

Priča o Pravilima

Pravila programiranja su rođena iz ogorčenja.

Proveo sam oko deceniju vodeći programske timove u Microsoftu, a zatim suosnivao kompaniju za video igre Sucker Punch 1997. Obe kompanije su bile uspešne – velikim delom zbog svoje sposobnosti da regrutuju i razviju vrhunske programerske timove. U Sucker Punchu, to je dovelo do 25-godišnjeg niza uspešnih igara. Postojale su tri igre *Sly Cooper*, koje su omogućile deci svih uzrasta da iskuše uzbudljiv život glavnog kradljivca rakuna Sly Coopera i njegovih prijatelja. Bilo je pet *inFamous* igara, koje su igračima dale supermoći i izbor da ih koriste za dobro ili zlo. A tu je i ono što je do sada naš veliki opus, *Ghost of Tsushima*, gde igrači igraju usamljenog samuraja koji se bori protiv invazije na Japan 1274. godine.¹

Veliki deo strategije regrutovanja i u Microsoftu i u Sucker Punchu je zapošljavanje pametnih mladih programera, a zatim njihovo obučavanje na način profesionalnih programera. Ova praksa je bila neosporno uspešna, ali je vodila i do određene frustracije.

Stalno sam nailazio na isti problem iznova i iznova. Dovedi bismo nove programere u tim, često sa tek završenim fakultetom. Pregledao bih neku novu funkciju koju su planirali da dodaju u kôd, obično za rešavanje veoma jednostavnog problema – da bih otkrio da su napisali kôd koji pokušava da reši mnogo veći problem, onaj u kome je veoma jednostavan i konkretan problem samo mali podslučaj.

Grrr! Nije nam potrebno da se reši taj veći problem, svakako ne sada! Kao i uvek rešenje većeg problema je osrednje rešenje jednostavnog problema koga imamo – komplikovano za korišćenje, komplikovano za razumevanje i moguće je da krije mnogo grešaka. Ali kada bi im u recenziji koda rekli² – da nam nije potrebno rešavanje većeg problema, da treba da pokušaju da reše problem koji razumeju – bilo bi uzalud. Nastavljali su to da rade.

1 Oštrooki među vama su možda primetili da sam izostavio *Rocket: Robot on Wheels*, prvu igru Sucker Puncha. To je zato što je retko ko od vas nju igrao; ako ste jedan od retkih, imate moju zahvalnost.

2 Sav kôd projekta Sucker Puncha prolazi recenziju koda. Pogledajte Pravilo 6 za više detalja.

Zbog frustracije, rekao sam dosta. „U redu. Evo novog pravila. Dok nemate tri primera problema, nije vam dozvoljeno da napišete opšte rešenje“.

Na moje iznenađenje i oduševljenje, ovo je zaista dalo rezultate! Pretvaranje opšte filozofije u specifično pravilo sa specifičnim kriterijumima bio je efikasan način da se poruka prenese. Naravno, većina naših novih programera bi jednom napravila grešku preuranjene generalizacije, ali im je pravilo pomoglo da izbegnu da je ponovo naprave. Pomoglo im je i da prepoznaju kada je *došlo* vreme da generalizuju. Manje od tri primera? Ne generalizujte. Tri ili više? Počnite da tražite mogućnosti.

Pravilo je funkcionisalo jer ga je bilo lako zapamtiti, a situacije u kojima se primenjuje bilo je lako prepoznati. Kada su koderi mogli da vide da su počeli da prelaze granice dobro definisanog problema, mogli su da naprave korak unazad, izbroje broj konkretnih primera te vrste problema na koje bi naišli i doneli bolju odluku o tome da li generalizovati ili ne. Pisali su bolji kôd.

Vremenom smo pronašli druge važne delove filozofije Sucker Puncha koji bi mogli da se destiluju u lako pamtljive fraze – aforizme, da budemo precizni. Duga je istorija aforizama – tih kratkih, sadržajnih izjava koje obuhvataju neku suštinsku istinu. Kladam se da možete da izvučete popriličan broj iz sećanja. Mogao bih da vas ograničim na aforizme *u vezi sa pticama*, a vi biste ipak mogli da smislite bar dva! Ponudiću nekoliko:

- Ne brojite svoje piliće pre nego što se izlegu.³
- Bolje vrabac u ruci nego golub na grani.
- Ptica koja se prva izleže uzima crva.
- Ne stavljajte sva jaja u jednu korpu.

Aforizmi postoje jer funkcionišu. Oni su viralni, u savremenom smislu – hiljadama godina aforizmi „inficiraju“ ljude delićima mudrosti.⁴ Nije iznenađujuće što su efikasan način zaražavanja novih članova tima sa filozofijom kodiranja Sucker Puncha.

Dakle, mic po mic, ono što je nekada bilo jedno pravilo preraslo je u listu pravila: Pravila programiranja opisana u ovoj knjizi. Ona predstavljaju mnoge od najvažnijih aspekata inženjerske kulture Sucker Puncha: stvari za koje verujemo da

3 U svom originalnom obliku, iz *Novi soneti i lepi pamfleti (New Sonnets and Pretty Pamphlets)* Thomas Howella (1570): „Ne računaj svoje neizležene kokoške...“ (Count not thy chickens that unhatched be...) Mala demonstracija trajnosti dobrog aforizma.

4 Samu reč *aforizam* je skovao Hipokrat oko 400. godine pre nove ere. Pa, strogo govoreći, reč koju je skovao bila je *Ἀφορισμός*. To je bio naslov njegove knjige pravila za medicinsku dijagnozu i lečenje, od kojih su neka i dalje prisutna milenijumima kasnije, poput aforizma 13 iz odeljka 6 – „napad kijanja će izlečiti slučaj štucanja“. Tako istinito.

su dovele do našeg uspeha, ideje koje novi koderi u timu moraju da usvoje da bi bili efikasni. Stvari na koje čak i stariji koderi poput mene treba da se podsete bar s vremena na vreme!

Svako od poglavlja koje sledi opisuje Pravilo, sa mnogo primera koji ilustruju ideju iza njega. Nakon čitanja poglavlja, trebalo bi da imate jasnu predstavu o praksi kodiranja koju Pravilo podstiče i situacijama u kojima se primenjuje.

Da li su Pravila isto tako zarazna i u obliku knjige? Hajde da otkrijemo.