



Scrum je pomogao Omega softwareu stvoriti jake timove međusobno povezanih kolega koji posjeduju kompetencije, ali i kreirao kompetitivnu okolinu, ne samo unutar tima, nego i između timova

Pet godina iskustva razvoja softwarea *scrum* metodologijom

Scrum je nastao devedesetih godina prošlog stoljeća, kao rezultat analize japanske kompanije Honda i njihova jedinstvenog stila u razvoju novih proizvoda. *Scrum* tehnika, koja je općeprihvaćena u razvoju softvera, počiva na nekoliko naizgled jednostavnih načela, poput empirijske kontrole procesa (umjesto teoretskog razumijevanja), samoorganizacije timova, kolaboracije, prioritizacije utemeljene na vrijednosti za korisnika, strogim vremenskim okvirima (sprintevima) i iterativnom razvoju rješenja

Bilo da ste trgovačko društvo ili javna institucija, u poslovanju morate koristiti različite softverske alate za upravljanje i organizaciju. Najčešće su to ERP sustavi (Enterprise Resource Planning), koji, ako su ispravno parametrirani i implementirani, upravljaju svim resursima, neovisno o tome jesu li u pitanju imovina, zalihe ili zaposlenici, a kao posljedicu bilježenja svih poslovnih procesa, u pozadini kreiraju kontinuiranu i ažurnu knjigovodstveno-računovodstvenu sliku poslovanja organizacije.

Prednosti korištenja ERP sustava mnogobrojne su, i implementacija svakako pomaže i osnažuje prihvaćanje najboljih poslovnih praksi koje su se iskristalizirale tijekom desetljeća razvoja ERP sustava. Unatoč fleksibilnosti, ponekad je za rješenje specifičnog poslovnog problema jednostavno potrebno naručiti posebni *custom made*, odnosno nadogradnju softvera koji će pratiti, i na odgovarajući način, adresirati specifični poslovni izazov koji je postavljen pred organizaciju.

STANDARDIZIRANA RJEŠENJA

Zagrebačka Omega software već desetljećima razvija različita standardizirana softverska rješenja, poput ERP sustava ili rješenja za uredsko poslovanje, rješenja koja konkuriraju inozemnim rješenjima, usvojena od nekih najvećih domaćih institucija i trgovačkih društava. Manje poznati dio poslovanja Omega softwarea je i razvoj specijaliziranih *custom* rješenja za poznatog naručitelja, u kojima razvojni timovi Omega softwarea do detalja razlažu poslovni problem na njegove dijelove, kreiraju popise funkcionalnosti, i potom kreću u razvoj rješenja.

BRZINA JE SRCE RJEŠENJA

Odavno su prošla vremena kada su se softverska rješenja planirala i isporučivala u višegodišnjim rokovima. Moderno poslovanje, neovisno o tome govorimo li o trgovačkim društvima ili državnoj i javnoj upravi, pod kontinuiranim je pritiskom svojih korisnika, pa jednostavno nema više toliko vremena na raspolaganju. Poslovni problem treba riješiti brzo, učinkovito, učiniti ga što prije dostupnim korisnicima, te prema novim potrebama i očekivanjima korisnika i naručitelja, dograđivati i adaptirati "u hodu". Čak ni metodologija "vodopada", koju još uvijek neki koriste (ali ne i u Omega softwareu), a koja je predviđala inkrementalni razvoj softverskih rješenja, nije više prihvatljiva jer se softver metodologijom "vodopada" razvija linearno, i dovršenje jedne razvojne faze preduvjet je za početak sljedeće.

Razvojni stručnjaci Omega softwarea, a njih je više od pedeset, podijeljeni su u devet razvojnih timova, koji funkcioniraju prateći agilnu metodologiju razvoja softvera. Tih devet timova u razvoju poslovnih rješenja za svoje korisnike koriste *scrum* tehniku, koja je prije svega fokusirana na isporuku rješenja za svojeg korisnika i naručitelja u najkraćem mogućem vremenskom roku. *Scrum* je prema svojoj prirodi, kolaborativni proces, koji omogućuje da više timova istodobno rješava različite aspekte istog poslovnog izazova, kako bi u konačnici zajedno isporučili rješenje kompleksnog problema.

Korištenje *scrum* tehnike razvoja softvera, u Omega softwareu omogućuje da veći broj razvojnih stručnjaka istodobno radi na problemu, čime se vrijeme razvoja drastično skraćuje, a nove funkcionalnosti mogu se isporučiti korisnicima u kratkim vremenskim intervalima, koji tipično traju dva ili četiri tjedna, pa se zato s dobrim razlogom nazivaju *sprintevi*. U *scrum* tehnici, svi se

Klaudia Teković,
Voditeljica sektora
konzaltinga i
implementacije



članovi tima, ili čak više timova, kreću prema zadanom cilju istodobno, pa je i rezultat moguće isporučiti mnogo brže.

