

2. Analiza i specifikacija zahtjeva

2.2. Smjernice za pisanje zahtjeva

Sažetak

Dobro strukturiran dokument specifikacije zahtjeva olakšava nam organizaciju, pisanje, provjeru i razumijevanje skupa zahtjeva za softversko rješenje. Međutim, ukoliko su pojedinačni zahtjevi napisani loše, dobra struktura neće garantirati da će zahtjevi biti ispravno interpretirani, a samim tim niti ispravno realizirani u aktivnostima dizajna, implementacije i testiranja. U ovoj lekciji se bavimo temom pisanja kvalitetnih pojedinačnih zahtjeva i skupa zahtjeva uz pomoć široko prihvaćenih INCOSE smjernica ¹. Smjernice definiraju skup karakteristika koje pojedinačnih zahtjevi i skupovi zahtjeva trebaju posjedovati, te skup pravila koje trebamo slijediti kako bismo postigli navedene karakteristike.

Uvod

Da bi specifikacija zahtjeva osigurala čvrst temelj za daljnje aktivnosti procesa razvoja softvera, dokument zahtjeva treba biti jasno strukturiran i lako razumljiv. Tako organiziran dokument nam omogućava da lako pronađemo zahtjeve, razumijemo njihov kontekst, uočimo propuste ili redundantne zahtjeve, eliminiramo konfliktne ili loše napisane zahtjeve, i sl. Međutim, još jedan ključan faktor kvalitete specifikacije zahtjeva je i kvaliteta pojedinačnih zahtjeva. Sama struktura dokumenta nam neće mnogo pomoći ukoliko pojedinačni zahtjevi nisu napisani tako da budu jasni, konzistentni, i precizni u izričaju. Toga su bili svjesni i autori predložaka za definiranje specifikacije zahtjeva (npr. Volere i IEEE 830-198 SRS), pa su osim za strukturu cjelokupnog dokumenta dali i općenite smjernice za oblikovanje pojedinačnih zahtjeva. Ipak, mnogo detaljnije i brojnije smjernice za oblikovanje pojedinačnih zahtjeva možemo naći u INCOSE smjernicama za pisanje zahtjeva, na koje se ova lekcija posebno fokusira.

Prije nego se usredotočimo na smjernice za pisanje zahtjeva, osvrnuti ćemo se na kriterije po kojima možemo zahtjeve razvrstavati. Jedan od tih kriterija je razina apstrakcije na kojoj je zahtjev pisan. U tom smislu možemo govoriti o nekoliko vrsta zahtjeva (poredanih od veće razine apstrakcije do manje):

¹ Incose.org

- Izjava o potrebi,
- **Dionički (korisnički) zahtjevi,**
- **Sustavski (softverski) zahtjevi,**
- Zahtjevi komponente sustava,
- Zahtjevi komponente podsustava.

S obzirom na razinu složenosti problemske domene i softverskog rješenja kojima ćemo se mi baviti, za naše potrebe će biti dostatno govoriti o korisničkim zahtjevima (tj. što korisnik želi postići korištenjem softvera), te softverskim zahtjevima (tj. što softver mora implementirati). Te dvije vrste zahtjeva smo spominjali i u prošloj lekciji kada smo govorili o specifikaciji korisničkih zahtjeva i specifikaciji softverskih zahtjeva.

Druga važna klasifikacija softverskih zahtjeva koju ćemo spomenuti, i ujedno klasifikacija na koju ćemo se usredotočiti tokom ove lekcije, uzima u obzir svrhu zahtjeva. U tom smislu razlikujemo funkcionalne zahtjeve i nefunkcionalne zahtjeve. Funkcionalni zahtjevi softvera definiraju što softver treba raditi, kako reagira na stimulanse iz okoline i interakciju s korisnikom. Takvi zahtjevi se ponekad nazivaju i mogućnostima (engl. capabilities) ili opcijama (engl. features). Koji će se funkcionalni zahtjevi naći u specifikaciji zahtjeva uvelike ovisi o problemskoj domeni kojom se softver bavi, vrsti softvera, korisnicima softvera i njihovim potrebama i očekivanjima, i sl.

