

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

Милица Д. Симић

**Модел пројектно-заснованог учења у
области развоја софтвера
заснован на примени
генеративне вештачке интелигенције**

докторска дисертација

Београд, 2026

UNIVERSITY OF BELGRADE
FACULTY OF ORGANIZATIONAL SCIENCES

Milica D. Simić

**A model of project-based learning in
software development based on the application
of generative artificial intelligence**

Doctoral Dissertation

Belgrade, 2026

Ментор:

др Зорица Богдановић, редовни професор

Универзитет у Београду - Факултет организационих наука

Чланови комисије:

др Маријана Деспотовић-Зракић, редовни професор

Универзитет у Београду - Факултет организационих наука

др Душан Бараћ, редовни професор

Универзитет у Београду - Факултет организационих наука

др Саша Лазаревић, редовни професор

Универзитет у Београду - Факултет организационих наука

др Душан Савић, ванредни професор

Универзитет у Београду - Факултет организационих наука

др Слађана Јанковић, редовни професор

Универзитет у Београду - Саобраћајни факултет

Датум одбране: _____

Модел пројектно-заснованог учења у области развоја софтвера заснован на примени генеративне вештачке интелигенције

Сажетак:

Предмет истраживања докторске дисертације је развој модела пројектно-заснованог учења (PBL) у области развоја софтвера, заснованог на примени генеративне вештачке интелигенције (GenAI). Истраживање је усмерено на интеграцију GenAI алата у наставни процес кроз све његове фазе, од планирања наставе и креирања наставних материјала, преко менторског рада и подршке студентима у развоју софтвера, до процеса формативног и вршиљачког оцењивања.

Централни истраживачки проблем огледа се у недостатку систематизованих и методолошки утемељених приступа за интеграцију генеративне вештачке интелигенције у образовање у области развоја софтвера. Иако је у индустрији употреба GenAI алата већ широко распрострањена и значајно утиче на продуктивност, у академском окружењу њихова примена је често спонтана, недовољно контролисана и без јасних педагошких, технолошких и етичких оквира. Оваква пракса резултира недовољним искоришћавањем потенцијала генеративне вештачке интелигенције у настави, уз истовремено постојање изазова који се односе на очување академске честитости, развој критичког размишљања и ризик од прекомерног ослањања студената на аутоматизоване алате.

Главни циљ истраживања је развој структурираног методолошког оквира који омогућава ефикасну, етички одговорну и педагошки одрживу интеграцију генеративне вештачке интелигенције у пројектно-засновано учење у области развоја софтвера. Овим приступом се унапређују исходи учења, постиже се већа мотивација и ангажованост студената, као и развој техничких и меких вештина неопходних за рад у мултидисциплинарним тимовима. Поред тога, студенти су кроз практичан рад на пројектима стекли искуства блиска реалним професионалним ситуацијама, што их боље припрема за изазове савремене ИТ индустрије.

У раду се разматрају кључни концепти пројектно-заснованог учења у области развоја софтвера и његове образовне предности, са посебним освртом на улоге наставника и студената у овом приступу. Анализирани су савремени домени примене генеративне вештачке интелигенције у образовању, укључујући постојеће алате и технике попут prompt engineering-а, релевантни међународни оквири попут UNESCO стандарда за AI компетенције и европских регулатива, уз осврт на етичке аспекте. Посебна пажња посвећена је интеграцији генеративне вештачке интелигенције у различите фазе пројектно-заснованог учења, уз нагласак на педагошке приступе и методе оцењивања, што представља основу за развој предложеног методолошког оквира. Резултати овог истраживања доприносе унапређењу наставних процеса у области развоја софтвера и интеграције савремених технологија, посебно генеративне вештачке интелигенције, у пројектно-засновано учење.

Кључне речи: генеративна вештачка интелигенција, пројектно-засновано учење, развој софтвера, GenAI у образовању

Научна област: Информациони системи и технологије

Ужа научна област: Електронско пословање

A model of project-based learning in software development based on the application of generative artificial intelligence

Abstract:

The research subject of this paper is the development of a project-based learning (PBL) model in the field of software development, grounded in the application of generative artificial intelligence (GenAI). The study focuses on the integration of GenAI tools into the educational process across all its phases, from lesson planning and creation of instructional materials, through mentoring and student support in software development, to formative and peer assessment processes.

The central research problem lies in the lack of systematic and methodologically grounded approaches for integrating generative artificial intelligence into software development education. Although the use of GenAI tools is already widespread in the industry and significantly impacts productivity and innovation, their application in academic settings is often spontaneous, insufficiently controlled, and lacking clear pedagogical, technological, and ethical frameworks. This practice leads to underutilization of the potential of generative artificial intelligence in teaching, while simultaneously raising challenges related to maintaining academic integrity, developing critical thinking skills, and the risk of students over-relying on automated tools.

The main objective of the research is to develop a structured methodological framework that enables the effective, ethically responsible, and pedagogically sustainable integration of generative artificial intelligence into project-based learning in software development education. This approach improved learning outcomes, increased student motivation and engagement, and foster both technical and soft skills necessary for working in multidisciplinary teams. Furthermore, through practical project work, students gained experience close to real professional situations, better preparing them for the challenges of the modern IT industry.

The paper examines key concepts of project-based learning in software development education and its educational benefits, with particular attention to the roles of instructors and students in this approach. It analyzes current domains of generative artificial intelligence application in education, including existing tools, techniques such as prompt engineering, relevant international frameworks such as the UNESCO standards for AI competencies, and European regulations, alongside ethical considerations. Special emphasis is placed on the integration of generative artificial intelligence into various phases of project-based learning, with a focus on pedagogical approaches and assessment methods, forming the foundation for the proposed methodological framework. The results of this research may contribute to the improvement of educational processes in software development and the integration of modern technologies, particularly generative artificial intelligence, into project-based learning.

Keywords: *generative artificial intelligence, project-based learning, software development, GenAI in education*

Scientific field: *Information systems and technologies*

Scientific subfield: *E-business*

САДРЖАЈ

1	УВОД	1
1.1	Дефинисање предмета и циљева истраживања	1
1.2	Полазне хипотезе	4
1.3	Методологија истраживања	5
1.4	Структура и организација рада	6
2	ОБРАЗОВАЊЕ У ОБЛАСТИ РАЗВОЈА СОФТВЕРА	8
2.1	Педагошки приступи у области развоја софтвера	8
2.2	Иновације у области развоја софтвера	11
2.2.1	Развој софтвера применом low-code и no-code платформи	12
2.2.2	Развој софтвера уз подршку вештачке интелигенције	13
2.2.3	Софтверско инжењерство засновано на AI агентима	18
2.3	Усаглашавање образовања у области развоја софтвера са SWEBOOK стандардом	19
3	ПРОЈЕКТНО-ЗАСНОВАНО УЧЕЊЕ У ОБРАЗОВАЊУ У ОБЛАСТИ РАЗВОЈА СОФТВЕРА	23
3.1	Појам и основе пројектно-заснованог учења	23
3.2	Образовне предности пројектно-заснованог учења	23
3.3	Примена пројектно-заснованог учења у образовању у области развоја софтвера	24
3.4	Улоге наставника и студената у пројектно-заснованом учењу	25
4	ГЕНЕРАТИВНА ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА У ОБРАЗОВАЊУ	26
4.1	Генеративна вештачка интелигенција	26
4.1.1	Појам и основе генеративне вештачке интелигенције	26
4.1.2	Преглед и анализа алата генеративне вештачке интелигенције	31
4.1.3	Технике prompt engineering-a	38
4.2	AI агенти	38
4.3	UNESCO оквир за AI компетенције наставника и студената	44
4.4	Анализа постојећих оквира за примену генеративне вештачке интелигенције у образовању	48
4.4.1	4E оквир за усвајање генеративне вештачке интелигенције у образовању	48
4.4.2	LAIK оквир за интеграцију генеративне вештачке интелигенције у наставу	50
4.4.3	Колаборативни модел интеграције наставника и генеративне вештачке интелигенције	52
4.4.4	GenAI-TRACK оквир за развој компетенција наставника за интеграцију генеративне вештачке интелигенције	54
4.4.5	Упоредна анализа постојећих оквира	56
4.5	Регулатива, етички аспекти и изазови	57
5	ИНТЕГРАЦИЈА ГЕНЕРАТИВНЕ ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ У ОБРАЗОВАЊУ У ОБЛАСТИ РАЗВОЈА СОФТВЕРА	60
5.1	Примена генеративне вештачке интелигенције у развоју курикулума	61
5.2	Педагошки приступи за интеграцију вештачке интелигенције у наставу	62
5.3	Методе оцењивања засноване на генеративној вештачкој интелигенцији	63
6	МЕТОДОЛОШКИ ОКВИР ЗА УНАПРЕЂЕЊЕ ОБРАЗОВАЊА У ОБЛАСТИ РАЗВОЈА СОФТВЕРА ИНТЕГРАЦИЈОМ АЛАТА ГЕНЕРАТИВНЕ ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ У ПРОЈЕКТНО-ЗАСНОВАНО УЧЕЊЕ	65
6.1	Идентификација потреба и изазова у настави у контексту пројектно-заснованог учења	65
6.2	Развој методолошког оквира за унапређење образовања у области развоја софтвера интеграцијом генеративне вештачке интелигенције у пројектно-засновано учење	66
6.3	Развој истраживачког модела за евалуацију предложеног оквира	69

6.4	Развој алата за евалуацију предложеног оквира	71
6.4.1	Архитектура AI агента за анализу и евалуацију студентских софтверских пројеката	72
6.4.2	Ток рада AI агента (Agent Workflow)	74
6.4.3	Технологије и структура пројекта	75
6.4.4	Функционалности AI агента	76
6.4.5	Приказ примене AI агента на студентском софтверском пројекту	82
6.4.6	Резултати анализе и повратне информације AI агента	85
7	ПРИМЕНА И ЕВАЛУАЦИЈА РАЗВИЈЕНОГ МЕТОДОЛОШКОГ ОКВИРА У НАСТАВНОМ ПРОЦЕСУ	89
7.1	Примена методолошког оквира у фазама анализе корисничких захтева и управљања софтверским пројектом	89
7.1.1	Контекст истраживања и хипотезе	89
7.1.2	Процедура истраживања	92
7.1.3	Учесници	95
7.1.4	Инструменти	95
7.1.5	Анализа резултата	96
7.1.6	Дискусија	102
7.2	Примена методолошког оквира у фазама пројектовања, имплементације и тестирања софтвера	107
7.2.1	Контекст истраживања и хипотезе	107
7.2.2	Процедура истраживања	110
7.2.3	Учесници	112
7.2.4	Инструменти	112
7.2.5	Анализа резултата	113
7.2.6	Дискусија	118
7.3	Импликације и ограничења истраживања	122
7.3.1	Практичне импликације	122
7.3.2	Ограничења	124
8	НАУЧНИ И СТРУЧНИ ДОПРИНОС	125
9	БУДУЋА ИСТРАЖИВАЊА	128
10	ЗАКЉУЧАК	130
	ЛИТЕРАТУРА	134
	ЛИСТА СЛИКА	149
	ЛИСТА ТАБЕЛА	150
	ЛИСТА АКРОНИМА	151
	ПРИЛОЗИ	152
	Прилог 1 - Структура упитника (анализа корисничких захтева)	152
	Прилог 2 - Структура упитника (управљање софтверским пројектом)	153
	Прилог 3 - Структура упитника (пројектовање софтвера)	154
	Прилог 4 - Структура упитника (имплементација и тестирање софтвера)	155
	БИОГРАФИЈА	156
	СПИСАК ОБЈАВЉЕНИХ РАДОВА	157
	ИЗЈАВЕ	161
	Изјава о коришћењу генеративне вештачке интелигенције	161
	Изјава о ауторству	162
	Изјава о истовестности штампане и електронске верзије докторског рада	163
	Изјава о коришћењу	164

1 Увод

1.1 Дефинисање предмета и циљева истраживања

Предмет истраживања ове докторске дисертације је развој модела пројектно-заснованог учења (енг. *Project-Based Learning - PBL*) у области развоја софтвера, заснованог на примени генеративне вештачке интелигенције (енг. *Generative Artificial Intelligence - GenAI*). Истраживање се бави интеграцијом генеративних AI модела (попут оних заснованих на архитектури великих језичких модела) у све аспекте наставног процеса - од планирања наставе, преко развоја наставних материјала и курикулума који укључују рад са алатима генеративне вештачке интелигенције, до реализације наставе. Посебна пажња посвећена је подршци студентима, са акцентом на менторски рад и коришћење *GenAI* алата у развоју софтвера. Такође, истражују се могућности примене генеративне вештачке интелигенције у процесу формативног и вршњачког оцењивања (енг. *peer assessment*), кроз аутоматизовано праћење напредовања, давање повратних информација и процену исхода учења у складу са постављеним циљевима учења.

У оквиру високошколских програма усмерених на образовање у области развоја софтвера, најчешће се користе приступи који се ослањају на активно учење, где су студенти не само пасивни примаоци информација, већ активни учесници у процесу учења кроз решавање реалних проблема, рад на пројектима и примену стечених знања у практичном контексту [1]. Један од најефективнијих модела активног учења у образовању у области развоја софтвера је пројектно-засновано учење. Овај приступ подстиче студенте да кроз тимски или индивидуални рад на пројектима повезују теоријска знања са практичним вештинама. Пројектно-засновано учење доприноси развоју техничких компетенција, као и тзв. меких вештина као што су комуникација, критичко размишљање, управљање временом и решавање проблема. Осим што припрема студенте за реалне изазове у ИТ индустрији, овај приступ омогућава наставницима да кроз улогу ментора боље прате напредак студената и прилагоде подршку у складу са индивидуалним потребама [2]–[4].

Упркос значајним предностима пројектно-заснованог учења, савремени наставни процес у ИТ области све чешће поставља нове изазове повезане са применом савремених технологија, пре свега генеративне вештачке интелигенције. Развој генеративне вештачке интелигенције одвија се брзим темпом, што је довело до њене спонтане, често стихијске примене у образовању [5]. Високошколске институције генерално касне у формулисању смерница и политика за употребу ових алата, док наставници и студенти често немају довољно јасну представу о етичким аспектима примене вештачке интелигенције у настави. Са друге стране, привреда већ интензивно користи алате генеративне вештачке интелигенције у свакодневном раду, а од нових ИТ стручњака се очекује да буду компетентни у ефикасној и одговорној употреби ових алата [6]–[8]. У том контексту, све више међународних тела истиче потребу за системским приступом. *UNESCO* је формулисао оквири дигиталних компетенција за наставнике и студенте у вези са употребом вештачке интелигенције у образовању [9], [10]. Европска комисија, кроз *Digital Education Action Plan 2021–2027 - Action 6* препознаје значај развоја дигиталне

писмености, укључујући критичко разумевање и одговорну употребу AI технологија у настави [11].

Иако још увек не постоји јединствена методологија за увођење генеративне вештачке интелигенције у образовни процес, све више истраживања и иницијатива препознаје приоритетне области њене примене у високом образовању. Генеративна вештачка интелигенција се користи у различите сврхе: од подршке наставницима у креирању наставних садржаја, преко персонализованог учења [12], до подршке процесима оцењивања [13], [14]. У развоју курикулума све више се користи модел ко-креације, у којем наставници, студенти и AI заједнички развијају наставне материјале и пројектне задатке [15]. Учење се све више ослања на алате који омогућавају резимирање текста, прераду сложених концепата или развој упутстава прилагођених индивидуалним потребама студената. Посебан напредак остварен је у примени алата за подршку формативном и вршњачком оцењивању, као и за анализу и повратне информације на студентске радове [16].

Истовремено, брзи успон генеративне вештачке интелигенције у образовању изазвао је и забринутост међу наставницима [17]. Иако се у академском домену генеративна вештачка интелигенција неформално користи у настави, учењу и администрацији [18], постоје озбиљне дискусије о њеном утицају на исходе учења и дугорочно задржавање знања (енг. *knowledge retention*) [19]. Посебно се истичу изазови у вези са академском честитошћу, критичким мишљењем, као и могућношћу прекомерног ослањања студената на аутоматизоване алате [20], [21]. Истраживања показују да студенти алата генеративне вештачке интелигенције, попут *ChatGPT*-а, виде као корисне за проналажење идеја, сажимање текстова или формулисање првих верзија радова, али постоји опасност да овакви алати постану замена за дубинско разумевање сложених концепата [22].

У контексту образовања софтвер инжењера, генеративна вештачка интелигенција има широку примену, посебно у задацима повезаним са генерисањем, анализом и објашњавањем кода [23]–[26]. Алати као што су *GitHub Copilot* или *ChatGPT* показали су значајан утицај на продуктивност програмера и студената. Они се користе у фазама анализе корисничких захтева [27], пројектовања, имплементације, тестирања, одржавања софтвера и ИТ операција [28], [29]. Примена генеративне вештачке интелигенције треба да се посматра у ширем контексту еволуције софтверског инжењерства, које се све више ослања на интегрисане и аутоматизоване процесе развоја. Методологија као што је *DevOps* постаје стандард у индустрији, чиме се мењају не само технички аспекти развоја, већ и улоге и компетенције инжењера [30], [31]. Савремени софтверски инжењери, поред знања из програмирања, морају разумети процесе дигиталне трансформације и иновативне методе као што је *design thinking* (скр. *DT*) [32], [33]. Ове промене су препознате и у релевантним индустријским документима и препорукама, попут *Software Engineering Body of Knowledge (SWEBOOK)*, који све више укључује области као што су колаборативни развој, континуирана испорука, аутоматизација, управљање променама и интеграција технолошких и пословних захтева [34]. У том контексту, високо образовање мора пратити развој индустрије и системски укључити нове домене знања у наставне планове како би студенте припремило за рад у ИТ индустрији.

Циљ овог истраживања је развој структурираног методолошког оквира који омогућава ефикасну и педагошки одрживу интеграцију генеративне вештачке интелигенције у пројектно-засновано учење у образовању у области развоја софтвера. Предложени модел дефинише принципе, техничке компоненте и педагошке приступе који омогућавају примену алата генеративне вештачке интелигенције у настави на предметима који се баве развојем софтвера у различитим доменима савременог дигиталног пословања. Овакав приступ доприноси већем ангажовању студената, бољем разумевању сложених концепата, као и укупном побољшању квалитета наставног процеса. Интеграцијом генеративне вештачке интелигенције у пројектно-засновано учење, студенти стичу практична искуства у коришћењу савремених технологија, што их на реалан и применљив начин припрема за тржиште рада. Оваквим моделом образовања подстиче се развој техничких и меких вештина неопходних за рад у мултидисциплинарним тимовима, брже прилагођавање технолошким иновацијама и активније учешће у процесима дигиталне трансформације у привреди.

Истраживање на развоју модела пројектно-заснованог учења у области развоја софтвера заснован на примени генеративне вештачке интелигенције интелигенције обухвата:

- дефинисање концептуалног модела пројектно-заснованог учења уз подршку генеративне вештачке интелигенције;
- дефинисање методолошког оквира за интеграцију алата генеративне вештачке интелигенције у наставни процес у области развоја софтвера;
- идентификацију модела примене генеративне вештачке интелигенције у свим фазама животног циклуса развоја софтвера, према областима дефинисаним у *SWEBOK*-у;
- дефинисање модела ко-креирања наставних садржаја и пројектних задатака уз активно учешће наставника, студената и алата генеративне вештачке интелигенције;
- анализу технолошких, етичких и педагошких аспеката употребе генеративне вештачке интелигенције у образовању у области развоја софтвера;
- изградњу ИТ инфраструктуре и софтверских компоненти за имплементацију пројектно-заснованог учења уз подршку генеративне вештачке интелигенције;
- дефинисање модела за евалуацију утицаја генеративне вештачке интелигенције на ангажовање студената, развој компетенција и постигнуте исходе учења.

Најважнији циљеви који се постижу имплементацијом развијеног модела пројектно-заснованог учења у области развоја софтвера су:

- унапређење исхода учења;
- повећање мотивације и ангажованости студената кроз интерактиван и персонализован приступ учењу;
- припрема студената за реалне индустријске изазове путем рада на пројектима уз примену савремених алата;
- развој критичког размишљања, тимског рада и вештина решавања проблема;
- ефикасније коришћење наставничких ресурса кроз делимичну аутоматизацију одређених наставних активности.

Задаци истраживања, с обзиром на постављене циљеве су:

- анализа постојећих приступа интеграције генеративне вештачке интелигенције у образовање, посебно у домену ИТ и софтверског инжењерства;
- идентификација и компаративна анализа најпогоднијих алата генеративне вештачке интелигенције и њихових функција у оквиру пројектно-заснованог учења;

- моделирање наставних сценарија који укључују коришћење алата генеративне вештачке интелигенције од стране студената током различитих фаза пројекта;
- дефинисање индикатора перформанси система (нпр. ангажованост студената, тачност кода, квалитет решења);
- евалуација резултата у реалном наставном окружењу и анализу утицаја на процес учења и наставе.

У процесу евалуације, развијени модел је примењен и тестиран на Факултету организационих наука у Београду, у оквиру предмета Катедре за Електронско пословање који се баве развојем софтвера за екосистеме електронског пословања, развојем софтвера заснованог на интеграцији различитих технологија и сл.

Резултати овог истраживања доприносе унапређењу наставних процеса у области развоја софтвера. Примена развијеног модела омогућава иновативан облик пројектно-заснованог учења, што повећа ангажовање и интересовање студената. Интеграцијом генеративне вештачке интелигенције у наставни процес, студенти кроз реализацију практичних пројеката добијају персонализовану подршку у анализи захтева, писању и анализи кода, као и у разјашњавању сложених концепата. На тај начин стичу искуства блиска реалним професионалним ситуацијама.

Научни циљ овог истраживања огледа се у развоју модела пројектно-заснованог учења у области развоја софтвера заснован на примени генеративне вештачке интелигенције. Коначни резултати доприносе формализацији и стандардизацији приступа који омогућава ефикасно, етички одговорно и педагошки оправдано коришћење алата генеративне вештачке интелигенције у образовању.

1.2 Полазне хипотезе

На основу дефинисаног предмета и циљева истраживања, у докторској дисертацији биће испитане следеће хипотезе:

Х.1: Могуће је обезбедити ефективну и етички одговорну примену генеративне вештачке интелигенције у развоју софтвера уколико се њена интеграција спроводи на системски и плански начин, уз примену одговарајућег методолошког оквира који подржава развој релевантних дигиталних и професионалних компетенција код студената.

Х.2: Студенти су спремни да прихвате употребу генеративне вештачке интелигенције као подршку у пројектно-заснованом учењу и настави у области развоја софтвера.

Х.3: Примена генеративне вештачке интелигенције у настави доприноси побољшању исхода учења студената.

Х.4: Генеративна вештачка интелигенција као алат за ко-креирање курикулума и наставних садржаја, уз активно учешће наставника и студената, доводи до веће релевантности и прилагођености образовних материјала потребама савременог ИТ тржишта.

Х.5: Применом генеративне вештачке интелигенције у формативном и вршњачком оцењивању се унапређује квалитет повратне информације и омогућава правовремено праћење напредовања студената.

1.3 Методологија истраживања

У истраживању које се спроводи у оквиру овог рада, користи се комбинација општих научних метода и специфичних методолошких приступа за интеграцију генеративне вештачке интелигенције у пројектно-засновано учење у области развоја софтвера. Од општих научних метода примењене су:

- метода анализе и синтезе, у циљу прегледа и анализе постојеће научне и стручне литературе из области генеративне вештачке интелигенције, пројектно-заснованог учења и образовања у области развоја софтвера. У оквиру ових метода примењен је *PRISMA* протокол [35], који омогућава структурирану идентификацију, укључивање и анализу релевантних извора;
- аналитичко-дедуктивна метода, за испитивање постојећих пракси и идентификовање могућности примене генеративне вештачке интелигенције у настави у области развоја софтвера;
- метода моделирања, која је примењена при развоју концептуалног и методолошког оквира за интеграцију алата генеративне вештачке интелигенције у различите фазе животног циклуса софтверских пројеката у наставном контексту;
- хипотетичко-дедуктивна метода, за формулисање и проверу хипотеза везаних за утицај генеративне вештачке интелигенције на исходе учења, ангажовање студената и развој компетенција;
- статистичка метода, која се користи за обраду и интерпретацију података добијених током експерименталне евалуације оквира.

У оквиру експерименталног дела рада, развијени методолошки оквир је примењен у реалном наставном окружењу, у оквиру наставе на предметима у области развоја софтвера за електронско пословање. Студенти су радили на пројектима уз подршку алата генеративне вештачке и под менторством наставника, са могућностима генерисања програмерских задатака, објашњења кода, тестирања решења итд. Резултати експеримента су прикупљени кроз анализу учешћа, задовољства, перформанси и повратних информација студената и наставника.

Анализа добијених података обухвата квалитативне и квантитативне методе, укључујући статистичку обраду, поређење резултата и графичко представљање у облику табела, дијаграма и графикана.

Истраживање је интердисциплинарног карактера, обједињујући области као што су: информатика, рачунарске науке, образовне технологије, педагогија, и вештачка интелигенција. Оваквим приступом се обезбеђује целовито разумевање изазова и могућности примене генеративне вештачке интелигенције у образовању.

1.4 Структура и организација рада

Ова докторска дисертација се фокусира на унапређење образовања у области развоја софтвера кроз интеграцију генеративне вештачке интелигенције у пројектно-засновано учење. Структура рада је осмишљена тако да пружи свеобухватан преглед теоријских основа, практичних приступа и методолошких оквира за постизање дефинисаних циљева истраживања.

У другом поглављу се разматра образовање у области развоја софтвера, са посебним освртом на педагошке приступе који се примењују у настави софтверског инжењерства. Анализирају се иновације које утичу на начин планирања и реализације наставног процеса, укључујући развој софтвера заснован на примени вештачке интелигенције и *AI* агената. Посебна пажња посвећена је усаглашавању образовних програма са препорукама и областима знања дефинисаним у оквиру *SWEBOK*-а.

У трећем поглављу се обрађује пројектно-засновано учење у контексту образовања за развој софтвера. У уводном делу дефинисани су основни појмови и принципи пројектно-заснованог учења, уз нагласак на његове образовне предности. Посебна пажња посвећена је примени овог приступа у образовању за развој софтвера, истражујући како овај педагошки модел омогућава студентима да стекну практичне вештине и знања релевантна за тржиште рада. Такође, анализирају се улоге наставника и студената у овом приступу, укључујући њихове одговорности и динамику сарадње током пројектних активности.

Четврто поглавље посвећено је генеративној вештачкој интелигенцији и њеној улози у образовању. Уводни део објашњава основне концепте генеративне вештачке интелигенције, укључујући преглед и анализу релевантних алата и технике *prompt engineering*-а. Посебан део поглавља односи се на *AI* агенте, њихове карактеристике и архитектуру, *LangChain* екосистем и примену агентских система у софтверском инжењерству. Поглавље даље детаљно обрађује *UNESCO* оквир за *AI* компетенције наставника и студената. Анализирају се постојећи оквири за примену генеративне вештачке интелигенције у образовању, као и регулатива, етички аспекти и изазови повезани са овом технологијом.

У петом поглављу разматра се интеграција генеративне вештачке интелигенције у пројектно-засновано учење. Фокус је на примени ове технологије у развоју наставних планова и програма, уз истраживање педагошких приступа који омогућавају ефикасну интеграцију *AI* алата у наставне процесе. Поглавље такође детаљно разматра методе оцењивања које користе ове технологије како би се унапредила евалуација студентских радова.

Шесто поглавље предлаже методолошки оквир за унапређење образовања у области развоја софтвера кроз интеграцију генеративне вештачке интелигенције у пројектно-засновано учење. У овом делу идентификују се кључне потребе и изазови у настави, са фокусом на пројектно-засновано учење. Циљ је креирање одрживог система који подржава унапређење компетенција студената у складу са савременим захтевима образовања и тржишта рада. У овом поглављу представља се и развијени *AI* агент, који има улогу алата за аутоматизовану анализу студентских пројеката и подршку евалуацији предложеног методолошког оквира.

У седмом поглављу приказују се примена и евалуација развијеног методолошког поступка у наставном процесу. Истраживање се реализује кроз више итерација које обухватају различите фазе развоја софтвера, од прикупљања и анализе корисничких захтева до пројектовања и имплементације софтверских решења. За сваку итерацију приказују се прилагођавање методолошког поступка конкретном образовном контексту, хипотезе, процедура истраживања, структура учесника, примењени инструменти и методе анализе података. На основу прикупљених података анализирају се резултати истраживања, испитују постављене хипотезе и разматрају ограничења, практичне импликације и могућности даљег унапређења предложеног приступа.

Научни и стручни доприноси докторске дисертације су представљени у осмом поглављу, док су у деветом поглављу приказани правци будућих истраживања.

2 Образовање у области развоја софтвера

2.1 Педагошки приступи у области развоја софтвера

Образовање у области развоја софтвера захтева примену педагошких приступа који превазилазе искључиво преношење теоријских знања и омогућавају студентима активно учешће у наставном процесу [36], [37]. Специфичност образовања у области развоја софтвера огледа се у потреби да студенти, поред усвајања концепата и метода развоја софтвера, развијају професионалне компетенције које обухватају примену стечених знања, решавање инжењерских проблема, тимски рад, комуникацију, управљање задацима и сл [38]. Због тога се у образовању у области развоја софтвера примењују активни педагошки приступи усмерени на рад студената на пројектима, решавању проблема и реалних изазова, међу којима се издвајају пројектно-засновано учење, учење засновано на решавању проблема и учење засновано на изазовима [39].

Пројектно-засновано учење (енг. *Project-Based Learning*), учење засновано на решавању проблема (енг. *Problem-Based Learning*) и учење засновано на изазовима (енг. *Challenge-Based Learning*), представљају сродне педагошке приступе у којима се наставни процес организује око реалних или реалистично постављених проблема, пројектних задатака и изазова, уз активно и практично ангажовање студената [39], [40]. Заједничке карактеристике ових приступа огледају се у централној улози студената, самосталном и колаборативном решавању постављеног проблема, повезивању нових знања са конкретним задатком, као и у избору, образлагању и преиспитивању предложених решења [4], [40], [41]. Иако се у наставној пракси могу међусобно комбиновати, разликују се према основном покретачу процеса учења, очекиваном резултату и ширини контекста у којем се примењују. Код учења заснованог на решавању проблема полазиште представља проблем који треба анализирати и разумети [42], код пројектно-заснованог учења процес је организован око реализације пројекта и развоја одређеног резултата [2], док је код учења заснованог на изазовима акценат на ширем реалном изазову и развоју применљивог решења које може имати професионални или друштвени утицај [43].

Учење засновано на решавању проблема заснива се на схватању учења према којем студенти активно стичу знање кроз рад на јединственим проблемима и интеракцији са другим учесницима у наставном процесу [41], [42], [44]. За разлику од традиционалног приступа, у којем се првобитно наставни садржај излаже, а затим примењује на одговарајућим задацима, код овог приступа проблем се представља на почетку и служи као покретач процеса учења. Проблеми су најчешће сложени, реални и недовољно структурирани, због чега захтевају анализу различитих информација, разматрање више перспектива и примену различитих знања и стратегија [41], [42].

Процес учења започиње анализом проблема, издвајањем познатих чињеница и активирањем претходног знања студената. Студенти затим формулишу могућа објашњења и решења, идентификују недостатке у сопственом знању и одређују питања и циљеве који усмеравају даље учење [41], [42], [45]. Након самосталног прикупљања и проучавања релевантних

информација, студенти се враћају групи, представљају резултате истраживања и преиспитују претходно формсулисане претпоставке. Процес се завршава применом новостеченог знања на проблем и рекапитулацијом примењених стратегија учења [41], [45]. У оквиру овог приступа студенти преузимају већу одговорност за сопствено учење, избор извора информација и допринос раду групе, док наставник напушта доминантну улогу преносиоца знања и усмерава процес учења [41], [42], [44]–[46]. Наставник је у овом случају одговоран за [46], [47]:

- осмишљавање проблема усклађених са исходима учења,
- праћење рада групе,
- постављање подстицајних питања,
- усмеравање студената и сл.

Применом учења заснованог на решавању проблема развијају се способности примене знања, решавања сложених проблема, сарадње и критичког размишљања [41], [42], [46]. Рад у мањим групама додатно подржава развој вештина комуникације, преговарања, доношења заједничких одлука и одговорности за остваривање заједничких циљева [45], [46].

У образовању у области развоја софтвера овај приступ омогућава студентима да апстрактне концепте и методе повежу са реалистичним ситуацијама из професионалне праксе. Примене у настави софтверског инжењерства показују да студенти кроз анализу конкретних сценарија могу развијати разумевање захтева, моделовање токова података, пројектовања система и других активности развоја софтвера, уз истовремени развој вештина сарадње, комуникације и управљања задацима [47], [48]. Међутим, студентима са мање искуства може бити потребан виши ниво усмеравања од стране наставника, због чега се учење засновано на решавању проблема у пракси често прилагођава конкретном наставном контексту и комбинује са другим педагошким приступима [47], [48].

