

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Savjeti za oblikovanje studentskih radova
(radna verzija, 9.3.2010.)

Zagreb, ožujak 2010.

Sadržaj

Predgovor	1
1. Osnovni predložak strukture rada	2
1.1. Oblikovanje dokumenta	5
1.2. Česte pogreške u pisanju	8
1.2.1. Gramatika, pravopis i stil	8
1.2.2. Navođenje izvora literature i poštivanje autorstva	9
1.2.3. Organizacija i podjela sadržaja dokumenta	9
1.3. Vremensko planiranje	11

Predgovor

Neovisno o zadanoj temi koju obrađuju, određene kategorije tehničkih dokumenata kao što su seminarski rad, završni rad, diplomski rad, magistarski rad pa i doktorska disertacija, kao i znanstveni radovi koji se objavljuju na konferencijama i u časopisima uvijek imaju vrlo sličnu strukturu.

Ovaj dokument namijenjen je prvenstveno studentima preddiplomskog i diplomskog studija Fakulteta elektrotehnike i računarstva, jer je i nastao na temelju iskustava skupljenih kroz godine mentorskog rada i iskustva, kako osobnih, tako i onih zajedničkih, rado i manje-rado preporučanih “FER-ovskih legendi”. Zato se dokument, govoreći o studentskim radovima, uglavnom referencira na “diplomski rad” ili “završni rad”. Naravno, veselit će nas ako se pokaže korisnim i drugima.

Motivacija je, dakle, krenula od studenata. Za mnoge je taj diplomski (ili, po FER-2, završni) rad prvi “ozbiljni” tehnički dokument te veličine koji moraju napisati samostalno i “od nule”. Susrevši se s time, ispada iznenađujuće koliko vremena i truda pisanje zahtijeva te koliko je ponekad teško sadržaj rada, i onda kad je sam zadatak već uspješno riješen, smisleno i logičkim redoslijedom složiti u cjelinu. Cilj je ovog dokumenta olakšati studentima taj posao, s jednostavnim i – nadamo se – lako primjenjivim savjetima o organizaciji sadržaja koji čini pisani rad, kao i nekim najčešćim pogreškama koje treba izbjegavati. Na kraju su se tu uvukli i neki savjeti o vremenskom planiranju aktivnosti vezanih uz pisanje, bilo bi lijepo vidjeti i njihovu primjenu u praksi.

Ovaj dokument nije zamjena za komunikaciju s mentorom, niti za upute i pravila koja objavljuje Fakultet.

Prof. dr. sc. Maja Matijašević

1. Osnovni predložak strukture rada

Za potrebe ovog izlaganja, studentski radovi se općenito mogu podijeliti u dvije skupine, na one kod kojih je problem, odnosno zadatak, prvenstveno “sinteza” (1) i one kod kojih je problem prvenstveno “analiza” (2). Svaki problem, naravno, može uključivati elemente i jednog i drugog, ali je osnovnu “orijentaciju” lako uočiti već iz formulacije zadatka.

Evo nekoliko ilustrativnih primjera zadataka:

Primjer diplomskog zadatka tipa (1) – oblikovanje programske podrške – rezultat: MojRaspored!

Naslov rada: Web-aplikacija za rezervacijsku uslugu

Opis zadatka: Troslojne arhitekture aplikacija na Webu proširuju osnovni model klijent-poslužitelj uvođenjem odvojenog podatkovnog poslužitelja. Takve aplikacije posebno su pogodne za usluge sa složenim rukovanjem s podacima te onima s velikim brojem istovremenih zahtjeva, kao npr. kod rezervacijskih usluga.

Vaš zadatak je proučiti arhitekturu i performanse troslojnih aplikacija na Webu i razraditi model aplikacije za pružanje rezervacijskih usluga termina laboratorijskih vježbi. Osmislite podatkovni model i uvedite kontrolu pristupa zasnovanu na ulogama. Izradite korisničko i administracijsko sučelje pomoću dinamičkih Web stranica. Pri izradi programskog rješenja koristite skriptni jezik PHP i sustav upravljanja bazom podataka MySQL.