S druge strane, nefunkcionalni zahtjevi softvera predstavljaju ograničenja koja vrijede za cijeli softver. Oni utječu na cjelokupnu arhitekturu sustava, a ne samo na individualne komponente. Zbog tog izrazitog utjecaja na kvalitetu softverskog rješenja, često se nazivaju i zahtjevima kvalitete, te mogu biti kritičniji od funkcionalnog zahtjeva. Naime, ukoliko propustimo identificirati, a samim tim i implementirati funkcionalni zahtjev, možemo ga uz razuman trud implementirati i naknadno. Međutim, s obzirom da potencijalno prožima cijelu arhitekturu softverskog rješenja, naknadna implementacija nefunkcionalnog zahtjeva je izrazito zahtjevnija, a ponekad i nemoguća unutar razumnih rokova i resursa. Nefunkcionalni zahtjevi se obično bave temama performansi, pouzdanosti, dostupnosti, prenosivosti, lakoće korištenja, sigurnosti, privatnosti, hardvera i sl. Iako je i za funkcionalne zahtjeve bitno da budu precizno izraženi, za nefunkcionalne zahtjeve je to još bitnije jer je za njih teže objektivno utvrditi jesu li ispunjeni ili ne. Zbog toga nefunkcionalni zahtjevi često sadrže i kvantitativnu komponentu koja će biti pokazatelj ispunjenosti zahtjeva (npr. broj transakcija u sekundi, prosječno vrijeme do greške, vrijeme odziva, megabajti memorije, broj istovremenih korisnika i sl.). Na taj način se ostavlja manje prostora za subjektivnu interpretaciju uspjeha ili neuspjeha u ispunjavanju zadanog zahtjeva.

U predlošku specifikacije softverskih zahtjeva koji smo započeli popunjavati u prethodnoj lekciji (poglavlja 1 i 2), svoje mjesto imaju i funkcionalni (poglavlje 3 predložka) i nefunkcionalnih zahtjevi (poglavlje 4). Nažalost, kao i kod većine aktivnosti koje spadaju u domenu kreativnog rada, ne postoji točno propisan niz koraka koji trebamo slijediti kako bismo imali zagarantirano ispravno formuliране zahtjeve. Ipak, smjernice kao što su INCOSE smjernice za pisanje zahtjeva nude heuristiku koja nam može u tome pomoći. Naravno, kao i svaka druga heuristika, i ove smjernice se trebaju oprezno i smisleno primjenjivati, a ne doslovno bez kritičke prosudbe.

Smjernice za pisanje zahtjeva

INCOSE smjernice su u praksi često korištene i u literaturi citirane smjernice za pisanje zahtjeva. Započinju s definiranjem 9 karakteristika (C1 – C9) koje bi svaki pojedinačni zahtjev trebao posjedovati da bismo ga smatrali ispravno napisanim. Osim njih, smjernice definiraju i 5 karakteristika (C10 – C14) koje treba posjedovati skup zahtjeva kako bi bio smatran ispravnim. Neovisno o tome radi li se o karakteristikama pojedinačnog zahtjeva ili skupa zahtjeva, svakoj karakteristici je pridruženo pravilo ili više pravila koje pomažu autorima zahtjeva da postignu propisane karakteristike. Postoji ukupno 41 pravilo (R1-R41), pri čemu svako pravilo može doprinositi postizanju jedne ili više karakteristika. S obzirom da su smjernice u originalu na engleskom jeziku, neka od pravila su specifična za engleski jezik i nisu mogla biti smisleno primijenjena na hrvatski jezik. Takva pravila su izostavljena (npr. R3 i R5).

Karakteristike zahtjeva

#	Karakteristika	Opis karakteristike	Pravila
C1	Potreba	Zahtjev treba definirati neophodnu mogućnost, karakteristiku, ograničenje ili faktor kvalitete. Kada bi ovaj zahtjev bio izostavljen iz skupa zahtjeva, njegov nedostatak ne bi bilo moguće nadomjestiti ostalim zahtjevima iz skupa.	R30
C2	Prikladnost	Specifična namjera i količina detalja navedenih u zahtjevu treba biti prikladna s obzirom na razinu apstrakcije na kojoj se u tom trenutku nalazimo.	R31
C3	Nedvosmislenost	Zahtjev treba biti napisan tako da se može interpretirati samo na jedan način, neovisno o tome tko ga interpretira.	R1-R37
C4	Kompletnost	Zahtjev opisuje neophodnu mogućnost, karakteristiku, ograničenje ili faktor kvalitete na način da pruža dovoljno informacija za razumijevanje, tj. nisu potrebne dodatne informacije kako bi se zahtjev razumio.	R6-R9, R18, R24, R25, R33, R34, R35, R39
C5	Singularnost	Zahtjev se treba odnositi na točno jednu mogućnost, karakteristiku, ograničenje ili faktor kvalitete.	R9, R18-R23, R39
C6	Izvedivost	Zahtjev treba biti izvediv s obzirom na postojeća ograničenja (npr. budžet, rok, oprema, legislativa, sigurnost i sl.) te uz prihvatljiv rizik.	R26, R33

