

13. EKSTREMNO PROGRAMIRANJE

(Engl. **Extreme Programming – EP**) (1999.) - Struktura metodologije se može opisati kroz njene osnovne elemente, a koji se tiču načina organizovanja posla i uređivanja razvojnog ciklusa. **Osnovni strukturni elementi EP su:**

- Razvojni ciklus;
- Korisničke celine;
- Izdanja;
- Iteracije;
- Zadaci i
- Testovi.

Razvojni ciklus - jedan od osnovnih ciljeva EP je **drastično skraćivanje razvojnog ciklusa**. Specifičnost je **podela na dve osnovne vrste ciklusa: iteracije i izdanja**. Razvojni ciklus projekta započinje planiranjem korisničkih celina, nakon čega se ulazi u prvo izdanje. **Jedno izdanje može da se sastoji od više iteracija, a po njegovom završetku se započinje novo izdanje ili se projekat završava.**

Korisničke celine (engl. User stories) imaju sličnu ulogu kao slučajevi korišćenja (engl. Use Cases) u OO metodologijama, ali se u mnogo čemu od njih razlikuju. Korisničke celine predstavljaju **okvirnu specifikaciju zahteva. Piše ih klijent i obično nemaju tehničku preciznost** kao slučajevi korišćenja. A često su opisi veoma kratki. Korisničke celine služe kao relativno površne reference na **stvarne zahteve, koji će se detaljnije razmatrati tek kada dođu na red za implementaciju**. Osnovna namena korisničkih celina je da pomognu pri pravljenju okvirnog plana i praćenju toka projekta. One predstavljaju osnovne elemente za pravljenje planova iteracija. Zbog toga im se obično dodeljuju prioriteti i grube procene resursa potrebnih za implementaciju. Pošto celine nisu dovoljno detaljne, a uz to mogu i da se menjaju tokom vremena, na osnovu njih ne mogu da se prave tačne procene. Teži se da se korisničke celine definišu relativno brzo na početku projekta, pa se njihovom inicijalnom oblikovanju ne posvećuje mnogo vremena. Naravno, one se kasnije u toku rada na projektu mogu i menjati i dodavati ili uklanjati. Iako se ustanovljavaju u relativno kratkom roku, ipak se očekuje da svaka korisnička celina ima neke bitne elemente. Pre svega, korisnička celina mora da bude jasna i proverljiva, tj. mora da bude razumljiv i nedvosmislen poslovni cilj koji bi korisničkom celinom trebao da se ostvari, kao i kriterijumi kojima će moći da se ustanovi da li je celina ostvarena ispravno i dovoljno kvalitetno. U idealnom slučaju korisnička celina bi trebalo da može da se implementira u relativno kratkom vremenskom periodu, odnosno u okviru jedne iteracije. Ako to nije slučaj, onda se korisnička celina (prilikom planiranja iteracija) prevodi u skup manjih zadataka tako da svaki zadatak može da stane u jednu iteraciju.

Izdanje je osnovni deo razvojnog ciklusa, čiji je cilj da se izabran **skup korisničkih celina implementira** i ako je moguće **pusti u produkciju**. Izdanje može da se planira na dva osnovna načina: 1. **klijent izabere prioritete korisničke celine i zatim razvojni tim proceni potrebno angažovanje i okvirno trajanje izdanja**, ili 2. **najpre razvojni tim u dogovoru sa klijentom definiše trajanje izdanja i proceni raspoložive razvojne resurse**, a zatim klijent (uz konsultovanje sa razvojnim timom) izabere korisničke celine koje mogu da se implementiraju u zadatim okvirima. Prvi pristup više liči na klasičan pristup razvoju softvera, dok je drugi karakterističan za agilni pristup i poželjniji je u okviru EP. **Skup izabranih korisničkih celina se obično bira tako da predstavlja jednu veću zaokruženu celinu koja može klijentu da pruži veoma jasnu i prepoznatljivu korist**. Korisničke celine koje se dovrše u okviru izdanja mogu da uđu u produkciju, bilo odmah posle završetka izdanja bilo nakon što prođe period testiranja isporučenog izdanja od strane klijenta. **Jedno izdanje ne bi trebalo da traje više od nekoliko meseci. Ono se sastoji od više iteracija**. Sastavni deo planiranja izdanja je planiranje iteracija, pri čemu se okvirno planira šta će od izabranih celina da ide u koju iteraciju, ali se detaljno planira samo prva iteracija.

