

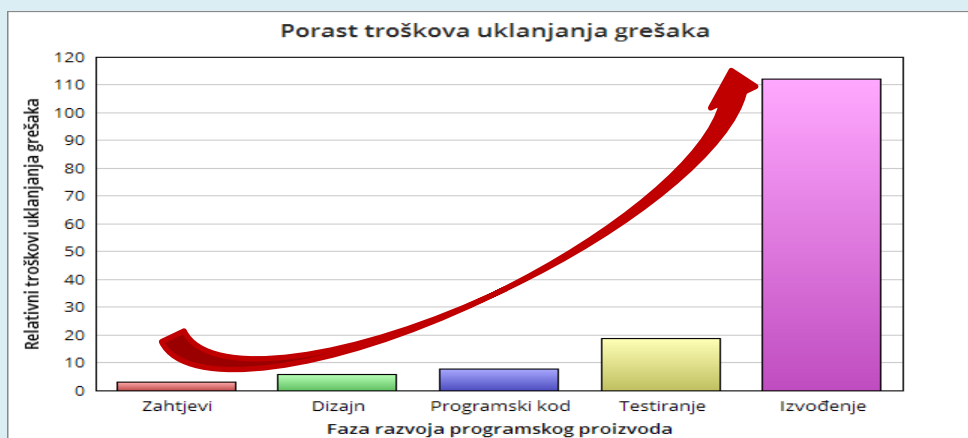
VERIFIKACIJA, VALIDACIJA I TESTIRANJE PROGRAMSKE OPREME

1. Zašto VVT?

Programska oprema (PO) srce je svakog informacijskog sustava. Ona postaje sve složenija, metode i alati njenog razvoja sve zahtjevniji. U nju se ugrađuje sve veći broj programskih komponenti, a konkurencija među softverskim kućama raste, dok je vrijeme u kojem je nužno isporučiti nove programske proizvode sve kraće. Međutim, **ona je najslabiji segment IT-a**. PO izrađuju ljudi, zbog čega su anomalije u njihovoj izradi neizbježne. Anomalije su skupni naziv za kvarove (errors, mistakes), pogreške (faults, bugs, defects) i ispade (failures) u njihovom korištenju. **Kvar je krivi rad u oblikovanju i programiranju, koji uzrokuje pogreške** kojima se oni unose u projektnu dokumentaciju, a u trenutku korištenja PO izazivaju ispade ili zatajenja. One nastaju u svim fazama proizvodnje PO, u svim programskim kućama bez obzira na njihovu veličinu i kvalitetu i bez obzira na korisnike za koju se proizvode.

Broj anomalija u PO kreće se od 30 – 80 pogrešaka / 1.000 linija isporučenog koda. Smatra se da je njihov industrijski prosjek 50 – 60, a raspodjela izvora njihovog nastanka je: 60 % tijekom specifikacija + oblikovanja i 40 % tijekom programiranja. Oko 66 % pogrešaka otkrije se tek nakon isporuke PO. S druge strane cijena otklanjanja pogreške u PO eksponencijalno raste s vremenom njenog otkrivanja (zakonitost 1 : 10 : 100).

Mogućnost unapređenja ovog stanje jest primjena konzistentnog procesa verifikacije, validacije i testiranja (VVT) PO.



Ovaj proces moguće je oblikovati na različite načine, koristiti različite pristupe i metode verifikaciji i validaciji, te primijeniti različite strategije i metode testiranja. Sve ove aktivnosti moraju biti dokumentirane, a na način kako to sve ostvariti, upućuju VVT norme (standardi). Alati za ovo područje, mogu ga značajno pospješiti. No, u današnjoj praksi **VVT je još uvijek najslabija karika životnog ciklusa razvoja PO**. Ovaj problem može se promatrati s dva glavna stajališta: **proizvođača PO i korisnika PO**.

