Predgovor

Projektovanje softvera je inženjerska disciplina i jedna od važnih aktivnosti u razvoju softvera. Ono se zasniva na određenim principima koji se odnose na modelovanje strukture i ponašanja softvera, kao i na primenu dobro poznatih projektnih uzoraka (šablona, obrazaca) u rešavanju problema. Kao i svaki drugi inženjerski proizvod, softver u svom nastajanju prolazi kroz prepoznatljive faze i ponavljajuće aktivnosti u svim fazama, sve do instalacije na ciljne mašine. Instalacijom tek počinje životni vek softvera kao proizvoda. Život svakog netrivijalnog softverskog proizvoda se sastoji od njegovog korišćenja (eng. *usage*) i održavanja (eng. *maintenance*). Održavanje uključuje popravke i unapređenja. Praktično isti principi projektovanja softvera se primenjuju kako na razvoj proizvoda, tako i na održavanje tog proizvoda u toku njegovog životnog veka.

Primenu principa projektovanja softvera omogućava odgovarajuće izražajno sredstvo projektanata – jezik za modelovanje softvera. Jezik UML (*Unified Modeling Language*) je standardizovan grafički jezik za vizuelizaciju, specifikaciju, konstruisanje i dokumentovanje sistema u kojima je softver bitna komponenta. Specifikacija (standard) jezika UML propisuje trinaest vrsta dijagrama, od koji je šest vrsta namenjeno modelovanju strukturnih aspekata sistema, a sedam vrsta namenjeno modelovanju ponašanja sistema.

Tema ove knjiga je modelovanje strukturnih aspekata objektno orijentisanog softvera i relevantnih aspekata hardvera na kojem se softver izvršava. Autor se odlučio na pisanje knjige nakon više od dvadaest godina iskustva držanja predmeta Projektovanje softvera (ranije pod nazivom Objektno orijentisani softver) na modulu Računarska tehnika i informatika i studijskom programu Softversko inženjerstvo na Elektrotehničkom fakultetu Univerziteta u Beogradu. Na predmetu se izučavaju principi projektovanja softvera uz njegovo modelovanje na jeziku UML i primenu poznatih projektnih uzoraka. Knjiga, čija je tema modelovanje strukturnih aspekata, pokriva samo jedan deo građe za ovaj predmet.

Osim stečenih iskustava u prenošenju znanja u oblasti projektovanja softvera, autor je sticao i praktična iskustva na velikom broju softverskih projekata čiji je razvoj vodio. Na tim projektima su razvijeni softverski proizvodi (aplikacije i sistemi) od kojih se neki, na svetskom tržištu, već više od dve decenije uspešno primenjuju, održavaju i unapređuju. Primarni motiv autora za pisanje ove knjige je bio da svoja iskustva stečena kroz proces razvoja softvera, a naročito iskustva u prenošenju znanja na predmetu koji se bavi projektovanjem softvera, objedini i prenese čitaocu. Ipak, autor se čuvao zamki da lična iskustva predstavi kao univerzalne principe. Svoja iskustva je autor imao u vidu samo kao proveru široko prihvaćenih principa i standarda kojih se čvrsto drži ova knjiga. Ukoliko je reč o ličnom stavu, autor je nastojao da to posebno naglasi. Nedostatak adekvatne literature koja pokriva gradivo predmeta Projektovanje softvera, na kojem se izučavaju principi i praksa tog segmenta softverskog inženjerstva, a prvenstveno nedostatak literature na srpskom jeziku, je svakako predstavljao autoru dodatni motiv za pisanje ove knjige.

Knjiga je namenjena svima koji nameravaju da se u karijeri bave projektovanjem softvera na profesionalan način. To su prevashodno studenti softverskog ili računarskog inženjerstva, kao i studenti drugih usmerenja u širokoj oblasti računarstva. Ali knjiga je namenjena i onima koji, iako već odavno nisu studenti, žele da prošire, unaprede i sistematizuju svoja znanja iz projektovanja softvera. Tu spadaju analitičari, projektanti, programeri, testeri i drugi neposredni učesnici u razvoju i održavanju softvera. Delovi knjige bi bili od koristi i rukovodiocima projekata čiji je cilj razvoj softverskog proizvoda ili uređaja čija je važna komponenta softver. Činjenica je da i danas mnoga preduzeća koje se bave razvojem softvera

svoje proizvode ili uopšte ne modeluju, već neposredno implementiraju, ili ih modeluju na osnovu relativno uskih ličnih iskustava arhitekata softvera i programera, ne oslanjajući se na šira iskustva formulisana kroz principe projektovanja i standardni jezik UML, koje ova knjiga nastoji da približi čitaocima. Autor je uveren da bi ova knjiga mogla da bude korisno štivo čak i onima koji vode preduzeća, odeljenja ili projektne timove koji se bave proizvodnjom softvera, a ubeđeni su da je modelovanje softvera samo gubljenje vremena.

