



Agilni manifest: Pojedinaac i njegov tim planiraju zajedno

Agilne metode rada su se pojavile početkom 1990-tih godina kod razvoja softwera i stekle prvu popularnost sa Extreme Programming (Kent Beck). Značaj agilnih metoda rada znatno se povećao tijekom posljednjih godina. Ovo, prije svega – ali ne i isključivo – važi za IT-sektor: 62% projekta za razvoj softwera radi u Njemačkoj djelomično agilno, čak 16% kontinuirano, dok samo 22% svih projekata rade isključivo sa konvencionalnim metodama.

Agilna filozofija naglašava sposobnost pojedinca i samoorganizaciju tima za razvoj. „Agilnim metodama“ se želi prevladati nedostatke prethodnih razvojnih modela softwera, koji se smatraju previše krutim. Kako kruto i kontinuirano planiranje više nije moguće, zahtjevi koji se postavljaju za softwera nisu od početka određeni. Oni se, umjesto toga, razvijaju sa rastućim razumijevanjem softwera-sustava, što bi se trebalo postići intenzivnom komunikacijom s kupcima. U cilju olakšanja komunikacija, Extreme Programming se, između ostalog, koristi za razumijevanje takozvane „User Stories“, znači korisničkih priča za opisivanje svojstava sustava odnosno bitnih funkcija. Planiranje treba provesti u kooperaciji, to nije posebna odluka menadžera projekta. Uskladiti jednu skupinu za planiranje obujma sljedeće (softver-) dozvole ili stavljanja na raspolaganje, naziva se Planning Games, a procjena projektnog napora se ponavlja ciklički (iterativan proces).

Na jednom sastanku u Utah-u 2001. god. je za ovu vrstu razvoja softwera odabran naziv „agilan“ i formuliran je „agilni manifest“ (www.agilemanifesto.org/iso/de/manifesto.html): „Razvijamo bolje načine razvoja softwera, tako što to sami radimo, i drugima pri tome pomažemo. Kroz ovu djelatnost smo naučili cijeniti sljedeće vrijednosti:

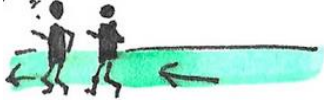
- **Individue i interakcije** su iznad procesa i alata
- **Funkcionirajući softwera** je iznad sveobuhvatne dokumentacije
- **Suradnja s kupcem** je iznad pregovora o ugovorima
- **Reagiranjem na promjene** je iznad praćenja plana.”

Agilni principi su vodeća načela za agilan rad. Ponekad se agilni principi također nazivaju metodama. **U agilnom manifestu je navedeno dvanaest principa.**¹

- Zadovoljiti kupca kroz ranu i kontinuiranu isporuku vrijednog softwera
- Agilni procesi koriste promjene (čak i u kasnom razvoju) za konkurentsku prednost kupca
- Isporuka funkcionirajućeg softwera u redovitim, po mogućnosti kratkim vremenskim razdobljima (par tjedana ili mjeseci)
- Gotovo svakodnevna suradnja između stručnjaka i programera tijekom projekta (n.pr. zajedničko vlasništvo Code-a (Collective Code Ownership)
- Raspoloživost okruženja i potpore, što je potrebno motiviranim pojedincima za ostvarenje zadaća
- Prijenos informacija, po mogućnosti u razgovoru licem u lice.
- Kao najvažnija mjera napretka važi funkcionalnost softwera
- Održavati ravnomjeran radni tempo nalogodavca, programera i korisnika za održiv razvoj
- Stalno pratiti tehničku izvrsnost i dobar dizajn
- Jednostavnost je neophodna
- Samoorganizacija timova kod planiranja i provedbe
- Samo-refleksija timova o vlastitom ponašanju kod prilagodbe u smislu povećanja efikasnosti

Princip održivog tempa: Nema prekovremenog rada

1) MAINTAIN A SUSTAINABLE PACE



Jedan od dvanaest agilnih principa je održiv tempo (sustainable pace). Kod Beck-a postoji „40-hour week“ (40-satni radni tjedan) kao jedna od dvanaest praksi Extreme Programmings. Prekovremeni rad ukazuje na ozbiljne probleme u projektu, koje treba lokalizirati i riješiti. Uslijed

stalnog prekovremenog rada programeri nisu više u stanju, kodirati jasno i pregledno. Postaju neproduktivni i programiraju više grešaka. Roman Pichler kao istaknuti predstavnik piše u svezi s tim: Za Scrum – najčešća agilna metoda – je redoviti prekovremeni rad neprihvatljiv.

»Agilni razvoj software-a je skupni pojam za niz metoda i praksi, koje baziraju na vrijednostima i principima manifesta agilnih razvoja software-a.« (Agile Alliance, 2018)

U **Scrum-u**, najrasprostranjenijem agilnom pristupu, postoje uloge Product Owner, Team i Scrum Master. Product-Owner preuzima gledište korisnika i u takozvanom Product Backlog opisuje zahtjeve koji moraju biti ispunjeni, kako bi se postigao cilj projekta. Taj dokument obrađuje kontinuirano i postavlja prioritete zahtjeva, ovisno o cjelokupnom napretku.

Zahtjevi se provode u tzv. sprintovima (2-4 tjedna). U Sprint Planning se razvojni tim dogovara sa Product Owner o zahtjevima. Tim odabira zadaće, koje će obrađivati u sljedećem sprintu, i procjenjuje potrebni napor, kako bi se osigurala izvodivost u odabranom razdoblju sprinta.

Rezultati Sprint-a se predstavljaju Product Owner-u u Sprint Review (po potrebi i kupcima/korisnicima). Samo testirane i funkcionalne funkcionalnosti se smatraju gotovim.

Tim se za cijeli proces sam organizira i mora raspolagati određenim resursima i sposobnostima. U retrospektivi nakon sprinta, tim odražava proces rada i suradnju, kako bi odredio mjere za poboljšanje. Scrum Master, kao stručnjak za tu metodu, podržava tim u cilju očuvanja i primjene pravila koja su sami postavili i uklanjanja prepreka sa puta.

Dr. Nadine Müller, ver.di (DE)