

APLIKAZIOEN PROBA PLANAK

1. PROBA PLANA OSATU

Proba planean hainbat puntu hartu beharko dira kontuan, hauek, besteak beste:

- Egin behar diren proben antolaketa hierarkikoa.
- Probak egingo dituzten arduradunak.
- Proben helburuak.
- Probak egiteko beharko diren baliabideak: giza baliabideak eta baliabide teknikoak.

Jarduera honetan probak garatzeko ingurunea zehaztu behar da eta probetan erabiliko diren datu eta transakzio bolumenak estimatu behar dira. Bolumen handiko kasuetan, probak, sistemaren ingurune errealean egin behar dira.

Sistema bakoitzari proba mota ezberdinak egiten zaizkio:

- Unitarioak: osagai bakoitzari egiten zaizkionak.
- Integraziozkoak: osagaien arteko integrazioa probatzeko egiten direnak.
- Sistemarenak: sistema osoa probatu nahi denean egiten direnak.
- Onartzekoak: erabiltzaileak egiten dituenak, sistemak bere betebeharrak betetzen dituela ikusteko.
- Erregresiokoak: sistemaren atalen bat aldatzean berriro egin behar diren probak.

Probak egiteko oinarritzko printzipio batzuk kontuan hartzea komeni da:

- Probak programatzailea izan ez den beste batek egitea komeni da. Pertsona berri batek sistemaren bestelako ikuspegi bat izango du eta proba horien emaitzak hobeagoak izango dira, akats gehiago topatzeko aukera egongo delako.
- Probak, espero diren zein espero ez diren baldintzetan egin behar dira.
- Esperientziaren arabera, programa zati batean zenbat eta errore gehiago aurkitu, ondorengoak aurkitzeko aukerak ere handitu egiten dira, eta alderantzizkoa.

2. PROBAK DISEINATZEKO TEKNIKAK

Proben diseinu aldetik, bi mota nagusi bereizten dira:

- Kutxa zuriko probak: programaren barruko egitura aztertzeo aukera ematen dutenak.
- Kutxa beltzeko probak: proba kasuak, sarrera eta irteerak soilik kontsideratuz diseinatzen dira, barruko egituraz arduratu gabe.

2.1. Kutxa zuriko probak

Sistemaren osagaien barruko egitura aztertzea dute helburu nagusia. Bide logiko multzo bat aukeratzeko da eta proba datuak horren azterketa egiteko sortzen dira. Horretarako, bide horien egikaritzapena duten balioak zehaztu behar dira.

Proba hauek egiteko era askotako irizpideak erabil daitezke:

❖ Sententzien probak

Modulu baten barruko sententziak, gutxienez behin probatzeko, proba kasuak sortu behar dira.

❖ Baldintzen probak

Baldintza bakoitzaren balio guztiak kontsideratzen dituzten proba kasuak sortu behar dira. Orokorrean, baldintza guztiak betetzen dituen proba kasu multzo batek, sententzien probak ere betetzen ditu.

❖ Zenbaki Ziklomatikoa

Beste irizpide bat, McCabe-k proposatutako zenbaki ziklomatikoa da, kontrolerako fluxu diagrametan oinarritzen dena. Formula honetan du bere oinarria:

$$ZZ = e - n + 2 \quad e = \text{konexio kopurua} \quad n = \text{nodo kopurua}$$

Honen ondorioz, ZZ handiago den heinean, programa konplexuago izango da. McCabe-ren proposamenak, ZZ konplexutasuna duen programa batentzat, ZZ bide desberdin behintzat jorratzea da.

Kutxa zuriko probek eskaintzen duten abantaila nagusia aukeratutako irizpideak asetu direla egiaztatzea da. Halere, hainbat eragozpen aurkitu daitezke: proba bide kopuru altua lortzea, programen zuzentasun osoa ezin frogatzea edo proba kasuen diseinuak programaren egiturarekiko menpekoak izatea.