Пројектно-засновано учење представља педагошки приступ у којем се процес учења организује око реализације сложеног пројекта током дужег временског периода. Студенти кроз истраживање, планирање и практичну примену знања развијају конкретан производ, прототип, извештај или другу врсту пројекта [2], [49]. Пројекат при томе не представља само завршни задатак након обраде наставног садржаја, већ основни оквир у којем студенти стичу и повезују нова знања. Овај приступ најчешће подразумева активно учешће студената, одређени степен њихове самосталности, тимски рад и рад на реално постављеним задацима [2], [49].

Пројектно-засновано учење посебно је погодно за образовање у области развоја софтвера јер студентима омогућава да теоријске концепте и методе примене током развоја софтверског решења. Рад на пројекту може обухватити анализу корисничких захтева, пројектовање, имплементацију, тестирање и писање пројектне документације, уз развој компетенција као што су комуникација, управљање временом, расподела улога и одговорност за заједнички резултат [39], [50]. Пројектно-засновано учење има значајну улогу и у смањивању јаза између академског образовања и потреба ИТ индустрије. Систематски преглед који су спровели *Aliyah* и коаутори [51] показује да се овај приступ убраја међу најчешће примењиване моделе у образовању софтверских инжењера, јер студентима омогућава рад на аутентичним развојним задацима, упознавање са животним циклусом софтвера, тимским радом и професионалним

алатима и процесима. Истовремено, аутори истичу да најбоље резултате даје повезивање пројектно-заснованог учења са агилним праксама и сарадњом са индустријом.

Будући да пројектно-засновано учење представља основни педагошки приступ примењен у истраживању спроведеном у оквиру ове дисертације, његове теоријске основе, кључне карактеристике, образовне предности, улоге наставника и студената, као и специфичности примене у образовању у области развоја софтвера, детаљније су размотрене у трећем поглављу.

Учење засновано на изазовима представља педагошки приступ у којем се процес учења организује око отвореног и сложеног изазова из реалног окружења. Изазов најчешће има шири друштвени или технолошки значај и захтева повезивање знања из различитих области, сарадњу више учесника и развој применљивог решења. За разлику од приступа у којима наставник унапред прецизно дефинише проблем, садржај и поступак рада, код учења заснованог на изазовима студенти имају већу слободу у одређивању конкретног проблема у оквиру ширег изазова, избору начина истраживања и обликовању решења. Карактеристика овог приступа је сарадња студената са наставницима, стручњацима, представницима привреде и локалне заједнице, при чему се настоји да развијено решење, поред образовне, има и практичну или друштвену вредност [40], [43], [52], [53].

Процес учења заснованог на изазовима најчешће се организује кроз три међусобно повезане фазе [54]:

1. фаза ангажовања (енг. *engage*),
2. фаза истраживања (енг. *investigate*) и
3. фаза деловања (енг. *act*).

У фази ангажовања студенти полазе од шире идеје, формулишу суштинско питање и из њега изводе конкретан изазов који представља улазну тачку за фазу деловања. Током фазе истраживања дефинишу питања која усмеравају учење, одређују потребне активности и изворе, прикупљају и анализирају информације и разматрају могућа решења. У фази деловања студенти развијају и примењују изабрано решење, испитују његову успешност и, према добијеним резултатима, врше његово даље унапређивање. Током свих фаза се могу спроводити активности као што су документовање активности, формативно оцењивање, размена повратних информација и рефлексција о процесу учења [52], [54], [55].

У оквиру овог приступа студенти преузимају одговорност за усмеравање сопственог учења, дефинисање циљева, расподелу задатака, доношење одлука и процену резултата. Наставник има улогу ментора и сарадника који помаже студентима да управљају сложеностју изазова, повежу практичне активности са исходима учења и критички преиспитују предложена решења. У процес могу бити укључени и спољни актери који обезбеђују професионални контекст, релевантне информације и повратне информације о применљивости резултата. Овакво окружење може допринети развоју доменског знања, критичког и креативног размишљања, вештина комуникације, одлучивања и способности деловања у неизвесним и мултидисциплинарним ситуацијама [40], [53], [56].

У образовању у области развоја софтвера учење засновано на изазовима омогућава студентима да примене техничка знања у развоју решења за потребе реалних корисника и заједница. Изазови могу бити организовани у облику развоја различитих софтверских производа, хакатона, такмичења и других активности које захтевају истраживање корисничких потреба, развој и вредновање решења и представљање резултата различитим заинтересованим странама [57]. Примена овог приступа на предмету који се бави развојем мобилних апликација показала је да може допринети бољем разумевању проблема и могућих решења, као и мотивацији студената, али је истовремено показала потребу за његовим повезивањем са одговарајућим процесом развоја софтвера и додатном подршком студентима који немају довољно претходног искуства [55].

Иако се разликују према начину организације наставног процеса, улози проблема, пројекта или изазова и очекиваним резултатима, наведени педагошки приступи заједнички су усмерени на активно учешће студената, примену знања у реалистичним ситуацијама, сарадњу и развој професионалних компетенција. Њихов значај у образовању у области развоја софтвера додатно расте услед брзих промена у професионалној пракси, које мењају начин на који се софтвер пројектује, имплементира и одржава. Појава *low-code* и *no-code* платформи, *GenAI* алата и *AI* агената уводи нове облике аутоматизације, чиме се мењају и вештине које будући софтверски инжењери треба да стекну.

2.2 Иновације у области развоја софтвера

Савремени развој софтвера обележава постепено померање од искључиво ручног писања програмског кода ка приступима у којима се различите развојне активности подржавају или аутоматизују специјализованим платформама и системима заснованим на вештачкој интелигенцији.

Low-code и *no-code* платформе комбинују визуелно моделовање, коришћење готових компоненти и механизме за аутоматизацију рутинских активности, чиме смањују потребу за традиционалним програмирањем и омогућавају учешће корисника са различитим нивоима техничког знања у развоју софтверских решења [58]. Истовремено, велики језички модели и други алати вештачке интелигенције примењују се у различитим активностима софтверског инжењерства, укључујући анализу корисничких захтева, пројектовање, генерисање и објашњавање програмског кода, тестирање, отклањање грешака, документовање и одржавање софтвера [59]. Даљи корак у развоју ових технологија представљају *AI* агенти, који могу да користе алате и податке из спољног окружења, планирају и извршавају низ међусобно повезаних корака и сарађују са другим агентима током решавања сложених задатака [60].

Наведене иновације не мењају само алате који се користе, већ и организацију развојног процеса, „расподелу активности“ између човека и технологије и компетенције које се очекују од будућих софтверских инжењера.

2.2.1 Развој софтвера применом low-code и no-code платформи

Low-code и *no-code* платформе представљају развојна окружења чији је циљ смањивање ручног писања програмског кода и убрзавање имплементације, постављања и измене софтверских апликација [61]. Уместо непосредног програмирања већег дела система, развој се заснива на визуелном моделовању, декларативном дефинисању правила, конфигурисању готових компоненти и повезивању са постојећим изворима података и сервисима. *Low-code* платформе задржавају могућност проширивања решења традиционалним програмским кодом и најчешће су намењене сениор програмерима и корисницима са вишим нивоом техничког знања. Насупрот томе, *no-code* платформе настоје да омогуће израду апликација искључиво визуелним и конфигурационим поступцима, због чега су првенствено намењене пословним корисницима без формалног програмерског образовања [58], [62], [63].

Иако се термин *low-code* интензивније користи од средине претходне деценије, основне идеје на којима се ове платформе заснивају нису у потпуности нове. Оне се ослањају на раније приступе, као што су брзи развој апликација, визуелно програмирање, *CASE* алати, развој вођен моделима, језици четврте генерације и поновна употреба софтверских компоненти. Њихова специфичност огледа се првенствено у интеграцији различитих развојних механизма у јединствено окружење, чиме се смањује потреба за коришћењем и усклађивањем више независних алата [58], [64].

Основне карактеристике *low-code* и *no-code* платформи обухватају следеће [58], [62], [64]:

- визуелно дефинисање модела података и корисничког интерфејса;
- употребу готових компоненти, шаблона и декларативно дефинисаних правила;
- повезивање са спољним изворима података и сервисима посредством програмских интерфејса;
- моделовање пословних процеса и токова рада;
- дефинисање корисничких улога и права приступа;
- прилагођавање апликација различитим уређајима;
- поједностављено тестирање, постављање и одржавање апликација;
- могућност проширивања решења традиционалним програмским кодом, пре свега код *low-code* платформи.

На тај начин се у једном окружењу обједињују механизми који се у традиционалном развоју често реализују употребом већег броја одвојених алата. Повећање продуктивности не произлази нужно из примене потпуно нових технологија, већ првенствено из аутоматизације рутинских активности, поновне употребе компоненти и смањивања потребе за ручним усклађивањем различитих развојних делова [58], [64]. Међу често коришћеним платформама издвајају се *OutSystems*, *Mendix* и *Microsoft Power Apps* као примери *low-code* окружења, док *Bubble* представља пример *no-code* платформе. У литератури се такође наводе *Google AppSheet*, *Appian*, *Salesforce*, *Oracle APEX* и *Zoho Creator*. Анализа коју су спровели *Silva и Avelino* [65] показала је да су *OutSystems*, *Mendix* и *Power Apps* биле најчешће разматране платформе у анализираним научним радовима.

Главне предности *low-code* и *no-code* платформи односе се на смањивање времена потребног за развој апликација, повећање продуктивности, бржу израду прототипова и лакше увођење

измена. Аутоматизација рутинских активности и поновна употреба компоненти омогућавају програмерима да већи део времена посвете пословној логици и специфичним захтевима система. Истовремено, визуелна окружења омогућавају активније учешће доменских стручњака и такозваних „грађанских програмера“ (енг. *citizen developers*) у изради решења, чиме се може побољшати комуникација између учесника из пословне и техничке области и убрзати валидација захтева. У прегледу који су спровели *Silva и Avelino* [65], смањење времена развоја и повећање продуктивности представљали су најчешће наведене предности ових платформи, након којих следе доступност ширем кругу корисника и смањење трошкова [66], [67].

Low-code и *no-code* платформе највећу вредност остварују када се примењују у пројектима чији се захтеви могу реализовати у оквиру расположивих компоненти и унапред дефинисаних механизма платформе. У таквим условима оне могу представљати допуну традиционалном развоју и омогућити брже креирање пословних апликација, прототипова и решења умерене сложености. Даљи развој ових окружења све више је повезан са применом генеративне вештачке интелигенције, која омогућава формулисање захтева природним језиком и аутоматско генерисање делова апликације [67].

2.2.2 Развој софтвера уз подршку вештачке интелигенције

Ранија примена вештачке интелигенције у софтверском инжењерству углавном је подразумевала развој специјализованих модела намењених појединачним задацима, као што су предвиђање софтверских грешака, класификација захтева, откривање сличности између делова програмског кода и препознавање безбедносних недостатака [68]. Развој великих језичких модела обучених на обимним скуповима природног језика и програмског кода омогућио је шири облик подршке, у којем се исти модел може применити на различите активности, укључујући анализу захтева, пројектовање, имплементацију, тестирање, документовање и одржавање софтвера [59].

За разлику од класичних *low-code* и *no-code* платформи, код којих корисник решење претежно развија визуелним моделовањем и конфигурисањем унапред дефинисаних компоненти, генеративна вештачка интелигенција омогућава да се намера корисника изрази природним језиком. На основу задатог промпта модел може да предложи структуру решења, генерише програмски код, објасни постојећу имплементацију или изврши тражену измену. Развојни процес тиме добија итеративни и „конверзациони“ карактер: корисник описује жељени резултат, процењује генерисани излаз и додатним инструкцијама захтева његово прецизирање или исправљање. Међутим, генерисани резултат није нужно тачан, потпун или усклађен са свим функционалним и нефункционалним захтевима, због чега његова примена захтева људску проверу и комбиновање са традиционалним методама софтверског инжењерства [68], [69].

Велики језички модели се могу применити у различитим активностима развоја софтвера, као што су [59], [70]:

- анализа и прецизирање корисничких и системских захтева;
- генерисање *user stories*, сценарија употребе и критеријума прихватања;
- генерисање, допуњавање, преводјење и објашњавање програмског кода;
- откривање грешака, исправка кода и рефакторисање;
- генерисање тестова;
- анализа квалитета програмског кода;
- генерисање коментара и пројектне документације;
- подршка одржавању софтвера;
- брза израду прототипова и генерисање већих делова апликације на основу описа природним језиком.

У фази анализе, велики језички модели могу да обрађују текстуалне описе, сажимају обиман материјал, издавају релевантне информације и превде неформалне описе у структуриране софтверске концепте. Могу се користити за формулисање *user stories* и критеријума прихватања, размену идеја (енг. *brainstorming*) и идентификовање могућих нејасноћа или недоследности [71], [72].

У фази пројектовања, генеративна вештачка интелигенција може да помогне у идентификацији компоненти система и описивању односа између њих. На основу захтева може предложити почетну организацију апликације, структуру базе података, програмске интерфејсе и начин комуникације са спољним сервисима. Такви предлози могу послужити као полазиште за дискусију, али не представљају замену за анализу архитектуре, јер моделу често није доступан потпун контекст који обухвата постојећу инфраструктуру, очекивано оптерећење, организационе стандарде и дугорочне циљеве система [59], [68].

Највидљивији облик примене вештачке интелигенције присутан је **у фази имплементације**. Велики језички модели могу да довршавају започете линије или блокове кода, генеришу функције на основу текстуалног описа, превде код из једног програмског језика у други и објашњавају непознате делове програма. Подршка може обухватити и предлагање начина употребе програмских библиотека, генерисање конфигурационих датотека, рефакторисање и усклађивање кода са задатим правилима и стилем пројекта [59]. Када алат има приступ ширем контексту репозиторијума, генерисани резултат може бити боље прилагођен структури и конвенцијама постојећег пројекта. Ипак, ограничена количина доступног контекста и недовољно разумевање целокупне структуре и зависности унутар пројекта и даље представљају значајне изазове у практичној примени ових алата [73].

Генеративна вештачка интелигенција примењује се и **у фази тестирања** софтвера. На основу имплементације, захтева или описа очекиваног понашања могу се генерисати тест сценарији, јединични тестови, граничне вредности и тестни подаци. Модели могу да анализирају поруке о грешкама, могуће узроке неуспеха и генеришу почетне верзије исправки. Систематски преглед примене великих језичких модела у тестирању показује да се међу најзаступљенијим областима налазе припрема тест сценарија. Ипак, генерисани тестови могу бити непотпуни

или засновани на истој погрешној претпоставци као и генерисана имплементација, па њихово постојање само по себи не представља доказ исправности система [74].

У оквиру активности **фазе одржавања** софтвера, модели могу сажимати пријављене проблеме, објашњавати постојећи код и измене настале између верзија, идентификовати делове репозиторијума који би могли бити повезани са грешком и генерисати пројектну документацију. Посебно су корисни у ситуацијама у којима програмер треба да се упозна са недовољно документованим или за њега непознатим системом. Међутим, сложене измене у постојећим системима захтевају разумевање ширих зависности и пословних правила, што се не може увек поуздано реконструисати само на основу делимично доступног кода [59], [68].

Примена вештачке интелигенције у софтверском инжењерству кроз различите фазе развоја софтвера приказана је на Слици 1 [75].



Слика 1. Примена вештачке интелигенције у развоју софтвера (прилагођено према [75])

Поред примене у појединачним фазама животног циклуса софтвера, генеративна вештачка интелигенција се у савременом развојном процесу најчешће примењује кроз *AI* асистенте у развоју софтвера и платформе за развој апликација на основу природног језика.

AI алати у развоју софтвера

Посебну групу алата представљају *AI* алати интегрисани у развојна окружења или доступни кроз конверзационе интерфејсе. Они могу да користе контекст активне датотеке, више датотека или ширег репозиторијума како би генерисали и допуњавали код, одговарали на питања о пројекту, предлагали рефакторисање и генерисали тестове. У ову категорију могу се сврстати *GitHub Copilot*, *Cursor*, *Claude Code*, *Gemini Code Assist* и слични алати, али предмет анализе није поређење њихових појединачних функција, већ начин на који њихова примена мења рад програмера. Ови алати су детаљно обрађени у поглављу 4.1.2.

Barke и коаутори [76] издвајају два основна начина интеракције програмера са *AI* системом за генерисање кода: режим „убрзавања“ и режим истраживања.. У режиму „убрзавања“ програмер већ зна шта треба да имплементира и користи *AI* алат како би брже добио планирани резултат. У режиму истраживања програмер нема унапред јасно решење, већ користи предлоге система за испитивање могућих приступа и упознавање са непознатим технологијама. У првом случају *AI* првенствено смањује количину рутинског рада, док у другом може утицати на сам ток решавања проблема и избор техничког решења.

Истраживања употребе *AI* асистената указују на то да се они примењују приликом имплементације нових функционалности, писања тестова, анализе грешака и рефакторисања кода. У истраживању које је обухватило 481 програмера, генерисање и сажимање кода припадали су најзаступљенијим облицима употребе, док су недостатак поверења, организационе политике и недовољно разумевање контекста целокупног пројекта наведени као значајне препреке широј примени [73]. Такође, потенцијал за повећање продуктивности зависи од задатка, искуства програмера и контекста употребе. У контролисаном експерименту који су спровели *Peng* и коаутори [23], учесници са приступом *GitHub Copilot*-у завршили су задатак имплементације *HTTP* сервера 55,8% брже од контролне групе. Овај резултат показује потенцијал алата приликом извршавања јасно дефинисаних и временски ограничених задатака, али се не може непосредно пренети на сложене пројекте, тимски рад и дугорочно одржавање софтвера. С друге стране, у новијем истраживању, искусни програмери који су решавали задатке у познатим репозиторијумима били су у просеку спорији када им је била дозвољена употреба *AI* алата, иако су пре и после експеримента процењивали да су им алати уштедели време. Овај резултат указује на разлику између субјективног осећаја продуктивности и стварног времена потребног за разумевање, проверу и исправљање генерисаног резултата [77].

Истраживања такође указују на то да програмери *AI* предлоге често користе као почетну основу, али да могу имати потешкоће приликом разумевања, измене и отклањања грешака у дужим генерисаним деловима кода. Разлози за одбацивање предлога обухватају неиспуњавање функционалних или нефункционалних захтева и недовољну контролу над излазом система. Због тога уштеда времена остварена генерисањем почетног кода може бити умањена временом потребним за његову проверу, интеграцију и исправљање [69], [78]. Квалитет добијеног резултата зависи и од начина формулисања задатка, количине доступног контекста и прецизности ограничења која су достављена моделу. Промпт може обухватити опис функционалних и нефункционалних захтева, технологија које треба користити, структуре очекиваног излаза и критеријума за проверу. Ове технике се детаљније разматрају у поглављу 4.1.3 докторске дисертације.

Платформе за развој апликација на основу природног језика

Поред алата који генеришу појединачне делове програмског кода, развијене су платформе у којима корисник природним језиком описује целокупну апликацију или њен већи део. Опис може да обухвати намену решења, корисничке улоге, екране, модел података, пословна правила, визуелне карактеристике и спољне интеграције. Систем затим генерише почетну верзију апликације, која се покреће, прегледа и итеративно мења кроз даљу конверзацију.

Овакав приступ се разликује од класичних *no-code* платформи. Код *no-code* развоја корисник углавном ручно бира, повезује и конфигурише готове визуелне компоненте. Код платформи заснованих на генеративној *AI*, велики део структуре, интерфејса и имплементације динамички се генерише на основу природног језика. Иако интеракција може деловати као *no-code* развој, у позадини се ствара стварни програмски код, који може бити доступан за преглед, преузимање и даље измене.

Међу репрезентативним примерима могу се издвојити следеће платформе, чије се карактеристике наводе на основу њихове званичне документације [79]–[82]:

1. *Lovable* омогућава креирање, итеративно унапређивање и постављање *full-stack* веб апликација на основу природног језика, при чему генерисано решење може обухватити кориснички интерфејс, пословну логику, базу података, аутентификацију и интеграције;
2. *Hercules* омогућава конверзациони развој веб сајтова, *SaaS* решења, интерних пословних алата, електронских продавница и мобилних апликација, уз подршку за базе података, пословну логику и аутентификацију.
3. *Base44* омогућава генерисање веб и мобилних апликација у интегрисаном окружењу, у којем *AI* систем планира пројекат, пише код и помаже приликом отклањања проблема;
4. *Replit Agent* на основу описа природним језиком поставља пројекат, планира реализацију, креира апликацију, проверава резултат, исправља уочене проблеме и подржава њено постављање.

Описане платформе показују постепено померање од *AI* асистента који предлаже појединачни фрагмент кода ка интегрисаном окружењу које генерише и повезује више софтверских компоненти. Корисник се више не бави искључиво писањем функција, већ формулише циљеве, описује понашање система, процењује генерисану апликацију и захтева њене измене. На овај начин се смањује почетна техничка препрека и омогућава брза израда прототипова, али се истовремено повећава ризик да корисник прихвати решење чију унутрашњу структуру, безбедност и дугорочну одрживост не може да процени.

Потребно је нагласити да се наведене платформе брзо развијају и да поједине већ поседују изражене агентске функције. *Replit Agent*, на пример, не ограничава се на давање текстуалног одговора, већ поставља пројекат, проверава свој рад и исправља проблеме, док *Hercules* користи агенте за планирање, измену кода и тестирање апликације.

У приступима представљеним у овом поглављу човек и даље има централну улогу у покретању интеракције, формулисању задатака, избору предлога и провери резултата. Даљи степен аутоматизације остварује се у системима у којима *AI* самостално разлаже сложени задатак,

планира редослед активности, користи развојне алате, покреће код и тестове и на основу добијених резултата врши нове измене. Овакви системи представљају основу софтверског инжењерства заснованог на *AI* агентима.

2.2.3 Софтверско инжењерство засновано на *AI* агентима

Софтверско инжењерство засновано на *AI* агентима (енг. *Agentic Software Engineering - ASE*), које се често означава и као софтверско инжењерство 3.0 (*SE 3.0*), представља прелазак са развоја софтвера уз подршку вештачке интелигенције ка примени аутономних *AI* система који преузимају одговорност за активности у свим фазама животног циклуса развоја софтвера [83], [84]. За разлику од традиционалних *AI* асистената, који углавном генеришу програмски код као одговор на појединачни захтев корисника, *AI* агенти могу да анализирају циљ, формулишу план, користе развојне алате, провере резултат и прилагоде наредне активности на основу добијених повратних информација [85].

AI агент у софтверском инжењерству се најчешће заснива на великом језичком моделу, има приступ развојним алатима и одређени степен самосталности у њиховој употреби. Агент може да претражује репозиторијум, мења програмски код, покреће тестове, користи компајлер и алате за анализу, а затим на основу резултата настави или измени свој план рада. Због тога се софтверско инжењерство засновано на *AI* агентима не може свести само на *prompt engineering*, већ обухвата планирање, употребу алата, извршавање активности и итеративну проверу резултата [86].

Систематски преглед литературе, који обухвата период од 2022. до 2025. године, указује на значајан пораст аутономности софтверских система. Део истраживања у којима су представљени системи са „високим степеном аутономности“, способни да извршавају сложене задатке уз минималан надзор човека, повећао се са 18% у 2022. години на 73% у 2024. години [87]. Овај прелазак ка вишем степену аутономности приметан је и у образовном контексту, у којем интеграција алата заснованих на генеративној вештачкој интелигенцији мења начин на који се извршавају задаци у области софтверског инжењерства [88].

Структура и категоризација приступа софтверског инжењерства заснованог на *AI* агентима приказана је на Слици 2.



Слика 2. Таксономија приступа софтверског инжењерства заснованог на *AI* агентима (прилагођено према [87])

Према овој таксономији, софтверско инжењерство засновано на *AI* агентима може се класификовати у пет основних домена [87]:

1. Системи са једним агентом (енг. *Single-Agent Systems*) - усмерени су на аутономно генерисање програмског кода и самостално откривање и исправљање грешака [89], [90].
2. Системи са више агената (енг. *Multi-Agent Systems*) - заснивају се на сарадњи тимова агената са различитим улогама, као што су софтвер архитекта, програмер и тестер, чији се рад координира посредством платформи као што су *ChatDev* [91] и *MetaGPT* [92].
3. Сарадња човека и агента (енг. *Human-Agent Collaboration*) - обухвата интерактивни преглед програмског кода и програмирање у пару уз подршку вештачке интелигенције, при чему човек задржава улогу коначног валидатора [83], [93].
4. Агенти проширени алатима (енг. *Tool-Augmented Agents*) - подразумевају интеграцију агената са алатима у развојним окружењима, *CI/CD* процесима и изолованим окружењима за извршавање и проверу задатака [94], [95].
5. Хибридни приступи (енг. *Hybrid Approaches*) - комбинују више модела, као што је приступ *Mixture of Experts (MoE)* [96], као и адаптивне оквире за решавање разноврсних и хетерогених задатака [92].

Ипак, примена *AI* агената у софтверском инжењерству праћена је бројним изазовима, међу којима се посебно издвајају питања поверења, етички аспекти, промене у понашању модела током времена и сложеност координације у системима са више агената [93].

2.3 Усаглашавање образовања у области развоја софтвера са *SWEBOK* стандардом

Софтверско инжењерство представља област чији се садржај непрекидно проширује услед развоја нових технологија, метода, развојних окружења и начина организације софтверских пројеката. Због тога је приликом дефинисања образовних програма неопходно обезбедити равнотежу између фундаменталних знања, практичних вештина и савремених професионалних компетенција. Један од најзначајнијих референтних оквира за систематизацију знања у овој области представља *SWEBOK* (енг. *Guide to the Software Engineering Body of Knowledge*), који развија *IEEE Computer Society* [34].

SWEBOK представља међународно признат стандард и референтни оквир у области софтверског инжењерства, који обухвата систематизацију знања и пракси неопходних за професионалну примену у овој области [34]. Основни циљеви *SWEBOK*-а обухватају успостављање уједињеног међународног разумевања софтверског инжењерства, одређивање граница ове области у односу на сродне дисциплине, систематизацију садржаја професионалне области, омогућавање тематског приступа релевантним знањима и обезбеђивање основе за развој наставних програма, професионалну сертификацију и лиценцирање [34]. *SWEBOK* је намењен различитим групама корисника, укључујући студенте, наставнике, истраживаче, организације које развијају образовне програме и тела задужена за сертификацију и акредитацију.

Развој *SWEBOK*-а прати развој софтверског инжењерства као професионалне дисциплине. Након издања из 2004. године и верзије три (v3) из 2014. године, верзија четири (v4) објављена је 2024. године са циљем унапређивања актуелности, доследности, читљивости и

употребљивости оквира. Све области знања су ажуриране како би обухватиле савремене методе, праксе и стандарде. Агилни приступи и *DevOps* интегрисани су у већи број области знања, док су машинско учење и вештачка интелигенција укључени у фундаменталне области. У односу на претходну верзију, као посебне области додате су софтверска архитектура, безбедност софтвера и операције софтверског инжењерства [34]. *SWEBOK* верзија 4 обухвата укупно 18 области знања (енг. *Knowledge Areas - KA*) које су приказане на Слици 3.



Слика 3. *SWEBOK* области знања

Првих петнаест области непосредно се односи на софтверско инжењерство, док последње три представљају рачунарске, математичке и опште инжењерске основе на којима се ова дисциплина заснива [34], [97], [98]:

1. **Прикупљање корисничких захтева** (енг. *Software Requirements*) подразумева прикупљање, анализу и спецификацију функционалних и нефункционалних захтева. Циљ је да се прецизно дефинише шта систем треба да ради и у којим условима ће функционисати.
2. **Архитектура софтвера** (енг. *Software Architecture*) односи се на структуру софтверског система, компоненте архитектуре и њихове односе, архитектуралне стилове, документовање и вредновање архитектуре.
3. **Пројектовање софтвера** (енг. *Software Design*) подразумева превођење захтева у структурална и детаљна решења. Пројектовање обухвата моделовање система, дефинисање интерфејса, структуре података и алгоритама, као и креирање *UML* дијаграма.
4. **Имплементација софтвера** (енг. *Software Construction*) односи се на реализацију фазе пројектовања кроз писање и интеграцију кода. У ову фазу спадају кодирање, верзионисање кода и управљање конфигурацијама.
5. **Тестирање софтвера** (енг. *Software Testing*) је фаза чији је циљ провера испуњености захтева и квалитет система. У овој фази врши се развој и извршавање различитих типова тестова - јединичних, интеграционих, и др.
6. **Операције софтверског инжењерства** (енг. *Software Engineering Operations*) односе се на постављање, пуштање у рад и оперативно управљање софтвером, мониторинг, управљање инцидентима, доступност, континуитет и аутоматизацију оперативних активности.
7. **Одржавање софтвера** (енг. *Software Maintenance*) представља фазу у којој се софтвер, након пуштања у рад, прилагођава новим захтевима, исправљају се грешке и врши се

оптимизација перформанси. Одржавање чини најдужу фазу животног циклуса, јер систем мора да остане употребљив у променљивом окружењу.

8. **Управљање конфигурацијом софтвера** (енг. *Software Configuration Management*) обухвата идентификацију конфигурационих јединица, контролу измена, управљање верзијама, праћење стања конфигурације, ревизије и управљање издањима.
9. **Управљање софтверским пројектом** (енг. *Software Engineering Management*) односи се на планирање и праћење пројектата, процену трошкова, времена и ресурса, управљање ризицима, квалитетом и реализацијом пројектата.
10. **Процес софтверског инжењерства** (енг. *Software Engineering Process*) обухвата моделе животног циклуса, дефинисање, прилагођавање, мерење, процену и унапређивање процеса развоја софтвера.
11. **Модели и методе софтверског инжењерства** (енг. *Software Engineering Models and Methods*) обухватају принципе и језике моделовања, структурне моделе, прототиповање, формалне и хеуристичке методе и агилне приступе.
12. **Квалитет софтвера** (енг. *Software Quality*) односи се на планирање, обезбеђивање, контролу и мерење квалитета, атрибуте квалитета, верификацију и валидацију и активности унапређивања квалитета.
13. **Безбедност софтвера** (енг. *Software Security*) обухвата безбедносне захтеве, безбедно пројектовање и конструисање, безбедносне обрасце, тестирање безбедности и управљање рањивостима.
14. **Професионална пракса софтверског инжењерства** (енг. *Software Engineering Professional Practice*) обухвата етичка и правна питања, професионалну одговорност, стандарде, тимски рад, комуникацију, сарадњу са заинтересованим странама и поступање у условима неизвесности и сложености.
15. **Економика софтверског инжењерства** (енг. *Software Engineering Economics*) односи се на економске аспекте доношења одлука, процену трошкова и користи, продуктивност, пословне моделе, вредновање алтернатива и оправданост улагања у софтверска решења.
16. **Рачунарске основе** (енг. *Computing Foundations*) обухватају архитектуру рачунара, структуре података и алгоритме, програмске језике, оперативне системе, базе података, рачунарске мреже, људске факторе, вештачку интелигенцију и машинско учење.
17. **Математичке основе** (енг. *Mathematical Foundations*) обухватају логику, скупове, релације, функције, графове, вероватноћу, нумеричке методе и друге математичке концепте неопходне за разумевање и формализацију софтверских проблема.
18. **Инжењерске основе** (енг. *Engineering Foundations*) обухватају инжењерски процес и пројектовање, апстракцију, моделирање, симулацију, мерење, емпиријске и експерименталне методе, стандарде и анализу основних узрока проблема.

Наведене области не треба посматрати као потпуно независне целине. Развој софтвера подразумева њихово међусобно повезивање, јер одлуке донете у једној области непосредно утичу на друге области. На пример, захтеви утичу на архитектуру, пројектовање, имплементацију и тестирање, док управљање конфигурацијом, квалитетом и безбедношћу прожима већи део животног циклуса. *SWEBOK* зато није организован искључиво према линеарним фазама развоја, већ обухвата техничке, организационе, управљачке, професионалне и фундаменталне аспекте софтверског инжењерства. Свака област знања у оквиру *SWEBOK*-а има хијерархијску структуру и разлаже се на подобласти. Увод у област одређује њен обухват и однос са другим областима, након чега следи систематизација кључних процеса, метода, техника, алата, улаза и излаза. На крају сваке области налазе се препоручене референце, додатна литература и матрица која повезује појединачне теме са релевантним изворима.

SWEBOK садржи и прилоге са правилима за опис области знања, прегледом одговарајућих *IEEE* и *ISO/IEC* стандарда и обједињеним списком литературе.