Proizvođač PO ima dva krupna razloga za korištenje VVT-a: **1. što ranije otkriti i otkloniti pogrešku u razvoju PO i 2. razviti i prodati programski proizvod primjerene kvalitete, jer ona određuje njegovu reputaciju.** No, i kvaliteta ima svoju cijenu. Morala bi postojati ravnoteža između troškova preventive (primjene normi kvalitete i troškova procjene kvalitete) i troškova kurative (troškova otkrivanja i otklanjanja pogrešaka). **Najbolja praksa pokazuje da troškovi preventive PO danas ne bi smjeli biti manji od 10 - 15 % troškova njene proizvodnje, čime se pozitivno utječe na smanjenje troškova kurative barem za oko 80 %.** No, u praksi se na VVT gleda kao na dodatni trošak. Budući se u natjecajima za kupovinu nove PO uglavnom primjenjuje kriterij najniže cijene, VVT proces u životnom ciklusu razvoja PO najčešće se svodi na najmanju moguću mjeru, donekle se primjenjuje verifikacija, testiranje se provodi minimalno, a validacija se gotovo niti ne primjenjuje. Ali takvo stanje nije održivo na duže vrijeme, budući da raste svijest korisnika o potrebi postizanja više kvalitete PO. Na strani proizvođača PO ključne uloge za odlučivanje o problematici kvalitete PO imaju **direktor (CEO) programske kuće i voditelj razvoja PO**, dok su u **postupku njenog razvoja ključni: voditelj projekta, projektanti, programeri i tester.**



Audit kvalitete proizvodnje naše PO pokazao je da u razvoju nemamo predviđene aktivnosti preventive (ovi troškovi su < 1 %), a da su troškovi pronalaženja i otklanjanja pogrešaka izrazito visoki, što nam umanjuje i kvalitetu naše PO. Konkurenti su uspješniji od nas, pobjeđuju nas na natjecajima.... Što trebam pokrenuti da tvrtka ne dođe u neugodnu poziciju?

Korisnici PO bi trebali biti svjesni da njeni proizvođači i distributeri prodaju i isporučuju programske proizvode koji tijekom razvoja najčešće nisu bili dovoljno provjereni – verificirani, validirani i testirani. U takvim proizvodima često se nalazi još veliki broj razina pogrešaka, koji se otkriju tek tijekom njihove operativne primjene, i one se uklanjaju naknadno kroz tzv. održavanje. **Odnos između troškova razvoja i troškova održavanja PO izrazito je nerazmjern, jer su troškovi održavanja kumulativno 5, 10, pa i 20 puta veći od troškova razvoja.** Kako u sadašnjim uvjetima troškove razvoja snosi proizvođač programske opreme, a troškove njenog održavanja njihov korisnik, jasan je i interes ovog drugog da se sadašnje stanje poboljša. To je moguće postići **pooštravanjem načina ugovaranja** nabavke nove PO, **striktnijim testiranjem prihvatljivosti** kad dođe do njene isporuke i stavom da troškove otklanjanja pogrešaka u fazi primjene PO snosi proizvođač. Da bi se to postiglo, nužna je podići spoznajnu razinu korisnika o ovoj potrebi. Ključne uloge o ovom problemu na korisničkoj strani imaju: **glavni korisnik i CIO** (ukoliko postoji interna služba za informatiku).



Poželjna VS. moguća reakcija korisnika

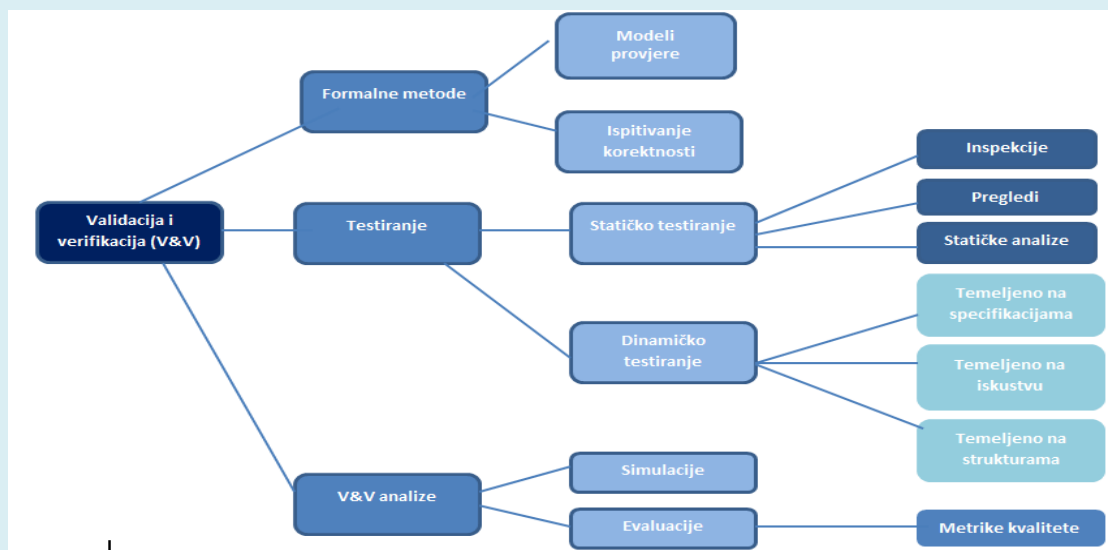
Navedeno stanje korisnik može značajno poboljšati. Njegova odgovornost za to, vrlo je velika, a konkretne mogućnosti koje mu stoje na raspolaganju su: **jasno specificiranje funkcionalnih i nefunkcionalnih zahtjeva za novom PO, definiranje strukture održavanja (preventivno i korektivno), određivanje načina validacije PO, određivanje načina testiranja prihvatljivosti, te način ugovaranja, u kojem navedene specifikacije trebaju biti sastavni dio.** Ako poslovni sustav raspolaže vlastitom službom za informatiku, tada je i reakcija njenog **direktora (CIO-a)** izrazito važna.