Primjer diplomskog zadatka tipa (1) – nadogradnja zadanog open source programskog alata

Naslov rada: Ostvarivanje višemedijske sjednice u sustavu IMS uz dodavanje podatkovne medijske komponente

Opis zadatka: U mreži treće generacije zasnovanoj na sustavu IMS (engl. IP Multimedia Subsystem) upravljanje višemedijskom sjednicom provodi se pomoću protokola Session Initiation Protocol (SIP).

Vaš zadatak je proučiti sustav IMS i protokol SIP, osnovne scenarije upravljanja višemedijskom sjednicom putem SIP-a i programski alat IMS Communicator. Osmislite i razradite način ostvarivanja sjednice koja, uz audio i video koji su uključeni u IMS Communicator, kao dodatni medij može uključiti nepomičnu sliku ili podatke. Programski izvedite proširenje alata IMS Communicator kroz biblioteku funkcija za takvu podatkovnu komponentu. Rješenje demonstrirajte na odabranim primjerima.

Primjer diplomskog zadatka tipa (2) – programska simulacija i analiza protokola pomoću alata ns-2

Naslov rada: Analiza protokola usmjeravanja u ad-hoc mreži

Opis zadatka: Ad-hoc mreža je dinamički organizirana mreža autonomnih pokretnih čvorova koji međusobno komuniciraju bez pomoći stacionarne infrastrukture. Za primjenu u ad-hoc mrežama razvijeni su brojni protokoli usmjeravanja. Jedna od mogućih klasifikacija dijeli ih na proaktivne i reaktivne, ovisno o načinu osvježavanja staza usmjeravanja. Poznati čimbenici koji utječu na pogodnost određenog protokola za zadanu primjenu su broj i pokretljivost čvorova u mreži, te opterećenje mreže.

Vaš zadatak je proučiti i opisati način komuniciranja u ad-hoc okruženju općenito, te posebno opisati protokol usmjeravanja Destination-Sequenced Distance Vector (DSDV), kao primjer proaktivnog, i Ad-hoc On Demand Distance Vector (AODV), kao primjer reaktivnog protokola usmjeravanja. Pomoću simulacijskog alata ns-2 ispitajte odabrana svojstva tih dvaju protokola s obzirom na broj čvorova u mreži, njihovu pokretljivost i opterećenje mreže, te ih usporedite međusobno i s rezultatima u literaturi.

Primjer diplomskog zadatka tipa (2) – mjerenje i analiza rezultata u laboratorijskom prototipu

Naslov rada: Vrednovanje performanci SIP-signalizacije u laboratorijskom prototipu

Opis zadatka: U mreži treće generacije zasnovanoj na sustavu IMS (engl. IP Multimedia Subsystem) uvodi se potpora višemedijskim uslugama i upravljanje kvalitetom usluge, pri čemu se parametri usluge prenose signalizacijom s kraja na kraj pomoću protokola Session Initiation Protocol (SIP).

Vaš zadatak je proučiti u literaturi opisane metrike za vrednovanje performanci SIP-signalizacije s kraja na kraj te laboratorijski prototip napredne višemedijske usluge virtualnog okruženja koji za signalizaciju koristi protokol SIP. Posebno proučite alate za automatizirano generiranje SIP-zahtjeva kojima bi se mogla simulirati okolina s većim brojem aktivnih korisnika. Osmislite i pomoću odabranih alata provedite mjerenja u laboratorijskoj mreži na prototipu te obradite i komentirajte rezultate.

