

Tema 3. Principiile și axiomele testării

Principii ale testării (Myers)

Continuând ideea temei precedente observăm că cea mai importantă problemă în testare este totuși cea psihologică. Reieșind din aceasta pot fi formulate niște principii ale testării.

Principiul 1: *Un caz de test trebuie să definească neapărat ieșirea sau rezultatul dorit.*

Formularea acestui principiu este bazată pe psihologia umană. Dacă rezultatul corect nu este predefinit, șansa că orice rezultat va fi interpretat ca corect este destul de mare. Avem aici fenomenul „Vedem ceea ce dorim să vedem”. Cu alte cuvinte noi subconștient așteptăm rezultatul corect. O cale de combatere a acestui fenomen este încurajarea examinării detaliate a tuturor ieșirilor. Conform acestui principiu un caz de test trebuie să fie compus din:

1. Descrierea instanței (datelor de intrare) unui program.
2. Descrierea exactă a rezultatului sau a ieșirii programului pentru această instanță.

Principiul 2: *Un programator ar trebui să evite să-și testeze propriul program.*

Orice scriitor cunoaște, sau cel puțin trebuie să cunoască, că este o idee nu prea bună să-și corecteze și să-și redacteze singur manuscrisele. Iarăși este vorba de un fenomen psihologic. Autorul nu dorește să descopere erori în propria lucrare. După ce programatorul a lucrat constructiv la design și scrierea programului este foarte dificil să ia o atitudine distructivă față de propria operă. Plus la aceasta programatorul subconștient nu dorește să găsească erori de teama criticii din partea clientului sau beneficiarului. În afară de această problemă psihologică mai este una importantă: Programul poate avea erori din cauza neînțelegerii de către programator a sarcinii puse sau a specificației produsului. Astfel programatorul va transmite aceleași neînțelegeri și testelor executate. Aceasta nu înseamnă că programatorul nu trebuie sub nici o formă să-și testeze propriul program, însă testarea de către o altă persoană va fi mai eficientă. Acest principiu nu poate fi aplicat și depanării. Acest proces este mai bine efectuat de către programatorul care a scris programul.

Principiul 3: *Companiile de programare nu ar trebui să-și testeze produsele proprii.*

Argumentele acestui principiu sunt similare cu cele de la principiul precedent. Aici pot avea loc probleme de planificare: produsul trebuie livrat la timp, orice întârziere costă bani. Este destul de dificil ca o companie să fie obiectivă față de calitatea propriului produs. Grupul de testeri ar trebui să fie altul decât cel de dezvoltare, iar întreținerea unui departament de testare este destul de costisitoare și nu orice companie poate să-și permită aceasta.

Principiul 4: *Fiecare rezultat al testului trebuie examinat foarte minuțios.*

Acesta este probabil cel mai evident principiu. Cât de minuțios nu ar fi făcută testarea este posibil să se strecoare diferite erori. Erorile depistate într-un stadiu tardiv de testare deseori se dovedesc a fi cele omise în stadii incipiente de testare.

Principiul 5: *Cazurile de test trebuie să fie scrise atât pentru condiții de intrare valide cât și pentru cele invalide și neașteptate.*

Este normală tendința de testare a produsului pentru date valide de intrare, neglijând datele incorecte. Este însă foarte important comportamentul produsului când sunt introduse date incorecte sau ilegale. Acest principiu nu poate fi neglijat atunci când sunt elaborate cazurile de test.

Principiul 6: *Trebuie testat că produsul face ce trebuie și nu face ce nu trebuie.*

Putem considera acest principiu ca un corolar al principiului precedent. Programele trebuie supuse și examinate pentru efecte neașteptate. Mai ales astfel de teste trebuie elaborate pentru produsele cu destinație financiară.

Principiul 7: *Trebuie de păstrat și refolosit cazurile de test.*

Acest principiu este important și este respectat de specialiștii în testare. Orice produs poate fi îmbunătățit și dacă dorim să vedem dacă schimbările nu au avut un efect negativ asupra altor componente ale sistemului atunci este bine de a testa cu testele vechi noile versiuni. Astfel se testează integritatea și funcționalitatea produsului. Refolosirea cazurilor de test este binevenită și atunci când au fost corectate unele erori deja găsite.

Principiul 8: Nu se planifică procesul de testare presupunând că nu vor fi găsite erori.

Este o greșeală a managerilor de proiect când se folosește o definiție incorectă a testării. De exemplu: „Testarea e procesul prin care se demonstrează că produsul este performant și funcționează corect” în locul “Testarea e procesul prin care se execută un program cu intenția de a găsi erori”

Principiul 9: Probabilitatea de a găsi erori într-un fragment de cod este proporțională cu numărul de erori deja găsite.

Dacă avem de exemplu un program din două module A și B și în modulul A au fost găsite cinci erori iar în B numai una, și dacă modulul A nu a fost supus intenționat unui test riguros, atunci acest principiu ne spune că probabilitatea că în modulul A vor fi găsite mai multe erori decât în modulul B este destul de mare. La prima vedere acest principiu nu are prea mult sens, însă este un fenomen prezent în majoritatea programelor.

Principiul 10: Testarea este un lucru extrem de creativ și intelectual.

Este probabil adevărat că creativitatea cerută la testarea produselor serioase depășește creativitatea cerută în proiectarea acestora. Este clar că este imposibil de testat un produs suficient pentru a demonstra absența erorilor în acest produs. Metodologia pe care o vom studia în cadrul cursului va fi de un mare folos la elaborarea unui număr rezonabil de cazuri de test, însă această metodologie cere un nivel înalt de creativitate a testatorului.