Od čitaoca koji ima nameru da u potpunosti razume izloženu materiju se očekuje da mu čitanje ove knjige nije prvi kontakt sa objektno orijentisanim načinom razmišljanja u razvoju softvera. Studenti koji prate predmet Projektovanje softvera su prethodno pratili dva jednosemestralna predmeta sa odgovarajućim praktikumima iz proceduralnog programiranja i dva jednosemestralna predmeta iz objektno orijentisanog programiranja, na kojima su učili principe programiranja i programiranje na jezicima *Pyton* (ili ranije *Pascal*), *C*, *C++*, *Java* i *C#*. Ipak, ne očekuje se tako bogato predznanje od svih čitalaca. Bilo bi dovoljno da je čitalac upoznat sa osnovnim konceptima objektno orijentisanog programiranja i da ima neku praksu programiranja na barem jednom objektno orijentisanom jeziku.

Iako se ova knjiga ne bavi korišćenjem projektnih uzoraka, ukoliko se knjiga koristi kao udžbenik za kurseve iz projektovanja softvera (eng. *software design*), preporuka za predavače je da se na kursu simultano napreduje kroz principe modelovanja na jeziku UML i primenu projektnih uzoraka. Slušaocima kurseva projektovanja softvera je potrebno razviti veštinu prepoznavanja projektnih uzoraka u ranim fazama razvoja softvera. Kako se u notaciji jezika UML projektni uzorci mogu strukturno opisati već na dijagramima klasa, ne postoji prepreka da se odmah po izlaganju lekcije o ovim dijagramima jezika UML obrade i prvi projektni uzorci. Kombinovanjem ovih tematika slušaoci dobijaju priliku da odmah vežbaju projektovanje softvera modelovanjem na jeziku UML uz primenu projektnih uzoraka, što bi trebalo da bude i cilj izučavanja projektovanja softvera.

Nijedna veština se ne može steći bez vežbanja. U ovoj knjizi se izlaže materija koju treba razumeti i naučiti, ali da bi čitalac stekao adekvatnu veštinu projektovanja softvera, neophodno je da vežba da projektuje softver. Iz tog razloga čitaocu se, kao prateća literatura uz ovu knjigu,

preporučuje *Zbirka zadataka iz projektovanja softvera*, čiji su autori Laslo Kraus i Igor Tartalja, koja se koristi na predmetu Projektovanje softvera na Elektrotehničkom fakultetu u Beogradu. Ona je metodološki organizovana tako da postupno uvodi čitaoca u materiju i dobro prati predloženo simultano napredovanje kroz tematike modelovanja na jeziku UML i primene projektnih uzoraka.

Autor zahvaljuje svima koji su ga nagovarali da napiše ovu knjigu i svima koji su ga odgovarali od tog posla. Bez prvih knjiga verovatno nikada ne bi bila započeta. Bez drugih nikada ne bi bila završena. Ipak, najveću zahvalnost autor duguje onima koji su već pročitali rukopis knjige. Bez njih knjiga nikada ne bi bila objavljena u ovom obliku. Bez njihovih korisnih zamerki i sugestija čitaoci bi imali mnogo više muka u razumevanju i savladavanju materije izložene u knjizi.

U uverenju da će knjiga biti od koristi njenim čitaocima, autor se potajno nada da im neće biti ni smrtno dosadna. Moli ih da čitaju pažljivo, a za sve greške i nejasnoće koje otkriju i prijave mu, biće im veoma zahvalan.