C7	Provjerljivost	Zahtjev treba biti formuliran na način da se može provjeriti je li zahtjev ispunjen ili nije.	R1-R10, R15, R17, R18, R24, R26, R28, R32-R35
C8	Točnost	Zahtjev treba biti točna reprezentacija potrebe korisnika na temelju koje je napisan.	R6, R32, R33, R36
C9	Sukladnost	Zahtjev treba biti napisan u skladu sa odobrenim predloškom, stilom ili smjernicama za pisanje zahtjeva.	R12, R18, R30, R36, R37-R40

Karakteristike skupa zahtjeva

#	Karakteristika	Opis karakteristike	Pravila
C10	Kompletnost	Skup zahtjeva treba u dovoljnoj mjeri opisivati potrebne mogućnosti, karakteristike, ograničenja i faktore kvalitete softverskog sustava. Da bismo razumjeli skupa zahtjeva nije potreban niti jedan drugi skup zahtjeva.	R3, R29, R41
C11	Konzistentnost	Skup zahtjeva treba sadržavati pojedinačne zahtjeve koji su jedinstveni, nisu u međusobnom konfliktu, ne preklapaju se, te koriste konzistentan izričaj.	R4, R29, R30, R36-R41
C12	Izvedivost	Skup zahtjeva treba biti izvediv s obzirom na postojeća ograničenja (npr. budžet, rok, oprema, legislativa, sigurnost i sl.) te uz prihvatljiv rizik.	R26, R29, R30, R33, R34
C13	Razumljivost	Skup zahtjeva treba biti napisan na način da je jasno što se očekuje.	R4, R18, R36-R41
C14	Provjerljivost	Skup zahtjeva treba biti formuliran na način da se može provjeriti je li skup zahtjeva ispunjen ili nije.	R3, R4, R36-R41