Napomena: Svaki mentor ima svoj stil pisanja zadatka, tako da navedene primjere nipošto ne treba shvatiti kao pravilo. Oblik zadatka koji zadaje mentor je slobodan, ali zbog ograničenog prostora u propisanom obrascu mora biti sažet i pokriti ono najvažnije: *što* student treba napraviti i osnovne smjernice o tome *kako* to treba napraviti. Dobra je komunikacija s mentorom ključ za razumijevanje i rješavanje zadanog problema, neovisno o pisanoj formulaciji zadatka.

Iz navedenih primjera može se uočiti glavne karakteristike po vrstama zadataka:

- (1) Oblikovanje i razvoj programske podrške – glavni je cilj rada dobiti programski proizvod ili komponentu. Može se raditi o novom, samostalnom rješenju nekog problema (na primjer, programska izvedba algoritma, usluge ili njene

komponente, korisničko sučelje i sl.) ili proširenju postojećeg programskog rješenja dodatnom funkcionalnošću (na primjer, grafičkim sučeljem). Često se prilikom programske izvedbe koriste i gotovi elementi, na primjer, baza podataka, web-poslužitelj, web-klijent, biblioteke funkcija, i drugo. Naglasak je na sintezi, dakle, samostalnom oblikovanju i razvoju softvera, od opisa i analize zahtjeva preko formalnog modela do izvedbe. Vrednovanje rješenja provodi se prema zahtjevima postavljenima na početku rada, a demonstracija rezultata sastoji se u izvođenju odabranih primjera, odnosno, slučajeva uporabe.

- (2) Primjena metodologije na problem – glavni je cilj dobiti (nova) saznanja o odabranim svojstvima zadanog predmeta ispitivanja, pri čemu on može biti stvaran (uređaj, mreža, komunikacijski protokol, usluga, ...), ili pak izveden kao simulacija ili emulacija, koja se rabi kad je stvarni sustav nedostupan, presložen ili preskup da bi se na njemu eksperimentiralo. (Razlika između simulacije i emulacije je u načinu na koji se postiže ponašanje sustava: kod simulacije se ono temelji na matematičkom modelu (fizikalnog sustava, prometa u mreži, i sl.), dok je kod emulacije najvažnije uspješno oponašanje vanjskog ponašanja ili odlika sustava). Zadani sustav može biti, na primjer, laboratorijska mreža, korisnička aplikacija, programski alat, uređaj, laboratorijski prototip i slično. Odabrana svojstva ispitivanja ovise o sustavu: na primjer, može nas zanimati kašnjenje, propusnost, gubici (za mrežu), opterećenje CPU-a, prostor na disku (za aplikaciju), subjektivna procjena (za kvalitetu zvuka i slike), jednostavnost uporabe, korisničko sučelje (za programski alat), itd. Dobivene rezultate treba obraditi i samostalno interpretirati. Demonstracija rezultata može uključivati izvođenje odabranog scenarija simulacije ili mjerenja uživo (pod pretpostavkom da je izvedivo s obzirom na uvjete demonstracije).

Kod usporedne analize, kao što sam naziv kaže, analizira se više sustava uz primjenu zadanih kriterija. Zadatak može uključivati i samostalan odabir ili definiranje kriterija usporedbe. Cilj usporedbe obično je odabrati postojeće rješenje koje se najbolje može primijeniti za zadani problem pa se usporedna analiza često zadaje kao tema seminarskog rada ili projekta i prethodi kasnijoj primjeni odabranog rješenja kroz završni rad (iz bilo koje skupine).