2.2. Kutxa beltzeko probak

Kutxa beltzeko probetan, erazagupen funtzionala abiapuntu gisa izanik, proba kasuak eta proba datuak diseinatzeko dira.

Proba mota hauekin, zer probatu nahi da?

- Sistema betetzen dituen funtzioak.
- Sistemaren helburuak betetzen direla.
- Sistemaren erantzuna kanpoko eraginei.
- Sistemak kudeatutako transakzioak.

Kutxa beltzeko probak gauzatzeko ere irizpide batzuk definitu daitezke:

❖ Baliokidetasun partizioak

Proba kasuen diseinuan, programaren sarrera posible guztiak jasotzen dituen multzoak osatzea da arazo nagusia. Proba kasuak bi pausotan gauzatzen dira:

1. Baliokidetasun klaseak identifikatzea: Sarrera baldintza bakoitza bi edo multzo gehiagotan banatzen da. Taula lagungarri bat erabili ohi da, sarrera baldintza, baliokidetasun klase balidoak eta baliogabeak baliokidetasun klaseak errepresentatuz.
2. Proba kasuak definitu baliokidetasun klase bakoitzarentzat: Klase bakoitzari zenbaki bat asignatzen zaio. Klase guztiak proba kasuren batek probatzen dituen arte proba kasuak idazten dira. Kasu bakoitzean ahalik eta klase gehien probatuko dira. Klase balidoak eta baliogabeak probatu behar dira.

❖ Balio limiteen azterketa

Proba kasuen diseinuan beste irizpide hau erabiltzen denean, mugetan dauden balioak aztertzen dira. Aldagai baten balioa hein batean dagoenean, erroreak hein horren mugetan egon ohi dira. Adibidez, balioak 20 eta 10000 artean direnean, proba kasuetarako hartu beharreko balioak, 19, 20, 21, tarteko edozein balio, 9999, 10000 eta 10001 izatea komeni da.

Gainera, baliabideen arabera, beste proba gehigarri hauek egitea komeniko litzateke, balio negatiboak, 19 baino txikiago edozein balio eta 10001 baino handiagoa den beste balioren bat erabiliz.

❖ Errore balio tipikoak

Proba datuen artean errore balio tipikoak zehazten dira. Hots, errorea ematea espero diren balioak. Helburua beraz, arazoak emango dituzten balioak zeintzuk diren zehaztea da.

❖ Ezinezko balioak

Sistemaren barnean ezinezko moduan definitzen diren datuak dira. Beharbada sistema berak sor ditzake. Balio hauen azterketa egingo da, sistemaren funtzionamendu desegokia edo gelditzea ekar dezaketelako.

Alde batetik, balio hauek detektatzeko proba kasuak sortuko dira. Bestetik, konponezinak diren erroreak sortu gabe sistemak funtzionamenduan jarraitzeko aukera duen ala ez aztertuko da.

3. PROBA MOTAK

3.1. Proba unitarioak

Programa bati, behar den moduan funtzionatzen duela egiaztatzeko egiten zaizkio probak. Hauen helburua erroreak topatzea da, ondoren zuzenduak izan daitezen.

Proba unitarioak sistemaren osagai bakoitzaren gainean egiten dira, besteak beste, kutxa zuriko eta kutxa beltzeko teknikak erabiliz.

3.2. Integrazio probak

Osagai bakoitzaren probak egin ondoren, beraien arteko integrazio probak egiten dira, beraien arteko interfazeak ondo funtzionatzen dutela bermatzeko. Gainera, definitutako prozedura guztiak teknikoki bateragarriak direla ere bermatzen da.

Sistemaren osagaien arteko integrazioa probatzea da atal honen helburua, sistema egikaritu daitekeela probatzea da. Integrazio probetan garapen taldeak (programatzaileak) parte hartzen dute, ez erabiltzaileak. Bertan sistemaren funtzionamendu egokia aztertuko da, erabiltzaileen parte hartzea berriz, onartze probetan nabarmenduko da.