Pravila za definiranje zahtjeva

Pravilo	Opis
R1: Struktura rečenice	Definirajte zahtjev kao cijelu rečenicu oblika kao što je npr. „Sustav će omogućiti <funkcionalnost> <objekt> uz <ograničenja>“.
R2: Aktivni oblik	Koristite aktiv prilikom definiranja zahtjeva.
R4: Definirani pojmovi	Koristite pojmove koje ste prethodno definirali u specifikaciji zahtjeva.
R6 : Jedinice mjere	U slučaju da se u zahtjevu koriste kvantitativni pokazatelji, navedite ispravne jedinice mjere.
R7: Nejasni pojmovi	Izbjegavajte korištenje nejasnih pojmova, kao što su: otprilike, dovoljno, obično, približno, nešto, bilo koji, i sl.
R8: Izrazi bijega	Izbjegavajte korištenje klauzula bijega kao što su: ukoliko je moguće, ako je potrebno, sl.
R9 – Otvoreni izrazi	Izbjegavajte otvorene izraze kao što je „i tako dalje“.
R10 – Suvišne riječi	Izbjegavajte korištenje suvišnih riječi samo da biste „ukrasili“ zahtjev.
R12 – Ispravna gramatika-	Zahtjev definirajte gramatički ispravno (npr. ispravan padež, ispravno lice,...).
R14 – Interpunkcija	Ispravno koristite interpunkcijske znakove.
R16 – Negacija	Izbjegavajte korištenje apsolutnih negacija. Zahtjeve kao što je „Sustav NEĆE sadržavati greške.“ je vrlo teško ili čak nemoguće provjeriti.
R17 – Simbol „/“	Izbjegavajte korištenje simbola „/“ jer može imati mnogo interpretacija („i“, „ili“, ...).
R18 – Jedna rečenica	Zahtjev napišite kao jednu rečenicu koja sadrži jednu misao.
R19 – Veznici	Izbjegavajte korištenje veznika prilikom definiranja zahtjeva. Veznici kao što su <i>i</i> , <i>pa</i> , <i>te</i> , <i>ni</i> , <i>niti</i> , <i>a</i> , <i>ali</i> , mogu značiti da se radi o više zahtjeva a ne o jednom. Isto vrijedi i za „ <i>međutim</i> “, „ <i>uz to</i> “, „ <i>s druge strane</i> “, „ <i>dok</i> “.
R20 – Svrha	Izbjegavajte korištenje riječi ili izraza koji ukazuju na svrhu zahtjeva (npr. „ <i>zbog</i> “, „ <i>kako bi</i> “, i sl.). Svrha, odnosno odgovor na pitanje zašto je zahtjev bitan, treba specificirati u zasebnom atributu <i>Obrazloženje</i> , a ne u samoj rečenici zahtjeva.
R21: Zgrade	Izbjegavajte korištenje zagrada prilikom definiranja zahtjeva. Informacije u zagradama su često suvišne, ili nedovoljno jasne, pa ih treba ili maknuti ili dodatno raspraviti s korisnikom i navesti u atributu <i>Obrazloženje</i> .
R22: Nabranje	Ako se zahtjev odnosi npr. na skup operacija ili entiteta, potrebno je eksplicitno navesti elemente tog skupa a ne samo navesti naziv skupa. Npr. umjesto „Sustav će omogućiti izradu izvještaja.“, potrebno je eksplicitno navesti o kojim se izvještajima radi. Vrlo vjerojatno će za svaki element skupa trebati definirati zaseban zahtjev.
R23: Kontekst	Ako se zahtjev odnosi na izrazito složenu strukturu ili ponašanje, radi jasnoće bi bilo poželjno priložiti i dijagram (npr. UML dijagram).
R24: Zamjenice	Izbjegavajte korištenje zamjenica (taj, to, ono, on, oni...). Umjesto toga, radije ponovite imenicu na koju se referirate. Također, izbjegavajte korištenje i neodređenih zamjenice kao što su <i>svi</i> , <i>bilo koji</i> , <i>ostali</i> , <i>bilo koji</i> , <i>svaki</i> , i sl., jer stvaraju mogućnost više interpretacija istog zahtjeva.
R25: Naslovi	Zahtjev ne smije biti definiran tako da ovisi o naslovu poglavlja ili potpoglavlja u kojem se nalazi kako bi mogao biti interpretiran. Zahtjev mora biti jasan sam po sebi.
R26: Apsolutne vrijednosti	Izbjegavajte korištenje nedostižnih apsolutnih vrijednosti. Npr. „Sustav će biti online 100% vremena.“ nije dostižan cilj. Osim kvantitativnih apsolutnih pokazatelja, treba izbjegavati i izraze koji sadrže riječi „ <i>svi</i> “, „ <i>uvijek</i> “, „ <i>nikad</i> “ i slično.
R27: Uvjeti	U slučaju da je zahtjev primjenjiv samo pod određenim uvjetima, te uvjete treba eksplicitno navesti.
R29: Klasifikacija	Klasificirajte zahtjeve sukladno aspektima problemske domene ili samog rješenja na koji se zahtjevi odnose (npr. funkcionalni zahtjevi, nefunkcionalni zahtjevi, sigurnost, privatnost, ...).

R30: Izrazi jedanput	Svaki zahtjev treba biti izražen jednom i samo jednom. Pojava dupliciranih zahtjeva dovodi do problema u predstojećim aktivnostima procesa razvoja softvera.
R31: Bez rješenja	Izbjegavajte navođenje ili sugeriranje rješenja prilikom definiranja zahtjeva. Odgovor na pitanje kako realizirati zahtjeve trebaju dati aktivnosti dizajna i implementacije, a ne aktivnost definiranja zahtjeva.
R33: Raspon vrijednosti	Ukoliko zahtjev nije moguće kvantificirati s konkretnom vrijednosti, koristite raspon vrijednosti (npr. „Sustav će očitavati vrijednosti temperature s preciznošću od +/- 0.1°C“).
R34: Mjerljivost	Ukoliko se zahtjev može kvantificirati, definirajte ga na način da bude mjerljiv. Izbjegavajte riječi koje nejasno ili paušalno kvantificiraju zahtjev, kao npr. <i>odmah</i> , <i>brzo</i> , <i>uobičajeno</i> , <i>maksimalno</i> , <i>minimalno</i> , <i>optimalno</i> , <i>lako</i> , i sl.
R35: Vremenska nedefiniranost	Izbjegavajte riječi koje nejasno definiraju vremenski aspekt zahtjeva, kao npr. prije, poslije, nekad, u jednom trenutku, simultano i sl. Umjesto toga budite eksplicitni u izražavanju vremenskog aspekta.
R36: Konzistentni pojmovi	Koristite pojmove i jedinice mjere konzistentno u cijelom skupu zahtjeva.
R37: Akronimi	Ako zahtjevi sadrže akronime, koristite konzistentan skup definiranih akronima.
R38: Skraćenice	Izbjegavajte korištenje nestandardnih skraćenica jer mogu dovesti do različitih i pogrešnih interpretacija.
R39: Stilske smjernice	Koristite smjernice za pisanje zahtjeva kako bi standardizirali oblik i sintaksu zahtjeva u timu ili poduzeću.
R40: Grupiranje	Grupirajte srodne zahtjeve na temelju tipa, scenarija ili drugih pogodnih karakteristika.
R41: Struktura	Držite se definirane strukture ili predložka prilikom pisanja zahtjeva za sve zahtjeve u skupu.