1.1. Oblikovanje dokumenta

Najlakše je početi od osnovnog predloška ili “kostura” dokumenta, koji se onda postupno popunjava. Predložak možemo shvatiti kao okvirni sadržaj rada, čiji su (sadržajno bitni) dijelovi sljedeći:

1. Uvod
2. Pregled literature
3. Opis formalnog modela i metodologije
4. Opis konkretne primjene modela ili metodologije
5. Rezultati i diskusija
6. Zaključak
7. Literatura

(Dodaci ili Prilozi, po potrebi)

Poglavlja se, osim uvoda i zaključka, najčešće ne zovu ovako “generički”, već im se daju smisleni(ji) naslovi. Također, ponekad je zgodno pregled literature razdvojiti na dva (rijetko i više) poglavlja. Na primjer, ako je tema rada “Pregled programske podrške za internetske usluge u IPv6 mreži”, onda će vjerojatno biti zgodno da se “pregled literature” razdvoji na dva poglavlja: jedno koje će obraditi protokol IPv6, a drugo internetske usluge općenito, te da će se svakom dati odgovarajući naslov (npr. “Osnove protokola IPv6”, “Internetske usluge”).

Osim navedenih dijelova, rad može sadržavati i druge opcionalne ili propisane elemente, kao što su, na primjer, zahvala, popis slika, popis tablica, popis kratica, popis ključnih riječi, sažetak, rječnik pojmova i drugo. Tim elementima se u ovom izlaganju nećemo baviti, jer su opcionalni elementi pitanje autorovog izbora, a propisani jednostavno obveza koju treba ispuniti.

Potreba za dodacima (odn. priložima) obično proizlazi iz praktičnih razloga, radi bolje preglednosti. Primjeri sadržaja koji se često stavlja u dodatak su: izvorni kod, dugački ispisi (npr. algoritama, rezultata izvođenja programa, rezultata mjerenja), upute za instalaciju, konfiguraciju i pokretanje softvera te, općenito, sve što je bolje imati ili sažeti na jednom mjestu, ili pak ono što je preopsežno da bi se uključilo u glavni tekst dokumenta. Dodatke koji svojom veličinom graniče s duljinom glavnog dokumenta

najbolje je pohraniti u elektroničkom obliku na CD-ROM na kojem se arhivira rad, ili ih uvezati u samostalni dokument, odvojeno od samog rada.

Pogledajmo detaljnije što – sadržajno – treba biti gdje, po dijelovima teksta.

<p>Uvod</p> <ul style="list-style-type: none"> • što je zadatak (problem) • zašto ga rješavamo (motivacija) • kako ga rješavamo (pristup) • opis organizacije ostatka dokumenta (najčešće, po poglavljima) • za ovo obično ne treba više od 1 stranice 	
<p>Pregled literature</p> <ul style="list-style-type: none"> • sažeti pregled područja koje se u radu obrađuje – poznato znanje i iskustva (što su napravili “drugi”) na temelju literature: <ul style="list-style-type: none"> ○ članci ○ knjige ○ drugi dokumenti (<i>technical report, whitepaper</i>) ○ razna izdanja normizacijskih tijela (RFC-ovi, ITU-T preporuke,...) ○ on-line izvori ○ nastavni materijali ○ prethodni studentski radovi 	
<p>Model</p> <ul style="list-style-type: none"> • ako se radi o proširenju postojećeg sustava, njegov opis • specifikacija zahtjeva (željena funkcionalnost budućeg sustava, ako se radi o novom sustavu ili proširenja, ako se radi o postojećem) • prijedlog i razrada vlastitog rješenja <ul style="list-style-type: none"> ○ neformalni opis (tekst, skice) ○ formalni opis (npr. UML dijagram, relacijski model baze podataka) 	<p>Metodologija</p> <ul style="list-style-type: none"> • opis postupaka (mjerenje, analiza, simulacija, emulacija) i hipoteza • važno je dobro objasniti što se radi/mjeri/promatra (koji parametar/i, koji kriterij), i kako se to radi (način, učestalost, i sl.) • ako se radi usporedba: opis kriterija usporedbe
<p>Programska izvedba</p> <ul style="list-style-type: none"> • smjernice za programsku izvedbu: vlastiti razvoj, integracija s postojećim rješenjima (naziv, verzija, opis) • programski alati (naziv, verzija, opis) • opis laboratorijske infrastrukture (računala, softver, mreža), instalacije i konfiguracije * • opis vlastitog programskog rješenja 	<p>Primjena metodologije / postupka</p> <ul style="list-style-type: none"> • opis mjerne opreme ili softvera • opis laboratorijske okoline • opis provođenja postupka u zadanoj okolini, primjena na problem • primjena opisanih kriterija na zadani problem • rezultati (simulacije, emulacije, mjerenja, analize, ...) • ako se radi usporedba: sažeti prikaz rezultata
<p>Rezultati i diskusija</p>	<p>Rezultati i diskusija</p>