У образовном контексту *SWEBOK* може да се користи као референтна основа за проверу обухвата наставног програма, дефинисање исхода учења и идентификовање области које нису довољно заступљене. Међутим, његово усаглашавање са студијским програмом не подразумева једноставно увођење по једног предмета за сваку област знања. Поједине области могу бити распоређене кроз више предмета, лабораторијских вежби и студентских пројеката, док један предмет може истовремено обухватати теме из више области. *SWEBOK* првенствено одређује шта припада општеприхваћеном знању софтверског инжењерства, али не прописује како, којим редоследом и на којем нивоу то знање треба подучавати. *Alarifi* и коаутори [99] предложили су процес развоја и евалуације наставних програма софтверског инжењерства *SECDEP* (енг. *Software Engineering Curricula Development and Evaluation Process*), у којем се *SWEBOK* користи као основа за систематичну процену покривености области знања. Предложени процес повезује анализу наставног програма са објективним и субјективним показатељима, као и са захтевима на ИТ тржишту, чиме се омогућава континуирано идентификовање недостатака, преклапања и могућности за унапређивање предмета и исхода учења.

Само присуство одређене теме у наставном програму није довољно за процену остварених компетенција. Потребно је размотрити и дубину њене обраде и ниво на којем студенти могу да примене стечена знања. *Dolog* и коаутори [48] су у ту сврху повезали *SWEBOK* области знања са Блумовом таксономијом [100] и анализирали предмете и студентске пројекте у оквиру студијског програма софтверског инжењерства. Њихов приступ показује да се знање о одређеној области може кретати од познавања и разумевања основних концепата до њихове примене, анализе, повезивања и критичког вредновања у сложенем пројектном контексту. Резултати њихове анализе указују на посебан значај пројектно-заснованог учења за постизање виших когнитивних нивоа. Кроз вишесеместралне тимске пројекте студенти не усвајају само изолована теоријска знања, већ анализирају захтеве, бирају технологије и методе, пројектују и имплементирају решења, тестирају их и образлажу донете одлуке.

3 Пројектно-засновано учење у образовању у области развоја софтвера

3.1 Појам и основе пројектно-заснованог учења

Пројектно-засновано учење представља наставни приступ који ставља студента у центар образовног процеса. Засновано је на реалним проблемима и питањима, а обухвата колаборативни рад, постављање циљева, размену идеја и израду конкретних пројеката или презентација. Наставник у овом моделу има улогу ментора, док студенти преузимају већу одговорност за сопствено учење и напредак [2], [49]. *PBL* доприноси повећању мотивације, критичког размишљања, комуникационих и социјалних вештина, као и дубљем разумевању наставних садржаја. Ипак, примена у пракси може бити праћена бројним изазовима јер захтева добро управљање временом, пажљив избор тема, систематично праћење и вредновање рада студената, као и адекватне ресурсе. *PBL* се издваја као приступ који обогаћује наставни процес и подстиче студенте да развијају компетенције потребне за савремено радно окружење [101].

Да би пројектно-засновано учење било заиста делотворно, потребно је да пројекти садрже неколико кључних елемената [102]:

1. Потреба за знањем - пројекат се покреће активностима које изазивају интересовање
2. Покретачко питање - јасно и атрактивно питање које усмерава рад и даје сврху пројекту
3. Активно учешће студената - могућност да студенти самостално одлучују о теми, приступу или начину представљања резултата
4. Савремене вештине - колаборативни рад, развој критичког размишљања, комуникационих вештина и способност коришћења технологије
5. Истраживање и иновација - истраживање, формулисање нових питања и креирање оригиналних решења
6. Повратне информације и ревизија - континуирано унапређивање рада кроз повратне информације и савете
7. Јавно представљање - завршни резултат који се излаже или користи у ширем контексту, изван учионице

3.2 Образовне предности пројектно-заснованог учења

Пројектно-засновано учење је приступ који омогућава студентима да активно уче кроз решавање аутентичних проблема и развој конкретних пројеката. Једна од најзначајнијих предности овог приступа јесте развијање и когнитивних и „практичних“ компетенција. Студенти не усвајају само теоријска знања већ их примењују у контексту који одражава реалне изазове, што доводи до дубљег разумевања и дугорочног памћења градива [103].

Могу се издвојити четири кључне фазе пројектно-заснованог учења у образовном контексту [104]:

1. Иницијална фаза (енг. *initiation*) - укључује преглед теоријских концепата, дефинисање проблема или изазова, постављање истраживачких питања и развој потребних вештина. У овој фази студенти добијају основне информације и мотивишу се за рад на пројекту

2. Фаза планирања (енг. *planning*) - обухвата организацију тимова, расподелу улога и одговорности, као и израду плана активности и задатака. Ово је тренутак када се дефинишу конкретни кораци за реализацију пројекта
3. Фаза извођења (енг. *execution*) - подразумева пројектовање и развој решења, прикупљање података, израду прототипа, тестирање и постепено унапређивање. У овој фази се пројекат финализује и припрема за презентацију
4. Завршна фаза (енг. *closing*) - обухвата евалуацију коначног решења, добијање повратних информација, као и рефлексију о процесу и усвојеним знањима. Овај корак омогућава студентима да сагледају шта су постигли и да извуку поуке за будуће пројекте

Истраживања показују да *PBL* развија вештине тимског рада, комуникације и решавања проблема, које су све чешће међу кључним захтевима савременог тржишта рада. Студенти истичу да овакво учење подстиче креативност и способност интеграције знања из различитих области [105]. Поред тога, рад на пројектима јача мотивацију јер студенти препознају значај онога што уче и добијају прилику да донекле самостално управљају процесом учења [106]. Дугогодишње искуство примене *PBL*-а у инжењерском образовању показује да овај приступ омогућава развој техничких, личних и друштвених компетенција кроз рад на стварним проблемима у сарадњи са индустријом и заједницом [107].

3.3 Примена пројектно-заснованог учења у образовању у области развоја софтвера

Развој софтвера је област у којој је пројектно-засновано учење нарочито погодан приступ, јер омогућава студентима да се суоче са проблемима и практичним задацима сличним онима из професионалног окружења. На овај начин долази до интеграције теоријских знања и њихове непосредне примене у развоју софтверских решења [108].

У том процесу посебно је важно да студенти раде у тимовима, где се кроз поделу улога и сарадњу симулира организација стварних развојних тимова. Оваква динамика не само што омогућава стицање техничких компетенција већ развија и вештине комуникације, координације и заједничког решавања проблема, које су кључне у савременом софтверском инжењерству [109].

Поред тога, интеграција пројектно-заснованог учења у наставном процесу у области развоја софтвера показала се ефикасном у повећању мотивације и развоју професионалних вештина студената. *PBL* подстиче студенте да активно преузму одговорност за сопствено учење, истражујући различите улоге у тиму и пролазећи кроз фазе развоја стварних пројеката. Ротација улога и подела на мање потпројекте омогућили су им да стекну искуство у лидерству, вођењу документације и контроли квалитета, док је комбинација наставничке и вршњачке процене подстакла критичко размишљање и самосталну процену рада. Резултати су показали да овакво структурирано *PBL* искуство не само да повећава ангажованост и мотивацију студената, већ и доводи до бољих укупних академских постигнућа [50]. Интегрисани *PBL* приступи показују да пројекти у настави софтверског инжењерства могу покривати читав животни циклус развоја софтвера.

Главни фокус је на [110]:

- прикуљању захтева (формулисање, моделирање и валидација захтева)
- пројектовању и имплементацији софтвера
- тимском раду (сарадња, расподела улога и развијање способности за рад у већем контексту)
- управљању пројектом (израда плана пројекта и координација активности)
- документацији (припреми стандардизованих извештаја и спецификација).

3.4 Улоге наставника и студената у пројектно-заснованом учења

У пројектно-заснованом учењу улоге наставника и студената значајно се разликују од оних у традиционалној настави. Наставници више не делују искључиво као предавачи, већ преузимају улогу ментора и водича у процесу учења. Они усмеравају студенте ка истраживању, пружају подршку у решавању проблема и подстичу критичко размишљање, али не намећу готова решења [111]. Наставник у овом моделу мора да поседује низ компетенција [112]:

- лидерске способности за усмеравање тима и доношење кључних одлука,
- страст за менторством и спремност да подстиче студенте,
- способност прилагођавања различитим нивоима знања унутар групе,
- вештине управљања временом ради праћења пројектних рокова,
- отвореност за експериментисање и увођење иновација,
- емпатију и инспирацију као подршку студентима у процесу учења,
- креативно размишљање како би подстакли развој иновативних решења.

Поред улоге ментора и водича, наставници у *PBL*-у имају и кључну одговорност у области формативног оцењивања. За разлику од оцењивања које се спроводи на крају наставне јединице, формативно оцењивање је континуиран процес праћења учења и давања повратних информација током пројекта. Оно омогућава наставницима да на основу студентских одговора и активности идентификују ниво разумевања, уоче погрешне концепције и благовремено прилагоде наставу [113]. Студенти, са друге стране, преузимају активну и одговорну улогу у процесу учења. Од њих се очекује да формулишу проблеме, креирају планове пројектата, сарађују у тимовима и процењују сопствени и групни рад. Овај вид активности доприноси развоју самопоуздања и јачању осећаја ефикасности [114]. Поред тога, укључивање у заједничко решавање комплексних задатака подстиче унутрашњу мотивацију студената и развија осећај припадности групи [115]. Студенти постају склонији да преузимају изазове, истрајније раде на комплексним задацима и показују већу одговорност за сопствено учење. Овакви исходи указују да *PBL* не само да утиче на квалитет усвојених знања, већ и на развој личних и професионалних вештина студената [116].

Истраживања показују да у *PBL* окружењу наставници обликују услове који охрабрују истраживање и примену знања, док студенти кроз тај процес стичу не само академска већ и социјална искуства. На пример, рад [117] показује да је управо педагогија усмерена на сарадњу и заједничко истраживање у *PBL* срединама кључна за развој разумевања и дугорочне мотивације студената. Тако се улоге наставника и студената преплићу у заједничком процесу истраживања и стварања знања, где је наставник подршка, а студент активни конструктор сопственог учења.

4 Генеративна вештачка интелигенција у образовању

Вештачка интелигенција (енг. *artificial intelligence*) је грана рачунарске науке која се фокусира на стварање система способних да обављају задатке који обично захтевају људску интелигенцију. Ови задаци укључују учење, решавање проблема, разумевање језика, перцепцију и доношење одлука. *AI* има за циљ да развије алгоритме и modele који омогућавају машинама да имитирају когнитивне функције, прилагођавају се новим информацијама и раде аутономно у сложеним окружењима [118], [119].

Вештачка интелигенција се може класификовати према различитим критеријумима, при чему је најчешћа подела према нивоу способности која обухвата следеће категорије [120]:

1. Вештачка интелигенција у ужем смислу (енг. *narrow AI*) - системи дизајнирани за обављање специфичних задатака унутар ограниченог домена, попут препознавања лица, обраде говора или генерисања препорука. Ови системи се истичу у извршавању унапред дефинисаних функција, али не поседују свест, опште разумевање или способности изван свог програмирања [121].
2. Вештачка интелигенција у општем смислу (енг. *general AI*) - хипотетички ниво *AI* који би могао да разуме, учи и примењује знање на широком спектру задатака, опонашајући когнитивне способности човека. Овај ниво интелигенције још увек није достигнут и представља предмет интензивних истраживања у области *AI*, у фокусу развоја система који могу генерализовати знање сличним људима [122].
3. Супер вештачка интелигенција (енг. *super AI*) - теоријски концепт који описује *AI* систем који надмашује људску интелигенцију у свим областима, укључујући креативност, социјалну интелигенцију и доношење одлука. Ови системи би показивали висок степен аутономије и когнитивних способности, што отвара бројне филозофске, технолошке и етичке дилеме. Суперинтелигенција се често сматра футуристичким циљем, праћеним расправама о безбедности и етичким импликацијама [123].

4.1 Генеративна вештачка интелигенција

4.1.1 Појам и основе генеративне вештачке интелигенције

Генеративна вештачка интелигенција обухвата методе и системе способне да стварају нови садржај као што су текст, слика, звук или код на основу образаца научених из постојећих података [124]–[126]. Да би се разумеле основе ове технологије, потребно је представити концепте на којима она почива. У њих спадају машинско учење (енг. *machine learning*) као широка методолошка основа, дубоко учење (енг. *deep learning*) као доминантан технички приступ у савременим системима, и генеративна вештачка интелигенција као скуп техника усмерених на стварање новог садржаја.

Машинско учење (енг. *Machine Learning* - ML)

Машинско учење представља једно од кључних подручја вештачке интелигенције и дефинише се као скуп метода и алгоритама који омогућавају системима да аутоматски побољшавају своје перформансе кроз искуство, односно анализу података, без експлицитног програмирања сваког корака [127]–[129]. Суштина машинског учења огледа се у способности алгоритама да идентификују обрасце у подацима и да их користе за доношење предвиђања или доношење одлука у новим ситуацијама. Овај приступ омогућио је примену вештачке интелигенције у бројним доменима, укључујући медицину, финансије, индустрију и образовање, где традиционални програмски приступи нису били довољно ефикасни [130].

Основне парадигме машинског учења могу се поделити на [130]–[132]:

1. Надгледано машинско учење (енг. *supervised learning*) - модел се обучава на подацима који имају унапред познате ознаке (енг. *labels*). Циљ је предвиђање излаза за нове, неозначене улазе. Типични примери су класификација докумената или регресија за предвиђање цена.
2. Ненадгледано машинско учење (енг. *unsupervised learning*) - алгоритам ради са подацима који немају ознаке и настоји да открије структуру унутар њих. Овај приступ је посебно значајан у откривању скривених образаца у великим и неозначеним корпусима. Најчешће технике укључују кластеровање и редукцију димензионалности.
3. Машинско учење са подршком (енг. *reinforcement learning*) - модел учи кроз интеракцију са окружењем. Понашање система се оптимизује на основу повратних информација у виду награда и казни, што омогућава доношење све бољих одлука у динамичним условима.

Захваљујући овим парадигмама, машинско учење представља полазиште за све напредније облике вештачке интелигенције, укључујући дубоко учење и генеративне моделе. Оно пружа основе за изградњу алгоритама који не само да анализирају и предвиђају, већ и омогућавају системима да се прилагођавају и еволуирају у динамичким и комплексним окружењима.

Дубоко учење (енг. *Deep Learning* - DL)

Дубоко учење представља један од најзначајнијих приступа у оквиру машинског учења и заснива се на примени вишеслојних вештачких неуронских мрежа ради моделирања сложених и нелинеарних односа у подацима. Основни градивни елемент ових система су вештачки неурони, који су повезани у слојеве и омогућавају постепено апстраховање информација, од једноставних карактеристика на нижим слојевима до апстрактних и комплексних концепата на вишим слојевима [133]–[135].

Захваљујући способности да уче репрезентације директно из података, дубоке неуронске мреже постале су темељ већине савремених система у областима као што су препознавање слика, аутоматска обрада говора, обрада природног језика и, нарочито, генеративни задаци [129], [136].

Дубоко учење обухвата више архитектура, међу којима се издвајају [135], [137]:

1. Конволуционе неуронске мреже (енг. *Convolutional Neural Network - CNN*) - ове мреже су инспирисане радом визуелног кортекса и најчешће се примењују у задацима компјутерског вида. Њихова структура обухвата конволуционе и пулинг слојеве који постепено издвајају карактеристике из улазних података, од ивица и облика до комплексних објеката. Познате варијанте су *LeNet*, *AlexNet*, *VGG*, *ResNet* и *Inception*.
2. Рекурентне неуронске мреже (енг. *Recurrent Neural Networks - RNN*) - за разлику од *CNN*-а, ове мреже су намењене секвенцијалним подацима и захваљујући повратним везама могу да задрже информације о претходним стањима. Модерније верзије попут *LSTM* (енг. *Long Short-Term Memory*) и *GRU* (енг. *Gated Recurrent Unit*) превазилазе проблем нестајања градијената и омогућавају успешно моделирање дугорочних зависности у тексту, говору и временским серијама.
3. Аутоенкодери - архитектуре које уче компресовану представу улазних података и затим покушавају да их реконструишу. На тај начин мрежа издваја битне карактеристике и елиминише шум. Посебно су корисни за смањење димензионалности, претпроцесирање података и детекцију аномалија.
4. Генеративне супарничке мреже (енг. *generative adversarial neural network - GAN*) - састоје се од два подсистема: генератора, који ствара нове податке, и дискриминатора, који их разликује од стварних. Њихова интеракција доводи до стварања веома реалистичних синтетичких података.

Посебно место у савременом развоју дубоког учења заузима трансформер (енг. *transformer*) архитектура. Представљен 2017. године у раду „*Attention is All You Need*“ [138], трансформер је заснован на механизму *self-attention*, који омогућава моделу да обрађује све елементе секвенце истовремено и да ефикасно учи односе између њих. За разлику од *RNN*-а, трансформери немају ограничења у погледу дугорочних зависности и знатно су бржи захваљујући паралелној обради података. Ова архитектура постала је основа модерних великих језичких модела као што су *BERT*, *GPT* и *T5*. Трансформер се данас сматра једном од најзначајнијих архитектура дубоког учења, јер је омогућио значајан развој мултимодалних система и напредне генеративне моделе [139].

Обрада природног језика (енг. *Natural Language Processing - NLP*)

Обрада природног језика представља интердисциплинарну област која проучава начине на које рачунари могу да разумеју, анализирају и генеришу људски језик у писаном или говорном облику. Циљ *NLP*-а је да обезбеди „приближно људско“ процесирање језика ради остваривања широког спектра задатака, од машинског превођења и препознавања говора, до апликација као што су системи за претраживање информација, интелигентни дијалогски системи и аутоматско сажимање текста [140]–[142]. Савремени *NLP* заснива се на обради језика на више нивоа, који се често комбинују у зависности од конкретног задатка [142]–[144]:

- морфолошки ниво - анализа речи и њихових граматичких облика,
- лексички ниво - одређивање значења речи и њихових категорија,
- синтаксички ниво - утврђивање структуре реченице и односа између речи,
- семантички ниво - интерпретација значења речи и реченица,
- дискурсни ниво - анализа повезаности између реченица у тексту,
- прагматички ниво - разумевање значења у контексту и у складу са знањем о свету

У технолошком смислу, *NLP* је еволуирао од симболичких метода заснованих на правилима и граматикама, преко статистичких модела и техника машинског учења као што су *n*-грамови, класификатори и *Хидени Марковљеви* модели, до дубоког учења и трансформер архитектура које доминирају савременим приступима [145].

Примене *NLP*-а су бројне и разноврсне. Најчешће се користи у задацима као што су [141], [143]:

- машинско превођење (нпр. *Google Translate*),
- препознавање и синтеза говора (нпр. *Siri*, *Alexa*),
- системи за претраживање и извлачење информација,
- аутоматско сажимање текста и класификација докумената,
- анализа сентимента и препознавање емоција,
- интелигентни дијалошки системи и *chatbot*-ови

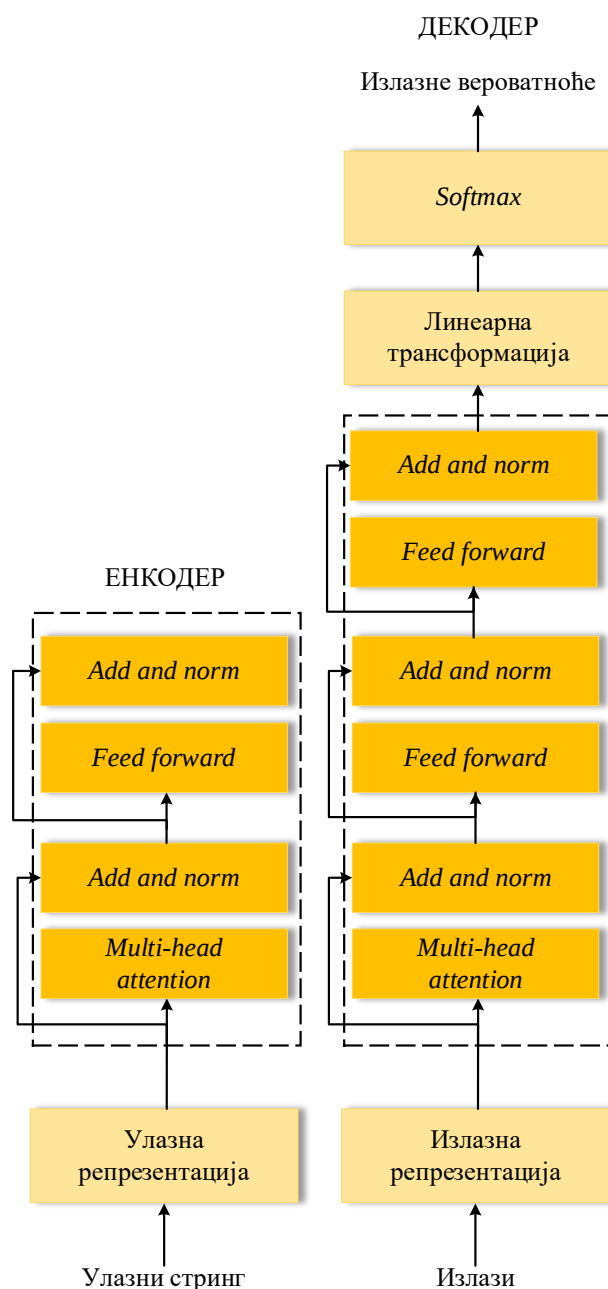
Посебан помак у развоју обраде природног језика остварен је кроз појаву великих језичких модела који представљају најзначајнији резултат примене дубоког учења и трансформер архитектура у области *NLP*-а.

Велики језички модели (енг. *Large Language Models - LLMs*)

Велики језички модели омогућавају аутоматску обраду и генерисање природног језика на нивоу који је до недавно био сматран доменом искључиво људске интелигенције. Они су засновани на трансформер архитектури и тренирани на огромним корпусима текстуалних података, што им омогућава да уче статистичке односе између речи и контекста и да их користе у задацима као што су машинско превођење, резимирање, одговарање на питања, генерисање кода и дијалошка интеракција [146], [147]. Њихова основна функција је предвиђање наредног токена у низу, али у пракси ова једноставна поставка доводи до појаве комплексних способности као што су резонување, доношење одлука и учење у контексту, што их чини изузетно моћним алатима за различите домене примене [146]. Истраживања су показала да повећање обима параметара и количине података доводи не само до бољих резултата већ и до квалитативних скокова у могућностима модела, што се често описује као „емергентне способности“ [148].

Посебан значај у развоју генеративних *LLM*-ова има породица модела *GPT* (енг. *Generative Pretrained Transformer*). *GPT-1* је 2018. године први пут показао снагу претходног тренирања на великим корпусима без надзора, док је *GPT-2* из 2019. значајно унапређен повећањем броја параметара и способности да решава разноврсне *NLP* задатке. Прекретница је био *GPT-3* са 175 милијарди параметара, који је показао да модел може да производи текст упоредив са људским и да решава задатке без експлицитног фино подешавања, искључиво на основу упутстава у облику *prompt*-ова. *GPT-4* уводи мултимодалност, што значи да поред текста може обрађивати и друге типове података, попут слика, чиме се додатно проширују његове могућности [149]. На Слици 4 је приказана *GPT-3* архитектура [150], [151]. Слика приказује како се улазни текст токенизује и претвара у векторске репрезентације, након чега се обрађује кроз низ слојева самопажње (енг. *self-attention*) и *feed-forward* неуронских мрежа. Посебан акценат је стављен на скалабилност: повећање броја слојева и параметара довело је до

значајног побољшања у капацитету модела за генерацију природног језика. Овај приказ наглашава улогу дубоких слојева у хватању дугорочних зависности у тексту и показује зашто је управо *GPT-3* постао прекретница у развоју *LLM*-ова [150].



Слика 4. *GPT-3* архитектура

Најновији модел, *GPT-5*, објављен је у августу 2025. године и представља најзначајнији искорак у развоју генеративних језичких модела. За разлику од претходних верзија, *GPT-5* користи унификовану архитектуру која омогућава динамичко пребацивање између брзог одговарања и дубоког резонувања у зависности од сложености упита, чиме се постиже бољи баланс између ефикасности и прецизности [152]. Модел показује значајно унапређене способности у програмирању, математици и медицини, где у мултимодалним задацима дијагностике надмашује *GPT-4* [153].

У практичној примени *GPT* модели пролазе кроз два кључна корака: прво се тренирају на масивним корпусима јавних података, а затим се фино подешавају и усклађују са људским повратним информацијама (*reinforcement learning with human feedback - RLHF*), како би били корисни, тачни и безбедни у интеракцији са корисницима [148]. Управо је овај приступ омогућио настанак апликација попут *ChatGPT*, које данас представљају основ савремених генеративних система.

Упркос својим импресивним могућностима, *LLM*-ови се суочавају са бројним изазовима. Међу најважнијим су питања транспарентности и објашњивости, јер се често описују као „црне кутије“, затим појава пристрасности у одговорима, као и феномен „халуцинација“ - генерисање садржаја који делује уверљиво, али је фактички нетачан [154]. Све ово показује да ће *LLM*-ови, а нарочито генеративни модели попут *GPT*-а, остати у фокусу будућих истраживања и практичних примена, као кључни покретачи даљих иновација у области вештачке интелигенције.

У наставку рада биће дат систематски преглед и критичка анализа алата генеративне вештачке интелигенције који су одабрани на основу учесталости примене у пракси и препознате релевантности у стручним публикацијама.

4.1.2 Преглед и анализа алата генеративне вештачке интелигенције

У овом поглављу биће представљени и анализирани следећи алати генеративне вештачке интелигенције:

1. *ChatGPT*
2. *Google Gemini*
3. *GitHub Copilot*
4. *Claude Code*
5. *Tabnine*
6. *Perplexity AI*
7. *DeepSeek* и
8. *Grok AI*.

ChatGPT (OpenAI)

ChatGPT, који је развио *OpenAI*, представља један од најзначајнијих генеративних језичких модела, изграђен на архитектури трансформера, са способношћу да производи кохерентан и контекстуално прилагођен текст [150], [155]. Од када је представљен крајем 2022. године, овај модел постао је централна тачка у дебатама о употреби вештачке интелигенције у академским и образовним контекстима, али и у другим доменима [156]. *ChatGPT* омогућава корисницима да у интерактивном облику добију одговоре на питања, да генеришу текстове различитог стила, па чак и да симулирају дијалог у складу са задатим параметрима. Његова примена је вишеструка: од подршке у писању и истраживању до едукативних и пословних активности [157].

Развој модела унапредио је и начин функционисања, па тако најновија верзија, *GPT-5*, представљена у августу 2025. године, доноси низ нових функционалности [158]. *GPT-5* је

унапређен кроз јединствени систем који комбинује брзи модел за једноставније захтеве, напредни *reasoning* модел за сложене задатке и динамички механизам усмеравања који током дијалога одлучује о избору оптималног модула. Овај приступ омогућава ефикасније и прилагођеније одговоре у зависности од комплексности упита. Значајан напредак односи се и на увођење тзв. *thinking mode* функције, која омогућава моделу да разликује ситуације у којима је потребан брз одговор од оних које захтевају дубље размишљање, чиме се постиже експертски ниво у решавању комплексних проблема. Поред тога, *GPT-5* је у већој мери мултимодалан у односу на претходне генерације, јер поред текста може обрађивати и слике, као и одређене аудио и видео садржаје, што значајно проширује опсег примене. Посебна пажња посвећена је смањењу такозваних халуцинација, односно генерисања нетачних или измишљених података, што је било једно од кључних ограничења ранијих верзија. Новина је и побољшана могућност генерисања програмског кода, као и подршка за рад са обимним контекстом који достиже стотине хиљада токена, што омогућава анализу великих докумената или дуготрајних разговора без губитка кохерентности. *GPT-5* је такође увео персонализоване стилове комуникације (енг. *personalities*) и додатне механизме за безбедност и етику, уз истовремено јачање интеграције са алатима за продуктивност као што су *Microsoft 365 Copilot* и други пословни системи [158].

Према бројним истраживањима, домени примене овог алата обухватају образовање, здравство, пословање, истраживање, као и свакодневну комуникацију и подршку корисницима. *ChatGPT* већ мења академску реалност кроз подршку истраживачима у генерисању нових идеја, проналажењу извора и припреми научних радова, али истовремено отвара питања академског интегритета и поузданости генерисаног садржаја [157], [159]. У образовању се користи за побољшање наставног процеса, подстицање критичког мишљења и олакшавање сарадње између наставника и студената [160], [161]. У раду [162] је наглашен значај пажљивог планирања интеграције овог алата у наставу како би се избегли етички и педагошки проблеми, док се у раду [163] истиче да *ChatGPT* може подстицати продуктивност и креативност, али и смањити аутономију студената.

Google Gemini

Google Gemini, развијен од стране компаније *Google DeepMind*, представља напредну генерацију великих мултимодалних модела, дизајнираних да комбинују текстуалне, визуелне и друге облике података у јединственом систему [164]. Као наследник модела *Bard* [165], *Gemini* је осмишљен да превазиђе ограничења класичних језичких модела интеграцијом дубоког разумевања различитих модалитета и могућношћу решавања сложених когнитивних задатака [166]. *Gemini* омогућава снажну подршку за сложено резоновање, генерисање кода, као и боље управљање дугим контекстима, чиме се позиционира као конкурент *ChatGPT*-у и другим генеративним системима [167].

Функционалности модела обухватају напредно мултимодално резоновање, способност обраде текста, слике и звучних података, као и интегрисано решавање комплексних проблема у реалном времену [164]. У односу на претходне генеративне моделе, *Gemini* поседује побољшану ефикасност у радним окружењима која захтевају тачност и прецизност, што је посебно значајно у медицинским и научним дисциплинама. У поређењу са *ChatGPT*-ом,

показало се да *Gemini* може пружити квалитетнију подршку у клиничком доношењу одлука, нарочито у хируршким случајевима у офталмологији [168]. Поред тога, у раду [169] се указује да *Gemini* има потенцијал да буде ефикаснији од *ChatGPT*-а у процени квалитета истраживачких радова, чиме доприноси унапређењу научне продуктивности.

У области образовања, *Gemini* је већ препознат као значајан алат који може подржати нове моделе наставе и учења. У раду [170] истичу да *Gemini* представља „следећу генерацију образовних алата“, са потенцијалом да унапреди персонализовано учење, пружи студентима интерактивне и прилагођене материјале, као и да наставницима понуди нове начине за процену знања. *Gemini*, у поређењу са другим моделима за генерисање текста, показује напредније способности у разумевању контекста и одржавању конзистентности у дужим дијалозима [171].

Док се *ChatGPT* показао као изузетно користан у генерисању академских текстова и подршци студентима кроз интерактивни дијалог [157], [160], *Gemini* доноси предност у мултимодалности и прецизнијем резоновању у специфичним професионалним контекстима као што су медицина и процена научног квалитета [168], [169]. У раду [171] закључују да, иако *ChatGPT* остаје доминантан у образовним и општим истраживачким сценаријима, *Gemini* показује потенцијал да преузме водећу улогу у задацима који захтевају дубље когнитивне способности и интеграцију више модалитета.

GitHub Copilot

GitHub Copilot, развијен у сарадњи између *GitHub*-а и *OpenAI*-а, представља један од првих масовно доступних алата за програмирање уз подршку генеративне вештачке интелигенције. Заснован на великим језичким моделима обученим на обимном коду из јавних репозиторијума, *Copilot* функционише као *AI pair programmer* који програмерима нуди сугестије кода у реалном времену, аутоматско довршавање функција и генерисање нових програмских сегмената [172], [173]. Његова основна сврха је убрзавање процеса програмирања, смањење рутинског рада и омогућавање програмерима да се фокусирају на пројектовање и логичку структуру софтверских решења [174].

Истраживања показују да *Copilot* може значајно олакшати учење програмирања, посебно код почетника. У наставном контексту овај алат може деловати као ментор који пружа тренутне предлоге и примере, чиме се смањује когнитивно оптерећење студената и омогућава брже савладавање синтаксе [175]. Овај алат може ефикасно решавати једноставне програмске проблеме, али важно је нагласити да квалитет генерисаног кода није увек на нивоу који би био прихватљив за сложеније задатке. Често је потребна људска интервенција ради побољшања читљивости и оптимизације кода [176], [177].

У поређењу са генеративним моделима опште намене, као што су *ChatGPT* или *Gemini*, *GitHub Copilot* је уско специјализован и дизајниран за конкретан домен, програмирање. Док су *ChatGPT* и *Gemini* фокусирани на мултимодалност, генерацију текста и широку примену у образовању, медицини или науци [157], [164], *Copilot* показује највећи потенцијал као практичан алат за повећање продуктивности у развоју софтвера. Његова предност лежи у директној интеграцији са развојним окружењима (*IDE*) и могућности да у реалном времену

реагује на захтеве програмера, али истовремено захтева сталну контролу и критички надзор како би се избегли проблеми са поузданошћу и безбедношћу кода.