Axiomele testării.

Axiomele testării (Patton)

1. Este imposibilă testarea completă a unui program.

Această axiomă este adevărată din următoarele motive:

- Numărul intrărilor posibile este foarte mare.
- Numărul rezultatelor posibile este foarte mare.
- Numărul drumurilor într-un program este foarte mare.
- Specificațiile programului sunt subiective.

Exemplul calculatorului (1)

2. Testarea software este un exercițiu de apreciere a riscurilor.

Dacă ai luat decizia să nu testezi toate scenariile de test posibile, atunci ți-ai asumat un oarecare risc. De exemplu, dacă ai decis să nu testezi în calculator cazul $1024+1024=2048$, iar programatorul accidental a greșit această combinație, nimerind acest produs la consumator, care este un contabil și a găsit această eroare, ce se va întâmpla în așa caz? Această eroare va costa foarte mult. Sună destul de urât aceasta, dar pe de altă parte este imposibil de testat tot, dar dacă nu se testează totul este posibil să se strecoare diferite erori. Ce se poate de făcut în astfel de cazuri? Un element important este că testatorul trebuie să învețe cum se poate reduce domeniul imens a datelor de intrare până la niște seturi realizabile de teste și cum să ia decizii ce este important de testat și ce nu este.

3. Testarea nu poate arăta că produsul nu are erori.

Testarea poate arăta că erorile există într-un produs, dar nu poate arăta că ele nu sunt în el. Pot fi perfecționate testele, pot fi găsite și raportate erorile, dar nu este posibil de garantat că erori în produs nu se vor mai găsi. Unicul lucru care se poate de făcut este de a continua testarea și posibil se vor mai găsi erori.

4. Cu cât mai multe erori găsești, cu atât mai multe sunt.

Erorile au tendința de a se găsi în grupuri. Dacă ai găsit una, poți fi sigur că în vecinătatea ei vor mai fi și altele. Deseori, trece destul de mult timp până când testerul găsește o eroare. Însă dacă a fost găsită una el o va găsi și pe a doua și pe a treia etc. Există motive serioase pentru aceasta:

- Programatorul a avut o zi grea. Codul scris într-o zi poate fi perfect, iar unul scris în altă zi poate avea o sumedenie de erori. O eroare găsită ne poate spune că mai sunt și altele prin zonă.
- Programatorii foarte des fac aceleași erori. Un programator care a făcut aceiași greșeală de mai multe ori o va repeta iar și iar.

- Fiecare eroare este ca un aisberg. Deseori proiectul și arhitectura software au probleme serioase. Testerul ar trebui să le depisteze pe acestea în primul rând, dar de obicei ele sunt detectate după ce au provocat multe probleme serioase.

5. *Paradoxul pesticidelor: erorile devin rezistente la teste*

Paradoxul pesticidelor în testare descrie fenomenul: cu cât mai mult testezi cu atât mai imun față de teste devine softul. Analogic cu insectele și pesticidele: dacă sunt aplicate aceleași pesticide de fiecare dată insectele devin imune la ele și chiar mai puternice decât au fost. Pentru a găsi erori noi e nevoie de teste noi.

6. *Nu toate erorile găsite vor fi corectate.*

O realitate tristă atestării este, că după ce s-a lucrat destul de mult și greu pentru a depista erorile, nu toate din ele sunt fixate și corectate. Șefii echipelor de testare trebuie să facă compromisuri și să ia decizii bazate pe riscuri în ceea ce privește fixarea anumitor erori.

Acest lucru are motive serioase:

- Nu este suficient timp pentru aceasta.
- Într-adevăr nu este o eroare.
- Este prea riscant de fixat(fixarea ei poate provoca mai multe erori).
- Nu are nici o valoare.

Procesul de luare a deciziilor față de fixarea erorilor îi implică pe testori, managerii de proiect și pe programatori și fiecare are opinia lui față de eroarea respectivă însă nimeni nu știe cine are dreptate, așa că decizia luată este bazată pe anumite riscuri (exemplu Procesorul Intel Pentium)

7. *E greu de spus când o eroare e o eroare ...*

În afară de acele afirmații care alcătuiesc definiția erorii testării pot apela și la opinia altor persoane poate chiar a clientului, ce reprezintă pentru el o eroare și își poate formula propria definiție a erorii.

8. *Specificațiile produselor nu sunt niciodată definitive.*

Industria software se dezvoltă foarte rapid. În același timp produsele soft devin tot mai complexe, iar termenii de predare mai limitați din cauza concurenței. Aceste două aspecte sunt în conflict permanent și rezultatul conflictului este schimbarea constantă a specificațiilor. Un alt motiv este perfecționarea unui produs care a fost scos pe piață cu ceva timp în urmă specificațiile au fost rescrise, dar nu conțin și funcționalitățile vechi ale produsului.

9. *Testorii nu sunt cei mai populari membri ai echipei de proiect.*

Ne amintim de scopul unui testor. Câteva sfaturi utile pentru testori:

- Găsiți erorile cât mai devreme. Toți vor fi mulțumiți dacă o să găsiți erorile serioase cu două luni, dar nu cu o zi înainte de sfârșitul perioadei de testare.
- Temperați-vă entuziasmul. Cât de bucuros nu ai fi că ai găsit o eroare critică, fii diplomat când i-o spui programatorului.