<ul style="list-style-type: none"> • scenariji izvođenja programa, prema predviđenim slučajevima uporabe • komentar svakog slučaja • završni osvrt na početno postavljene zahtjeve i diskusija 	<ul style="list-style-type: none"> • obrada rezultata mjerenja (tablice, grafovi, statistička obrada) • interpretacija rezultata i diskusija
<p>Zaključak</p> <ul style="list-style-type: none"> • sažeti prikaz napravljenog <u>s naglaskom na (odabrani) rezultat</u> • za ovo obično ne treba više od 1 stranice • u diplomskom se radu student vlastoručno potpisuje ispod zaključka 	
<p>Literatura</p> <ul style="list-style-type: none"> • svi izvori trebaju biti pobrojani • za svaki izvor treba navesti dovoljno podataka da je a) jasno o čemu se radi, i b) da zainteresirani čitatelj može doći do tog izvora 	
<p>Dodaci</p> <ul style="list-style-type: none"> • mjesto gdje se, po potrebi, mogu dodati dugački ispisi, npr. ispisi ulaznih, izlaznih ili konfiguracijskih datoteka, slike s ekrana i sl. koje nisu neophodne za razumijevanje samog rada, ali dodatno pojašnjavaju ili dokumentiraju napravljeno, upute za instalaciju i korištenje softvera i slično 	

U vremenskom nizu, obično se prvo piše pregled literature. To je prilika da se sistematizira i sažeto pokaže znanje stečeno njenim proučavanjem. Uz pisanje pregleda literature, naravno, ide i briga o poglavlju Literatura, gdje treba pažljivo navesti sve izvore korištene u tekstu i podatke o njima. Nije dobro ostavljati taj posao za kraj, jer se do tada lako zaborave i pomiješaju izvori, odnosno reference.

Sljedeća tri dijela, dakle, Model – Programska izvedba – Rezultati i diskusija, odnosno Metodologija - Primjena metodologije / postupka – Rezultati i diskusija, obično se pišu tim redosljedom, ali u nekoliko iteracija. S obzirom na to da se pisanje i stvarni rad na rješavanju zadanog problema preklapanju, logično je da se u tom procesu pojavljuju nova saznanja, iskustva i rezultati koje treba ugraditi u rad. S vremenom se poglavlja “stabiliziraju”, opet istim redosljedom kako su navedena, tako da pri kraju preostaje samo opisati rezultate i osvrt ili diskusiju.

Uvod i zaključak najlakše je napisati na kraju, kad je posao i pisanje privedeno kraju i kada je jasno što čini sadržaj rada. Često se zaboravlja ono ključno – da je smisao poglavlja Uvod – uvod *u rad*, a Zaključka – zaključak *rada*. Uvod “uvodi” čitatelja u problem, i daje

mu naznake o onome što slijedi. Zaključak sažeto opisuje ono što je napravljeno. U zaključku nema novih informacija!

Najčešća pogreška u uvodu je preopćenit, a često i predugačak tekst iz kojeg nije jasno što je zadatak i kako je riješen. Uvod mora biti kratak i jasan – treba opisati problem, što se dobiva njegovim rješenjem i kako se pristupilo problemu te ukratko opisati što slijedi, odnosno kako je organiziran dokument (najjednostavnije je po poglavljima). U zaključku je najčešća greška da se ne komentira svoj rad i rezultate, da se otvaraju novi problemi umjesto da se opiše rješenje onog koji je (bio) zadan, da se posvećuje pažnja temama izvan rada kao takvog (“svijetla budućnost (telekomunikacija, tehnologije X, Interneta sl.).