Troškovi naših aplikacija (plaće ljudi + licence) došli su do 75 % našeg IT budžeta i onemogućavaju nam naš daljnji razvoj. Nešto ozbiljno trebam poduzeti glede ovog problema. Što trebam učiniti?

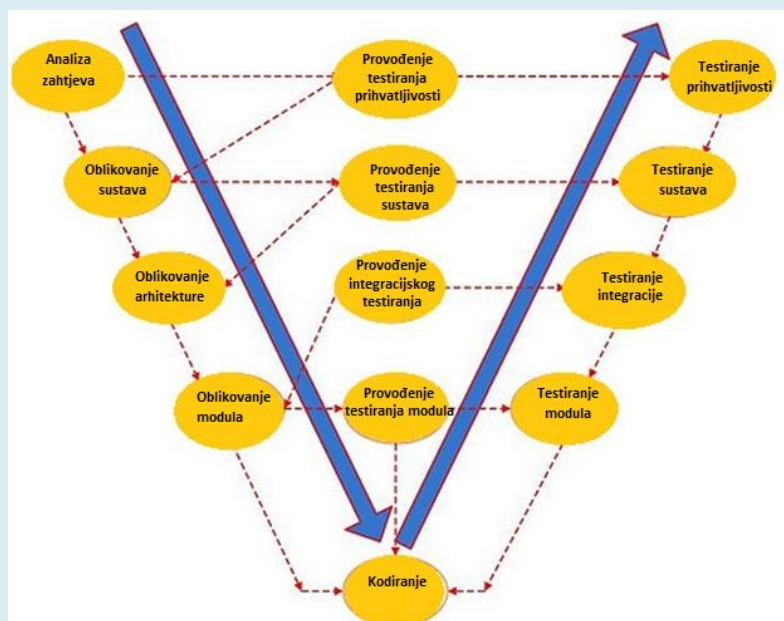
2. PRIMJENA VVT-a U PROIZVODNJI PROGRAMSKE OPREME

Postoji niz načina kako unaprijediti područje VVT-a primjenom suvremenih metoda i normi. Vodeće svjetske organizacije za normizaciju ISO, IEC i IEEE, usvojile su norme serije ISO 29119, koje na cjelovit i transparentan način „rješavaju“ ovu problematiku, prvenstveno sa stajališta testiranja PO, dok je za problematiku validacije i verifikacije nužno koristiti određene IEEE norme ili odgovarajuće norme ISO/IEC 25000. Osnovni pristup koji ZIH primjenjuje u VVT-u PO, temelji se na normi **ISO/IEC/IEEE 29119-1** koja propisuje područja VVT aktivnosti, na način prikazan na slici.



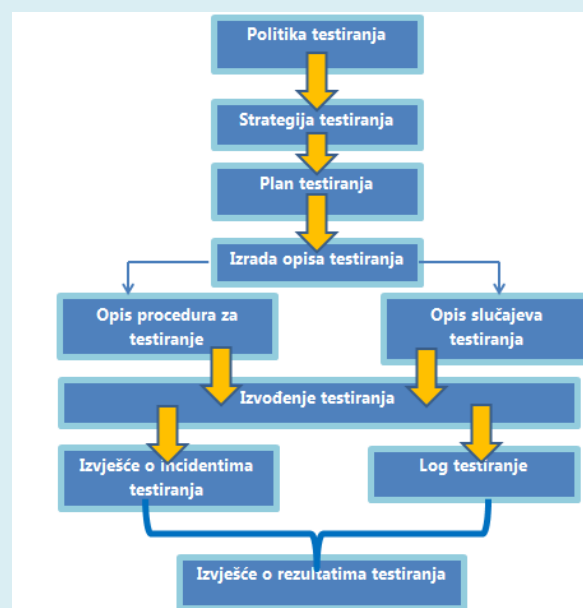
Verifikaciju PO ZIH može provesti za cjelokupni tijek procesa razvoja PO, za cijeli programski sustav ili njegove komponente, kako bi se odredilo da li svaka faza razvoja zadovoljava standarde struke, specifikacije, dobru praksu ili konvencije. To se čini: pregledom ugovorne i projektne dokumentacije, inspekcijama, kontrolnim listama, itd., kako bi se odredila korektnost, kompletnost, konzistentnost i točnost razvojnih postupaka i njihovih rezultata i dobio odgovor na pitanje – da li se programski proizvod radi ili se radio na dovoljno dobar način.

