  - nem termelésben használják
- üzemeltetői átvételi teszt.

2024.04.08

Barna
Eric
BJFATSO

2.,
A JUnit rendszer használatával a programok minősége javítható
A JUnit 4.x a legelterjedtebb verziója. Alap eszközei a tesztosztály és a
tesztmetódus. A tesztelő kódot metódusokban lehet megadni.

Tesztelés részei:
~~A tesztelés~~ Egy teszt futtatása során három eredmény lehet:
sikeres végrehajtás, sikertelen végrehajtás, hiba.
Assert osztály "alapvető" építőelemek, amelyekkel logikai állításokat fogalmazhatóak meg pl: assertTrue(boolean), assertFalse(boolean), assertNull(Object):

Annotációs jegyei:
- @Test public void method: egy metódust jelöl meg
- @Test(~~ex~~ expected = Exception.class): elvárt kivétel megadása
- @Test (timeout): elvárt idő" után hibát dob
- @Before: teszt metódus előtt fut le
- @After: teszt metódus után fut le
- @Ignore: hiányozza az adott tesztet.

Paraméterezált teszt:
- Célja ugyanazon tesztesetek többszöri lefuttatása de más értékekkel
- @RunWith(Parameterized.class)
- @Parameters annotációval ellátott publikus statikus metódus, eredménye egy egy dimenziós tömb.

2021.01.08.

Burka Emil
BFATSO

3y

Tesztelési modellek:

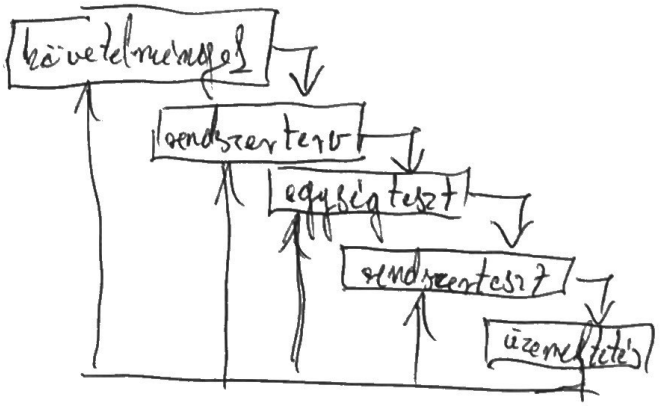
- Vízesés modell
- V-modell
- Prototípus modell
- Inkrementális modell
- Gyors alkalmazásfejlesztés
- Agilis fejlesztés
- Extrém fejlesztés

Vízesés modell:

A szoftverfejlesztés első" publikált modellje, más tervezői modellek ből származnak. Egyszerű, ~~fázisok~~ a fázisok egy dokumentumot képeznek. Egy fázis csak akkor indulhat, ha az előző befejeződött, ~~stb~~ Rugalmatlan korai szakaszokban komoly döntéseket kell hozni.

5 fázisra bontható:

- 1. fázis a követelmények ~~elemzés~~ elemzése és meghozása
- 2. fázis a rendszer- és szoftver terv (architektúra kialakítása)
- 3. fázis implementáció és egységteszt.
- 4. fázis integráció és rendszerteszt
- 5. fázis működtetés és karbantartás



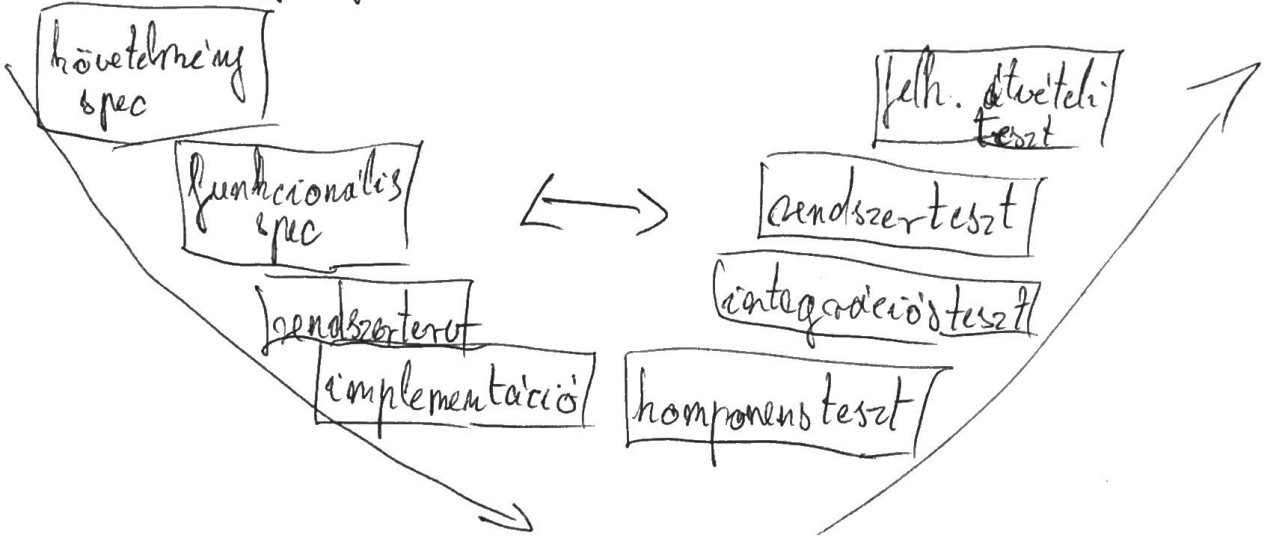
2021.01.08

Bunka Ézi
B7ATJ0

3./

V-modell:

~~Két~~ Két szára van, hasonlít egy V betűre). Fejlesztési része a vízesés modellből öröklődött. A vízesés modell kiegészítése teszteléssel (fejlesztés majd tesztelés). Az egy szinten lévő fejlesztési és tesztelési lépések összetartoznak.
Elterjedt, de nagyon merev. Ha nem változnak a követelmények a fejlesztés alatt akkor jól ~~a~~ alkalmazható. Ha változik akkor érdemes vagy iteratív vagy agilis modellre váltani.



2021.01.09

Burka Erik
BFATSO

4,

Integrációs stratégiák:

- „Big - Bang"
- Inkrementációs
- Top - down
- Bottom - up

Big - Bang: minden egység rendelkezésre áll és ezekből 1 egy teljes rendszert építünk fel.

Inkrementáció: a rendszer elemeit fokozatosan integrálja minden integrációs szinten tesztelet kg. kay.t végez

Top - down: legfelső elem tesztelésével kezdi, szimulálói, ideiglenes elemekét szükségesek.