Beograd, avgust 2022. Autor

Kratki sadržaj

Predgovor ..... *i*

Kratki sadržaj ..... *v*

Sadržaj ..... *vi*

Slike, tabele i kod ..... *xiii*

1. Uvod *1*

2 O jeziku UML *43*

3 Opšti pojmovi *75*

4 Dijagrami klasa *91*

5 Dijagrami paketa *157*

6 Dijagrami objekata *171*

7 Dijagrami slučajeva korišćenja *179*

8 Dijagrami složene strukture *193*

9 Dijagrami komponenata *211*

10 Dijagrami raspoređivanja *225*

11 Mehanizmi proširivosti jezika UML *235*

12 Literatura *243*

Sadržaj

Predgovor ..... *i*

Kratki sadržaj ..... *v*

Sadržaj ..... *vi*

Slike, tabele i kod ..... *xiii*

1. Uvod *1*

1.1 Analiza, projektovanje i implementacija 4

1.1.1 Analiza 4

1.1.2 Projektovanje 5

1.1.3 Implementacija 6

1.2 Osobine softvera 9

1.3 Principi objektne orijentacije softvera 10

1.3.1 Ključni principi 11

1.3.1.1 Apstrakcija 11

1.3.1.2 Kapsulacija 13

1.3.1.3 Hijerarhija 14

1.3.1.4 Polimorfizam 16

1.3.1.5 Modularnost 17

1.3.2 Sekundarni prinicipi 18

1.3.2.1 Konkurentnost 18

1.3.2.2 Perzistencija 19

1.4 Proces razvoja softvera 20

1.4.1 Metod vodopada 22

1.4.2 Metod spirale 23

1.4.3 Metod objedinjenog procesa 24

1.4.3.1 OpenUP 27

1.4.4 Agilni metodi 28

1.4.4.1 Ekstremno programiranje 32

1.4.5 Radni okvir *Essence* ... 33
1.5 Primena projektnih uzoraka ... 40
2 O jeziku UML ... *43*
2.1 Model i modelovanje ... 44
2.2 Notacija za modelovanje ... 46
2.3 Istorija jezika UML ... 49
2.4 Definicija jezika UML ... 50
2.5 Notacija za opis sintakse jezika UML ... 51
2.6 Korisnici jezika UML ... 53
2.7 Pregled jezika UML ... 55
2.7.1 Stvari ... 56
2.7.1.1 Stvari strukture ... 56
2.7.1.2 Stvari ponašanja ... 62
2.7.1.3 Stvari grupisanja ... 62
2.7.1.4 Stvari anotacije ... 63
2.7.2 Relacije ... 64
2.7.2.1 Zavisnost ... 64
2.7.2.2 Asocijacija ... 64
2.7.2.3 Generalizacija ... 65
2.7.2.4 Realizacija ... 65
2.7.3 Dijagrami ... 66
2.7.3.1 Vrste dijagrama ... 66
2.7.3.2 Klasifikacija dijagrama ... 69
2.7.4 Opšte podele ... 72
3 Opšti pojmovi ... *75*
3.1 Ključne reči ... 75
3.2 Ugrađeni primitivni tipovi ... 77

3.3 Identifikatori ... 78
3.4 Pravila jezika ... 79
3.5 Dijagrami ... 81
3.5.1 Notacija dijagrama ... 81
3.6 Mehanizmi za proširivanje jezika ... 82
3.6.1 Stereotipi ... 83
3.6.2 Oznake ... 83
3.6.3 Ograničenja ... 84
3.6.4 Profili ... 84
3.7 Metamodel ... 85
3.7.1 Metaklasa ... 85
3.7.2 Klasifikator ... 86
3.7.2.1 Notacija klasifikatora ... 88
4 Dijagrami klasa ... 91
4.1 Klasa ... 92
4.1.1 Simbol klase ... 93
4.1.2 Osobine klase ... 95
4.1.3 Vidljivost odlika klase ... 96
4.1.4 Atributi ... 97
4.1.4.1 Sintaksa atributa ... 97
4.1.4.2 Multiplikativnost ... 98
4.1.4.3 Osobine atributa ... 99
4.1.5 Operacije ... 101
4.1.5.1 Sintaksa operacije ... 102
4.1.5.2 Osobine operacije ... 103
4.1.6 Tip podatka ... 105
4.1.7 Interfejs ... 107
4.1.8 Aktivna klasa ... 109
4.1.9 Signali i prijem signala ... 112