Validaciju PO ZIH provodi na kraju životnog ciklusa njenog razvoja, da bi se procijenilo da li ona zadovoljava vanjske i unutarnje propise, poslovna pravila i potrebe korisnika, te da li su specifikacije zahtjeva dobro postavljene. Evaluiraju se rezultati svake faze razvoja PO, počevši od specifikacije zahtjeva korisnika, planiranja kvalitete, specifikacije zahtjeva za PO, oblikovanja, kodiranja, testiranja, instalacije, pa do operativnog rada i održavanja. Ispituje se da li u specifikacijama zahtjeva ima pogrešaka, što su njihove posljedice, te da li postoji konzistentna sljedivost tijekom njenog razvoja, Rezultat validacije, osim evaluacije rezultata razvoja PO, može biti zahtjev promjene specifikacija zahtjeva.



VVT za vodopadni razvoj PO

Testiranje PO, kao dio verifikacije i validacije, snažno je inkorporirano u proces njenog razvoja, a obvezatne faze su prikazane na slici. Proizvođač PO trebao bi imati svoju politiku testiranja kao i strategiju testiranja. Strategija testiranja usmjerava na načine donošenja plana testiranja i oblikovanje okvira testiranja. Mora postojati procedura testiranja, opis način pripreme testnih slučajeva, opis načina izvršenja testiranja, izrade izvješća o incidentima, izrade testnih logova, te završno izvješćivanje o svim provedenim postupcima i rezultatima.



3. USLUGE ZIH-a U PODRUČJU TESTIRANJA PO

3.1. Treninzi / Seminari / Prezentacije

- Motivacijska prezentacija poslovdstvu
- Kako uspostaviti sustav testiranja programske opreme?
- Primjena programskih alata u testiranju programske opreme
- Što korisnik treba znati o testiranju prihvatljivosti programske opreme?
- Priprema testera za polaganje ISTQB ispita
- Specifične teme iz testiranja programske opreme



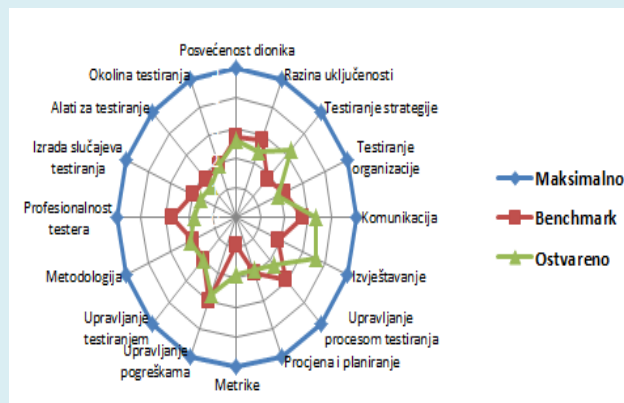
3.2. Konzalting usluge iz testiranja programske opreme

3.2.1. Analiza i ocjena postojećeg načina testiranja PO te izrada prijedloga mogućih / potrebnih poboljšanja

Prije no što se pristupi cjelovitijem rješenju razvoja cjelovitog sustava testiranja PO u nekom poslovnom sustavu, iskustvo najbolje prakse preporuča načiniti analizu postojećeg načina testiranja. ZIH to provodi na slijedeći način.

1. *Priprema za određivanje postojeće razine zrelosti testiranja PO.* Analiza troškova nekvalitete. Dogovor s menadžmentom o metodi procjene. Mogućnosti su: Ad hock metoda, metoda dobre prakse, metoda ciklusa poboljšanja, metoda upravljanja razvojem zrelosti i metoda primjene standarda. Izrada plana procjene, priprema tima za procjenu i prikupljanje potrebne dokumentacije.
2. *Praktična procjena u skladu s odabranom metodom procjene.* Slika u nastavku, prikaz je malog dijela mogućih rezultata – procjena 20 ključnih područja testiranja i ocjenjene razine zrelosti.
3. *Prepoznavanje slabosti, motiva, ciljeva i potencijala unapređenja postojeće razine testiranja u skladu s rezultatima procjene.* Izrada plana poboljšanja, potrebne organizacije i razvoja kompetencija za testiranje.
4. *Prezentacija rezultata menadžmentu.* Rasprava o rezultatima procjene i prijedlozima poboljšanja.