Claude Code

Claude Code, који је развила компанија *Anthropic*, представља агентски алат за програмирање намењен раду директно у окружењу пројекта. За разлику од класичних *AI* асистената, који углавном предлажу или допуњују појединачне сегменте кода, *Claude Code* може самостално да анализира структуру репозиторијума, чита и мења датотеке, покреће команде у терминалу, извршава тестове и комуницира са спољним сервисима. Његово функционисање засновано је на итеративној петљи у којој модел анализира задатак, бира одговарајуће алате, извршава акције, процењује добијене резултате и понавља поступак све док задатак не буде завршен. На тај начин, *Claude Code* превазилази улогу алата за аутоматско довршавање кода и приближава се систему способном да реализује вишекорачне активности током различитих фаза развоја софтвера [178].

Прилагођавање понашања алата конкретном софтверском пројекту врши се, између осталог, посредством конфигурационе датотеке *CLAUDE.md*. У њој програмери могу дефинисати архитектуру система, структуру директоријума, стандарде писања кода, команде за изградњу и покретање апликације, правила тестирања и препоручени начин рада. *Claude Code* чита ова упутства приликом обраде корисничких захтева, чиме се повећава вероватноћа да ће генерисане измене бити усклађене са постојећим техничким и организационим правилима пројекта. Анализа 328 конфигурационих датотека из јавних пројеката показала је да су најчешће дефинисана правила повезана са архитектуром софтвера, развојним смерницама, описом пројекта, тестирањем и командама за рад са пројектом. Правила за архитектуру пројекта била су присутна у 72,6% анализираних датотека, што указује на значај експлицитног усмеравања агента приликом рада на сложеним системима [179].

Истраживања *Claude* модела за генерисање кода показују да његова успешност зависи од домена примене и врсте задатка. У истраживању генерисања *Java* кода за управљање роботским системима, *Claude 3.5 Sonnet* успешно је решио 19 од 20 задатака, односно остварио стопу успешности од 95%. Модел је показао добре способности просторног резонувања, праћења инструкција кроз више корака и генерисања добро структурисаног кода [180]. Са друге стране, поређење *ChatGPT*-а, *Claude Sonnet*-а и *DeepSeek*-а на задацима имплементације алгоритама сортирања показало је да је *Claude* генерисао функционалан код, али са нешто већом потрошњом меморије, дужим временом извршавања и нижим просечним резултатом читљивости у односу на друга два модела [181].

Tabnine

Tabnine представља један од пионирских генеративних алата специјализованих за аутоматско допуњавање кода, који користи дубоке моделе машинског учења обучене на великим корпусима отвореног кода са дозвољеним лиценцама, а ради унутар популарних развојних окружења. За разлику од општих генеративних језичких модела као што су *ChatGPT* или *Gemini*, *Tabnine* је фокусиран на контекстуално предлагање фрагмената кода и тиме омогућава програмерима да раде ефикасније и продуктивније у току свакодневног развоја. Његове главне

функционалности укључују интелигентно аутоматско допуњавање, генерисање кода из упита у природном језику, као и ревизију кода са циљем повећања безбедности и приватности у развоју софтвера [182].

Систематска поређења показују да *Tabnine* производи краће и концизније сегменте кода у односу на конкуренте, али истовремено са нижим степеном коректности. У једном емпиријском истраживању које је упоредило више алата на сету од 100 *Python* задатака, *Tabnine* је имао најнижу стопу исправности (19%), али је генерисао најкраћи код у просеку што га чини погодним за брзе прототипе и мање задатке, док је за сложеније проблеме био мање поуздан. Истраживање такође указује да *Tabnine* производи код умерене структурне сложености и релативно брзо (3, 4 секунде), што га позиционира као алат који балансира између једноставности и ефикасности [183].

Када је у питању развој тестова, *Tabnine* показује капацитет да генерише *unit* тестове сличне онима које пишу програмери. Истраживања су показала да *Tabnine*, заједно са *Copilot*-ом и *ChatGPT*-ом, може да генерише тестове упоредивог квалитета са оригиналним, при чему често постиже преко 85% покривености у комплексним сценаријима. Упоређивања са ручно писаним тестовима откривају да је стил генерисаног кода другачији (нпр. употреба модернијих библиотека и др), али да функционално покрива потребне случајеве и може да допринесе бржој аутоматизацији тестирања [184].

Tabnine је, у поређењу са другим алатима, посебно погодан за организације које траже већу контролу над приватношћу кода, јер омогућава локално извршавање модела без слања података на екстерне сервере. Ова карактеристика га чини привлачним за компаније које развијају осетљив или власнички софтвер, где је безбедност података приоритет [182].

Perplexity AI

Perplexity AI представља генеративни алат који се развио као конверзациони претраживач, усмерен на побољшање корисничког искуства кроз интеграцију претраге и дијалошке интеракције. За разлику од класичних језичких модела, *Perplexity* је конципиран као *answer engine* који омогућава добијање одговора у реалном времену уз директно навођење извора и повезивање са релевантним онлајн садржајем [185]. Његова највећа предност у односу на друге моделе јесте транспарентност у приказивању извора и могућност да истовремено делује као систем за претраживање и као конверзациони агент, чиме се корисницима омогућава проверљивост генерисаних садржаја [186].

Истраживања указују да *Perplexity* може да генерише садржаје који су усклађени са темом и логички структурирани, при чему квалитет одговора зависи од детаљности и формулације упита. Захваљујући интеграцији са изворима, *Perplexity* омогућава корисницима да одмах провере кредибилност понуђених информација, што га чини специфичним у односу на друге генеративне системе [187].

DeepSeek

DeepSeek представља један од новијих модела вештачке интелигенције који је у кратком временском периоду стекао глобалну пажњу захваљујући комбинацији високих перформанси, нижих трошкова обуке и отвореног приступа. Његов развој означава значајан помак у парадигмама великих језичких модела, јер се уместо искључивог ослањања на повећање параметара и обима података акценат ставља на ефикасност, оптимизацију алгоритама и нове методе закључивања [188]. Архитектура *DeepSeek*-а укључује низ иновативних техника као што су *Multi-head Latent Attention (MLA)*, *Mixture-of-Experts (MoE)*, *Multi-Token Prediction (MTP)* и *Group Relative Policy Optimization (GRPO)*, чиме се постиже боља обрада комплексних задатака и унапређује скалабилност система [188], [189].

Модел је објављен у више итерација, међу којима се издвајају V3 и R1 серије. *DeepSeek-V3*, са 671 милијарди параметара, користи *MoE* архитектуру која омогућава да се појединачним токенима активира ограничен број експерата, чиме се значајно смањује рачунарски трошак. Ова верзија је у тестовима показала перформансе упоредиве са водећим затвореним моделима, али уз вишеструко мање трошкове тренирања, који су процењени на око шест милиона долара, у поређењу са више стотина милиона за конкуренте [189]. Поређења са *ChatGPT*-ом показују да *DeepSeek* остварује боље резултате у генерисању кода, посебно у алгоритамским задацима где постиже већу тачност при мањем броју покушаја. *ChatGPT* се са друге стране показује ефикаснијим у брзини и једноставности кода, али чешће захтева више итерација за добијање исправног решења. Ове разлике указују на то да је *DeepSeek* усмерен на дубље алгоритамско резонување, док *ChatGPT* остаје флексибилнији [190]. Слични налази у ширим компаративним студијама потврђују да *ChatGPT* предњачи у генералистичким и креативним задацима, док *DeepSeek* пружа прецизније одговоре у техничким и научним областима [191].

За разлику од већине западних модела, *DeepSeek* је развијен у кинеском контексту и одликује се изразито отвореним приступом, а неке верзије модела доступне су под *MIT* лиценцом. Такав приступ подстиче брзу усвојивост у академским и индустријским окружењима, али истовремено изазива забринутост око безбедности, геополитичких импликација и дугорочне одрживости у односу на комерцијално затворене моделе [189].

Grok AI

Grok AI је један од најновијих генеративних модела, развијен од стране компаније *xAI*, коју је основао Илон Маск. Први пут је представљен крајем 2023. године и убрзо интегрисан у платформу *X* (раније *Twitter*), где служи као интелигентни конверзациони агент са директним приступом подацима у реалном времену са *X* платформе [192]. Његова специфичност огледа се у дизајну који укључује елементе персонализованог изражавања, чиме се разликује од конвенционалних четбот система. Такав приступ омогућава интеракције које нису искључиво информативне, већ су структурисане тако да комбинују пренос података са облицима комуникације прилагођеним динамици друштвених мрежа [193].

На техничком плану, *Grok* је изграђен на трансформер архитектури, уз примену техника преноса знања (енг. *transfer learning*) и адаптивног финог подешавања (енг. *adaptive fine-tuning*), што му омогућава да генерише одговоре у контексту друштвених мрежа и прилагођава

се брзим променама у онлајн дискусијама [194]. Кључна предност овог система је његова интеграција са платформом *X*, што му омогућава да пружа увиде засноване на трендовима у реалном времену. Са друге стране, оваква усмереност представља и ограничење: зависност од једног извора података може резултирати информационим пристрасностима и смањити применљивост модела у општијим контекстима [193].

Упоредне анализе овог алата са другима као што су *ChatGPT*, *Gemini* и *DeepSeek* показују да сваки од ових система има различиту стратешку оријентацију. Док *ChatGPT* доминира у генералистичким задацима и образовању, а *Gemini* у мултимодалним применама и продуктивности, *Grok* се позиционира као модел који тежи већој слободи израза, мањим ограничењима у садржају и дубљој интеграцији са друштвеним мрежама [193], [194]. У том смислу, *Grok* не тежи да буде универзални замена за постојеће моделе, већ више представља специфичну алтернативу намењену корисницима којима је релевантност у реалном времену и интерактивност са елементима личности важнија од класичне кохерентности или мултимодалних капацитета [195]. У Табели 1 дат је приказ кључних карактеристика горе наведених алата.

Табела 1. Алати генеративне вештачке интелигенције - кључне карактеристике

Алат	Развијен	Кључне функционалности	URL адреса
ChatGPT	<i>OpenAI</i> (2022)	Генерасање текста, дијалог, кôд, мултимодалност (GPT-5)	https://chat.openai.com/
Google Gemini	<i>Google DeepMind</i> (2023)	Мултимодални модел, напредно резонување, интеграција са <i>Google</i> сервисима	https://gemini.google.com/
GitHub Copilot	<i>GitHub & OpenAI</i> (2021)	<i>AI pair programming</i> , аутоматско допуњавање кôда, интеграција у <i>IDE</i>	https://github.com/features/copilot
Claude Code	<i>Anthropic</i> (2025)	Анализа и измена кôда, покретање команди и тестова, <i>Git</i> интеграција	https://claude.com/
Tabnine	<i>Codota</i> (2019)	Аутоматско допуњавање кôда, фокус на приватност, концизност	https://www.tabnine.com/
Perplexity	<i>Perplexity AI</i> (2022)	<i>Answer engine</i> , интеграција претраге и дијалога, навођење извора	https://www.perplexity.ai/
DeepSeek	<i>DeepSeek AI</i> , (2023)	<i>MoE</i> архитектура, <i>MLA</i> , <i>MTP</i> , високе перформансе уз ниже трошкове, фокус на кинеско тржиште	https://chat.deepseek.com/
Grok AI	<i>xAI, Elon Musk</i> (2023)	Интеграција са <i>X</i> платформом, одговори у реалном времену, персонализовани израз	https://grok.com/

Разноврсност генеративних алата и њихове специфичне функционалности показују да квалитет и ефикасност добијених резултата у великој мери зависе од начина на који корисник формулише упите. Управо зато наредно поглавље посвећено је техникама *prompt engineering*-а, као кључном приступу за оптимално коришћење могућности ових алата.

4.1.3 Технике *prompt engineering*-а

Prompt engineering представља процес обликовања и прилагођавања улазних упита (енг. *prompts*) како би се понашање великих језичких модела усмерило ка жељеним резултатима. За разлику од класичног *fine-tuning*-а који подразумева измену параметара модела, *prompt engineering* делује искључиво на нивоу уноса, што га чини ефикаснијим и флексибилнијим у различитим доменима [196].

Кључни део ове дисциплине чине различите технике *prompt engineering*-а, од којих се издвајају следеће [196]–[199]:

1. *Zero-shot prompting* - модел добија инструкцију без примера, што омогућава брзу примену на широком спектру задатака
2. *Few-shot prompting* - модел добија неколико примера као смернице, на основу чега учи образац решавања
3. *Role/persona prompting* - модел преузима одређену улогу (нпр. наставник, програмер) како би се добили специфичнији одговори
4. *Chain-of-Thought (CoT) prompting* - модел се подстиче да разлаже своје размишљање у кораке (корисно у сложеним проблемима)
5. *Prompt chaining* - повезивање више промптова у низ, где излаз једног постаје улаз наредном
6. *Retrieval-augmented generation (RAG)* - комбинација улазних упита са екстерним изворима знања ради већег квалитета и релевантности резултата
7. *Self-criticism/self-consistency prompting* - модел се наводи да провери или упореди сопствене одговоре како би се добила поузданија решења
8. Примером вођени приступ - употреба апстракција или пажљиво одабраних примера за специфичне софтверске задатке

Поред техника, литература наглашава и одређене смернице за успешан *prompt engineering*. Најважније су јасноћа и прецизност инструкција, увођење експлицитних ограничења, као и итеративно тестирање и усавршавање упита [200]. Систематско усавршавање ових техника и развој стандарда за евалуацију указују на то да ће *prompt engineering* постати једна од дисциплина неопходна за оптималну примену генеративне вештачке интелигенције у различитим доменима, а посебно у софтверском инжењерству [201], [202].

4.2 AI агенти

Развој великих језичких модела омогућио је прелазак са система који само генеришу одговоре на корисничке упите ка AI агентима способним за самостално и циљно усмерено деловање [87], [89], [203]. За разлику од уобичајене примене генеративне вештачке интелигенције, засноване на једнократној размени упита и одговора, AI агенти могу да анализирају постављени циљ, планирају активности, користе различите изворе података и алате и

прилагођавају наредне кораке на основу резултата претходних активности. На тај начин, велики језички модел више не представља само средство за генерисање садржаја, већ постаје компонента сложенијег система који повезује расуђивање, доношење одлука и деловање у дигиталном или физичком окружењу [204], [205].

Иако су агенти присутни у области вештачке интелигенције знатно пре појаве великих језичких модела, њихове могућности су дуго биле ограничене на унапред дефинисане задатке, правила и контролисана окружења. Традиционални агенти могли су да опажају одређене промене у окружењу и да на њих реагују, али су углавном имали ограничену способност прилагођавања неструктурираним улазима, формулисања сложених планова и комуникације природним језиком. Велики језички модели проширили су овај концепт обезбеђујући флексибилнији механизам за тумачење инструкција, разумевање контекста, рашчлањивање задатака и избор активности. Савремени *AI* агент зато најчешће представља сложени систем у којем је језички модел повезан са меморијом, спољашњим изворима података, програмским алатима и механизмима за контролу тока извршавања [204], [206].

У литератури још увек не постоји потпуна терминолошка усаглашеност у погледу односа између појмова *AI* агент и „агентска вештачка интелигенција“ (енг. ***agentic AI***). Појам *AI* агент најчешће означава конкретан софтверски ентитет који опажа окружење, расуђује о расположивим информацијама и предузима активности ради извршавања одређеног задатка. Насупрот томе, поједини аутори појам *agentic AI* користе за сложеније системе састављене од више специјализованих агената који расподељују задатке, комуницирају, користе заједничку меморију и координирано раде на остваривању ширег циља [207], [208]. Други аутори овај појам користе у ширем значењу, за све *AI* системе који показују наглашену аутономију, проактивност, прилагодљивост и способност дугорочнијег остваривања циљева, без обзира на то да ли је реч о једном или више агената [206], [209].

Основне карактеристике савремених *AI* агената обухватају аутономију, циљно усмерено понашање, реактивност, проактивност, прилагодљивост и способност интеракције са окружењем. Аутономија подразумева да агент може да извршава одређене активности без непрекидних инструкција корисника, док циљна усмереност означава да се активности не предузимају изоловано, већ у функцији остваривања постављеног циља. Реактивност се односи на способност одговарања на промене и нове информације, док проактивност подразумева покретање активности које нису непосредно изазване новим корисничким упитом, али су неопходне за наставак рада. Прилагодљивост се огледа у способности промене стратегије на основу повратних информација, резултата претходних активности или новонасталих околности [209], [210]. Аутономија при томе не подразумева потпуно одсуство људске контроле. У пракси, она може бити ограничена правилима, овлашћењима, доступним алатима, временом извршавања и врстом одлука које агент сме да донесе. Ниво аутономије треба да буде усклађен са природом задатка и потенцијалним последицама одлуке. Код задатака ниског ризика агент може самостално извршити већи број корака, док је у осетљивим областима потребно обезбедити људску проверу пре предузимања активности које могу имати значајне последице. Због тога се савремени агентски системи често развијају као облици сарадње човека и вештачке интелигенције, при чему агент обавља анализу и припрему активности, а човек задржава надзор и одговорност за кључне одлуке [205], [211].

Основне функције агената заснованих на великим језичким моделима најчешће обухватају: опажање и прикупљање информација из окружења, планирање, управљање меморијом и интеракцију са спољним алатима [93], [203], [212]. У новијим прегледима као посебне функције издвајају се и расуђивање, извршавање активности, рефлексија, вредновање резултата и оркестрација рада једног или више агената [204], [208]. Иако се конкретне архитектуре разликују, већина њих заснива се на затвореној повратној петљи у којој агент прикупља информације, ажурира унутрашње стање, доноси одлуку, предузима активност и процењује њен резултат. **Опажање и прикупљање информација из окружења** представља почетну фазу овог процеса и обухвата прикупљање и тумачење релевантних сигнала из окружења. У зависности од примене, улазне информације могу обухватити текстуалне упите, документе, слике, аудио и видео садржаје, податке из информационих система, резултате претраге или токове података у реалном времену. Опажање се не своди само на техничко примање података, већ подразумева њихово повезивање са претходним контекстом и процену релевантности за текући задатак. На основу овако формираног приказа окружења агент може да утврди тренутно стање и изабере наредну активност [213]. **Расуђивање** представља механизам који повезује опажање са деловањем. Оно обухвата тумачење циља, анализу расположивих информација, поређење могућих активности и избор наредног корака. За разлику од једнократног генерисања одговора, агентско расуђивање најчешће се одвија кроз више повезаних фаза. Агент може да формулише почетни план, провери да ли су доступне све неопходне информације, активира одговарајући алат и затим ревидира план на основу добијеног резултата. На овај начин контрола тока извршавања није у потпуности унапред дефинисана, већ зависи од тренутног стања задатка и одлука које систем доноси током рада [213], [214]. **Планирање** омогућава претварање ширег циља у низ управљивих подциљева и активности. Уместо да покуша непосредно да генерише коначан резултат, агент најпре може да утврди које информације недостају, којим редоследом треба извршити активности и који критеријуми ће бити примењени за процену резултата. Код једноставнијих задатака план може бити кратак и линеаран, док сложенији задаци захтевају условно гранање, понављање корака и враћање на претходне фазе. Посебно значајна способност је поновно планирање, односно измена стратегије када изабрани поступак не доведе до очекиваног резултата или када се услови у окружењу промене [206], [211]. **Меморија** омогућава агенту да очува континуитет током интеракције и да претходно прикупљене информације користи приликом доношења наредних одлука. Краткорочна, односно радна меморија чува податке релевантне за тренутни задатак, као што су кориснички захтев, међурезултати и резултати активираних алата. Дугорочнији механизми меморије могу чувати раније интеракције, искуства, обрасце решавања задатака и информације о кориснику. У системима са више агената може постојати и заједничка меморија преко које агенти размењују резултате и одржавају заједнички приказ стања задатка. Трајнија меморија представља једно од важних обележја агентских система јер омогућава остваривање циљева током дужег временског периода и кроз више одвојених фаза интеракције [207], [211]. **Интеракција са спољним алатима** омогућава агенту да превазиђе ограничења знања садржаног у параметрима језичког модела. Алати могу обухватити претраживаче, базе података, системе за управљање документима, програмске интерфејсе, календаре, системе електронске поште, рачунарске интерпретере и друге специјализоване апликације. Језички модел у таквој архитектури не мора самостално да извршава све операције, већ може да одреди који је алат потребан, формулише одговарајући позив,

протумачи добијени резултат и искористи га у наставку рада. Повезивање са спољним изворима омогућава приступ актуелним и доменски специфичним информацијама, док повезивање са извршним алатима омогућава да агент не само предложи одређену активност већ и да је, у оквиру додељених овлашћења, непосредно изврши [204], [207].

Комбинација наведених функција омогућава агентима да превазиђу статичан модел рада заснован на директном повезивању улазног захтева и генерисаног одговора. Уместо тога, агент може да формира план, изврши низ међусобно повезаних активности, процени добијене резултате и, уколико је потребно, измени претходно изабран приступ. Овај процес се може представити као циклус опажање-расуђивање-деловање, који се понавља све док се не оствари постављени циљ, не испуни услов за прекид или не затражи интервенција корисника. Након сваке активности агент добија нове информације из окружења, ажурира меморију и процењује да ли је потребно наставити према постојећем плану или формулисати нови [213].

Значајну улогу у овом циклусу има **рефлексија**, односно способност система да процени квалитет сопственог резултата и препозна могуће недостатке. Рефлексија може подразумевати поређење резултата са постављеним критеријумима, проверу помоћу екстерног алата, анализу повратне информације корисника или укључивање другог агента задуженог за валидацију. Уколико резултат није задовољавајући, агент може да понови активност, промени промпт, употреби други алат или измени план. Ипак, самопроцена језичког модела није гаранција исправности, јер агент може поновити или додатно појачати почетну грешку. Због тога је потребно комбиновати рефлексију са независним изворима провере и јасно дефинисаним условима за заустављање извршавања [208], [211].

Значајан напредак у способности модела да решавају сложене задатке постигнут је применом приступа као што је „ланац мисли“ (енг. *Chain-of-Thought* - *CoT*), у оквиру којег модел генерише међукорак расуђивања пре формулисања коначног резултата [215]. Овај приступ може да олакша рашчлањивање задатка и структурирање процеса решавања, посебно када је потребно повезати више информација или извршити низ логички зависних операција. Ипак, генерисани међукораци не морају увек верно представљати механизам на основу којег је модел дошао до резултата, нити сами по себи обезбеђују његову тачност. Због тога се у агентским системима овај приступ комбинује са употребом алата, провером резултата и повратним петљама [204].

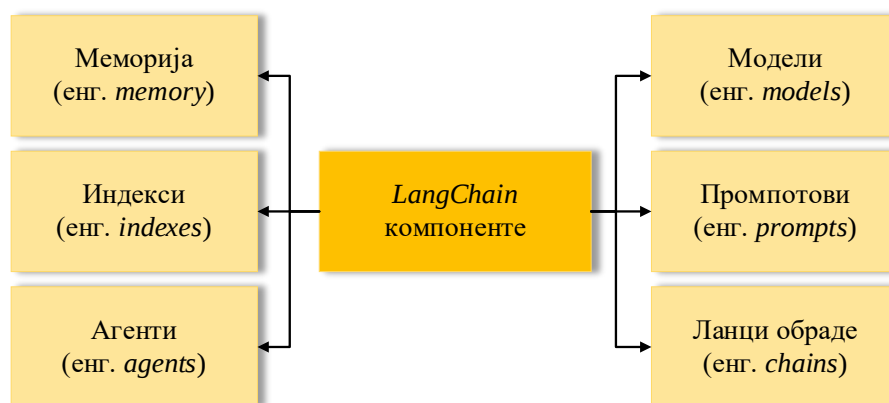
Даљи развој је постигнут концептом *ReAct* (енг. *Reasoning and Acting*), који повезује расуђивање и деловање агента [84], [212]. Применом овог концепта, агент наизменично анализира тренутно стање и активира одговарајуће алате, као што су претраживачи, интерпретери програмског кода, базе података или системи за превођење. Резултат активiranог алата постаје нова опсервација на основу које агент наставља расуђивање и бира наредну активност. На тај начин, резултат рада агента не заснива се искључиво на знању садржаном у параметрима језичког модела, већ и на информацијама које се динамички добијају из спољашњег окружења [94], [216]. *ReAct* тиме представља један од значајних образаца за реализацију затворене петље расуђивања, деловања и провере резултата [207], [208].

Агентски системи могу бити организовани као системи са једним агентом или као вишеагентски системи. У систему са једним агентом исти агент обавља планирање, употребу алата, анализу резултата и формулисање коначног одговора. Овакав приступ је погодан за релативно добро дефинисане задатке и једноставнији је за развој, праћење и контролу. Међутим, како се број активности и сложеност циља повећавају, један агент може бити оптерећен великим бројем инструкција, алата и међузависних информација, што повећава могућност губитка контекста и неконзистентног извршавања. Системи са више агената распоређују активности између већег броја специјализованих агената. Појединачни агенти могу бити задужени за планирање, претраживање извора, анализу података, генерисање решења, проверу резултата или комуникацију са корисником. Њихов рад може бити координисан централним оркестратором или организован кроз децентрализовану комуникацију. Специјализација улога омогућава расподелу сложених задатака и независну проверу појединих резултата, али истовремено уводи проблеме комуникације, синхронизације и усаглашавања циљева. Грешка једног агента може бити пренета другим агентима преко заједничке меморије, док нејасно дефинисане улоге могу довести до дуплирања активности или међусобно неусклађених одлука [204], [207], [211].

Због наведених карактеристика, процена успешности *AI* агената сложенија је од процене самог језичког модела. Тачност коначног резултата представља важан, али не и довољан критеријум. Потребно је разматрати и успешност остваривања циља, поузданост употребе алата, способност опоравка од грешке, број и оправданост предузетих корака, време извршавања, трошкове, безбедност и степен потребне људске интервенције. *Кароор* и коаутори [214] указују да сложеније агентске архитектуре не морају нужно бити успешније од једноставнијих приступа и да се повећање тачности понекад постиже великим бројем поновљених позива моделу, уз значајно веће трошкове. Поред тога, недостатак стандардизованих поступака евалуације и репродуктивности може довести до претерано оптимистичних закључака о стварним способностима агената. Зато евалуација треба истовремено да обухвати квалитет резултата, ефикасност, робусност и применљивост у реалним условима.

У образовању, *AI* агенти могу представљати основу за развој система који не пружају само једнократне одговоре, већ континуирано прате процес учења и предузимају активности у складу са напредовањем студента. Они могу да анализирају резултате претходних активности, препознају недостатке у знању, прилагоде редослед и тежину садржаја, обезбеде правовремене повратне информације и предложе додатне материјале или задатке. На институционалном нивоу, агенти могу подржати административне активности, саветовање студената, организацију материјала и друге рутинске процесе, чиме се наставницима оставља више времена за педагошки рад и непосредну интеракцију са студентима [208], [209].

За практичну реализацију система заснованих на *AI* агентима развијени су различити програмски оквири, међу којима значајно место заузима *LangChain*. Овај оквир обезбеђује модуларне компоненте за развој апликација и агената заснованих на великим језичким моделима [217], [218]. Основне компоненте *LangChain* оквира приказане су на Слици 5 [219].



Слика 5. *LangChain* компоненте

Компоненте приказане на Слици 5 обухватају:

1. Моделе (енг. *models*) - представљају стандардизоване интерфејсе за повезивање апликације са различитим моделима вештачке интелигенције. Ова компонента обухвата велике језичке моделе, који генеришу и обрађују текст, као и моделе за формирање векторских репрезентација текста, који се користе приликом семантичког претраживања.
2. Промптове и шаблоне промптова (енг. *prompts*) - дефинишу инструкције, контекст и формат у којем се задатак прослеђује језичком моделу. Шаблони омогућавају динамичко и виšekратно формирање промптова уносом различитих вредности, улога и контекстуалних података.
3. Ланце обраде (енг. *chains*) - повезују више компоненти и операција у јединствен ток извршавања, при чему резултат једног корака може представљати улаз у наредни корак. Ланци могу бити секвенцијални, паралелни или условни, у зависности од структуре задатка.
4. Меморију (енг. *memory*) - омогућава чување претходних интеракција, међурезултата и других података релевантних за даље извршавање задатка. На тај начин систем може да очува контекст разговора и прилагоди одговоре информацијама добијеним током претходних корака.
5. Индексе и механизме за преузимање података (енг. *indexes*) - омогућавају повезивање система са спољашњим изворима знања, као што су документи, веб-странице и базе података. Они обухватају читавање и поделу докумената, формирање и складиштење векторских репрезентација, као и проналажење делова садржаја који су најрелевантнији за постављени упит.
6. Агенте (енг. *agents*) - повезују језички модел са доступним алатима и механизмима за доношење одлука. За разлику од унапред дефинисаних ланаца, агент може динамички да одреди коју активност треба извршити, који алат треба употребити и да ли је потребно поновити неки од корака ради остваривања постављеног циља.

Њиховим комбиновањем могуће је развити систем у којем се кориснички захтев обрађује кроз више повезаних корака, уз приступ додатним изворима информација и очување контекста интеракције.

У оквиру *LangChain* екосистема развијен је и *LangGraph*, програмски оквир намењен оркестрацији сложенијих агентских токова применом управљања стањем и усмерених графова [203], [216]. За разлику од статичних и линеарних токова обраде, *LangGraph* подржава условно

гранање, понављање корака и циклично извршавање. Ове карактеристике посебно су значајне за задатке који захтевају итеративно побољшавање резултата, поновну процену добијеног одговора или враћање на неки од претходних корака [216], [220]. У *LangGraph* архитектури, чворови графа представљају појединачне функције, активности или специјализоване агенте, док гране одређују ток информација и услове под којима се прелази са једне активности на другу. Такође, *LangGraph* подржава и концепт „човек у петљи“ (енг. *human-in-the-loop*), у оквиру којег се аутономно извршавање може привремено зауставити како би корисник или други одговорни актер проверио резултат, унео додатне информације или одобрио наставак рада [203], [216].

4.3 UNESCO оквир за AI компетенције наставника и студената

UNESCO оквир за компетенције у области вештачке интелигенције представља глобални референтни алат намењен подршци образовним системима у припреми наставника и студената за изазове и могућности које доноси AI технологија. Овај оквир, развијен од стране Уједињених нација за образовање, науку и културу (UNESCO), фокусира се на дефинисање кључних знања, вештина и вредности неопходних за етичко и ефикасно коришћење вештачке интелигенције у образовању. Оквир је настао као потреба за увођењем односа наставник-AI-студент, где AI обрађује огромне количине информација, генерише нови садржај и подржава одлучивање кроз предиктивне анализе, али истовремено носи ризике попут угрожавања приватности, појачавања неједнакости и деградације људског учешћа. У наставку ће детаљно бити описани оквири компетенција за наставнике и за студенте, са нагласком на њихове димензије, нивое напредовања и примене у образовној пракси.

Садржај овог поглавља заснован је на препорукама и анализама из UNESCO оквира за компетенције у области вештачке интелигенције [9], [10].

Оквир за AI компетенције наставника

Нови модел односа наставник-AI-студент у образовању појавом вештачке интелигенције значајно утиче на улогу наставника, који више није искључиво преносилац знања, већ и посредник између технологије и студената. У том контексту UNESCO је развио Оквир за AI компетенције наставника (енг. *AI CFT*), који дефинише знања, вештине и вредности неопходне наставницима за рад у доба вештачке интелигенције.

Циљ овог оквира није само техничка обука, већ подстицање наставника да усвоје хуманистички и етички приступ вештачкој интелигенцији. На тај начин образовање остаје усмерено на људске вредности, оснаживање студената, њихову активну укљученост и развој критичког размишљања, као и на друштвену одговорност.

Оквир обухвата пет аспеката компетенција које представљају основне области развоја знања, вештина и вредности неопходних за одговорну употребу вештачке интелигенције у образовању. Аспекти су:

- 1. Приступ усмерен на човека** (енг. *human-centred approach to AI*) - овај аспект истиче потребу да наставници развијају вредности и критичке ставове у интеракцији са вештачком

интелигенцијом. *AI* не сме заменити педагога, већ треба да ојача његову улогу. Наставници би требало да критички процењују предности и ризике употребе вештачке интелигенције, при том водећи рачуна о људској одговорности и друштвеним последицама њене употребе.