1.2. Česte pogreške u pisanju

1.2.1. Gramatika, pravopis i stil

Uz postojeće alate za provjeru ispravnosti teksta, nema isprike za pogrešno pisanje i tipkarske pogreške koje se mogu ispraviti strojnom provjerom. Hrvatski akademski spelling checker – Hascheck (<http://hascheck.tel.fer.hr/>) “djeluje kao javna i besplatna usluga pravopisne provjere teksta pisanog hrvatskim jezikom”. Posebno se preporučuje i primjena Pravopisnog priručnika docenta dr. sc. Gordana Gledeca s odabranim primjerima (i dobrog i lošeg pisanja).

Stil pisanja studentskog rada treba biti izvještajan – neutralan, činjeničan, argumentiran. Neke od čestih pogrešaka su: nekritičko preuzimanje informacija (npr. “Ovaj moćni softver rješava sve sigurnosne probleme u vašoj mreži...”, “najbolji na tržištu...”), tvrdnje bez argumenata (“najpopularniji alat”, “najbolji alat”, i sl. – može, ali uz pouzdanu referencu!), kao i maskiranje činjenica (npr. “...rezultati nisu baš najbolji..”, ako se zapravo misli “rezultati su loši”, ili, “nema rezultata”). U opisima i izvještavanju najjednostavnije je pisati neutralno i bez obraćanja čitatelju. Na primjer, dobro je: “Format datoteke može se odabrati putem izbornika.”, ili “Korisnik putem izbornika bira format datoteke.”, a nije dobro: “U izborniku možete odabrati format spremljene datoteke.” (stil uputa i priručnika) ili “U izborniku možemo odabrati format spremljene datoteke”. (“udžbenički stil”, dobar na predavanju ili u udžbeniku, ali ne i za ovakav rad).

Posebna kategorija (vrlo iritantnih!) pogrešaka tiču se prijevoda. Osnovno pravilo je: kad se služimo literaturom na engleskom ili drugom stranom jeziku, treba paziti da prijevod

bude suvisao (redosljed riječi, subjekt-objekt, itd.), dosljedan i u duhu hrvatskog jezika. Poznati pojmovi su uglavnom već prevedeni u materijalima s predavanja i udžbenicima autora s FER-a. Za manje poznate izraze, kao i za one oko kojih (na žalost, već “tradicionalno”) postoje nesuglasice oko toga koji je izraz bolji, najsigurnije je potražiti savjet mentora.

1.2.2. Navođenje izvora literature i poštivanje autorstva

Služenje literaturom (neovisno o obliku (tiskani medij, on-line, audio/video i dr.) i jeziku) nipošto se ne smije svesti na prepisivanje ili doslovno prevođenje tuđeg teksta. Izvadak iz teksta, preuzeta definicija ili izravan prijevod označava se citiranjem, odnosno, ako se radi o većem dijelu teksta (npr. specifikacije) koja se preuzima u izvornom obliku, to se mora eksplicitno navesti, uz referenciranje na izvorni dokument u popisu literature. To vrijedi za svu literaturu, kako na hrvatskom, tako i na engleskom. U pravilu, tuđi tekst uvijek treba prepričati vlastitim stilom i riječima uz navođenje reference; inače se tekst može smatrati plagijatom i kršenjem autorskih prava stvarnog autora (sjetimo se da završni, odnosno diplomski, rad potpisuje student kao autor!, tako da se tekst smatra autorskim djelom ako nije drugačije navedeno). Slično vrijedi i za slikovne prikaze. Za svaku preuzetu sliku treba navesti izvor, a za one koje su zaštićene treba navesti izvor i dopuštenje za reprodukciju. Za slike koje su očigledno u javnoj domeni, ili se ne zna tko im je autor, uobičajeno je samo referencirati se na izvor, dok je generičke slike najbolje nacrtati na svoj način. Ako postoje anotacije na slici, one u pravilu, moraju biti na jeziku na kojem je napisan rad. Neki izuzeci od obveze prevođenja su “snapshoti” s ekrana, prikazi izvođenja programa čije je sučelje na stranom jeziku, prikazi za koje je jasno da su preuzeti iz poznatog izvora na stranom jeziku (npr., specifikacija, norma i sl. što nema smisla prevoditi).