Atributi zahtjeva

Kako bi bili lako razumljivi zahtjevi trebaju biti izraženi u obliku jednostavnih rečeničnih konstrukcija. Međutim, takav oblik sigurno nije dostatan za izražavanje svih potrebnih podataka zahtjeva. Umjesto da zahtjevi postanu duge, složene i teško razumljive rečenične konstrukcije, za definiranje dodatnih informacija o zahtjevima se često koriste tzv. *atributi*. Oni omogućavaju lakše razumijevanje, grupiranje, filtriranje i sortiranje zahtjeva. Koji konkretni atributi će biti upotrijebljeni za strukturiranje dodatnih informacija o zahtjevima je nešto što ovisi autorima specifikacije zahtjeva. Međutim, iako nema propisanog standarda, korist pojedinih atributa je prepoznata i u literaturi i u praksi. Neki od tih atributa su sljedeći:

Atribut	Oznaka
Identifikator/Oznaka	Jedinstvena oznaka zahtjeva. Nužna za jednoznačno identificiranje zahtjeva u specifikaciji zahtjeva.
Datum zahtjeva	Datum kreiranja zahtjeva.
Vrsta zahtjeva	Vrsta zahtjeva s obzirom na klasifikaciju koju autori koriste (npr. funkcionalni/nefunkcionalni).
Prioritet	Pokazatelj važnosti i/ili hitnosti zahtjeva. Može poprimiti kvantitativne (npr. 1 – 10), ili nominalne vrijednosti (npr. ključan, obvezan, neobvezan, poželjan...).

Izvor	Naziv dokumenta kojim je zahtjev propisan ili dionika koji je podnio zahtjev.
Status	Status u kojem se zahtjev nalazi (npr. predložen, prihvaćen, odbijen, ispunjen, i sl.
Obrazloženje	Obrazloženje zašto zahtjev postoji, odnosno zašto je potreban.
Način provjere	Kriterij provjere ili testni scenarij koji će omogućiti utvrđivanje je li zahtjev ispunjen ili nije.
Rizik	Razina rizika vezana uz realizaciju zahtjeva.

Praktični primjer

Specifikacija softverskih zahtjeva za Evaluation Manager

3. SPECIFIČNI FUNKCIONALNI ZAHTJEVI

Identifikator	FZ-1
Zahtjev	Sustav će dozvoliti pristup podacima samo autentificiranim korisnicima.
Obrazloženje	Evaluation Manager mora ograničiti pristup osjetljivim podacima studenata (rezultati i bodovi vrednovanja, te ocjena) tako da ih samo ovlaštene osobe (nastavnici na kolegiju) smiju čitati i unositi.
Način provjere	Upis ispravnih korisničkih podataka treba rezultirati uspješnom autentifikacijom i omogućiti korisniku daljnji rad u sustavu. U slučaju neispravnih korisničkih podataka autentifikacija treba biti neuspješna i neće biti moguć rad u sustavu.
Prioritet [1 -5]	1
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-2
Zahtjev	Sustav će omogućiti uvoz podataka o upisanim studentima iz datoteke izvezene iz ISVU sustava.
Obrazloženje	Na početku svake akademske godine potrebno je unijeti u Evaluation Manager veći broj studenata (očekivano oko 200). S obzirom da su studenti koji su upisali kolegij već evidentirani u sustavu ISVU, proceduru unosa podataka je moguće ubrzati uvozom podataka iz ISVU sustava.
Način provjere	Nakon provedenog uvoza podataka, svi studenti koji su izvezeni iz ISVU sustava trebaju biti trajno zapisani i vidljivi u Evaluation Manageru.
Prioritet [1 -5]	3
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-3
Zahtjev	Sustav će omogućiti ručni unos podataka o studentima koji pohađaju kolegij.
Obrazloženje	Na početku akademske godine studenti upisani u kolegij se uvoze iz ISVU sustava. Međutim, iz različitih razloga određeni broj studenata biva upisan u kolegij sa zakašnjenjem. Takve studente je tada potrebno unijeti ručno u sustav.
Način provjere	Ručno uneseni student treba biti trajno zapisan i vidljiv u sustavu.
Prioritet [1 -5]	3
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-4
Zahtjev	Sustav će omogućiti prikaz studenata upisanih na kolegij.
Obrazloženje	Nastavnici moraju u svakom trenutku moći vidjeti studente koji su upisani u kolegij kako bi mogli evidentirati njihove aktivnosti.