| Key Areas | Controlled | | | | | Efficient | | | | | Optimized | | | |
|--------------------------------|------------|---|---|---|---|-----------|---|---|---|---|-----------|----|----|----|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Test strategy | A | | | | | B | | | | | C | | | D |
| Life-cycle model | A | | | B | | | | | | | | | | |
| Moment of involvement | | A | | | | B | | | | C | | | | D |
| Estimating and planning | | A | A | | | | | | | | B | | | |
| Test specification techniques | A | | B | | | | | | | | | | | |
| Static test techniques | | | | A | | B | | | | | | | | |
| Metrics | | | | | A | | B | | B | | C | | D | |
| Test automation | | A | | A | | | B | | | C | | | | |
| Test environment | | | A | | | | B | | | | | | C | |
| Office environment | | | A | | | | | | | | | | | C |
| Commitment and motivation | A | | | | B | | | | | | | C | | |
| Testing functions and training | | | A | | | B | | | | | C | | | |
| Scope of methodology | | | | A | | | | | | B | | | C | |
| Communication | | A | | B | | | | | | | C | | | |
| Reporting | A | | | B | | C | | | | | | | D | |
| Defect management | A | | | | B | | C | | | | | | | D |
| Testware management | | A | | | B | | | | C | | | | | D |
| Test process management | A | | B | | | | | | | C | | | | |
| Evaluations | | | | | | A | | B | | | | | | |
| Low-level testing | | | | A | | B | | C | | | | | | |



3.2.2. Konzalting usluge ZIH-a u razvoju sustava testiranja PO

ZIH nudi usluge razvoja cjelovitog sustava testiranja PO, slijedeći glavne aktivnosti:

1. **Priprema tima korisnika**
2. **Oblikovanje procesa testiranja programske opreme** u ovisnosti od primijenjenog modela životnog ciklusa razvoja PO:

- VVT u vodopadnom modelu
- VVT u V modelu
- VVT u inkrementalnom modelu
- VVT u RAD modelu
- VVT u agilnom modelu
- VVT u iterativnom modelu
- VVT u spiralnom modelu

3. Izrada vršne dokumentacije za proces testiranja

- Politike testiranja i
- Strategije testiranja
- Izrada internog uputstva (priručnika) za testiranje. Izrada uputstava i kriterija za odabir razina i metoda testiranja.
- Razrada razine testiranja su: jedinično testiranje, testiranje integracije, testiranje sustava i testiranje prihvatljivosti.
- Što se tiče metoda testiranja, danas ih postoji oko 100, tako da je izuzetno značajno odabrati one koje su primjerene za potrebe konkretnog korisnika.

4. Izrada internog uputstva (priručnika) za testiranje:

Izrada uputstava i kriterija za odabir razina i metoda testiranja. Razrada razine testiranja su: jedinično testiranje, testiranje integracije, testiranje sustava i testiranje prihvatljivosti.

5. Izrada prijedloga operativne dokumentacije: Plana testiranja, Test Case specifikacije, Test Design Specification, Procedure za provođenje testiranja, Test Status Report-a, Test Completion Report-a, Test Data Requirement-a, Test Environment Requirement-a, Test Readines Report-a, Test Environment Readiness Report-a, Test Execution Log-ova, Procedure za Incident Report, Test Summary Report-a itd.

6. Oblikovanje prijedloga organizacije za testiranje i definiranja odgovornosti:

- Ustrojstvo tima za testiranje
- Odgovornosti CIO-a, voditelja razvoja, projekatana, programera, testera, auditora
- Audit sustava testiranja

7. Automatizacija testiranja PO: Prepoznavanje i odabir alata. Korištenje odabranog / odabranih alata u potpori procesu testiranja

8. Uvježbavanje primjene razvijene dokumentacije

3.2.3. Konzalting usluge ZIH-a u verifikaciji i validaciji PO

I za verifikaciju i validaciju PO ZIH ima razrađene postupke prema normama **IEEE 1012, ISO 25000 i ISO 29119**. Tako npr., validaciju PO ZIH može načiniti evaluacijom:

- Načina upravljanja projekta razvoja PO,
- Plana projekta,
- Faze specifikacije zahtjeva
- Faze oblikovanja PO
- Faze programiranja
- Faze implementacije i korištenja
- Načina održavanja PO