4.1.10 Šabloni ... 113
4.1.10.1 Šablonska klasa ... 113
4.1.10.2 Šablonska operacija ... 118
4.1.11 Izuzeci ... 118
4.1.12 Snažni tip ... 120
4.1.13 Standardni stereotipi klase i klasifikatora ... 121
4.2 Relacije ... 122
4.2.1 Zavisnost ... 123
4.2.1.1 Posebne kategorije zavisnosti ... 123
4.2.1.2 Standardni stereotipi zavisnosti ... 124
4.2.2 Asocijacija ... 126
4.2.2.1 Naziv asocijacije ... 127
4.2.2.2 Uloga kraja asocijacije ... 128
4.2.2.3 Multiplikativnost kraja asocijacije ... 128
4.2.2.4 Vidljivost kraja asocijacije ... 129
4.2.2.5 Navigabilnost asocijacije ... 131
4.2.2.6 Sadržanje ... 132
4.2.2.7 Vlasništvo kraja asocijacije ... 135
4.2.2.8 Izvedena uloga kraja asocijacije ... 136
4.2.2.9 Osobine kraja asocijacije ... 137
4.2.2.10 Klasa asocijacije ... 139
4.2.2.11 N-arna asocijacija ... 140
4.2.2.12 Kvalifikator kraja asocijacije ... 143
4.2.2.13 Specifikator interfejsa ... 145
4.2.3 Generalizacija ... 147
4.2.3.1 Generalizacioni skupovi ... 148
4.2.3.2 Tip generalizacionog skupa ... 151
4.2.4 Realizacija ... 152

4.2.5 Ugnežđenje ... 152
4.3 Saradnja ... 153
4.3.1 Šablonska saradnja ... 154
5 Dijagrami paketa ... 157
5.1 Paket ... 157
5.2 Relacije između paketa ... 160
5.2.1 Zavisnost ... 160
5.2.2 Uvoz ... 161
5.2.3 Spajanje ... 165
5.3 Principi modelovanja paketa ... 167
5.4 Posebne organizacione jedinice ... 168
6 Dijagrami objekata ... 171
6.1 Objekat ... 172
6.2 Veza ... 175
6.3 Primena dijagrama objekata ... 177
7 Dijagrami slučajeva korišćenja ... 179
7.1 Slučaj korišćenja ... 180
7.2 Scenario ... 183
7.3 Akter ... 184
7.4 Relacije ... 184
7.4.1 Komunikacija ... 185
7.4.2 Uključivanje ... 186
7.4.3 Proširivanje ... 187
7.4.4 Generalizacija ... 189
7.5 Okvir subjekta ... 190
8 Dijagrami složene strukture ... 193
8.1 Strukturirane klase i komponente ... 194
8.1.1 Deo i uloga ... 194

8.1.2 Port ... 195
8.1.3 Ponuđeni i zahtevani interfejs ... 197
8.1.4 Veznici ... 198
8.1.5 Multiplikativnost ... 200
8.1.6 Primeri klasa sa unutrašnjom strukturom ... 201
8.2 Saradnja sa unutrašnjom strukturom ... 202
8.2.1 Korišćenje saradnje ... 204
8.2.2 Složena saradnja ... 205
8.2.3 Primer složene saradnje ... 206
8.2.4 Predstavljanje projektnih uzoraka ... 208
9 Dijagrami komponenata ... 211
9.1 Komponenta ... 212
9.1.1 Ponuđeni i zahtevani interfejs ... 213
9.1.2 Odeljci komponente ... 214
9.1.3 Komponenta kao crna ili bela kutija ... 215
9.1.4 Zamenljivost i reupotrebljivost ... 215
9.1.5 Stereotipi komponente ... 215
9.2 Relacije ... 217
9.3 Složene komponente ... 219
9.4 Artefakti ... 220
9.4.1 Vrste artefakata prema načinu nastanka ... 221
9.4.2 Zavisnosti artefakata ... 222
9.4.3 Standardni stereotipi artefakata ... 222
9.5 Paketi ... 223
10 Dijagrami raspoređivanja ... 225
10.1 Čvor ... 226
10.2 Komunikaciona putanja ... 228
10.3 Relacija raspoređivanja ... 229

10.4 Specifikacija raspoređivanja....232
11 Mehanizmi proširivosti jezika UML....235
11.1 Stereotipi....236
11.2 Oznake....238
11.3 Ograničenja....239
11.4 Profili....240
12 Literatura....243