2. **Етички аспекти употребе *AI*** - овај аспект дефинише етичке вредности, принципе, регулације, институционалне законе и практична етичка правила која наставници морају да разумеју и примењују, извучена из брзо растућег корпуса знања о етици *AI* и њиховим импликацијама за образовање. Овај аспект дефинише постепено дубље разумевање наставника основне етике *AI*, вештине за безбедну и одговорну употребу *AI*, као и свеобухватне компетенције за учешће у адаптацији етичких правила. Наставници се подстичу да развију основно разумевање типичних етичких питања везаних за *AI* и интеракције човек-*AI*, у контексту заштите људских права, личних података и др.
3. ***AI* основе и примена** - аспект *AI* основе и примена дефинише концептуално знање и вештине које наставници морају да разумеју и примењују како би подржали селекцију, примену и креативну адаптацију *AI* алата за изградњу *AI* окружења за наставу и учење. Од наставника се очекује да стекну одговарајуће разумевање вештачке интелигенције, знање о томе како *AI* функционише, укључујући податке и алгоритме, главне категорије *AI* технологија и примера сваке, као и основне капацитете за процену прикладности специфичних *AI* алата за образовање и коришћење валидираних алата.
4. ***AI* педагогија** - овај аспект се односи на скуп компетенција неопходних за сврсисходну и ефикасну интеграцију *AI* у педагогију, покривајући свеобухватне компетенције за валидацију и селекцију одговарајућих *AI* алата и њихову интеграцију са педагошким методама за подршку припреми наставних програма, учењу, социјализацији, социјалној бризи и процени учења. Овај аспект подразумева да наставници развију способност да критички процењују када и како користити *AI* у настави и учењу на етички и одговоран начин, као и да планирају и имплементирају инклузивне *AI* праксе наставе и учења.
5. ***AI* за професионално учење** - овај аспект подразумева нове компетенције које наставници морају да изграде како би користили *AI* за покретање сопственог професионалног учења и колаборативног професионалног развоја у циљу модернизације наставне праксе. У одговору на брзи развој *AI*, наставницима је потребно упутство како да наставе свој професионални развој у образовним окружењима окарактерисаним растућом интеракцијом човек-*AI*. Ово укључује способност да се користи *AI* за процену потреба професионалног учења и неговање мотивације за доживотно учење и професионалну сарадњу.

Сваки аспект се развија кроз три нивоа прогресије који омогућавају постепени развој знања, вештина и вредности наставника у области *AI*. Ови нивои су осмишљени да подрже наставнике без обзира на њихово почетно искуство са технологијом, водећи их од основне дигиталне писмености до напредне, критичке и креативне примене *AI* у педагошкој пракси. Три кључна нивоа прогресије су:

1. **Стицати (енг. *Acquire*)** - уводни ниво на коме наставници стичу основна знања и вештине неопходне за разумевање и коришћење *AI* у образовању. Овај ниво је усмерен на развој основне дигиталне писмености у контексту *AI*, укључујући разумевање дефиниције *AI*, начина на који функционише, основних категорија *AI* технологија и њихових примена у образовању. Наставници уче да препознају типична етичка питања везана за *AI*, као што су заштита личних података и инклузија, и да користе валидиране *AI* алате на безбедан и одговоран начин. На пример, на овом нивоу наставници могу да процене прикладност *AI* алата за наставне активности и да их примене у једноставним педагошким сценаријима, попут коришћења *AI* за подршку припреми наставе или основним проценама знања. Овај ниво је приступачан свим наставницима, без обзира на њихово техничко предзнање и служи као основа за даљи развој.

2. **Продубити (енг. *Deepen*)** - подразумева дубље разумевање и примену *AI* у сложенијим образовним контекстима. Наставници на овом нивоу развијају способност да критички анализирају и прилагођавају *AI* алате. Они продубљују своје знање о *AI* техникама, укључујући разумевање података и алгоритама, и усавршавају вештине за интеграцију *AI* у педагошке методе, као што су планирање инклузивних *AI* наставних пракси или коришћење *AI* за подршку социјализацији и процени знања. Овај ниво такође укључује напредније разумевање етичких импликација *AI*, као и способност да се учествује у дијалогу о етичким правилима и регулативама. На пример, наставници могу да прилагоде *AI* алате за специфичне образовне циљеве, попут персонализације учења за студенте са различитим потребама, уз обезбеђивање усклађености са етичким стандардима.
3. **Стварати (енг. *Create*)** - највиши ниво прогресије, где наставници постају креатори иновативних *AI* решења и педагошких пракси. На овом нивоу наставници активно учествују у ко-креирању *AI* алата и стратегија, прилагођавајући их за унапређење наставе, учења и професионалног развоја. Они развијају компетенције за креирање нових *AI*-асистираних образовних пракси, попут осмишљавања иновативних метода за интеграцију *AI* у наставне планове или коришћење *AI* аналитике за професионални развој. Овај ниво укључује и способност да се критички истражују нове могућности *AI* технологија, уз промоцију инклузије, одрживости и етике. Наставници на овом нивоу делују као лидери у образовању, подстичући друге да усвоје етичке и иновативне *AI* праксе и доприносе развоју политика за одговорну употребу *AI* у образовању.

У Табели 2 је дат приказ димензија *AI* компетенција за наставнике по свим нивоима прогресије.

Табела 2. UNESCO *AI* компетенције наставника – димензије и нивои прогресије

<i>AI</i> аспект	Стицати	Продубити	Стварати
Приступ усмерен на човека	Разумевање утицаја <i>AI</i> на људска права, приватност и инклузију	Критичка анализа утицаја <i>AI</i> и промоција приступа усмереног на човека у пракси	Ко-креирање <i>AI</i> решења за подршку инклузији и одрживости у образовању
Етички аспекти употребе <i>AI</i>	Препознавање етичких питања и безбедна употреба <i>AI</i> алата	Примена етичких принципа и учешће у дијалогу о регулацијама	Активно креирање етичких правила и политика за <i>AI</i> у образовању
<i>AI</i> основе и примена	Разумевање дефиниције <i>AI</i> , података, алгоритама и основних категорија <i>AI</i> технологија	Прилагођавање <i>AI</i> алата наставним програмима и нивоима студија	Креирање нових <i>AI</i> алата и апликација за образовне сврхе
<i>AI</i> педагогија	Основна примена <i>AI</i> у настави, учењу и процени	Интеграција <i>AI</i> у сложене педагошке праксе и инклузивне наставне методе	Осмишљавање иновативних <i>AI</i> педагошких пракси
<i>AI</i> за професионално учење	Коришћење <i>AI</i> за основну процену потреба професионалног учења	Унапређење професионалног развоја коришћењем <i>AI</i> аналитике и алата	Креирање стратегија за доживотно учење уз подршку <i>AI</i>

У наставку су дати примери наставних активности организованих по нивоима компетенција (видети табеле 2, 3 и 4 у оригиналном документу за потпуне спецификације):

- Ниво Стицати - коришћење *AI* алата, попут генератора квизова, за креирање основних процена знања студената у учионици.
- Ниво Продубити - прилагођавање *AI* алата за персонализовано учење, као што је подешавање система за управљање учењем (*LMS*) за праћење напретка студената.
- Ниво Стварати - креирање програма обуке за колеге наставнике коришћењем *AI* за идентификацију потреба професионалног развоја.

Оквир за *AI* компетенције студената

UNESCO-ов оквир за *AI* компетенције студената (енг. *AI CFS*) дефинише знања, вештине и вредности неопходне студентима да критички разумеју, етички примењују и креативно обликују *AI* у складу са принципима људских права, инклузије и одрживости.

Циљ овог оквира је да подржи интеграцију *AI* у наставне програме, омогућавајући студентима да постану одговорни и будући лидери у *AI* ери. Оквир промовише критичко процењивање *AI* решења, разумевање етичких одговорности, основна знања за доживотно учење и развој инклузивних и одрживих *AI* система.

Оквир обухвата четири аспекта компетенција који представљају кључне области развоја знања, вештина и вредности неопходних за одговорну употребу вештачке интелигенције у образовању. Аспекти су:

- 1. Приступ усмерен на човека** (енг. *human-centred approach to AI*) - овај аспект наглашава потребу да студенти развију вредности и критичке ставове у интеракцији са вештачком интелигенцијом, фокусирајући се на заштиту људских права, приватности, културне разноликости и инклузије. Студенти се подстичу да разумеју друштвене и етичке импликације *AI*, укључујући ризике као што су нарушавање приватности или појачавање неједнакости. Они уче да критички процењују *AI* алате и њихову примену.
- 2. Етички аспекти употребе *AI*** - овај аспект дефинише етичке вредности, принципе и практична правила која студенти морају да разумеју и примењују, на основу корпуса знања о етици *AI* и њених импликација за образовање и друштво. Студенти развијају основно разумевање етичких питања, попут заштите личних података и избегавања пристрасности у *AI* системима, као и вештине за одговорну употребу *AI*.
- 3. *AI* технике и примена** - овај аспект обухвата концептуално знање и практичне вештине потребне студентима за разумевање основа *AI*, укључујући податке, алгоритме и главне категорије *AI* технологија, као и за њихову примену у образовним и другим контекстима. Студенти се обучавају да процењују прикладност *AI* алата за одређене задатке, користе валидиране алате на безбедан начин и прилагођавају их за специфичне потребе.
- 4. Пројектовање *AI* система** - овај аспект се фокусира на компетенције потребне за разумевање и учешће у процесу пројектовања *AI* система, са нагласком на инклузивност, одрживост и етичке принципе. Студенти уче основне концепте развоја *AI* система, укључујући процесе прикупљања података и тренирања модела, и развијају вештине за процену и прилагођавање *AI* система за образовне сврхе.

Сваки аспект се развија кроз три нивоа прогресије који омогућавају постепени развој знања, вештина и вредности студената у области *AI*. Ови нивои су осмишљени да подрже студенте

без обзира на њихово почетно искуство са технологијом, водећи их од основне *AI* писмености до напредне, критичке и креативне примене *AI* у образовању. Три кључна нивоа прогресије су:

1. **Разумети (енг. *Understand*)** - уводни ниво на коме студенти стичу основна знања и вештине неопходне за разумевање *AI* и њених примена. Овај ниво је усмерен на развој основне *AI* писмености, укључујући разумевање дефиниције *AI*, начина на који функционише (нпр. подаци и алгоритми), и основних категорија *AI* технологија. Студенти уче да препознају етичка питања, попут заштите приватности, и да користе једноставне *AI* алате на безбедан начин. На пример, могу да идентификују ризике прикупљања података у *AI* апликацијама или да користе *AI* алат за генерисање образовног садржаја, као што је квиз за учење.
2. **Применити (енг. *Apply*)** - подразумева дубље разумевање и практичну примену *AI* у образовним и другим контекстима. Студенти развијају вештине за критичку процену *AI* алата, прилагођавање постојећих решења за специфичне потребе и интеграцију *AI* у активности учења. Они такође продубљују разумевање етичких импликација, учествујући у дискусијама о праведности и инклузији у *AI* системима. На пример, могу да прилагоде *AI* алат за персонализовано учење или да анализирају *AI* систем за пристрасност у подацима.
3. **Стварати (енг. *Create*)** - највиши ниво прогресије, где студенти постају креатори иновативних *AI* решења. Они учествују у ко-креирању *AI* алата и система, развијајући нове апликације или методе за њихову примену у образовању и шире. Овај ниво укључује способност критичког истраживања нових могућности *AI*, уз промоцију инклузије, одрживости и етичких стандарда. На пример, студенти могу да креирају једноставну *AI* апликацију за подршку учењу или да развију смернице за етичку употребу *AI* у образовању.

UNESCO оквир за *AI* компетенције наставника и студената промовише приступ усмерен на човека, етички и инклузиван приступ вештачкој интелигенцији, оснажујући наставнике и студенте да критички користе и креирају *AI* решења за модернизацију образовања.

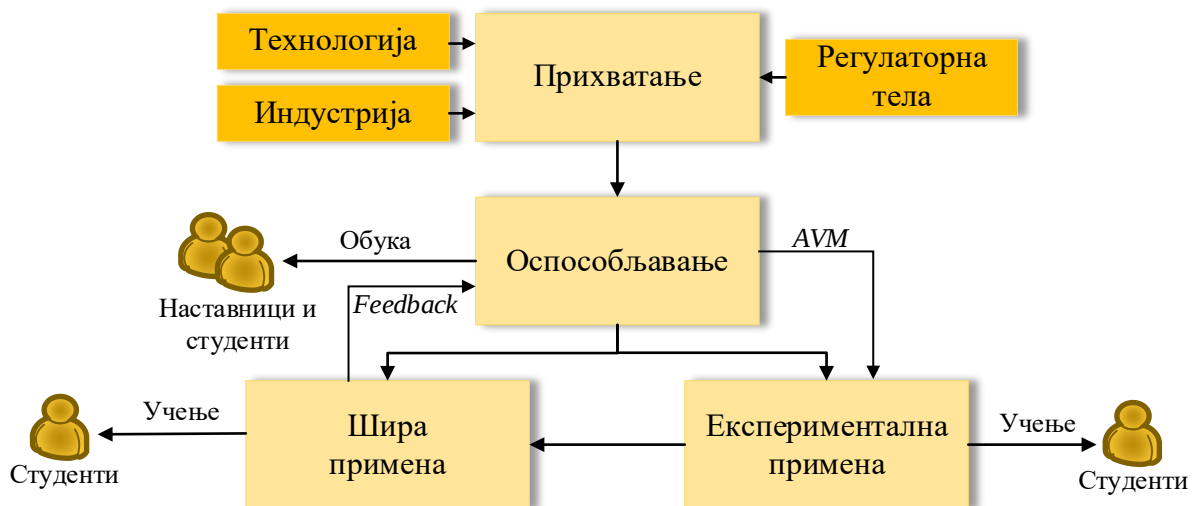
4.4 Анализа постојећих оквира за примену генеративне вештачке интелигенције у образовању

4.4.1 4Е оквир за усвајање генеративне вештачке интелигенције у образовању

Shailendra, Kadel и *Sharma* [221] предложили су 4Е оквир као систематичан приступ усвајању генеративне вештачке интелигенције у високом образовању. Оквир је усмерен на целокупан процес институционалне интеграције *GenAI*, од дефинисања стратешког опредељења и прилагођавања наставних програма до експерименталне примене и ширег увођења технологије.

Полазиште оквира јесте став да успешно увођење *GenAI* не може бити сведено на индивидуалну употребу алата од стране наставника и студената, већ захтева усклађено деловање различитих учесника образовног система. Аутори издвајају управу универзитета, академска тела и комисије, наставнике и координаторе предмета, студенте, ИТ особље, као и екстерне актере, попут државних органа, регулаторних тела, технолошких компанија и привреде. Њихове улоге обухватају дефинисање политика, развој наставних програма, обезбеђивање инфраструктуре и обука, примену *GenAI* у настави и праћење остварених резултата.

Назив 4Е потиче од четири међусобно повезане фазе: прихватање (енг. **Embrace**), оспособљавање (енг. **Enable**), експериментална примена (енг. **Experiment**) и шира примена (енг. **Exploit**). Ове фазе не представљају искључиво линеаран поступак, већ итеративан процес у којем се резултати примене и повратне информације користе за даље прилагођавање политика, наставних програма, обука и начина употребе *GenAI*. На Слици 6 је представљен 4Е оквир.



Слика 6. 4Е оквир за усвајање генеративне вештачке интелигенције у образовању [221]

Фаза прихватања (*Embrace*) односи се на прихватање генеративне вештачке интелигенције на стратешком и институционалном нивоу. У овој фази управа универзитета и надлежна академска тела утврђују намеру и правац њене примене, узимајући у обзир развој технологије, потребе привреде и смернице регулаторних тела. Резултат фазе представљају ажуриране политике, визија и мисија институције, као и јасно формулисано опредељење о улози *GenAI* у настави, учењу и истраживању. На тај начин се обезбеђује основа за рад академских тела и наставника у наредним фазама.

Фаза оспособљавања (*Enable*) обухвата стварање услова за практичну примену *GenAI*. У њој се анализирају наставни програми, исходи учења, наставне активности и начини оцењивања како би се утврдиле потребне измене. Истовремено се планира професионални развој наставника и обука студената за ефикасну, критичку и одговорну употребу *GenAI* алата. Посебна пажња посвећује се изради смерница које се односе на академски интегритет, приватност, етичку употребу и заштиту података. У овој фази се дефинише и академска евалуациона матрица (енг. *Academic eValuation Matrix - AVM*), намењена праћењу успешности интеграције на основу квалитативних и квантитативних показатеља.

У фази експерименталне примене (*Experiment*) предложене измене се испробавају у одабраним предметима, наставним активностима или студијским програмима. Координатори предмета и наставници примењују прилагођене наставне садржаје, методе рада и начине оцењивања, а затим анализирају њихову ефикасност и уочене проблеме. Применом *AVM* матрице могу се пратити, између осталог, спремност наставника и студената, њихово задовољство, остваривање исхода учења, доступност алата, као и појаве повезане са

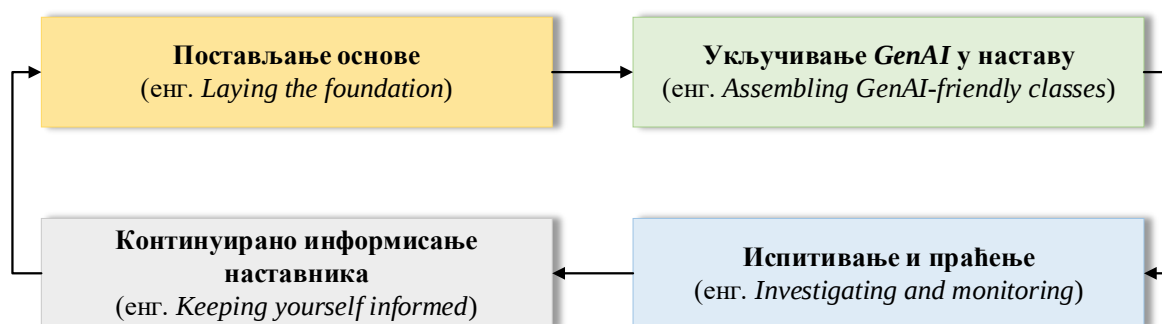
академским интегритетом, етиком и приватношћу. Добијени резултати враћају се у фазу *Enable*, у којој се наставни програми, обуке и смернице додатно унапређују. Итерације се понављају све док се не успоставе одговарајуће праксе и стандарди.

Фаза шире примене (*Exploit*) представља проширење претходно проверених решења на већи број предмета, програма или на институцију у целини. У овој фази наставници и студенти користе ажуриране наставне програме и *GenAI* алате у редовним образовним активностима, док ИКТ тимови обезбеђују потребну инфраструктуру и техничку подршку. Резултати примене се и даље прате, а повратне информације користе за периодично унапређивање обука, политика и наставних пракси. Аутори наглашавају да процена ефеката у овој фази захтева дужи временски период и више циклуса реализације истих предмета.

Основни допринос 4Е оквира огледа се у повезивању институционалног управљања, развоја наставних програма, оспособљавања учесника, контролисаног експериментисања и евалуације резултата. Оквир пружа холистички поглед на усвајање *GenAI* и јасно одређује актере и резултате сваке фазе. Ипак, његов примарни фокус је на институционалном процесу усвајања, док су конкретни начини повезивања *GenAI* алата са исходима учења и специфичним наставним активностима препуштени појединачним програмима и наставницима.

4.4.2 LAIK оквир за интеграцију генеративне вештачке интелигенције у наставу

Al-Ali, Tlili и *Al-Ali* [222] развили су *LAIK* оквир као практичан приступ интеграцији генеративне вештачке интелигенције у наставу високог образовања. За разлику од оквира усмерених првенствено на институционалне политике или оцењивање, *LAIK* је намењен наставницима и обухвата активности које треба предузети пре, током и након примене *GenAI* алата у учионици. Оквир је итеративно развијан на основу података прикупљених од наставника и студената, као и запажања током реализације наставе. *LAIK* оквир обухвата четири фазе које су приказане на Слици 7.



Слика 7. *LAIK* оквир за интеграцију генеративне вештачке интелигенције у наставу високог образовања [222]

Оквир почива на ставу да технологија не треба да буде централни елемент наставе, већ средство којим се подржавају учење и потребе студената. При томе се наглашава да *GenAI* не сме заменити интеракцију студената са наставником и другим студентима, већ треба да има помоћну улогу у наставном процесу.

Фазе *LAIK* оквира обухватају:

1. Постављање основе (енг. *Laying the foundation*)

Прва фаза подразумева припрему студената за ефикасну, критичку и одговорну употребу *GenAI*. Наставник представља изабрани алат, његове могућности, начин функционисања и ограничења, укључујући могућност генерисања нетачних или пристрасних резултата. Потребно је размотрити питања приватности, заштите података и етичке употребе, као и конкретне примере дозвољене и недозвољене примене у оквиру предмета. Ова фаза обухвата и развој вештина формулисања упита, критичког вредновања генерисаног садржаја и проверу информација. Наставник, по могућности у сарадњи са студентима, дефинише правила употребе *GenAI* на предмету и начин на који студенти треба да пријаве или документују његову примену.

2. Укључивање *GenAI* у наставу (енг. *Assembling GenAI-friendly classes*)

Друга фаза обухвата практично укључивање *GenAI* у наставне активности. Аутори предлажу више начина његове примене:

- покретање активности кроз активирање предзнања, разлагање задатка на кораке и генерисање идеја;
- додатно појашњавање појмова, превођење садржаја и моделовање могућих резултата;
- добијање повратних информација о напретку, структури, јасноћи или исправности рада;
- проширивање постојећих активности генерисањем визуелних садржаја, симулација и сценарија;
- подршку учењу кроз сажимање материјала, генерисање питања и објашњавање поступака решавања;
- симулацију интервјуа, усмених испита и професионалних ситуација;
- критичко исправљање *GenAI* резултата провером чињеница, уочавањем пристрасности и допуњавањем недостајућих перспектива.

Циљ није да *GenAI* преузме извршавање задатка, већ да студентима помогне да развију разумевање, креативност, самосталност и критичко мишљење.

3. Испитивање и праћење (енг. *Investigating and monitoring*)

Трећа фаза усмерена је на праћење квалитета и ефеката интеграције. Наставник посматра начин на који студенти формулишу упите, процењују резултате и користе *GenAI* у изради задатака. Посебно је важно утврдити да ли алат доприноси остваривању исхода учења или доводи до прекомерног ослањања на технологију. Поред праћења рада, наставник прикупља повратне информације студената о њиховој оспособљености, потешкоћама и предлозима за унапређење. На основу добијених података прилагођавају се наставне активности, правила употребе и начин подршке студентима.

4. Континуирано информисање наставника (енг. *Keeping yourself informed*)

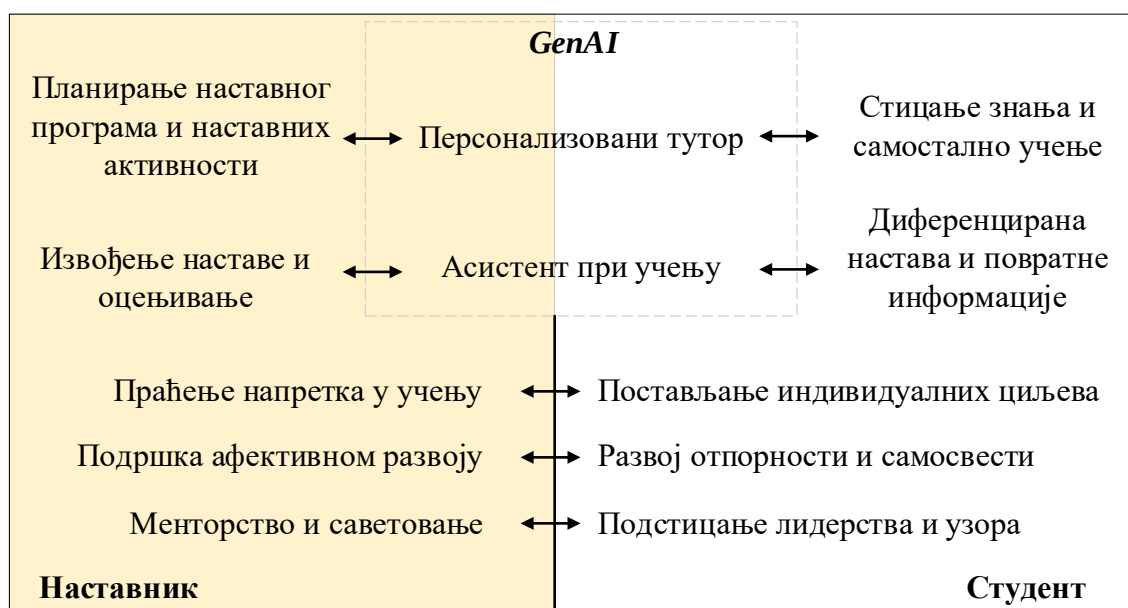
Последња фаза представља трајну активност која се одвија паралелно са осталим фазама. Наставници треба континуирано да унапређују компетенције кроз обуке, стручне конференције и размену искустава са другим наставницима. Истовремено је потребно пратити развој *GenAI* алата, нове функционалности, научна истраживања, институционалне политике и регулаторне смернице. На тај начин се постојеће наставне активности могу благовремено прилагођавати технолошким и нормативним променама.

Основни допринос *LAIK* оквира огледа се у његовој практичној оријентацији и усмерености на конкретне активности наставника и студената у учионици. Оквир повезује припрему студената, осмишљавање наставних активности, праћење резултата и континуирано усавршавање наставника. Међутим, он претпоставља да је одговарајући *GenAI* алат већ изабран, док поступак избора алата и његово систематско повезивање са специфичним исходима учења нису детаљно разрађени.

4.4.3 Колаборативни модел интеграције наставника и генеративне вештачке интелигенције

Метоп и *Kwan* [223] предложили су колаборативни модел интеграције наставника и генеративне вештачке интелигенције, заснован на ставу да *GenAI* не треба да замени наставника, већ да са њим оствари комплементаран однос. Циљ модела је расподела образовних активности у складу са различитим предностима наставника и *GenAI* алата, како би се унапредили исходи учења, персонализација наставе, ангажовање студената и ефикасност наставног процеса.

Предложени модел, улоге наставника, студената и *GenAI*, као и међусобне интеракције ових актера, приказани су на Слици 8.



Слика 8. Колаборативни модел интеграције наставника и генеративне вештачке интелигенције [223]

Модел обухвата три међусобно повезана актера: **наставника, студента и GenAI**. Наставник задржава одговорност за планирање наставног програма и часова, реализацију и вредновање наставе, праћење напретка студената, постављање индивидуалних циљева, менторство и подршку њиховом емоционалном и социјалном развоју. *GenAI* се укључује као персонализовани тутор и наставни асистент, који може да подржи стицање знања, самостално учење, диференцирану наставу и правовремено пружање повратних информација. Студент при томе остаје активан учесник наставног процеса, док наставник усмерава начин и обим употребе *GenAI*.

Сарадња се остварује посредством три врсте интеракција: између наставника и *GenAI*, студента и *GenAI*, као и наставника и студента. Наставник може користити *GenAI* за планирање наставе, креирање материјала, питања и задатака, прилагођавање садржаја и анализу напретка студената. Студенти могу употребљавати *GenAI* за добијање објашњења, генерисање идеја, сажимање материјала, уређивање нацрта, проверу разумевања и добијање непосредних повратних информација. Истовремено, директна интеракција наставника и студента остаје неопходна за развој критичког мишљења, мотивације, вредности, одговорности и других способности које захтевају емпатију и разумевање контекста.

Модел се може применити пре, током и након наставе. У активностима пре часа *GenAI* може имати улогу персонализованог извора за упознавање са појмовима и припрему студената. Током часа наставник може укључити *GenAI* у реализацију квизова, дискусија, групних задатака, симулација и других интерактивних активности. Након часа *GenAI* може подржати понављање, рефлексiju, додатне задатке и практичну примену усвојених знања. Аутори наглашавају да све активности морају бити усклађене са исходима учења и правилима академског интегритета.

Посебан елемент модела представља расподела образовних активности између наставника и *GenAI* на основу когнитивног, психомоторног и афективног домена Блумове таксономије [100]. Ниво њиховог учешћа не одређује се као фиксна подела, већ као распон од мањег до значајнијег ангажовања:

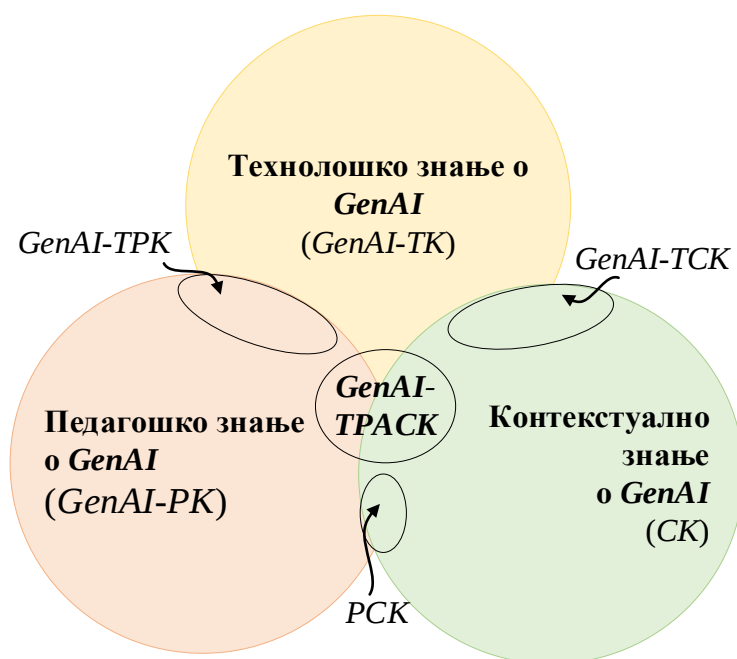
- у когнитивном домену предвиђено је мање до средње учешће наставника, а средње до значајно учешће *GenAI*;
- у психомоторном домену наставник има средње до значајно учешће, док је улога *GenAI* мања или помоћна;
- у афективном домену, који обухвата ставове, понашање, емоције, саморегулацију и социјалне вештине, наставник има кључну улогу, док је учешће *GenAI* ограничено.

GenAI је нарочито погодан за задатке који се односе на приступ информацијама, објашњавање основних појмова, генерисање примера, персонализовано учење и пружање непосредних повратних информација. Насупрот томе, наставник има значајнију улогу у развоју когнитивних способности вишег реда, практичних вештина, критичког мишљења, креативности, сарадње, комуникације и етичког расуђивања. Конкретна расподела задатака треба да зависи од области образовања, исхода учења, карактеристика студената и природе наставне активности.

Основни допринос овог модела огледа се у систематском повезивању улога наставника и *GenAI* са различитим доменима развоја студената. Уместо једнозначне поделе послова, модел предвиђа флексибилан однос у којем се ниво ангажовања прилагођава контексту и циљевима наставе. Међутим, реч је првенствено о концептуалном моделу који још увек није емпиријски проверен у различитим образовним областима, а његова успешна примена зависи и од оспособљености наставника, доступности технологије, заштите приватности, ублажавања пристрасности и обезбеђивања равноправног приступа *GenAI* алатима.

4.4.4 *GenAI-TPACK* оквир за развој компетенција наставника за интеграцију генеративне вештачке интелигенције

Belkina и коаутори [224] полазе од *TPACK* оквира (енг. *Technological Pedagogical Content Knowledge*) [225], који описује знања потребна наставницима за усклађивање технологије, педагогије и наставног садржаја. На основу систематског прегледа 21 емпиријске студије примене *GenAI* у високом образовању, аутори овај оквир прилагођавају специфичностима генеративне вештачке интелигенције и развијају *GenAI-TPACK* оквир. Његова сврха је да наставницима пружи структуру за процену и развој знања неопходних за ефикасну, педагошки оправдану и одговорну примену *GenAI* алата. Развијени оквир је приказан на Слици 9.



Слика 9. *GenAI-TPACK* оквир са кључним знањима и способностима наставника за примену генеративне вештачке интелигенције [224]

GenAI-TPACK задржава основну структуру *TPACK* оквира, која обухвата технолошко, педагошко и садржајно знање, али је проширује знањима непосредно повезаним са употребом генеративне вештачке интелигенције. Посебно је укључено и контекстуално знање, којим се у обзир узимају институционалне политике, етичка питања, регулаторни услови и практична ограничења примене.

Кључни домени оквира су:

1. Технолошко знање о *GenAI* (енг. *GenAI Technological Knowledge - GenAI-TK*)

Овај домен обухвата разумевање начина функционисања, могућности и ограничења *GenAI* алата. Наставник треба да зна како да комуницира са овим алатима, формулише одговарајуће упите и користи их за генерисање текста, слика, програмског кода и других врста садржаја. Поред практичних вештина, потребно је познавање непоузданости резултата, могућности појаве нетачних информација и граница применљивости конкретних алата.