SCRUM TEHNIKA

Omega software je *scrum* tehniku počela koristiti prije više od pet godina, a njihovi inženjerke i inženjeri tvrde kako im je rad u

malim agilnim timovima osigurao autonomiju, slobodu i fleksibilnost organiziranja rada u grupi.

Pomogao im je stvoriti jake timove međusobno povezanih kolega koji posjeduju kompetencije, ali i kreirao kompetitivnu okolinu, ne samo unutar tima, nego i između timova (koji su odabrali imena superheroja iz stripo-

DEFINICIJA SPREMNOSTI

Kako bi se to postiglo, za svaki pojedinačni zahtjev potrebno je zadovoljiti tzv. DoR (Definition of Ready), što je preduvjet da bi zahtjev mogao biti spreman postati dio razvojnog *backloga* – popisa funkcionalnosti spremnih za razvoj. Poslovna potreba za implementacijom zahtjeva mora biti jasno opisana, i mora se znati čemu tražena funkcionalnost služi, što se njome želi postići i zašto. Uz poslovnu potrebu, vrlo važan je i raspis kriterija prihvaćanja koji se promatra kroz izvedbeni model. U sklopu tog dijela opisuje se što je potrebno implementirati, tako da se kreiraju skice ekrana, raspisuju podatkovni model i ograničenja, te osiguravaju da je sve što je potrebno za implementaciju dio samog zahtjeva.

Trajanje *sprinta* nije određeno, ali najčešće traje dva ili četiri tjedna – ovisno o očekivanom trajanju, potrebnom za izvršenje svih zadataka planiranih u *backlogu*. *Scrum developeri* preuzimaju zadatke s *backloga* ili *sprint taskboarda*, poštujući prioritete i na dnevnim sastancima procjenjuju svoj napredak, te ostatku tima daju bitne informacije koje nisu vidljive samo iz pregleda *taskboarda*. Tom prilikom odlučuje se o tome postoji li potreba za dodatnim raspravama u obliku "parkiranih tema". Parkirane teme dodatna su pitanja ili rasprave nakon redovnih sastanaka za cijeli ili dio tima, a koje su se pokazale iznimno korisnima, jer tim ima priliku svaki dan, bez dodatnog ometanja kolega, u fokusiranom radu riješiti tekuće prepreke ili prokomentirati određenu temu. ◀



Omegin web: desetljećima razvijana standardizirana, ali i specijalizirana custom softverska rješenja i usluge

va i crtića, poput The Incredibles, Powerpuff Girls ili Scarlet Witch). Višegodišnje iskustvo rada uz *scrum* tehniku pokazalo je kako je učinkovitije imati permanentne *scrum* timove, koji se angažiraju za rješavanje pojedinačnih zadataka, nego da se timovi organiziraju i raspuštaju, ovisno o projektu.

ALATI ZA SPRINT

Iako većina rezultata proizlazi iz rada, odabir alata gotovo je jednako važan element uspjeha. U Omega softwareu, gotovo se tradicionalno oslanjaju na Microsoftove alate, ponajviše na Azure DevOps servis, koji sve timove i cjelokupnu organizaciju na odgovarajući način podržava u organizaciji rada na razvojnim projektima, u skladu s okvirom *scruma*, upravljanjem kôdom i drugim elementima razvoja, modeliranjem kanala za izgradnju i isporuku novih funkcionalnosti i pružanje tehničke korisničke podrške.

Odabir Microsoftove platforme nije slučajna: riječ je o jednoj od najboljih i najvećih dostupnih razvojnih platformi, a koja jamči dugoročnu i kvalitetnu podršku, ne samo razvojnim timovima, nego i krajnjem korisniku. Budući da se radi o industrijski standardnom alatu, lakše je privući i održati talente, jer svako poznavanje platforme bit će dugoročno korisno, a edukacijski materijali uvijek su dostupni. Sama platforma predmet je agilnog razvoja, pa se tako i alati redovito nadograđuju, čime se olakšava i ubrzo razvoju, a nove funkcionalnosti vrlo brzo nađu put do krajnjeg korisnika. Odabir platforme Azure DevOps jamči dugoročnu podršku, i u konačnici čuva ekonomsku vrijednost projekta na dugi rok.