Errore teknikoek sortzen duten mesfidantzagatik batez ere, integrazio probetan erabiltzaileak parte ez hartzea komeni da. Erabiltzaileak sistema ikusi baino lehen, errore gehienak konponduak egotea komeni da.

Integrazio probak egiteko hainbat estrategia jarrai daitezke. Osagaiak sortu eta integratuko diren sekuentzi zehatza oso garrantzitsua da: horretarako, hainbat aukera planteatzen dira:

❖ Goitik beherakoa (top-down)

Hierarkian antolatuta. Lehenik gainean dagoen osagaia sortu eta probatuko da eta ondoren beherantz jaitsiz jarraituko da. Abantaila nagusia, moduluen arteko interfazeen probak askotan eta hasierako fase batean egiten direla da.

❖ Behetik gorakoa (bottom-up)

Beheko moduluak lehenago sortzen dira eta goikoak simulatuko dituzten modulu laguntzaileak erabiltzen dira. Honen abantaila, paraleloan lana egiteko aukera gehiago dagoela, baina planifikazio handiagoa beharko da.

❖ Bien artekoa:

Aurreko bi estrategia batera erabiltzen dira. Estrategia honek planifikazio handia eskatzen du, erdiko moduluetan “topagunea” definitzen da eta bertara goitik nola behetik modu egokian ailegatu behar da.

Integrazio proben garapena bi ekintzatan banatzen da:

- Integrazio probak prestatu eta egin.
- Integrazio proben ebaluazioa eta dokumentazioa egin.

3.3. Sistemaren probak

Jarduera honetan sistemaren proben erazagupena egiten da. Horretarako, hainbat alderdi hartu beharko dira kontuan:

- Komunikazioak: interfazeen komunikazio egokia egiaztatzeko.
- Erantzun-denborak neurtu.
- Datu-bolumen handiekin sistemaren erantzuna aztertu.
- Datuak berreskuratu, akatsak daudenean.
- Segurtasuna.

Sistemaren probak egiteko beharrezkoak diren prozedurak eta baliabideak ere prestatzen dira.

Azkenik, sistemaren probak egiten dira, definitutako erazagupenei jarraituz. Beraz, jarduera honen garapenean hiru ekintza nagusi bereizten dira.

- Probak prestatu: Lehenengo ekintza honen helburua, sistemaren diseinu teknikoan definitutako sistemaren proba-plana berrikusi eta osatzea da, beharrezkoak diren probak gehituz. Horrela, sistemaren proba plana lortuko da.
- Proben ingurunea prestatu: Ekintza honen helburu nagusia sistemaren probak egiteko behar diren prozedurak eta baliabideak lortzea da: datu-baseak edo proba-fitxategiak, segurtasun kopiak, prozedurak, etab.
- Probak egin: Sistemak espero den moduan funtzionatzeko duela bermatu behar da. Proben emaitzak ebaluatu eta behar diren aldaketak zehaztu behar dira. Emaitza horien dokumentazioa ere egin behar da, batez ere historiko bat sortzeko asmoarekin.

3.4. Onartze-probak

Jarduera honetan erabiltzaileek sistema berria onar dezaten behar diren probak egiten dira. Helburu nagusia, erabiltzaileek eskatutako betebeharrak sistemak betetzen dituela egiaztatzea da.

Erabiltzaileek onartze-probetan parte hartzen dute, sistemak benetan behar dutena egiten duela ikusteko. Proben diseinua egitean hori kontuan hartu beharko da. Erabiltzaileen parte-hartzea nahitaezkoa izango da.

Horretarako, hiru ekintza bereizten dira:

- Onartze-probak prestatu: Erazagupenak berrikusi eta zuzendu egiten dira, eta onartze-probaren kasuak garatzeko beharko diren datuak prestatu behar dira.
- Produkzio-ingurunea prestatu. Baliabide fisikoak eta giza baliabideak prest daudela egiaztatu behar da.
- Onartze-probak egin. Probak egiten dira erabiltzaileen partaidetzarekin, lortutako emaitzak ebaluatu, beharrezko zuzenketak egin eta onartze formula lortu behar da.