Način provjere	Popis studenata upisanih u sustav treba biti vidljiv nastavniku.
Prioritet [1 -5]	1
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-5
Zahtjev	Sustav će omogućiti definiranje popisa elemenata praćenja na kolegiju.
Obrazloženje	S obzirom da se način izvedbe kolegija može mijenjati (npr. smanjiti ili povećati broj kolokvija i zadaća), potrebno je omogućiti i mijenjanje elemenata praćenja po kojima će se rad studenata vrednovati.
Način provjere	Promjene nastale unosom novog elementa praćenja, ili promjena i uklanjanje postojećeg su trajno zapisane i vidljive u sustavu.
Prioritet [1 -5]	4
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-6
Zahtjev	Za svaki element praćenja, sustav će omogućiti definiranje uvjeta koje student treba ispuniti kako bi ostvario pravo na potpis.
Obrazloženje	Nastavnici mogu odlučiti da postoje minimalni uvjeti koje student mora ispuniti za element praćenja da bi se moglo evidentirati da je „odslušao“ kolegij i da može pristupiti ispitu. U slučaju neispunjenja takvih minimalnih uvjeta, student gubi pravo na potpis i mora ponovno upisati kolegij.
Način provjere	U sustav je moguće unijeti minimalne uvjete za potpis, i ti uvjeti trebaju biti trajno zapisani i vidljivi.
Prioritet [1 -5]	4
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-7
Zahtjev	Za svaki element praćenja, sustav će omogućiti definiranje uvjeta koje student treba ispuniti kako bi bio pozitivno ocijenjen.
Obrazloženje	Nastavnici mogu odlučiti da postoje minimalni uvjeti koje student mora ispuniti za element praćenja kako bi bio pozitivno ocijenjen. U slučaju neispunjenja takvih minimalnih uvjeta, student će biti negativno ocijenjen.
Način provjere	U sustav je moguće unijeti minimalne uvjete za pozitivnu ocjenu, i ti uvjeti trebaju biti trajno zapisani i vidljivi.
Prioritet [1 -5]	4
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-8
Zahtjev	Sustav će omogućiti definiranje bodovne skale za ocjenjivanje studenata.
Obrazloženje	U ovisnosti o uvedenim promjenama u izvedbi kolegija, može biti potrebno korigirati bodovnu skalu u smislu proširivanja ili smanjivanja razreda za pojedinu ocjenu.
Način provjere	U sustav je moguće unijeti te korigirati bodovnu skalu. Navedene promjene trebaju biti trajno zapisane i vidljive.
Prioritet [1 -5]	4
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-9
Zahtjev	Sustav će omogućiti evidentiranje bodova ostvarenih prilikom evaluacije za svakog upisanog studenta po definiranim elementima praćenja.
Obrazloženje	Evidentiranje bodova po elementima praćenja je nužno kako bismo mogli evaluirati znanje studenata u teorijskom i praktičnom smislu, te
Način provjere	Nakon unosa bodova za neki element kontinuiranog praćenja studenta, bodovi moraju biti trajno zapisani i vidljivi.
Prioritet [1 -5]	1
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-10
Zahtjev	Sustav će za svakog studenta moći odrediti jesu li ostvareni uvjeti za potpis ili ne.
Obrazloženje	Student koji nije ostvario uvjete za potpis nije zadovoljio minimalne uvjete propisane modelom praćenja. Za takvog studenta će biti potrebno evidentirati zabranu potpisa, te mu na taj način onemogućiti izlazak na ispitne rokove iz kolegija.
Način provjere	Student za kojeg je sustav odredio da ima ostvarene uvjete za potpis mora imati evidentirane barem minimalne bodove iz svih modelom definiranih elemenata praćenja. Ukoliko nisu evidentirani bodovi iz svih modelom definiranih elemenata praćenja, ili su po barem jednom elementu praćenja evidentirani bodovi manji od minimalnih, student ne smije imati ostvareno pravo na potpis.
Prioritet [1 -5]	1
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-11
Zahtjev	Po završetku kontinuiranog praćenja, sustav će za studente koji su ostvarili uvjet za potpis predložiti ocjenu na temelju ukupnog broja ostvarenih bodova .
Obrazloženje	Kako bi se na roku za kontinuirano praćenje u ISVU mogla upisati ocjena sukladno ostvarenim bodovima na kontinuiranom praćenju, potrebno je za svakog studenta znati je li ostvario uvjet za upis ocjene i o kojoj se ocjeni radi.
Način provjere	Ocjena treba biti predložena samo u slučaju da je student ostvario uvjet za potpis, te mora biti u skladu sa definiranom bodovnom skalom.
Prioritet [1 -5]	1
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-12
Zahtjev	Sustav će izraditi izvještaj sa popisom studenata koji nisu ostvarili pravo na potpis.
Obrazloženje	Prije otvaranja roka za kontinuirano praćenje, nastavnik treba evidentirati zabranu potpisa u ISVU sustavu svim studentima koji nisu ostvarili pravo na potpis.
Način provjere	Na popisu moraju biti svi studenti koji nisu ostvarili pravo na potpis. Također, na popisu ne smije biti niti jedan student koji jeste ostvario pravo na potpis.
Prioritet [1 -5]	3
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