2. Технолошко-педагошко знање о *GenAI* (енг. *GenAI Technological Pedagogical Knowledge - GenAI-TPK*)

Овај домен односи се на разумевање начина на који *GenAI* може подржати различите наставне методе. Обухвата употребу алата за персонализовано и адаптивно учење, пружање повратних информација, праћење напретка студената, формативно оцењивање и креирање интерактивних наставних активности. Наставник треба да буде способан да процени када *GenAI* доприноси остваривању исхода учења, а када његова примена може довести до пасивизације студената или прекомерног ослањања на технологију.

3. Технолошко-садржајно знање о *GenAI* (енг. *GenAI Technological Content Knowledge - GenAI-TCK*)

GenAI-TCK подразумева способност примене *GenAI* у контексту конкретне научне или наставне области. Наставник треба да зна како да користи алате за проналажење, креирање и прилагођавање образовних материјала релевантних за одређену дисциплину. То подразумева и способност процене тачности, релевантности и примерености генерисаног садржаја, нарочито када се *GenAI* примењује на стручне појмове, проблеме и задатке специфичне за наставну област.

4. Технолошко-педагошко-садржајно знање о *GenAI* (енг. *GenAI Technological Pedagogical Content Knowledge - GenAI-TPACK*)

Централни домен оквира представља интеграцију технолошког, педагошког и садржајног знања. Наставник на овом нивоу не користи *GenAI* као изоловани алат, већ га систематски повезује са исходима учења, наставним садржајем и изабраним педагошким приступом. То подразумева осмишљавање наставе у којој *GenAI* доприноси ангажовању студената, персонализацији учења и развоју специфичних знања и способности, без нарушавања улоге наставника и аутентичности процеса учења.

5. Контекстуално знање о *GenAI* (енг. *GenAI Contextual Knowledge - GenAI-CK*)

Контекстуални домен обухвата разумевање ширег окружења у којем се *GenAI* примењује. Наставник треба да познаје институционалне политике, прописе и смернице о употреби генеративне вештачке интелигенције, као и питања академског интегритета, приватности, заштите података, правичности и инклузивности. Примена *GenAI* треба да буде усклађена са циљевима образовне институције, потребама студената и етичким принципима употребе технологије.

У оквиру су задржани и традиционални домени садржајног знања, педагошког знања и педагошко-садржајног знања, али их аутори не разрађују детаљно, јер је њихов фокус на областима које непосредно укључују *GenAI*. Повезивањем свих домена наглашава се да техничко познавање алата само по себи није довољно. Ефикасна интеграција захтева истовремено разумевање наставног садржаја, начина на који студенти уче и контекста у којем се технологија примењује.

Основни допринос *GenAI-TPACK* оквира огледа се у систематизацији компетенција које наставници треба да развију како би *GenAI* користили ефикасно и одговорно. Међутим, он превасходно описује потребна знања наставника, док не дефинише детаљан поступак интеграције *GenAI* у конкретне фазе наставног или пројектног процеса. Такође, повезивање појединачних *GenAI* алата са специфичним исходима учења и активностима у оквиру одређене дисциплине остаје задатак наставника.

4.4.5 Упоредна анализа постојећих оквира

Анализирани оквири приступају интеграцији генеративне вештачке интелигенције са различитих нивоа и из различитих перспектива. 4Е оквир примарно је усмерен на институционално усвајање *GenAI*, *LAIK* на практичну примену у настави, колаборативни модел на расподелу улога између наставника и *GenAI*, док *GenAI-TPACK* одређује знања и компетенције наставника потребне за њену педагошки оправдану примену. Због тога се наведени оквири могу посматрати као међусобно комплементарни, али ниједан од њих појединачно не обухвата све нивое и активности интеграције *GenAI* у образовни процес. Упоредни приказ основних карактеристика, доприноса и ограничења анализираних оквира дат је у Табели 3.

Табела 3. Упоредни приказ оквира за интеграцију генеративне вештачке интелигенције у образовање

Оквир	Примарни ниво и фокус примене	Кључни елементи	Основни допринос	Ограничења
4Е оквир	Институционални ниво; усвајање <i>GenAI</i> у универзитету	Прихватање, оспособљавање, експериментална и шира примена; <i>AVM</i> матрица	Систематизује институционално усвајање <i>GenAI</i>	Не разрађује конкретне наставне активности
<i>LAIK</i> оквир	Ниво предмета и наставе; практична подршка наставнику	Постављање основе, <i>GenAI</i> активности, праћење и информисање наставника	Нуди практичне смернице за примену у настави	Не обухвата избор алата и доменске специфичности
Колаборативни модел наставника и <i>GenAI</i>	Ниво улога и интеракције наставника, студената и <i>GenAI</i>	Когнитивни, психомоторни и афективни домен	Дефинише комплементарне улоге наставника и <i>GenAI</i>	Не прописује конкретан поступак примене
<i>GenAI-TPACK</i> оквир	Ниво знања и компетенција наставника	<i>GenAI-TK</i> , <i>GenAI-TPK</i> , <i>GenAI-TCK</i> , <i>GenAI-TPACK</i> и <i>GenAI-XK</i>	Систематизује компетенције наставника	Не операционализује примену кроз наставне фазе

Заједничку основу анализираних оквира представља став да *GenAI* не треба посматрати као замену за наставника, већ као средство за унапређење наставе и учења. У свим оквирима наглашавају се потреба за оспособљавањем наставника и студената, усклађивање примене са исходима учења, очување академског интегритета, критичко вредновање генерисаних резултата и континуирано праћење ефеката примене. Такође, интеграција се не посматра као једнократна интервенција, већ као флексибилан и итеративан процес који треба прилагођавати развоју технологије и потребама образовног контекста.

Разлике између оквира превасходно се односе на њихов обухват и јединицу анализе. 4Е оквир обезбеђује најшири институционални поглед, али се мање задржава на конкретној реализацији наставних активности. *LAIK* пружа детаљнију подршку наставнику у учионици, али остаје општи оквир применљив у различитим дисциплинама. Колаборативни модел јасније одређује комплементарне улоге наставника и *GenAI*, али не пружа разрађен поступак за планирање и реализацију интеграције. *GenAI-TPACK*, са друге стране, прецизира потребна знања наставника, али не операционализује њихову примену кроз фазе конкретног образовног процеса.

Посебно значајно за предмет овог истраживања јесте то што анализирани оквири нису доменски специфични за образовање у области развоја софтвера. Они не приказују на који начин *GenAI* треба интегрисати кроз фазе животног циклуса развоја софтвера, нити непосредно повезују исходе учења, развојне активности, избор и начин употребе *GenAI* алата, пројектне артефакте, менторску подршку и вредновање рада студената. Иако појединачно обухватају институционалне, педагошке, технолошке или колаборативне аспекте, недостаје јединствен приступ који би их операционализовао у оквиру пројектно-заснованог учења у области развоја софтвера.

Полазећи од наведених увида, у поглављу 6 ове докторске дисертације представљен је методолошки оквир за унапређење образовања у области развоја софтвера интеграцијом алата генеративне вештачке интелигенције у пројектно-засновано учење. Предложени оквир синтетиче релевантне елементе постојећих приступа и настоји да превазиђе ограничења која произлазе из њихове опште или парцијалне усмерености. Његова специфичност огледа се у систематској интеграцији *GenAI* кроз фазе животног циклуса развоја софтвера, при чему се непосредно повезују исходи учења, пројектне активности, примена *GenAI* алата и менторска улога наставника. На тај начин *GenAI* се не уводи као издвојено технолошко средство, већ као педагошки осмишљена подршка процесу пројектовања и развоја софтвера, уз задржавање активне улоге студената и одговорности наставника за усмеравање, праћење и вредновање процеса учења.

4.5 Регулатива, етички аспекти и изазови

Развој генеративне вештачке интелигенције поставља бројна етичка, правна и друштвена питања, посебно у контексту образовања. Иако ови алати могу унапредити процеса учења и персонализовану подршку студентима, јављају се ризици који укључују академску неискреност, смањену аутономију студената и прекомерну зависност од технологије [226]. Академски интегритет је посебно угрожен јер алати генеративне вештачке интелигенције могу

генерисати текстове или код који се тешко могу разликовати од радова људи, што захтева нова правила о ауторству, власништву и транспарентности у академској продукцији [227].

Етичке дебате у високом образовању крећу се између две крајности, односно од рестриктивних политика које настоје да ограниче или забране употребу ових алата до приступа који подстичу њихово одговорно укључивање у наставу. Како се истиче у раду [228], аргументи за строга ограничења, попут утицаја на животну средину, експлоатације рада или ризика по приватност, често не дају довољно снажне разлоге за забрану, већ упућују на потребу за уравнотеженим политикама које не занемарују образовне користи. У том контексту, кључним се показује увођење етичких принципа као што су поштовање интелектуалне својине, искреност, робусност система и културна одговорност [229].

Образовање представља једну од кључних области у којој примена генеративне вештачке интелигенције доноси највеће потенцијале, али и највеће изазове. Са једне стране, алати као што су *ChatGPT* или *Gemini* омогућавају студентима и наставницима приступ персонализованом учењу, креирање материјала и подршку у истраживачком раду, што може унапредити квалитет наставног процеса и учинити образовање инклузивнијим и доступнијим [230]. Са друге стране, прекомерно ослањање на ове алате може довести до смањења компетенција, угрожавања академског интегритета и смањења креативности [226], [231]. Потребно је развијати образовне стратегије које ће студенте учити како да критички процењују излазе модела, како да препознају пристрасности и како да интегришу алате у процес учења на начин који подстиче рефлексивност и самосталност [232].

Европски оквир кроз иницијативе *European Education Area* и *Digital Education Action Plan* (2021-2027) наглашава управо развој дигиталне писмености као централне компетенције за будућност, која подразумева не само техничко коришћење алата, већ и критичко разумевање њихових могућности и ограничења, укључујући одговорну и етичку употребу вештачке интелигенције у настави [11], [233]. Ови документи повезују дигиталне вештине са ширим образовним циљевима као што су једнакост, приступачност и инклузивност, што је посебно важно у контексту све веће улоге вештачке интелигенције у образовању. Нарочито је важно да дигитална писменост не буде схваћена само као техничка вештина, већ и као способност критичке процене извора, разумевања ограничења алгоритама и препознавања ризика као што су пристрасност, дезинформације и кршење приватности [230], [234].

Изазови у образовању огледају се и у потреби за балансирањем између подстицања креативности и иновације, с једне стране, и спречавања академског непоштења и површног усвајања знања, са друге. Истраживања показују да интеграција генеративне вештачке интелигенције у наставу захтева пажљиво пројектоване задатке, попут оних који подстичу валидацију излаза модела, колаборативне активности и укључивање етичких компетенција у курикулуме [226], [231]. Овакви приступи усмеравају студенте да користе алате као помоћ у процесу учења, а не као замену за сопствено знање и критичко размишљање.

Управо због ових изазова, образовне институције све више усвајају сопствене етичке и регулаторне оквири за употребу генеративне вештачке интелигенције. Пример је Факултет организационих наука Универзитета у Београду, који је 2024. године усвојио Опште смернице

за коришћење генеративне вештачке интелигенције. Смернице дефинишу правни, стратешки и етички оквир примене генеративне вештачке интелигенције, при чему наглашавају академски интегритет, транспарентност у коришћењу *AI* алата, поштовање ауторских права и заштиту приватности података. Оне постављају принцип да је генеративну вештачку интелигенцију дозвољено користити искључиво као помоћни алат, уз експлицитно навођење његове улоге у образовним и истраживачким активностима, чиме се спречава прикривена злоупотреба и осигурава очување интегритета наставног процеса [235].

У Табели 4 је дат систематизован приказ кључних ризика и мера које се односе на употребу генеративне вештачке интелигенције у образовању.

Табела 4. Преглед ризика и мера у примени генеративне вештачке интелигенције у образовању

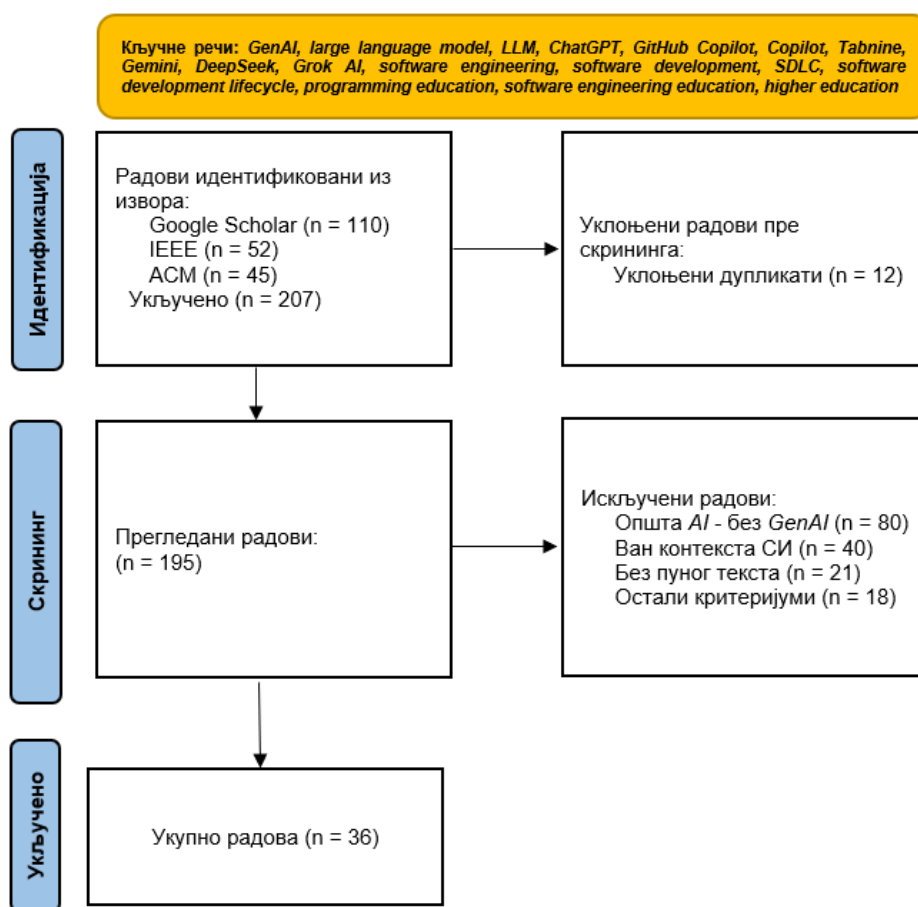
Ризик	Опис	Препоручене мере
Академска неискреност	Генерисање текста и кода без навођења	Увођење правила транспарентности, обавезно навођење улоге <i>AI</i> [226], [227]
Смањење аутономије студената	Прекомерно ослањање на алате уместо на самостално учење	Задаци који подстичу критичку валидацију излаза <i>AI</i> и колаборативно учење [231]
Пристрасности и дезинформације	Алати могу репродуковати пристрасне или нетачне податке	Обука студената за критичко читање и проверу извора [234]
Кршење ауторских права	Коришћење заштићених материјала без дозволе	Увођење политика цитирања и свести о интелектуалној својини [236]
Приватност података	Унос осетљивих информација у <i>AI</i> системе	Забрана уношења личних/поверљивих података у <i>AI</i> , усвајање институционалних смерница [235]
Неједнак приступ	Студенти из различитих средина немају исте могућности	Обезбеђивање инфраструктуре и једнаких услова приступа [11]

С обзиром на наведене ризике и мере, наредно поглавље биће посвећено начинима интеграције генеративне вештачке интелигенције у пројектно-засновано учење, као једном од приступа који може омогућити њену одговорну и сврсисходну примену у образовању.

5 Интеграција генеративне вештачке интелигенције у образовању у области развоја софтвера

За потребе систематског прегледа литературе коришћене су *PRISMA 2020* смернице [35]. Централни део истраживања је имао за циљ да идентификује и анализира радове који се баве применом генеративне вештачке интелигенције у образовању у домену развоја софтвера.

Критеријуми укључивања обухватили су радове објављене у периоду од 2020. до 2025. године, на енглеском или српском језику, који су доступни у пуном тексту и који експлицитно обрађују употребу генеративних *AI* алата (нпр. *ChatGPT*, *GitHub Copilot*, *Claude*, *Tabnine*, *Gemini*) у контексту софтверског инжењерства. Искључени су радови који се баве вештачком интелигенцијом уопште, без нагласка на генеративне моделе, као и радови без доступног пуног текста. Полазна база за анализу обухватала је укупно 207 референци. Након иницијалне провере уклоњено је 12 дупликата, те је на скрининг ушло 195 радова. Од тог броја, 159 радова је искључено као нерелевантно (фокус на општој *AI*, машинском учењу или другим доменима без директне везе са развојем софтвера, и др). Коначно, у преглед литературе је укључено 36 радова. Процес идентификације, скрининга и укључивања радова приказан је на *PRISMA flow* дијаграму (Слика 10).



Слика 10. Дијаграм тока анализе литературе за примену генеративне вештачке интелигенције у образовању у домену развоја софтвера

5.1 Примена генеративне вештачке интелигенције у развоју курикулума

Генеративна вештачка интелигенција све више постаје саставни део процеса редизајна наставних програма, јер омогућава већу персонализацију садржаја, ефикасније креирање наставних материјала и увођење нових педагошких пракси. Један од предложених модела је *GAIDE* оквир (енг. *Generative AI for Developing Education*) који представља структурирану методологију за примену *GenAI* у развоју курикулума. Оквир укључује фазе као што су [237]:

1. припрема - дефинисање наставних циљева и усклађивање са академским стандардима,
2. генерисање садржаја - употреба *GenAI* алата за креирање наставних материјала као што су студије случаја, задаци или квизови,
3. валидација и ревизија - провера академске поузданости и усаглашености генерисаног садржаја са педагошким принципима,
4. имплементација - интеграција материјала у наставу,
5. евалуација - прикупљање повратних информација од студената и наставника ради унапређења процеса

Овим приступом наставници могу систематски да користе *GenAI* за израду студија случаја, задатака и тестова, уз задржавање академске поузданости и усклађености са педагошким циљевима.

Други модел је *GenAI Competence Framework* [238], који дефинише примену *GenAI* у високошколским курикулумима кроз три компоненте:

1. дефинисање компетенција које студенти треба да развију,
2. стратегије интеграције *GenAI* у наставу и
3. механизме евалуације, односно коришћење индикатора за мерење утицаја *GenAI* на исходе учења.

Поред тога, у настави се развијају приступи који омогућавају наставницима да уз помоћ *GenAI* креирају персонализоване планове часова. Уношењем кључних наставних циљева, нивоа предзнања студената и расположивих ресурса, могуће је добити више верзија наставних планова прилагођених различитим стилима учења. Овај приступ показује посебну вредност у контексту инклузивног образовања, где је неопходно прилагођавати садржај специфичним потребама студената [239].

Као новина издваја се и ко-креирање курикулума, где студенти и наставници заједнички учествују у редизајну студијских програма. *GenAI* у том процесу функционише као посредник који систематизује идеје и повратне информације, подржавајући транспарентнији развој образовних садржаја [240], [241]. Недавна истраживања показују да *GenAI* може да прошири обим ко-креације тиме што студентима омогућава да активно учествују у изради наставних ресурса и да истовремено добијају аутоматизовану повратну информацију о свом раду. На пример, у раду [242], студенти су креирали задатке и питања за вежбу на платформи *RiPPLE*, док је *GenAI* пружао смернице за побољшање, истицао предности и проверавао тачност садржаја. Овакав приступ омогућава да се ко-креирање спроводи у већем обиму и уз већи ниво квалитета, а истовремено оснажује студенте да развијају и педагошке компетенције.

Поред примене у генерисању појединачних наставних материјала, генеративна вештачка интелигенција може имати значајну улогу и у ширем процесу пројектовања и редизајна курикулума. Истраживања показују да *GenAI* алати могу подржати формулисање исхода учења, организацију наставних садржаја, предлагање активности и развој елемената за процену знања [243]. Ипак, генерисани предлози захтевају стручну проверу и прилагођавање, због чега се улога наставника не умањује, већ се помера ка процени, селекцији и педагошком обликовању предложених садржаја.

Посебно значајан аспект представља динамичко прилагођавање курикулума променама у технологији и професионалној пракси. Уместо искључиво периодичног редизајна наставних програма, истиче се потреба за флексибилнијим приступом који омогућава редовно преиспитивање исхода учења, наставних активности и компетенција које студенти треба да развију. У том контексту предложени су механизми који обухватају модуларни дизајн курикулума, сарадњу са индустријом и истраживачким институцијама, развој *AI* писмености и веће ослањање на пројектно-засновано учење [244].

У процесу редизајна курикулума значајну улогу може имати и активно учешће студената. Уместо да буду искључиво корисници већ дефинисаних наставних садржаја, студенти могу учествовати у предлагању тема, наставних активности, примера и задатака, као и у процени њихове релевантности и примењивости. Савремени приступи ко-дизајну високог образовања укључују различите заинтересоване стране, међу којима су студенти, наставници, представници индустрије и други релевантни актери [245].

Овакав приступ је релевантан у образовању у области развоја софтвера, у којој технолошке промене непосредно утичу на знања и компетенције које се очекују од будућих софтверских инжењера. Развој и ширење *GenAI* алата мењају активности у процесу развоја софтвера, токове рада, начине сарадње и улогу самог софтверског инжењера, што истовремено ствара потребу за прилагођавањем образовних програма. Савремени курикулуми зато, поред традиционалних знања из области софтверског инжењерства, треба да обухвате компетенције за примену генеративне вештачке интелигенције, критичку процену њених резултата, етичку употребу и континуирано усвајање нових технологија [246].

5.2 Педагошки приступи за интеграцију вештачке интелигенције у наставу

Примена генеративне вештачке интелигенције у настави постаје све значајнија у контексту софтверског инжењерства, јер *GenAI* алати омогућавају нове облике подршке студентима у процесу учења. Истраживања показују да *GenAI* може да подстакне интерактивност и омогући студентима да кроз алате као што су *ChatGPT* или *GitHub Copilot* добију помоћ у програмирању, тестирању и документовању кода, чиме се побољшавају ефикасност и тачност њиховог рада [247]. Ови алати омогућавају студентима да брже прелазе почетне препреке у учењу програмирања, али и да развијају сложеније вештине кроз експериментисање са софтверским архитектурама и решавање реалних проблема [248].

У настави у области развоја софтвера посебан значај има интеграција *GenAI* у курикулуме, где се ови алати користе не само као подршка у изради задатака, већ и као средство за унапређење

критичког размишљања и процене различитих решења. *GenAI* може да функционише као „образовни асистент“, омогућавајући наставницима да усмере фокус на педагошке аспекте наставе, док рутинске активности попут креирања примера кода или објашњавања синтаксе препуштају моделима [249]. На овај начин, студентима се омогућава више времена за тимски рад и стваралачке активности које симулирају стварне пројектне ситуације [250].

Ипак, интеграција *GenAI* у наставни процес није без изазова. Морају се узети у обзир етички аспекти, ризик од прекомерног ослањања на алате и могућност смањења креативности студената ако се модели користе искључиво за брзо генерисање решења [251]. С друге стране, у раду [252] истичу да *GenAI* има потенцијал да промени саму природу наставних програма у софтверском инжењерству, јер омогућава нове методе у свим фазама развоја софтвера. Поред тога, употреба конверзационих агената у настави може да подстакне активније учешће студената и омогући им персонализовано искуство учења [253].

Истраживања показују да употреба *ChatGPT*-а у настави програмирања повећава мотивацију студената и јача њихову самоувереност приликом решавања задатака [254]. Ово је јако значајно у раним фазама образовања када студенти често имају потешкоће са усвајањем основних програмерских концепата. Поред тога, интеграција *GenAI* у различите фазе животног циклуса развоја софтвера показала је потенцијал да промени начин на који се пројекти реализују у образовном контексту јер, између осталог, може скратити време потребно за израду прототипа и повећати квалитет финалних пројеката [255].

5.3 Методе оцењивања засноване на генеративној вештачкој интелигенцији

Генеративна вештачке интелигенције проналази примену и у сегменту оцењивања. Због својих карактеристика и функционалности, може унапредити традиционалне метода процене знања и вештина студената. Класични облици оцењивања често су ограничени временским ресурсима наставника, док *GenAI* омогућава бржу, доследнију и персонализовану процену. Једна од кључних предности огледа се у креирању прилагођених задатака и повратних информација који одговарају различитим профилима студената. Ипак, при томе је неопходно водити рачуна о валидности, поузданости и правичности резултата, јер прекомерна персонализација може довести до неуједначености у критеријумима оцењивања [256].

Важан аспект развоја метода оцењивања је интеграција вршњачке процене са генеративним моделима. Вршњачка процена већ дуго представља један од начина за развој критичког размишљања код студената [257]. *GenAI* у овом контексту има потенцијал да унапреди цео процес кроз аутоматско вредновање квалитета и корисности коментара које студенти дају једни другима, што повећава ефективност повратних информација и подстиче студенте да их озбиљније схвате. Истраживања показују да су модели попут *GPT*-а у стању да поуздано процене колико су рецензије студената конструктивне, чиме се подиже квалитет интеракције у процесу вршњачке процене [258]. Додатно, интеграција *AI* у вршњачку процену не значи замену улоге студената, већ њено унапређење. Истраживања показују да *AI* може помоћи студентима да формулишу прецизније и конструктивније повратне информације, а наставницима омогућити увид у обрасце пристрасности или недоследности у процесу евалуације [259].

AI системи све више показују вредност и у генерисању повратних информација у реалном времену. На тај начин студенти добијају непосредну подршку која им омогућава да брже идентификују слабости у свом раду и одмах их коригују, што унапређује процес континуираног учења [260], [261]. Ово се показало корисним не само у формативном, већ и у финалном оцењивању (енг. *summative assessment*), јер системи могу обезбедити доследност током читавог наставног процеса [262].

Све већу пажњу добијају хибридни модели који комбинују вршњачко и *GenAI* оцењивање. Ови приступи омогућавају студентима да добију вишеструке перспективе чиме се подстиче развој способности самопроцене, али и обезбеђује већа доследност у резултатима [263]. Истовремено, интеграција аналитике учења и вештачке интелигенције показала се корисном у развоју транспарентних и поузданих система вршњачке процене, јер помаже да се ублажи ризик од субјективности и обезбеди веће поверење у резултате [264]. Оквир који аутори у раду [264] предлажу наглашава да AI може да подржи више фаза процеса, од доделе задатака и верификације рецензија, до генерисања повратних информација које студентима показују како да унапреде квалитет сопствених коментара.

Недавне компаративне студије додатно истичу потенцијал *GenAI* у односу на класичне методе оцењивања. У раду [265] упоређује резултате оцењивања које пружају наставници, вршњаци и *GenAI*, закључујући да модели могу обезбедити висок ниво тачности и доследности. Ипак, најбоље резултате даје комбинација људског и машинског оцењивања, посебно у сложеним задацима који захтевају разумевање контекста или процену креативности.

6 Методолошки оквир за унапређење образовања у области развоја софтвера интеграцијом алата генеративне вештачке интелигенције у пројектно-засновано учење

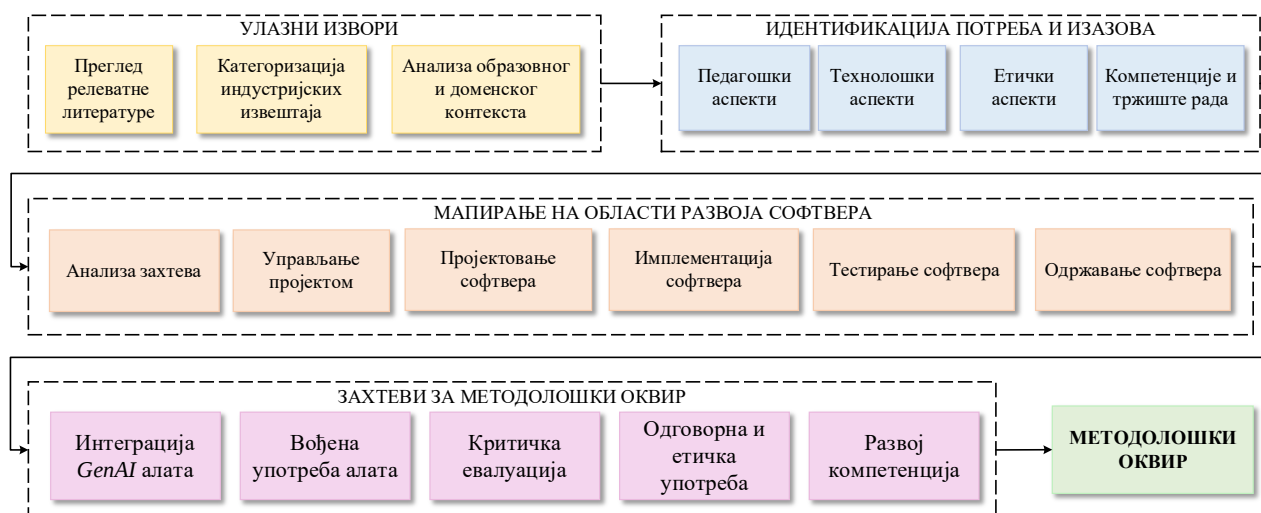
6.1 Идентификација потреба и изазова у настави у контексту пројектно-заснованог учења

У циљу развоја методолошког оквира за унапређење образовања у области развоја софтвера кроз интеграцију алата генеративне вештачке интелигенције у пројектно-засновано учење, најпре је извршена идентификација потреба и изазова који карактеришу савремени наставни процес. Полазну основу чинила је анализа релевантне научне литературе и индустријских извештаја о примени генеративне вештачке интелигенције у образовању и развоју софтвера, као и анализа образовног и доменског контекста у којем се предложени оквир примењује [22], [247]–[250], [252]–[254], [266]–[268].

Анализа је била усмерена на идентификовање потенцијала и ограничења примене *GenAI* алата у настави, потреба за развојем савремених дигиталних и *AI* компетенција, као и педагошких, технолошких и етичких аспеката који прате њихово укључивање у наставни процес. Посебно су разматране специфичности пројектно-заснованог учења у области развоја софтвера, у оквиру којег студенти кроз рад на конкретним пројектима повезују техничке активности развоја софтвера са анализом захтева, планирањем, организацијом и праћењем пројектних активности. На тај начин, примена генеративне вештачке интелигенције није посматрана само као подршка генерисању програмског кода, већ као технологија која може бити интегрисана у различите активности током развоја софтверског решења.

На основу синтезе анализираних извора и специфичности образовања у области развоја софтвера издвојени су захтеви које предложени методолошки оквир треба да адресира. Они обухватају интеграцију *GenAI* алата у конкретне активности развоја софтвера, вођену и постепену употребу алата у оквиру пројектно-заснованог учења, развој способности студената за критичку евалуацију и валидацију генерисаних резултата, одговорну и етичку примену генеративне вештачке интелигенције, као и развој компетенција релевантних за професионалну праксу.

Процес идентификације потреба и њиховог превођења у захтеве за развој методолошког оквира приказан је на Слици 11. Као што је приказано, процес полази од три комплементарна извора: прегледа научне литературе, анализе индустријских извештаја и анализе образовног и доменског контекста. На основу њих идентификовани су педагошки, технолошки и етички аспекти примене генеративне вештачке интелигенције, као и потребе које се односе на развој компетенција и захтеве тржишта рада. Њихово мапирање на релевантне области развоја софтвера омогућило је да се дефинишу захтеви који представљају основу за структуру и елементе методолошког оквира представљеног у наредном поглављу.