1.2.3. Organizacija i podjela sadržaja dokumenta

Na što treba paziti prilikom oblikovanja dokumenta? Česta pogreška je neusklađenost razine detalja i veličine prostora posvećenog pregledu literature (dakle, onog što su napravili drugi) i onog gdje se opisuje vlastiti rad – na štetu ovog drugog! Minimalni vlastiti udio ne smije biti manji od 50% sadržaja, neovisno o ukupnoj duljini dokumenta.

Druga česta pogreška je miješanje formalnog opisa i izvedbe, odnosno, metodologije i rezultata. (Ta pogreška je vjerojatno “inspirirana” predavanjima i primjerima u udžbenicima i priručnicima, gdje se uz teoriju paralelno provlači pokazni primjer. Treba

voditi računa da su ti i takvi materijali namijenjeni drugoj svrsi i nisu dobri kao predlošci za pisanje završnog rada!) U završnom radu treba pažljivo odvojiti dio koji je neovisan o tehnologiji (model, odnosno metodologiju) od onog ovisnog o tehnologiji (programsku izvedbu, odnosno primjenu metodologije na problem, recimo, provedbu mjerenja ili analize). To razdvajanje treba biti jasno i u tekstu – u dva odvojena poglavlja, kako je opisano ranije.

Na primjer, ako se radi o zadatku oblikovanja i razvoja programske podrške, onda formalni opis modela ne ovisi (niti smije ovisiti!) o tome hoće li programsko rješenje biti izvedeno primjenom programskog jezika C++ ili Java. Dakle, poglavlje koje opisuje model programske podrške ide u jedno poglavlje, a opis izvedbe u zadanom programskom jeziku u drugo.

Važno je uočiti da je formalni model tehnološki neutralan, iako *izbor* modela to ne mora biti – naime, jasno je da će pametan odabir modela ovisiti o tome što se modelira – proces, baza podataka, VLSI sklop – te da će tome biti što je moguće bolje prilagođen. U prethodnom primjeru, za model programske podrške za koju znamo da će biti izvedena nekim objektno orijentiranim programskim jezikom (C++ ili Java), vjerojatno bi dobar izbor bio Unified Modeling Language (UML). Za neke specifične primjene možda će postojati i bolje rješenje, ali time se ne treba opterećivati - izbor modela ili metodologije obično će biti zadan ili će smjernice doći od mentora.

Dobar test razdvojenosti modela (metodologije) od izvedbe (programske ili fizičke) je da se upitamo: može li se na temelju ovog modela (metodologije) napraviti programsko rješenje s pomoću drugog jezika (ili mjerenje s pomoću drugog programskog alata ili instrumenta). Ako je odgovor pozitivan, onda je uvjet zadovoljen.

Tablice 1 i 2 daju primjere za obje vrste rada.