Identifikator	FZ-13
Zahtjev	Sustav će omogućiti ispis rezultata kontinuiranog praćenja za studente koji su ostvarili potpis. Pri tome za svakog studenta treba biti vidljiv broj bodova pojedinačnih elemenata praćenja, ukupan broj bodova, te prijedlog ocjene.
Obrazloženje	Po otvaranju roka za kontinuirano praćenje nastavnik treba upisati ocjene u ISVU sustav svim studentima koji prijavljeni na ispitni rok (tj. koji su dobili potpis). Osim predložene ocjene, nastavniku je potreban i pregledan prikaz bodova po pojedinim elementima praćenja, kako bi lakše donio odluku u slučaju rubnih situacija i odgovaranja za veću ocjenu.
Način provjere	Na popisu moraju biti svi studenti koji su ostvarili pravo na potpis, i sukladno tome prijavljeni na rok za kontinuirano praćenje. Na popisu ne smije biti studenata koji nisu ostvarili pravo na potpis. Svi prikazani bodovi po elementima praćenja trebaju odgovarati činjeničnom stanju. Predložena ocjena borat biti u skladu sa bodovnom skalom.
Prioritet [1 -5]	2
Izvor	Nastavnici na kolegiju Programsko inženjerstvo

3.1. Dinamika realizacije zahtjeva

U inicijalnoj verziji softvera biti će realizirani samo zahtjevi sa najvećim prioritetom. To uključuje:

- **FZ-1** - Sustav će omogućiti pristup samo autentificiranim korisnicima.
- **FZ-4** - Sustav će omogućiti prikaz studenata upisanih na kolegij.
- **FZ-9** - Sustav će omogućiti evidentiranje bodova ostvarenih prilikom evaluacije za svakog upisanog studenta po definiranim elementima praćenja.
- **FZ-10** - Sustav će za svakog studenta moći odrediti jesu li ostvareni uvjeti za potpis ili ne.
- **FZ-11** - Po završetku kontinuiranog praćenja, sustav će za studente koji su ostvarili uvjet za potpis predložiti ocjenu na temelju ukupnog broja ostvarenih bodova .
- **FZ-13** - Sustav će omogućiti ispis rezultata kontinuiranog praćenja za studente koji su ostvarili potpis. Pri tome za svakog studenta treba biti vidljiv broj bodova pojedinačnih elemenata praćenja, ukupan broj bodova, te prijedlog ocjene.