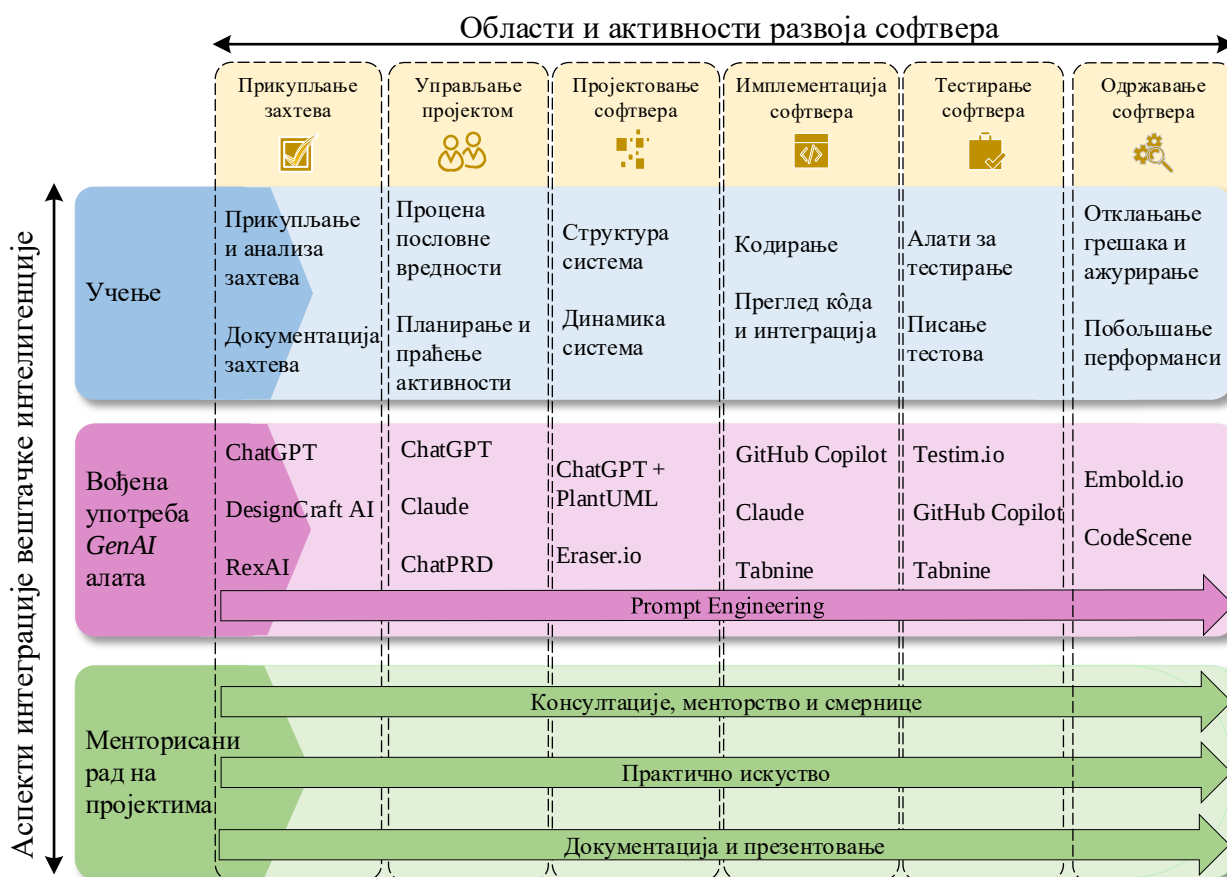


Слика 11. Процес идентификације потреба и захтева за развој методолошког оквира

6.2 Развој методолошког оквира за унапређење образовања у области развоја софтвера интеграцијом генеративне вештачке интелигенције у пројектно-засновано учење

Да би се *GenAI* алати успешно интегрисали у образовање софтвер инжењера и постигли жељени исходи учења, неопходно је систематски укључити генеративну вештачку интелигенцију у наставни процес. То захтева структуриран приступ који подржава студенте у развијању како техничких вештина, тако и разумевања процеса развоја и управљања софтверским пројектима уз подршку вештачке интелигенције. Интеграцијом алата генеративне вештачке интелигенције у наставни процес студентима се омогућава да развију и усавршавају способности ефикасне примене *AI* у различитим активностима развоја софтвера.

Полазећи од релевантних области софтверског инжењерства дефинисаних у *SWEBOK*-у, развијен је методолошки приступ који интегрише *GenAI* алате у наставни процес у домену развоја софтвера. Овај приступ приказан је на Слици 12 и заснива се на примени *GenAI* алата у различитим фазама развоја софтвера, при чему су наглашена три кључна аспекта: учење, вођена употреба *GenAI* алата и менторисани рад на пројектима [88].



Слика 12. Методолошки оквир за унапређење образовања у области развоја софтвера интеграцијом генеративне вештачке интелигенције у пројектно-засновано учење

Предложени методолошки приступ фокусира се на вођену употребу GenAI алата ради постизања дефинисаних исхода учења у области развоја софтвера у високошколским установама. То подразумева активно учешће наставника који, поред концепата софтверског инжењерства, студенте уче и о AI темама као што су *prompt engineering*, специфични GenAI алате, њихове одговарајуће функционалности и начини примене у оквиру предмета.

У почетној фази, студенти пролазе кроз свеобухватан процес учења у коме стичу знање о свим концептима релевантним за сваку фазу софтверског развоја. Након ове темељне фазе учења, студентима се пружа детаљан преглед о специфичним AI алатима. Знатан корак у овом аспекту представља укључивање *prompt engineering*-а, са циљем да студенти стекну знање неопходно за коришћење AI у практичним сценаријима током целокупног развоја софтвера. Ово је од суштинског значаја, јер квалитет употребе AI алата у свим фазама у великој мери зависи од ефикасних вештина креирања упита.

У овој фази студенти се упознају са GenAI алатима и уче о њиховој употреби и функционалностима уз подршку наставника. У Табели 5 приказани су различити алати који се могу користити у различитим активностима развоја софтвера. Избор алата за сваку фазу засниван је на најновијим индустријским извештајима који потврђују њихову широку примену [269], као и на прелиминарним интервјуима са професионалцима из области софтверског инжењерства који су идентификовали ове алате као најчешће коришћене у пракси.

Табела 5. Преглед *GenAI* алата по фазама животног циклуса развоја софтвера

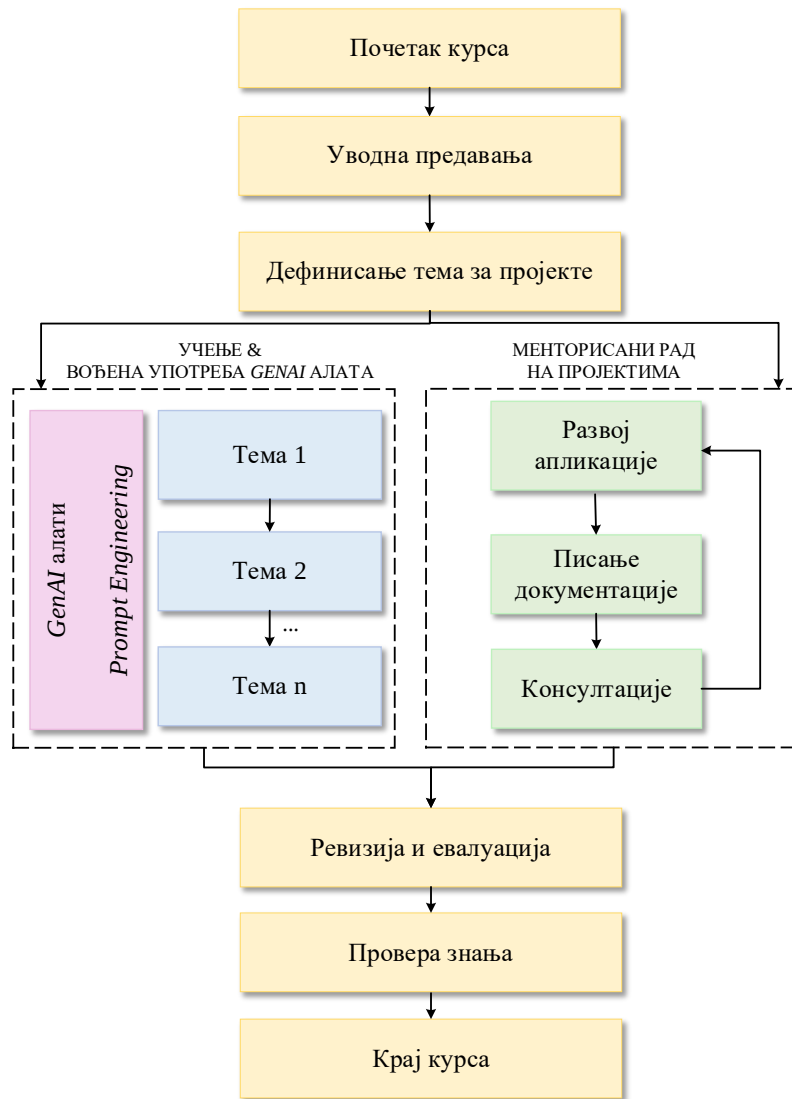
Фаза	<i>GenAI</i> алати
Анализа захтева	<i>ChatGPT, Gemini, DesignCraft AI, Rex AI</i>
Управљање пројектом	<i>ChatGPT, Claude, ChatPRD</i>
Пројектовање	<i>Software Architect GPT module with PlantUML, Eraser.io</i>
Имплементација	<i>GitHub Copilot, Claude, Tabnine</i>
Тестирање	<i>GitHub Copilot, Gemini, Tabnine, Testim.io</i>
Одржавање	<i>Embold.io, CodeScene</i>

Иако Табела 5 приказује алате који се користе у различитим фазама развоја софтвера, важно је нагласити да сви *GenAI* алати нису једнако ефикасни за исте задатке. На пример, у фази имплементације користе се и *GitHub Copilot* и *Tabnine*. Међутим, *Copilot* се често показује супериорним у погледу контекстуалног довршавања кода и разумевања образаца специфичних за пројекат, захваљујући интеграцији са *GitHub*-ом и напреднијим могућностима језичког моделирања [270].

Слично томе, у фази анализе захтева користе се алати попут *ChatGPT* и *Gemini*, који нуде опште могућности великих језичких модела, али се разликују по стилу интеракције и дубини резонувања. За *ChatGPT* се често истиче да пружа детаљније и нијансираније излазе, као и богатију конверзациону меморију [271], док *Gemini* нуди снажнију интеграцију са *Google* алатима и добру ефикасност у задацима резимирања и проналажења чињеница [167]. *Perplexity*, иако мање флексибилан у слободној генерацији, показује високе перформансе у пружању одговора са изворима података [272], што га чини погодним за валидацију захтева кроз спољну документацију. У тест фази, *Copilot* може помоћи у писању *unit* тестова [273], док *Testim.io* предњачи у аутоматизацији *end-to-end* и *UI* тестова кроз визуелне токове. То чини избор између ових алата веома зависним од нивоа тестирања и искуства тима.

Ови компаративни увиди деле се са студентима током наставе како би могли да доносе информисане одлуке приликом одабира алата, нарочито у реалним пројектним контекстима где је неопходно балансирали између употребљивости, тачности и перформанси. Надовезујући се на ово, сваки алат се студентима представља са нагласком на његове функционалности и најбоље праксе за ефикасну употребу у различитим задацима сваке фазе. Ово укључује и упозоравања на уобичајене грешке, као што су нејасне дефиниције класа у току фазе пројектовања или претерано ослањање на аутоматске сугестије у току имплементације. Поред препорука за употребу алата, студенти добијају и смернице за њихову оптимизацију током сваке фазе.

Највиши ниво примене методолошког оквира приказан је на Слици 13. Дијаграм је конципиран као *flowchart* који илуструје главне кораке у примени оквира у наставном процесу. Уместо детаљног приказа појединачних алата или активности, овај модел наглашава логички ток: од иницијалних наставних активности и дефинисања пројеката, преко ангажовања студената у пројектно-заснованом учењу уз менторску подршку, па све до евалуације исхода учења. Овим се на апстрактном нивоу приказује како се методолошки оквир спроводи у пракси кроз низ међусобно повезаних корака.



Слика 13. Примена методолошког оквира у пројектно-заснованом учењу у области развоја софтвера (највиши ниво апстракције)

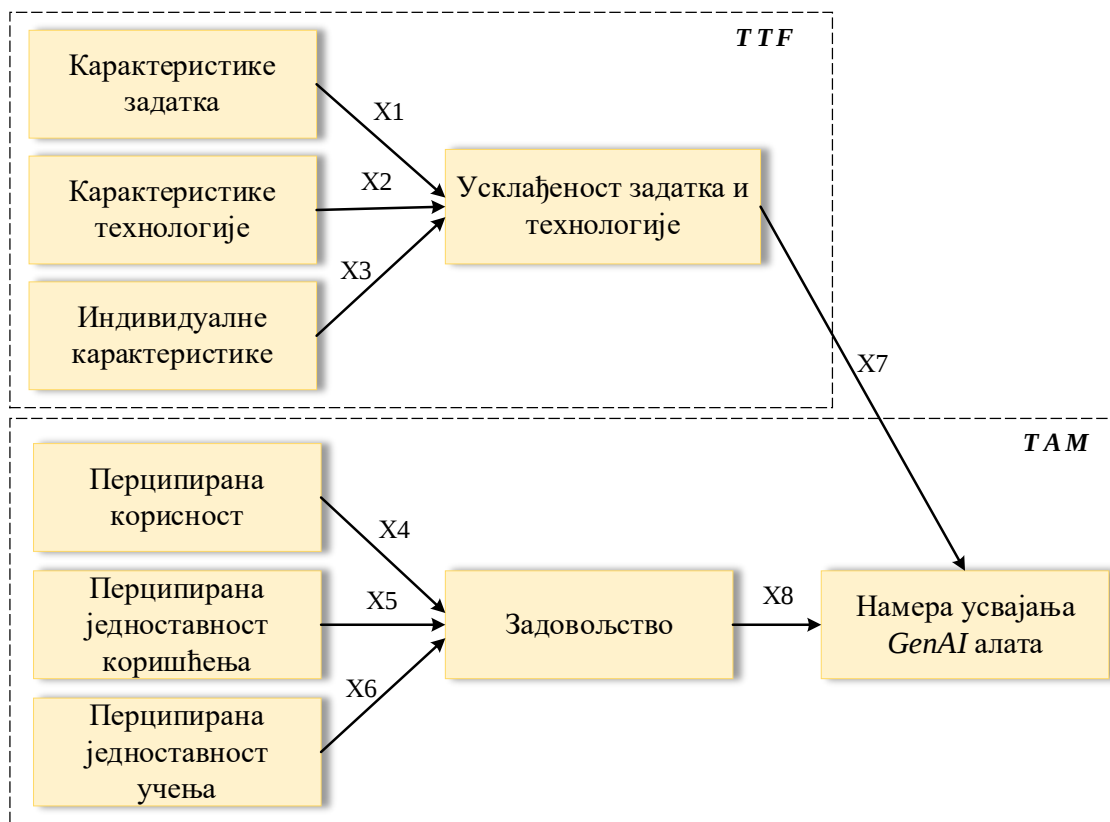
Овако структуриран приступ обезбеђује да студенти стекну и теоријска знања о интеграцији *GenAI* у процесе развоја софтвера и практична искуства у њиховој ефикасној примени. Комбинација ових вештина представља неопходну основу за припрему будућих стручњака у области софтверског инжењерства.

6.3 Развој истраживачког модела за евалуацију предложеног оквира

Ради евалуације предложеног модела, приступ је примењен на конкретним предметима како би се испитало да ли *GenAI* алати имају позитиван утицај на образовање софтвер инжењера и да ли ће студенти показати намеру да усвоје *GenAI* алате за учење у области развоја софтвера. За евалуацију су коришћена два истраживачка модела: *TAM* и *TTF*. *TAM* модел објашњава усвајање и коришћење технологије на основу два кључна фактора, перципиране корисности и перципиране једноставности употребе [274]. Овај модел је посебно релевантан јер обухвата психолошке и бихевиоралне факторе који утичу на прихватање технологије. У овом

истраживању, *TAM* је примењен са циљем да се испита спремност студената да усвоје *GenAI* алате.

С друге стране, *TTF* модел полази од претпоставке да усвајање технологије и постигнута ефикасност зависе од тога у којој мери су могућности технологије усклађене са специфичним захтевима одређеног задатка [275]. Овај модел је изабран јер наглашава практичну повезаност између задатака и технологије, нудећи комплементаран приступ у односу на *TAM*, будући да се фокусира на објективну усклађеност система и задатка, а не само на корисничку перцепцију. На основу теоријског оквира *TAM* и *TTF* модела, предложен је истраживачки модел приказан на Слици 14.



Слика 14. *TAM-TTF* истраживачки модел

Имплементација предложене методологије је усмерена на пет фаза животног циклуса софтвера:

1. Анализа корисничких захтева
2. Управљање софтверским пројектом
3. Пројектовање софтвера
4. Имплементација софтвера
5. Тестирање софтвера

У фазама анализе корисничких захтева и управљања софтверским пројектом студенти су користили *GenAI* алате у комбинацији са алатима *Notion* и *JIRA* за идентификацију, структурирање и управљање захтевима, формулисање корисничких прича, креирање и

приоритизацију листе функционалности, као и за доношење одлука о њиховој вредности. Алат *ChatPRD* коришћен је за израду спецификације захтева производа, док су решења заснована на *LangChain* оквиру и *AI* алати за тестирање примењени за дефинисање и валидацију функционалних и нефункционалних захтева. У фази пројектовања студенти су користили *ChatGPT* и *Google Gemini* алате, док су у фазама имплементације и тестирања користили *GitHub Copilot*, *Google Gemini* и *Tabnine* алате [276].

У поглављу 7 је приказана примена и евалуација предложеног методолошког оквира у два наставна контекста. У поглављу 7.1 разматрана је примена оквира у фазама анализе корисничких захтева и управљања софтверским пројектом, док је у поглављу 7.2 представљена његова примена у фазама пројектовања, имплементације и тестирања софтвера. У оквиру секција 7.1.1 и 7.2.1 извршено је прилагођавање општег *TAM-TTF* истраживачког модела специфичностима наведених наставних активности и формулисане су одговарајуће истраживачке хипотезе.

6.4 Развој алата за евалуацију предложеног оквира

У циљу провере хипотезе Х.5 ове докторске дисертације, према којој се генеративна вештачка интелигенција може ефикасно користити као подршка у формативном и вршњачком оцењивању, развијен је *AI* агент за анализу и евалуацију студентских софтверских пројеката. Алат је првенствено намењен пружању формативне подршке студентима током развоја веб апликација, тако што анализира јавно доступан репозиторијум пројекта и процењује његову усаглашеност са захтевима које су претходно дефинисали наставници. За развој и оркестрацију компоненти агента коришћен је *LangChain* оквир. На основу идентификованих доказа у програмском коду, конфигурационим фајловима, документацији, гранама и историји измена, агент генерише структурисану и контекстуално прилагођену повратну информацију о испуњеним, делимично испуњеним и неиспуњеним захтевима, уз конкретне препоруке за унапређење решења.

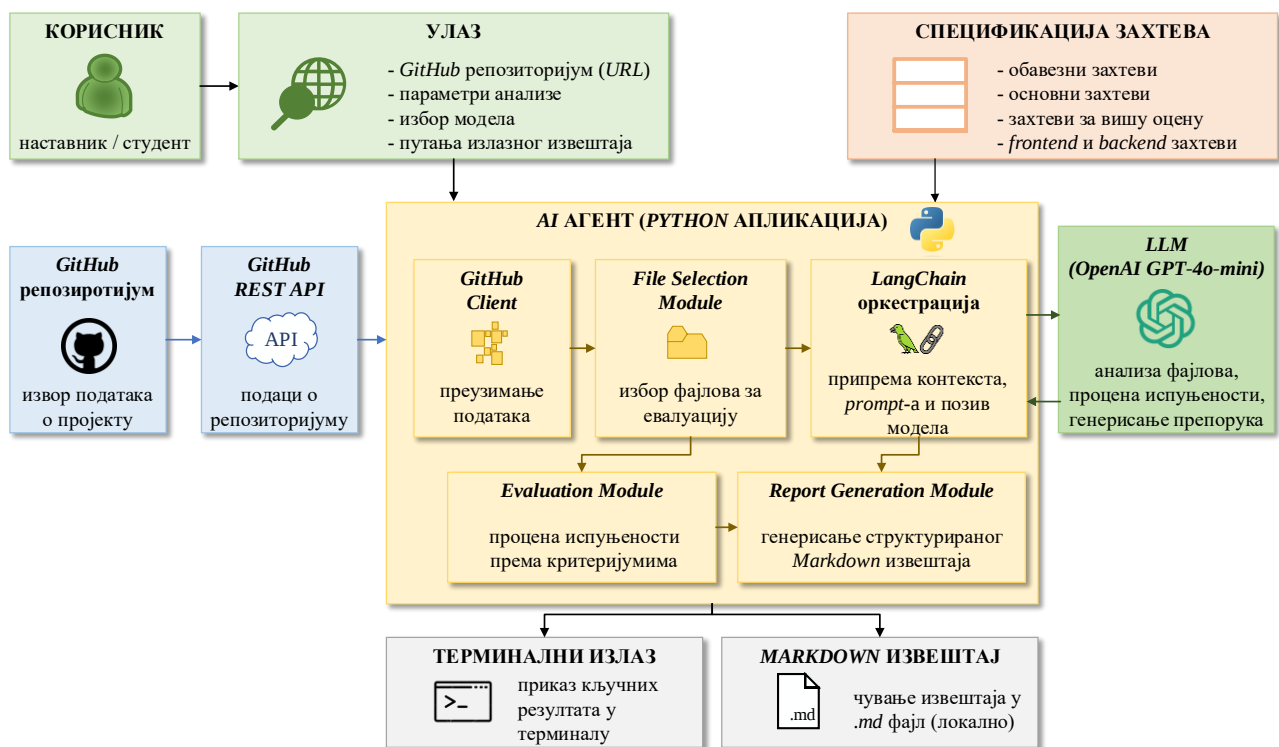
За иницијално тестирање развијеног алата коришћени су захтеви који се примењују у оквиру предмета Интернет технологије, на којем студенти у пару развијају веб апликацију. Ови захтеви обухватају различите аспекте развоја софтвера, укључујући организацију и верзионисање програмског кода, имплементацију *frontend* и *backend* апликације, рад са базом података, развој *REST API*-ја, аутентификацију и ауторизацију, тестирање, писање пројектне документације, *Docker* окружење, *CI/CD* механизме, примену екстерних *API*-ја, безбедност и постављање апликације у продукционо окружење. На тај начин је омогућено да се агент испита на реалном и садржински разноврсном скупу критеријума који одражавају очекиване исходе студентског пројекта и више фаза животног циклуса развоја софтвера.

Примена оваквог алата омогућава благовремено уочавање недостатака и континуирано праћење напретка студената током развоја пројекта. Алат не замењује наставничку процену, већ представља додатни механизам подршке који доприноси доступности, правовремености, доследности и квалитету повратне информације.

6.4.1 Архитектура AI агента за анализу и евалуацију студентских софтверских пројеката

Архитектура развијеног алата заснива се на модуларном приступу, у оквиру којег су раздвојене компоненте за приступ *GitHub* репозиторијуму, избор релевантних фајлова, оркестрацију поступка анализе, евалуацију испуњености захтева и генерисање повратне информације. За повезивање компоненти и управљање обрадом употребљен је *LangChain* оквир, док језички модел *OpenAI GPT-4o-mini* представља компоненту задужену за анализу прикупљених фајлова, процену усаглашености пројекта са дефинисаним критеријумима и формулисање препорука за његово унапређење.

Пројектовање система ослоњено је на основне компоненте архитектура AI агената заснованих на великим језичким моделима, које обухватају планирање активности, употребу алата, меморију и рефлексiju [89], [277]. У развијеној верзији алата ови принципи примењени су кроз оркестрацију поступка анализе, коришћење *GitHub REST API*-ја као спољашњег извора података, структурирање контекста који се прослеђује језичком моделу, чување генерисаног извештаја и могућност поновног покретања анализе након измена пројекта. Архитектура развијеног алата приказана је на Слици 15.



Слика 15. Архитектура AI агента за анализу и евалуацију студентских пројеката

Корисник, односно студент или наставник, покреће анализу уношењем *URL* адресе јавно доступног *GitHub* репозиторијума. Поред адресе репозиторијума, могуће је дефинисати параметре извршавања, као што су максималан број анализираних фајлова и преузетих *commit*-ова, избор језичког модела и путања за чување излазног извештаја. Захтеви пројектног задатка представљају одвојени улаз у систем и обухватају обавезне, основне, *frontend* и *backend*

захтеве, као и критеријуме за постизање више оцене. Ови захтеви су детаљно представљени у поглављу 6.4.4.

Подаци о студентском пројекту преузимају се из *GitHub* репозиторијума посредством *GitHub REST API*-ја. Компонента *GitHub Client* прикупља метаподатке о репозиторијуму, податке о гранама, историји *commit*-ова, ауторима и контрибуторима, као и структури и садржају фајлова. Након прикупљања података, модул за избор фајлова издваја елементе који су најрелевантнији за проверу дефинисаних критеријума. Предност се даје конфигурационим и изворним фајловима који могу садржати доказе о примени *Docker*-а, *CI/CD* механизма, тестова, *Swagger* или *OpenAPI* спецификације, *frontend* компоненти, *backend* модела, миграција, *API* рута, аутентификације и безбедносних механизма.

Централни део архитектуре представља *Python* апликација у оквиру које су интегрисани модули за прикупљање података, избор релевантних фајлова, оркестрацију, евалуацију и генерисање извештаја. *LangChain* компонента припрема контекст за анализу, повезује системске инструкције, спецификацију захтева и прикупљене доказе, а затим активира језички модел. Модел *GPT-4o-mini* анализира расположиве податке и за сваки захтев процењује степен испуњености. При томе је инструкцијама ограничен на коришћење достављених фајлова и не треба да проглашава захтев испуњеним уколико у анализираном садржају репозиторијума не постоји одговарајућа потврда.

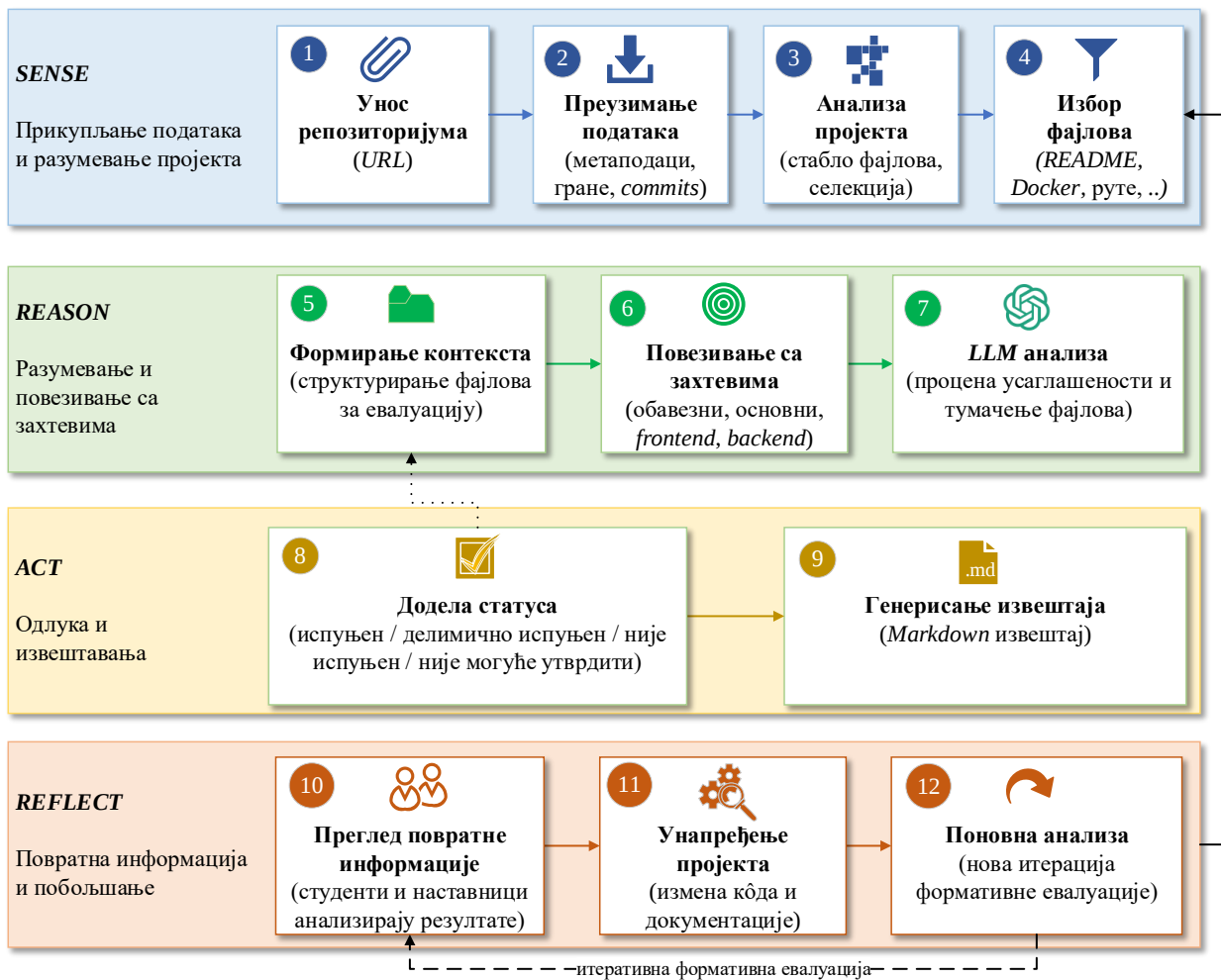
На основу резултата анализе модул за евалуацију додељује сваком захтеву један од предвиђених статуса: испуњен, делимично испуњен, није испуњен или није могуће утврдити. Потом модул за генерисање извештаја формира структурирану формативну повратну информацију која садржи основне податке о репозиторијуму, преглед статуса, доказе, образложења, препоруке и ограничења анализе.

Резултат рада система приказује се у терминалу и чува као *Markdown* извештај у локалном фајлу. Овакав начин чувања омогућава накнадни преглед и дељење резултата, као и поређење извештаја насталих у различитим итерацијама развоја пројекта. Извештај може да се користи за самопроцену студената, као додатни извор информација наставнику и за праћење напретка током реализације пројектног задатка.

Приказана архитектура раздваја приступ и припрему података од њихове интерпретације. *GitHub REST API* и компонента *GitHub Client* обезбеђују структуриране податке о пројекту, док модул за избор релевантних фајлова сужава контекст на елементе који могу бити значајни за проверу постављених критеријума. На основу тако припремљеног контекста, језички модел генерише повратну информацију засновану на идентификованим фајловима, а не искључиво на општем опису пројекта.

6.4.2 Ток рада AI агента (Agent Workflow)

Рад агента може се описати кроз петљу *Sense-Reason-Act-Reflect*, која представља један од често коришћених образаца у системима заснованим на великим језичким моделима [278], [279]. У развијеном алату овај образац не представља архитектуру потпуно аутономног агента, већ служи као погодан оквир за опис процеса прикупљања података, њихове интерпретације, генерисања повратне информације и поновне евалуације након измена пројекта. Ток извршавања AI агента приказан је на Слици 16.



Слика 16. Ток рада AI агента током формативне евалуације студентског пројекта

Процес започиње тако што корисник уноси URL адресу јавно доступног *GitHub* репозиторијума. На основу тог улаза систем преузима метаподатке о пројекту, податке о гранама, *commit*-овима и контрибуторима, а затим анализира структуру репозиторијума и формира иницијални скуп података за даљу обраду. Уместо да обрађује читав репозиторијум, агент издваја релевантне фајлове који највероватније садрже доказе о испуњености постављених захтева, као што су *README.md*, *Dockerfile*, *Docker Compose* конфигурација, *workflow* фајлови, тестови, руте, модели, миграције и други елементи значајни за евалуацију. На тај начин **фаза Sense** обухвата не само прикупљање података из окружења, већ и њихово почетно филтрирање и припрему за анализу [280].

Након тога следи **фаза Reason**, у којој се издвојени фајлови организују у структуриран контекст и повезују са унапред дефинисаним захтевима пројектног задатка. У овој фази *LangChain* оквир има улогу оркестрације: припрема улазни контекст, повезује фајлове и делове фајлова са критеријумима евалуације и усмерава језички модел ка процени релевантних аспеката пројекта. Тако се, на пример, конфигурациони фајлови доводе у везу са захтевима који се односе на *Docker* и *CI/CD*, модели и миграције са *backend* захтевима, а тестови, документација и безбедносни механизми са одговарајућим критеријумима квалитета. Уместо једноставног сажимања садржаја репозиторијума, модел овде има задатак да процени степен усаглашености пројекта са постављеним захтевима и да идентификује довољно јаке доказе за сваки појединачни закључак.

У **фази Act** агент на основу тако припремљеног контекста генерише структуриран извештај о евалуацији. За сваки захтев одређује се одговарајући статус - испуњен, делимично испуњен, није испуњен или није могуће утврдити, а затим се наводе конкретни докази из репозиторијума, кратко образложење и препорука за унапређење. Поред детаљне анализе, агент формира и табеларни преглед резултата, као и листу приоритетних препорука које студентима могу указати на најважније недостатке пројекта. На овај начин повезују се способности аутоматизованог прикупљања података и генеративне могућности језичког модела, што је у складу са приступима у којима се велики језички модели проширују употребом спољашњих алата.

Завршни корак односи се на **Reflect фазу**, која је у овој верзији алата реализована као спољашња, а не унутрашња петља. Генерисани извештај чува се у *Markdown* формату, што омогућава трајност резултата, њихов накнадни преглед и поређење различитих верзија пројекта. Студенти и наставници могу анализирати добијену повратну информацију, учити недостатке, унапредити програмски код, конфигурацију или пројектну документацију, а затим поново покренути анализу над ажурираном верзијом репозиторијума. Управо овај корак поновног покретања анализе даје процесу итеративни карактер и омогућава да алат буде коришћен као средство континуиране формативне евалуације, а не само као механизам за једнократну проверу завршног решења.