ANATOMIJA SPRINTA

Svaki razvojni *sprint* sastoji se od nekoliko

obaveznih dijelova. Iako se na prvi pogled možda ne čini tako, prilikom izgradnje velikih sustava nije uvijek jednostavno, ili čak moguće, definirati koje minimalne funkcionalnosti sustav treba imati – u tom kontekstu bitna je suradnja *product ownera* i ključnih korisnika. *Product owner* osoba je koja, u suradnji s klijentom i *scrum* timom, definira prioritete i slijed (roadmap) razvoja budućeg sustava.

U internim procesima *product owner* usko surađuje s cijelim timom, prije svega kroz postupak *refinements*, koji je prepoznat kao ključna aktivnost. *Refinement* je kontinuirana, redovita aktivnost, tijekom koje se poslovni zahtjevi, opisani korisničkim jezikom, susreću s razvojnim timom koji na korisnikovo "što" definira "kako", tj. implementacijske detalje, i tako daje bitne informacije o trajanju razvoja pojedine funkcionalnosti. *Refinement* je i idealan trenutak na kojem cijeli tim zajedno raspravlja o zahtjevima, i mora pokazati svoje razumijevanje korisničkih potreba, izvedbenog rješenja i vremena potrebnog za implementaciju, te provjerava ima li poznatih međuovisnosti – nepredviđenih, ili do tog trenutka neprimijećenih prepreka, koje bi mogle utjecati na razvoj i uspješni dovršetak *sprinta*.

Proces *refinements* je fleksibilan, i timovi ga prilagođavaju svojim potrebama, ali većina ih u tu aktivnost ulaže više sati tjedno. Naravno, svemu često prvo prethode fokusirane poslovne radionice, s ključnim korisnicima, kao i obrada zahtjeva raspisivanjem očekivanih funkcionalnosti i očekivanja korisnika na *backlogu*.

Tim Omega softwarea The Incredibles dosad je imao najduži kontinuitet rada u skladu s filozofijom i tehnikom *scruma*, i uspješno je završio jedan od projekata tijekom čak 124 dvotjedna *sprinta*. Gotovo

pet godina rada, u skladu s okvirom *scruma*, iznjedrilo je različite dobre prakse, a jedna od njih su i izolirane okoline za testiranje funkcionalnosti. Tako timovi osiguravaju još veću razinu kvalitete klijentima, jer se sâm zahtjev provjerava prije spajanja s potvrđenim i testiranim dijelom sustava. Takva provjera, osim revizije programskog kôda, uključuje i konkretno testiranje i potvrdu nove funkcionalnosti.

Na kraju svakog *sprinta* korisnicima se na provjeru, a potom i na produkcijsku okolinu, isporučuje funkcionalna nadogradnja sustava, a nakon osvrta, cijeli proces počinje ispočetka.

Bitni element *sprinta* je i osvrta, koji se nakon završetka *sprinta* događa između članova tima, a nerijetko i uz ključne korisnike, što je prilika za dobivanje redovitog uvida u rezultate rada tima. Predmet osvrta je i demonstracija svega što je napravljeno, objašnjenje i razlozi neizvršenja pojedinih zadataka, te raspravi o rizicima i preprekama na koje se naišlo, a koji su bitni za naredne *sprintove*. Osvrta je redovita aktivnost, u kojoj je klijent u prilici vidjeti novi komadić funkcionalnosti, i dati povratne informacije, koje se uzimaju u obzir prilikom planiranja sljedećih *sprintova*. Osim redovnih osvrta, korisnici povratne informacije uvijek mogu dati i kroz sustav podrške, koji im omogućuje da u realnom vremenu, sistematizirano, prijavljuju uočene nedostatke i predlažu daljnja poboljšanja.

POGLED UNATRAG

Scrum je nastao devedesetih godina prošlog stoljeća, kao rezultat analize japanske kompanije Honda i njihova jedinstvenog stila u razvoju novih proizvoda. *Scrum* tehnika, koja je općeprihvaćena u razvoju softvera, počiva na nekoliko naizgled jednostavnih načela, poput empirijske kontrole procesa (umjesto teoretskog razumijevanja), samoorganizacije timova, kolaboracije, prioritizacije, utemeljene na vrijednosti za korisnika, strogim vremenskim okvirima (*sprintevima*) i iterativnom razvoju rješenja.

Scrum u svojoj srži prihvaća da korisničke specifikacije neće uvijek biti razumljive, kako korisnik neće uvijek znati što točno želi prije nego što sustav uđe u produkciju (a ponekad ni tada), te kako složeni i interaktivni softverski sustav ne može biti detaljno opisan, ili čak u cijelosti testiran. Zbog svih tih razloga, iznimno je bitno redovito se podsjećati na načela *scrum* razvoja, razviti otvoreni i transparentni sustav komunikacije, ne samo između članova tima, nego i s krajnjim korisnicima sustava i vlasnicima procesa u organizaciji za koju se rješenje razvija. M