Tablica 1 – Primjer: Oblikovanje i razvoj programske podrške

*	Uvod	*
Neovisno o tehnologiji	Pregled literature	Tuđi rezultati (< 50%)
	Model	Vlastiti rezultati (>50%)
Ovisno o tehnologiji	Programska izvedba	
	Rezultati i diskusija	
*	Zaključak	*
*	Literatura	*

Tablica 2 – Primjer: Analiza (odabranih) performansi sustava

*	Uvod	*
Neovisno o tehnologiji	Pregled literature	Tuđi rezultati (< 50%)
	Metodologija mjerenja	
Ovisno o tehnologiji	Provedba mjerenja	Vlastiti rezultati (>50%)
	Rezultati i diskusija	
*	Zaključak	*
*	Literatura	*

1.3. Vremensko planiranje

Kod izrade diplomskog (završnog) rada, jako je važno započeti posao dobrom pripremom, i nastojati realno procijeniti vrijeme potrebno za različite faze posla, kao i završno pisanje i doradu rada.

Pripremna faza, proučavanje literature i razrada zahtjeva su izuzetno važni. Studenti se ponekad žure što prije započeti s programskim rješenjem ili mjerenjem (tzv. “praktičnim” dijelom rada), jer tako imaju osjećaj da se rad pomaknuo s mjesta, no od toga nema koristi ako se “trči u pogrešnom smjeru”. Početak mjerenja bez pregleda onoga što će se i kako mjeriti siguran je način da se dovede u situaciju da će mjerenje trebati ponavljati ili, još gore, raditi ispočetka. Početak razvoja softvera bez prethodne razrade zahtjeva ima za

posljedicu naknadne popravke i dopune koje je teško unijeti bez narušavanja postojećeg koda. Isplati se “potrošiti” par dana ili – ako treba – i tjedan više u početku na dobru pripremu, nego kasnije dane i tjedne na preinake i popravke. Rok od tri mjeseca za izradu diplomskog ili završnog rada nije dovoljan za višestruke pokušaje! Tablica 3 daje prijedlog okvirnog vremenskog rasporeda izrade rada kroz semestar.

Tablica 3 – Predloženi vremenski raspored

Vremenski raspored	Završni diplomski rad	
Tjedan 1	Zadavanje teme / preuzimanje zadatka	
Tjedan 2	Izrada diplomskog rada	Proučavanje/pregled literature (<i>vlastite bilješke, osobni plan rada</i>)
Tjedan 3		
Tjedan 4		
Tjedan 5		
Tjedan 6		
Tjedan 7		Model / metodologija (<i>vlastite bilješke, početni elektronički dokument</i>)
Tjedan 8		
Tjedan 9		Programska izvedba predloženog modela / primjena metodologije na zadani problem (<i>vlastite bilješke, dijelovi poglavlja o izvedbi i rezultatima, natuknice o rezultatima</i>)
Tjedan 10		
Tjedan 11		
Tjedan 12		
Tjedan 13		
Tjedan 14		Obrada rezultata & diskusija (<i>prva verzija rada</i>)
Tjedan 15	Dopune, ispravci, finalizacija (<i>konačna verzija rada</i>)	
Tjedan 16	<i>Predaja konačne verzije rada</i>	
Tjedan 17-18	Obrana	Priprema demonstracije i obrane (slajdovi)
Tjedan 19-20		Obrana

Tijekom rada jako je važno kontinuirano raditi vlastite bilješke. Bilježnica ili mapa “samo za to” pomaže i kao indikator vlastitog napredovanja. Osim što su osobne bilješke neprocjenjive kao podsjetnik, tijekom pisanja neke stvari i podsvjesno “sjednu na mjesto”. Dobro je ključne bilješke ubaciti u elektronički dokument i ne ostavljati unos teksta za sam kraj. Treba voditi računa (i) o tome da praktični rad mora biti na vrijeme gotov da bi se u pisani rad mogao unijeti detaljni opis, od ideje do izvedbe i rezultata, npr. *snapshots* korisničkog sučelja, rezultati mjerenja itd. – sve što spada u onaj drugi, “vlastiti” dio rada – kao i činjenici da mentor treba pregledati i odobriti rad prije predaje.

Konačno, u samoj završnici važno je predvidjeti dovoljno vremena za konačnu doradu nakon što se dobije povratna informacija od mentora te ispis i uvezivanje rada.