U narednim verzijama planirana je realizacija ostalih zahtjeva:

- **FZ-2** - Sustav će omogućiti uvoz podataka o upisanim studentima iz datoteke izvezen iz ISVU sustava.
- **FZ-3** - Sustav će omogućiti ručni unos podataka o studentima koji pohađaju kolegij.
- **FZ-5** - Sustav će omogućiti definiranje popisa elemenata praćenja na kolegiju.
- **FZ-6** - Za svaki element praćenja, sustav će omogućiti definiranje uvjeta koje student treba ispuniti kako bi ostvario pravo na potpis.

4. NEFUNKCIONALNI ZAHTEJEVI

4.1. Izgled softvera

NFZ-1 – Sustav će interakciju s korisnikom provoditi preko grafičkog sučelja.

NFZ-2 – Sustav će pratiti formalan/korporativni stil grafičkog sučelja.

4.2. Upotrebljivost softvera

NFZ-3 – Sustav će ponuditi mehanizme koji će smanjiti mogućnost grešaka prilikom unosa rezultata evaluacije od strane nastavnika.

4.3. Performanse softvera

NFZ-4 – Sustav će osigurati preciznost za decimalne brojeve na razini 2 decimalna mjesta.

NFZ-5 – Sustav će biti dostupan 24 sata za vrijeme trajanja kontinuiranog praćenja.

NFZ-6 – Sustav će osigurati mogućnost simultanog korištenja minimalno 5 korisnika.

4.4 Izvođenje softvera i okruženje

NFZ-7 – Sustav treba raditi na računalima s instaliranim Windows 10 ili novijim operacijskim sustavom.

4.5. Sigurnost i privatnost

NFZ-8 – Sustav će samo nastavnicima omogućiti pristup rezultatima praćenja studenata,

NFZ-9 – Sustav će upotrebljavati podatke o studentima u skladu sa odredbama GDPR-a.

4.6. Ostalo

Nema identificiranih dodatnih nefunkcionalnih zahtjeva.

5. SKICE ZASLONA

5.1. Skica zaslona za prijavu u sustav

Prijava u sustav	X
<p>Korisničko ime:</p> <input type="text"/>	
<p>Lozinka:</p> <input type="password"/>	
<p>Prijavi</p>	

5.2. Skica zaslona za prikaz studenata upisanih u kolegij

Studenci upisani u kolegij							X
Id	Ime	Prezime	Ocjena	
1	Ivan	Horvat	4	
2	Ana	Novak	5	
3	Marko	Lukić	3	
4	Stjepan	Perić	2	
...	

Generiraj izvještaj

Evaluiraj studenta

Dodaj studenta

Izmijeni studenta

Ukloni studenta

5.3. Skica zaslona za provedbu evaluacije studenata

Evaluacijska forma - Ivan Horvat		X
Aktivnost:	Kolokvij 1	<
Uvjet za potpis:		0/25
Uvjet za ocjenu:		11/25
Opis aktivnosti:	Pitanja tipa odaberi jedan između ponuđenih odgovora, odaberi više između ponuđenih odgovora, dopuni pojam koji nedostaje, spoji pojmove i sl., ili rješavanje zadataka uz dodatak teorijskih pitanja otvorenog tipa koja ispituju razumijevanje.	
Ocjenjivač:	Marko Mijač	
Datum evaluacije:	20.2.2022.	
...	-	
...	-	
...	-	
Broj bodova:	21	
Spremi		Odustani

5.4. Skica zaslona za prikaz rezultata kontinuiranog praćenja

Rezultati kontinuiranog praćenja									
Ime	Prezime	K1	K2	Z1	Z2	Z3	Ukupno	Potpis	Ocjena
Ivan	Horvat	20	24	11	12	10	77	DA	4
Ana	Novak	21	17	10	15	19	82	DA	4
Marko	Lukić	18	18	12	13	13	74	DA	3
Stjepan	Perić	13	16	8	10	9	56	DA	2
...

Pitanja i zadaci

1. Objasnite razliku između korisničkog zahtjeva i softverskog zahtjeva?
2. Objasnite razliku između funkcionalnog zahtjeva i nefunkcionalnog zahtjeva?
3. Zašto su nefunkcionalni zahtjevi često kritični, a funkcionalni zahtjevi ne?
4. Navedite koje karakteristike trebaju posjedovati pojedinačni zahtjevi?
5. Navedite koje karakteristike trebaju posjedovati skupovi zahtjeva?
6. Zašto koristimo dodatne atribute za opis zahtjeva?
7. Navedite nekoliko pravila za pisanje zahtjeva?
8. Za što su korisne skice zaslona u kontekstu specifikacije zahtjeva?