Iteracija je manji razvojni ciklus. Kao što se **ceo projekat deli na izdanja**, tako se **svako izdanje deli na iteracije**. **Cilj svake iteracije bi trebalo da bude implementacija jednog dela skupa korisničkih celina koje su izabrane za tekuće izdanje**. Izbor korisničkih celina za iteraciju se odvija na sličan način kao u slučaju izdanja, ali često razvojnog tima ima nešto veći uticaj na izbor nego u slučaju izdanja. **Po izboru celina, razvojni tim ih prevodi u zadatke**, a ako je neophodno (zbog neslaganja obima celine i obima iteracije) vrši i odabir zadataka koji će ući u iteraciju. **Zatim se programeri raspoređuju po zadacima za čiju su implementaciju zaduženi**. Nakon što završe svoje zadatke, programerima se dodeljuju novi zadaci, predviđeni za tu iteraciju. Uporedo sa razvojem, **klijent se bavi pravljenjem testova prihvatljivosti**. Isporučeni testovi se ugrađuju u projekat i izvode na implementiranim zadacima i celinama. Jedna iteracija bi trebalo da traje najviše nekoliko nedelja. Na njenom početku se detaljno analiziraju izabrane celine i zadaci. Rezultat rada iteracije, u načelu, može da ide i u produkciju, ali se do radi ređe nego u slučaju izdanja.

Zadaci predstavljaju elementarne delove korisničkih celina. Teži se da se zadaci oblikuju tako da **jedan zadatak može da se završi za nekoliko sati**. Naravno, to nije uvek moguće, pa neki zadaci mogu da budu i značajno većeg obima. Rad na zadatku počinje formiranjem para (praksa Programiranja u paru) i kratkim sastankom (oko 15 minuta) sa klijentom i programerima koji rade na neposredno povezanim zadacima. Rezultat tog sastanka bi trebalo da bude početni kratak ali sadržajan spisak testova koji bi morali da rade da bi se zadatak smatrao završenim (prakse Testovi prihvatljivosti - AT i Razvoj vođen testovima – TDD Test Driven Development). Nakon podele korisničke celine na zadake može da se desi da neki zadaci imaju strogo tehnički karakter i nemaju neposrednog dodira sa klijentom. U takvom slučaju nema potrebe da sastanku prisustvuje klijent, ali se umesto njega obično uključuje vođa tima. **Cilj rada na jednom zadaku je ispunjavanje definisanih testova**. Tokom rada se obično prave i proveravaju i novi detaljniji testovi (praksa TDD).

Testovi - Pored specifičnog oblika razvojnog ciklusa moglo bi se reći da su **testovi ključni element EP**. Testiranje softvera je tek sa EP postalo uobičajena praksa na svim nivoima razvoja i puštanja softvera u rad. Danas se raspoznaje veliki broj različitih vrsta testova i oni se sistematski projektuju i sprovode da bi se obezbedio visok kvalitet završnog proizvoda. U kontekstu priče o testiranju svi testovi se obično dele na one koje pravi (ako ne tehnički a onda bar konceptualno) klijent i koji služe da se proverí da li je završen proizvod (ili deo proizvoda) u skladu sa očekivanjima. Takve testove obično **nazivamo testovima prihvatljivosti** i oni služe da provere

ispravnost i kvalitet proizvoda ili njrgove komponente kao složene celine. Drugu grupu testova čine testovi koje prave programeri i čiji je cilj da testiraju pojedinačne strukturne elemente softvera. Na najnižem nivou pišu se i izvode **jedinični testovi**, koji služe da testiraju rad pojedinačnih i elementarnih jedinica koda - funkcija, metoda, klasa, događaja ili paketa. Iznad toga imamo **integracione testove i sistemske testove**, koji na tehničkom nivou proveravaju ispravnost rada većih funkcionalnih celina softvera, kao što su moduli, komponente, servisi i slično. Sve ove testove prave i izvršavaju testeri i programeri.