6.4.3 Технологије и структура пројекта

Алат је реализован у програмском језику *Python*, док је за оркестрацију комуникације са великим језичким моделом коришћен *LangChain* оквир. Приступ подацима о репозиторијуму остварен је посредством *GitHub REST API*-ја и библиотеке *requests*, док се конфигурациони параметри, као што су *OpenAI API* кључ и назив језичког модела, читавају из *.env* фајла применом библиотеке *python-dotenv*. За анализу и генерисање повратне информације употребљен је *OpenAI* језички модел, при чему је у тренутној верзији као подразумевани модел постављен *GPT-4o-mini*. Резултат анализе генерише се у *Markdown* формату, што омогућава његово једноставно читање, чување и накнадну употребу.

Пројекат је организован у више програмских модула, од којих сваки има јасно дефинисану улогу:

- **main.py** представља улазну тачку апликације. У овом фајлу обрађују се аргументи командне линије, преузима *URL* адреса *GitHub* репозиторијума и покрећу функције за прикупљање података и евалуацију пројекта. Поред тога, у њему се приказују основни подаци о прикупљеним фајловима и генерисани извештај чува у излазном *Markdown* фајлу.
- **github_client.py** садржи имплементацију клијента за комуникацију са *GitHub REST API*-јем. Овај модул преузима метаподатке репозиторијума, гране, *commit*-ове, податке о контрибуторима, стабло фајлова и садржај одабраних фајлова. У истом фајлу реализовани су механизми за филтрирање генерисаних и зависних фајлова, рангирање потенцијално релевантних путања и избор фајлова који могу садржати доказе о испуњености постављених захтева.
- **project_requirements.py** садржи централизовану спецификацију критеријума према којима се врши евалуација студентског пројекта. Захтеви су груписани на обавезне, основне, захтеве за вишу оцену, *frontend* и *backend* захтеве. У овом модулу дефинисани су и дозвољени статуси испуњености, чиме је омогућено да се критеријуми мењају независно од остатка логике апликације.
- **analyzer.py** представља централни модул за евалуацију. Он садржи системски *prompt*, правила анализе и очекивану структуру излазног извештаја. У њему се подаци прикупљени из репозиторијума форматирају и, посредством *LangChain* оквира, прослеђују језичком моделу заједно са дефинисаним захтевима. Резултат обраде је структурисана повратна информација на српском језику, која за сваки захтев обухвата статус, доказе, образложење и препоруку.
- **.env** служи за чување локалних конфигурационих вредности и осетљивих података, пре свега *OPENAI_API_KEY*, назива изабраног модела и, опционо, *GitHub* токена. Овај фајл се не поставља у јавно доступан репозиторијум.
- **requirements.txt** и **pyproject.toml** садрже податке о зависностима и конфигурацији *Python* пројекта, неопходне за инсталацију и репродукцију развојног окружења.

Током извршавања апликације генерише се фајл **project_evaluation.md**, који садржи комплетан извештај о анализи. Извештај обухвата основне податке о репозиторијуму, сумарни преглед статуса, детаљну процену сваког захтева, приоритетне препоруке и ограничења спроведене анализе.

6.4.4 Функционалности AI агента

Развијени *AI* агент обједињује функционалности прикупљања података из *GitHub* репозиторијума, избора релевантних доказа, процене испуњености пројектних захтева и генерисања формативне повратне информације. Корисник као основни улаз задаје *URL* адресу јавно доступног репозиторијума, након чега се поступак анализе извршава аутоматски.

Кључне функционалности агента обухватају:

1. Прикупљање података о репозиторијуму

Посредством *GitHub REST API*-ја агент преузима:

- основне метаподатке о репозиторијуму;
- податке о његовој видљивости и подразумеваној грани;
- листу расположивих грана;
- историју *commit*-ова и њихове поруке;
- податке о ауторима *commit*-ова и контрибуторима;
- садржај одабраних фајлова релевантних за евалуацију.

На тај начин омогућена је провера критеријума који се не могу утврдити искључиво анализом програмског кода, као што су јавна доступност репозиторијума, примена стратегије гранања, минималан број *commit*-ова и учешће чланова тима у развоју пројекта. Наведена функционалност реализована је методом ***fetch_repository_evidence()*** класе *GitHubClient*, која обједињује позиве метода за преузимање метаподатака, грана, *commit*-ове, контрибутора, стабла репозиторијума и садржаја одабраних фајлова. Део имплементације ове методе приказан је на Слици 17.

```
github_client.py > GitHubClient > fetch_repository_evidence
78 class GitHubClient:
100     def fetch_repository_evidence(self,
101                                   repository_url: str,
102                                   max_files: int = 30,
103                                   max_commits: int = 100,
104                                   max_chars_per_file: int = 12_000,) -> RepositoryEvidence:
105         """
106         Collects repository metadata and evidence needed for evaluating
107         the project against the predefined requirements.
108         """
109         repository = self.get_repository(repository_url)
110
111         owner = repository["owner"]["login"]
112         repo = repository["name"]
113         default_branch = repository["default_branch"]
114
115         tree = self._get(
116             f"/repos/{owner}/{repo}/git/trees/{quote(default_branch, safe='')}",
117             params={"recursive": "1"},
118         )
119
120         tree_paths = sorted(
121             item["path"]
122             for item in tree.get("tree", [])
123             if item.get("type") == "blob"
124             and not _looks_like_generated_or_vendor_file(item["path"])
125         )
126
127         branches = self.fetch_branches(owner, repo)
128         commits = self.fetch_commits(
129             owner=owner,
130             repo=repo,
131             max_commits=max_commits,
132         )
133         contributors = self.fetch_contributors(owner, repo)
```

Слика 17. Функција *fetch_repository_evidence()* - прикупљање и обједињавање података о *GitHub* репозиторијуму студентског пројекта

2. Избор фајлова релевантних за евалуацију

Систем филтрира генерисане фајлове, директоријуме са спољашњим зависностима и друге елементе који нису значајни за проверу постављених критеријума. Предност се даје фајловима и директоријумима који могу садржати конкретне доказе о имплементацији, као што су:

- *Dockerfile* и *Docker Compose* конфигурација;
- *GitHub Actions workflow* фајлови;
- *Swagger* и *OpenAPI* конфигурација; тестови;
- *frontend* странице, компоненте и руте;
- *backend* модели, миграције, контролери и *API* руте;
- механизми аутентификације и ауторизације;
- *middleware* компоненте и безбедносна конфигурација;
- фајлови повезани са *deployment*-ом, визуелизацијом података и екстерним *API*-јима.

Избор се реализује функцијом `select_requirement_relevant_paths()`, која рангира расположиве путање на основу вредности добијене функцијом `_requirement_path_score()`. На тај начин се пре позива језичког модела примењује детерминистичка селекција фајлова који највероватније садрже доказе релевантне за дефинисане захтеве, као што је приказано на Слици 18.



```

github_client.py > _requirement_path_score
393 def select_requirement_relevant_paths(file_paths: list[str], max_files: int = 30,) -> list[str]:
394     unique_paths = sorted(set(file_paths))
395     ranked_paths = sorted(
396         unique_paths,
397         key=lambda path: (
398             _requirement_path_score(path),
399             path.count("/"),
400             path.lower(),
401         ),
402     )
403     return ranked_paths[:max_files]
404
405 def _requirement_path_score(path: str) -> int:
406     normalized = path.lower()
407     filename = normalized.split("/")[-1]
408
409     exact_priority = {
410         "readme.md": 0,
411         "dockerfile": 0,
412         "docker-compose.yml": 0, "docker-compose.yaml": 0,
413         "compose.yml": 0, "compose.yaml": 0,
414         "package.json": 1,
415         "requirements.txt": 1,
416         "pyproject.toml": 1,
417         "composer.json": 1,
418         ".env.example": 1,
419         "openapi.yaml": 1, "openapi.yml": 1,
420         "swagger.yaml": 1,
421     }
422
423     if filename in exact_priority:
424         return exact_priority[filename]
425
426     if normalized.startswith(".github/workflows/"):
427         return 0
428
  
```

Слика 18. Функције `select_requirement_relevant_paths()` и `_requirement_path_score()` - избор и рангирање фајлова релевантних за евалуацију

3. Евалуација пројекта према дефинисаним захтевима

Централна функционалност агента јесте процена усаглашености пројекта са унапред дефинисаним критеријумима. За иницијално тестирање алата употребљени су захтеви пројектног задатка са предмета Интернет технологије, у оквиру којег студенти развијају *full-stack* веб апликацију. Захтеви су организовани у пет група.

Обавезни захтеви обухватају:

- докеризацију апликације применом *Docker* алата;
- употребу *Docker Compose* конфигурације;
- обезбеђивање *Swagger*, односно *OpenAPI* спецификације;
- укључивање *API* спецификације у пројектну документацију.

Захтеви за вишу оцену обухватају:

- употребу најмање два екстерна *API*-ја;
- постојање *README* фајла са описом и упутством за покретање;
- постојање *main* или *master*, *develop* и најмање две *feature* гране;
- имплементацију *CI/CD pipeline*-а;
- аутоматско покретање тестова на *push* и *pull request* догађаје;
- аутоматизовану изградњу *Docker image*-а;
- постављање продукционе верзије апликације на *Cloud* платформу;
- примену заштите од најмање три уобичајене безбедносне претње;
- постојање аутоматизованих тестова;
- реализацију визуелизације података или напредно подешене интерактивне мапе

Основни захтеви односе се на:

- употребу *Git*-а за верзионисање програмског кода;
- постојање најмање десет смислених *commit*-ова;
- учешће оба члана тима у развоју пројекта;
- јавну доступност репозиторијума;
- повезаност пројекта са *Elab GitHub* организацијом;
- постојање пратеће документације са скриншотовима екрана, описима функционалности, деловима програмског кода и одговарајућим објашњењима.

Frontend захтеви обухватају:

- употребу *React*, *Angular* или другог одговарајућег *frontend* оквира;
- постојање најмање три различите странице или *page-level* компоненте;
- најмање три *reusable* компоненте;
- примену *CSS*-а или другог начина стилизовања;
- реализацију најмање три функционалности;
- употребу *React hook*-ова код *React* пројектата;
- имплементацију *pipe*-а и атрибутске директиве код *Angular* пројектата;
- дефинисање одговарајућих *frontend* рута.

Backend захтеви обухватају:

- најмање пет међусобно повезаних модела;
- најмање три различита типа миграција;
- имплементацију *API* рута и контролера у складу са *REST* конвенцијама;
- враћање података и грешака у *JSON* формату;
- постојање најмање три типа *API* операција;
- дефинисање најмање три типа корисника или корисничке улоге;
- имплементацију регистрације, пријављивања и одјављивања корисника;
- ограничавање приступа заштићеним рутама на аутентификоване и одговарајуће ауторизоване кориснике.

Критеријуми евалуације издвојени су из програмске логике и централизовано дефинисани у фајлу ***project_requirements.py***, у оквиру променљиве ***PROJECT_REQUIREMENTS***. На тај

начин могуће је мењати, допуњавати или прилагођавати захтеве другом предмету и пројектном задатку без измене компоненти задужених за приступ *GitHub* репозиторијуму и комуникацију са језичким моделом. Део спецификације захтева приказан је на Слици 19.

```
project_requirements.py > ...
3 PROJECT_REQUIREMENTS = """
4 MANDATORY REQUIREMENTS
5
6 M1. The application must be containerized using Docker.
7
8 M2. The application must include a Docker Compose configuration.
9
10 M3. The project must provide an API specification generated with Swagger
11 or another OpenAPI-compatible tool.
12
13 M4. The API specification must be referenced or included in the project
14 documentation.
15
16
17 ADVANCED-GRADE REQUIREMENTS
18
19 A1. The project must use at least two external APIs.
```

Слика 19. Фајл *project_requirements.py* - део спецификације критеријума за евалуацију студентског пројекта

4. Додела статуса испуњености

За сваки појединачни захтев агент додељује један од следећих статуса:

- испуњен - у репозиторијуму постоје конкретни докази о испуњености захтева;
- делимично испуњен - пронађен је део тражене имплементације, али критеријум није испуњен у целини;
- није испуњен - доступни подаци јасно указују да захтев није реализован;
- није могуће утврдити - расположиви подаци нису довољни за доношење закључка.

Дозвољени статуси централизованом су дефинисани у фајлу ***project_requirements.py***, док се у системском *prompt*-у моделу задаје да за сваки захтев мора употребити тачно један од наведених статуса. Истовремено је дефинисан начин њиховог превођења у извештају намењеном студентима, као што је приказано на Слици 20.

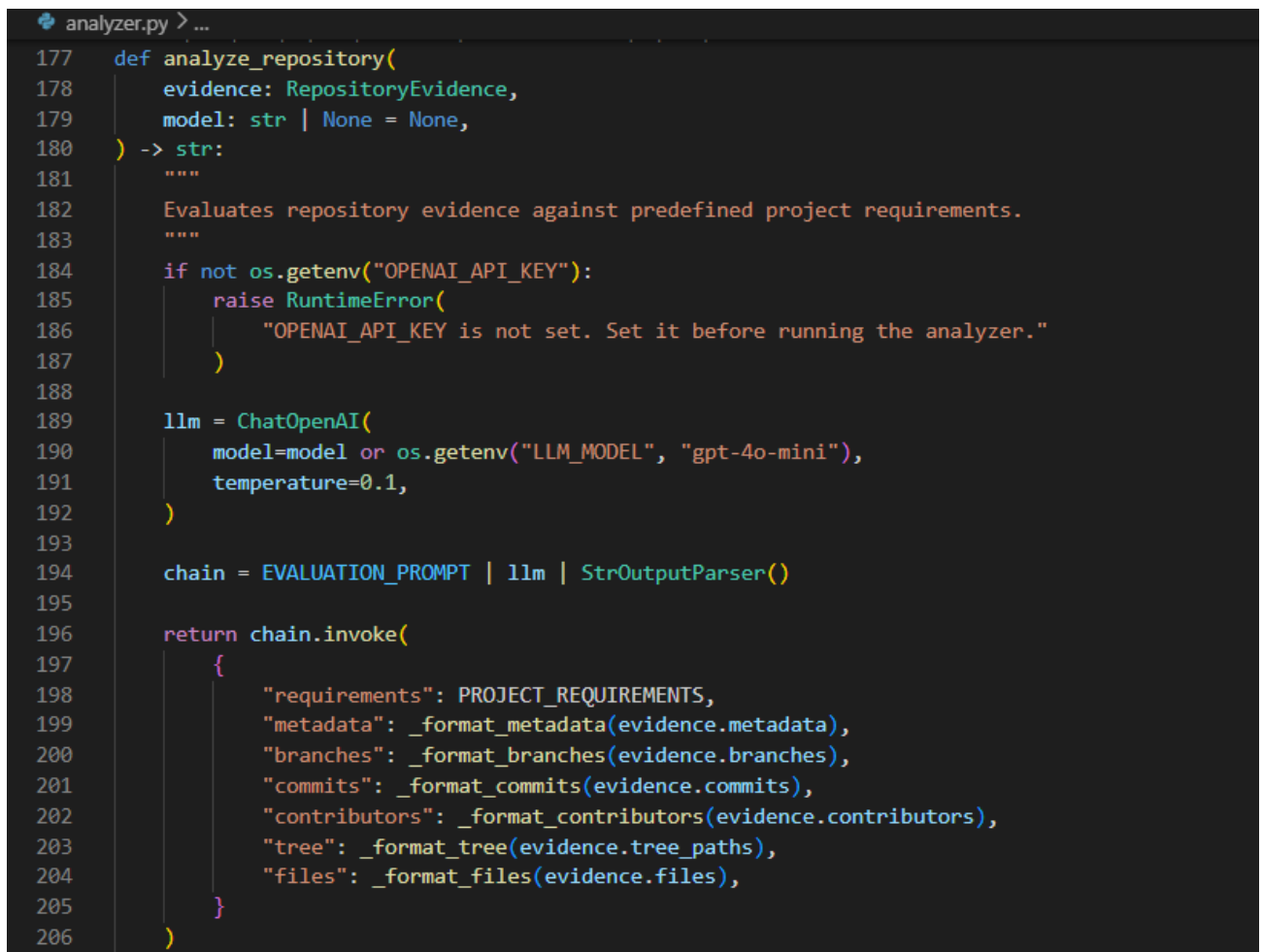
```
analyzer.py > ...
12 EVALUATION_PROMPT = ChatPromptTemplate.from_messages(
13     [
14         (
15             "system",
16             """
17 You are an AI agent for formative evaluation of student software projects.
18 Your task is to evaluate a GitHub repository against predefined project
19 requirements.
20 Use only the repository evidence provided in the user message. Do not assume
21 that a requirement is satisfied unless there is concrete evidence.
22 Evaluation rules:
23 1. Evaluate every requirement separately.
24 2. Use exactly one of these internal statuses:
25     - SATISFIED
26     - PARTIALLY_SATISFIED
27     - NOT_SATISFIED
28     - NOT_DETERMINABLE
29 """
30         )
31     ]
32 )
```

Слика 20. Системски *prompt* - дефинисање статуса испуњености захтева

5. Генерисање формативне повратне информације

На основу спроведене евалуације агент генерише структуриран извештај на српском језику. Извештај садржи основне податке о репозиторијуму, табеларни преглед свих захтева и њихових статуса, детаљну анализу сваког захтева, конкретне доказе из репозиторијума, образложење додељеног статуса, препоруке за унапређење, листу приоритетних активности, ограничења извршене анализе.

Функционалност генерисања формативне повратне информације реализована је у функцији `analyze_repository()` у фајлу `analyzer.py`. Ова функција иницијализује изабрани језички модел, формира *LangChain* ланац повезивањем објекта `EVALUATION_PROMPT`, језичког модела и парсера излазног резултата, а затим моделу прослеђује дефинисане захтеве и структуриране податке прикупљене из репозиторијума. Као резултат извршавања функције генерише се извештај на српском језику који садржи преглед испуњености захтева, образложења, препоруке и ограничења анализе. Део имплементације функције `analyze_repository()` приказан је на Слици 21.



```

177 def analyze_repository(
178     evidence: RepositoryEvidence,
179     model: str | None = None,
180 ) -> str:
181     """
182     Evaluates repository evidence against predefined project requirements.
183     """
184     if not os.getenv("OPENAI_API_KEY"):
185         raise RuntimeError(
186             "OPENAI_API_KEY is not set. Set it before running the analyzer."
187         )
188
189     llm = ChatOpenAI(
190         model=model or os.getenv("LLM_MODEL", "gpt-4o-mini"),
191         temperature=0.1,
192     )
193
194     chain = EVALUATION_PROMPT | llm | StrOutputParser()
195
196     return chain.invoke(
197         {
198             "requirements": PROJECT_REQUIREMENTS,
199             "metadata": _format_metadata(evidence.metadata),
200             "branches": _format_branches(evidence.branches),
201             "commits": _format_commits(evidence.commits),
202             "contributors": _format_contributors(evidence.contributors),
203             "tree": _format_tree(evidence.tree_paths),
204             "files": _format_files(evidence.files),
205         }
206     )

```

Слика 21. Функција `analyzer.py` - генерисање формативне повратне информације

6. Чување резултата и поновна анализа

Генерисани извештај чува се у *Markdown* формату, што омогућава његов накнадни преглед, дељење и поређење са резултатима наредних анализа. Студенти могу да измене програмски код, конфигурацију или документацију у складу са добијеним препорукама, а затим поново покрену агента над ажурираном верзијом репозиторијума. На тај начин алат подржава континуирану самопроцену и итеративно унапређивање студентског пројекта.

Чување резултата реализовано је у функцији `main()`, употребом класе `Path`. Након генерисања извештаја креира се потребна структура директоријума, а добијени текст се уписује у `Markdown` фајл применом `UTF-8` кодирања, као што је приказано на Слици 22.

```
main.py > ...
11 def main() -> None:
12     """Runs repository evidence collection and project evaluation."""
13     load_dotenv()
14
15     args = parse_args()
16     repository_url = (
17         args.repository_url
18         or input("GitHub repository URL: ").strip()
19     )
20
21     try:
22         client = GitHubClient()
23
24         print("\nCollecting repository evidence...\n")
25
26         evidence = client.fetch_repository_evidence(
27             repository_url=repository_url,
28             max_files=args.max_files,
29             max_commits=args.max_commits,
30         )
31
32         print_repository_overview(evidence)
33
34         print("\nEvaluating project requirements with the AI agent...\n")
35
36         report = analyze_repository(
37             evidence=evidence,
38             model=args.model,
39         )
40
41         output_path = Path(args.output)
42         output_path.parent.mkdir(parents=True, exist_ok=True)
43         output_path.write_text(report, encoding="utf-8")
44
45         print(report)
46         print(f"\nEvaluation report saved to: {output_path.resolve()}")
```

Слика 22. Функција `main()` - чување генерисаног извештаја у `Markdown` формату

Наведене функционалности омогућавају да се развијени алат користи као механизам за рано откривање неиспуњених захтева, пружање структурисане формативне повратне информације и праћење напретка током развоја пројекта. Истовремено, генерисани извештај наставницима може послужити као додатни извор информација, али не замењује стручни преглед програмског решења и коначну наставничку процену.

6.4.5 Приказ примене AI агента на студентском софтверском пројекту

Примена развијеног AI агента приказана је на примеру анализе `GitHub` репозиторијума студентског софтверског пројекта који је урађен на предмету Интернет технологије. Пре покретања апликације потребно је активирати виртуелно `Python` окружење и обезбедити вредност променљиве `OPENAI_API_KEY` у локалном `.env` фајлу. `GitHub` токен није обавезан за анализу јавно доступних репозиторијума, али се може користити ради повећања дозвољеног броја захтева ка `GitHub REST API`-ју.

Алат се може покренути на два начина. У првом случају *URL* адреса репозиторијума прослеђује се непосредно приликом покретања главног програмског фајла *python main.py https://github.com/OWNER/REPOSITORY*.

Уколико *URL* адреса није прослеђена као аргумент командне линије (Слика 23), апликација се покреће командом *python main.py*. Након тога се од корисника захтева да адресу репозиторијума унесе интерактивно у терминалу. Ова могућност олакшава коришћење алата корисницима који не желе да све параметре наводе у почетној команди.

```
milica@DESKTOP-JKJDDM3 MINGW64 ~/Desktop/elab/disertacija/4. disertacija/Agent/Agent (master)
$ source "/c/Users/milica/Desktop/elab/disertacija/4. disertacija/Agent/Agent/.venv/Scripts/activate"
(.venv)
milica@DESKTOP-JKJDDM3 MINGW64 ~/Desktop/elab/disertacija/4. disertacija/Agent/Agent (master)
$ python main.py
GitHub repository URL: https://github.com/elab-development/internet-tehnologije-2025-githubactivitydashboard_2022_0175
```

Слика 23. Интерактиван унос *URL* адресе након покретања *AI* агента

Поред основног начина покретања, корисник може подесити параметре којима се одређују број фајлова за анализу, избор језичког модела и локација излазног фајла. Аргументом *--max-files* задаје се максималан број релевантних фајлова чији ће садржај бити прослеђен језичком моделу. Аргумент *--max-commits* одређује максималан број *commit*-ова који се преузима из историје репозиторијума. Помоћу аргумента *--model* може се изабрати *OpenAI* модел, док се аргументом *--output* дефинишу назив и путања генерисаног *Markdown* извештаја. Наведени параметри нису обавезни. Уколико нису прослеђени, апликација користи подразумеване вредности дефинисане у програмском коду и *.env* фајлу. Подразумевано се анализира највише 30 одабраних фајлова, преузима до 100 *commit*-ова, користи модел наведен у променљивој *LLM_MODEL*, а извештај се чува под називом *project_evaluation.md*.

Након покретања, апликација проверава унету адресу и приступа јавно доступном *GitHub* репозиторијуму. У терминалу се приказује порука да је започето прикупљање података, након чега *GitHub* клијент преузима метаподатке репозиторијума, листу грана, историју *commit*-ова, податке о контрибуторима и стабло фајлова. Потом се примењује механизам филтрирања и рангирања, којим се издваја ограничен број фајлова релевантних за проверу дефинисаних захтева. По завршетку ове фазе апликација приказује основне информације о спроведеној анализи (Слика 24):

- назив анализираног репозиторијума;
- видљивост репозиторијума;
- подразумевану грану;
- број преузетих грана;
- број преузетих *commit*-ова;
- број идентификованих контрибутора;
- укупан број пронађених фајлова;
- број фајлова изабраних за анализу.

Ови подаци не представљају резултат евалуације, већ омогућавају кориснику да провери да ли је репозиторијум успешно учитан и колики је обухват прикупљених података.

```
milica@DESKTOP-JKJDDM3 MINGW64 ~/Desktop/elab/disertacija/4. disertacija/Agent/Agent (master)
$ python main.py
GitHub repository URL: https://github.com/elab-development/internet-tehnologije-2025-githubactivitydashboard_2022_0175

Collecting repository evidence...

Repository: elab-development/internet-tehnologije-2025-githubactivitydashboard_2022_0175
Visibility: public
Default branch: main
Branches retrieved: 7
Commits retrieved: 83
Contributors retrieved: 2
Repository files found: 68
Files selected for content analysis: 30

Selected files:
- docker-compose.yml
- backend/Dockerfile
- frontend/Dockerfile
- frontend/README.md
- backend/requirements.txt
- frontend/package.json
- backend/app_models/models.py
- backend/migrations/alembic.ini
- backend/migrations/env.py
- backend/migrations/README
- backend/migrations/script.py.mako
- backend/routes/activity_routes.py
```

Слика 24. Приказ прикупљених података из GitHub репозиторијума

Након формирања скупа података апликација покреће евалуацију пројекта помоћу AI агента. У овој фази дефинисани пројектни захтеви, прикупљени *GitHub* подаци и садржај одабраних фајлова прослеђују се језичком моделу посредством *LangChain* оквира. По завршетку обраде генерисани резултат се приказује у терминалу и истовремено чува у *Markdown* фајлу. На крају извршавања апликација исписује апсолутну путању сачуваног извештаја, чиме корисник добија потврду да је поступак успешно завршен. Назив и локација фајла зависе од вредности аргумента `--output`, док се у одсуству тог аргумента користи подразумевани назив *project_evaluation.md* (Слика 25).

```
milica@DESKTOP-JKJDDM3 MINGW64 ~/Desktop/elab/disertacija/4. disertacija/Agent/Agent (master)
$ python main.py
**Preporuka:**
Preporučuje se implementacija zaštite pristupa zaštićenim rutama.

## 4. Prioritetne preporuke

1. Implementirati Swagger ili OpenAPI specifikaciju za API.
2. Uključiti API specifikaciju u dokumentaciju.
3. Implementirati CI/CD pipeline.
4. Postaviti aplikaciju na cloud platformu.
5. Implementirati zaštitu od uobičajenih sigurnosnih pretnji.
6. Povezati repozitorijum sa Elab GitHub organizacijom.
7. Dodati podržavajuću dokumentaciju.
8. Razmotriti dodavanje vizualizacije podataka ili interaktivne mape.
9. Razmotriti korišćenje Angular-a ili ukloniti taj zahtev.
10. Implementirati zaštitu pristupa pristupa zaštićenim rutama.

## 5. Ograničenja analize

Nije moguće utvrditi zahteve koji se odnose na CI/CD pipeline, produkciju verziju aplikacije na cloud platformi, zaštitu od sigurnosnih pretnji, kao i povezanost sa Elab GitHub organizacijom, jer nema dovoljno dokaza u dostupnim fajlovima.

Evaluation report saved to: C:\Users\milica\Desktop\elab\disertacija\4. disertacija\Agent\Agent\project_evaluation.md
(.venv)
```

Слика 25. Чување генерисаног извештаја

Описани поступак показује да се примена агента своди на прослеђивање репозиторијума и, по потреби, додатних параметара анализе, док се прикупљање података, избор релевантног

контекста, позивање језичког модела и чување резултата извршавају аутоматски. На тај начин алат се може користити више пута током развоја студентског пројекта, без потребе за изменом програмског кода агента. Садржај генерисаног извештаја и облици повратне информације представљени су у наредном поглављу.

6.4.6 Резултати анализе и повратне информације AI агента

Резултат рада AI агента представља структуриран извештај о усаглашености анализираног студентског софтверског пројекта са унапред дефинисаним захтевима. Извештај се генерише у *Markdown* формату и на српском језику, док су интерне инструкције, критеријуми и правила евалуације дефинисани на енглеском језику због перформанси језичког модела. Извештај је организован у пет целина:

1. основни подаци о репозиторијуму;
2. сумарни преглед евалуације;
3. детаљна евалуација појединачних захтева;
4. приоритетне препоруке;
5. ограничења анализе.

У првом делу наводе се основни подаци који омогућавају идентификовање опсега спроведене анализе. Они укључују назив и *URL* адресу репозиторијума, његову видљивост, подразумевану грану, примарни програмски језик, број преузетих *commit*-ова, идентификоване контрибуторе и број фајлова чији је садржај анализиран (Слика 26).

```

project_evaluation.md > ** # Izveštaj o evaluaciji studentskog projekta
1 # Izveštaj o evaluaciji studentskog projekta
2
3 ## 1. Osnovni podaci o repozitorijumu
4
5 - **Naziv repozitorijuma:** internet-tehnologije-2025-githubactivitydashboard_2022_0175
6 - **URL repozitorijuma:** [https://github.com/elab-development/internet-tehnologije-2025-githubactivitydashboard_2022_0175](https://github.com/elab-development/internet-tehnologije-2025-githubactivitydashboard_2022_0175)
7 - **Vidljivost:** javni
8 - **Podrazumevana grana:** main
9 - **Primarni programski jezik:** JavaScript
10 - **Broj analiziranih commit-a:** 83
11 - **Identifikovani doprinosioci:** 2
12 - **Broj analiziranih fajlova:** 68
    
```

Слика 26. Основни подаци о репозиторијуму у генерисаном извештају

Следећу целину чини сумарна табела евалуације, у којој су обухваћени сви критеријуми дефинисани у фајлу *project_requirements.py*. За сваки захтев приказују се његова ознака, скраћени назив, додељени статус и сажет опис кључног фајла на основу којег је испуњеност критеријума процењена. Захваљујући табеларном приказу, студент може брзо да уочи захтеве којима треба посветити додатну пажњу, док наставник добија сажет преглед процењеног стања пројекта (Слика 27).

```
project_evaluation.md > *# # Izveštaj o evaluaciji studentskog projekta
1 # Izveštaj o evaluaciji studentskog projekta
14 ## 2. Sažetak evaluacije
15
16 | Oznaka zahteva | Skraćeni naziv | Status | Ključni dokaz |
17 |-----|-----|-----|-----|
18 | M1 | Docker | Ispunjen | backend/Dockerfile, frontend/Dockerfile |
19 | M2 | Docker Compose | Ispunjen | docker-compose.yml |
20 | M3 | API Spec | Nije ispunjen | Nema dokaza o Swagger ili OpenAPI specifikaciji |
21 | M4 | API Doc | Nije ispunjen | Nema dokaza o referenciranju API specifikacije u dokumentaciji |
22 | A1 | Eksterne API | Ispunjen | backend/services/github_service.py |
23 | A2 | README | Ispunjen | frontend/README.md |
24 | A3 | Grane | Ispunjen | develop, main, feature/activity-refactor, feature/auth-refactor, feature/schemas-cleanup,
feature/watchlist-cleanup |
25 | A4 | CI/CD | Nije ispunjen | Nema dokaza o CI/CD pipeline-u |
26 | A5 | Deployment | Nije ispunjen | Nema dokaza o produkcionalnoj verziji aplikacije na cloud platformi |
27 | A6 | Sigurnost | Nije ispunjen | Nema dokaza o zaštiti od tri uobičajene sigurnosne pretnje |
28 | A7 | Automatizovani testovi | Ispunjen | frontend/src/App.test.js |
29 | A8 | Vizualizacija | Nije ispunjen | Nema dokaza o vizualizaciji podataka ili interaktivnoj mapi |
30 | B1 | Git | Ispunjen | Repozitorijum koristi Git |
31 | B2 | Commits | Ispunjen | 83 commit-a |
32 | B3 | Doprinosioci | Ispunjen | Dva doprinosioca: una-s, anjaaasss |
33 | B4 | Javni repozitorijum | Ispunjen | Repozitorijum je javan |
34 | B5 | Elab organizacija | Nije ispunjen | Nema dokaza o povezanosti sa Elab GitHub organizacijom |
35 | B6 | Dokumentacija | Nije ispunjen | Nema dokaza o podršavajućoj dokumentaciji |
36 | F1 | Dinamički sajt | Ispunjen | frontend/src/App.js |
37 | F2 | Stranice | Ispunjen | frontend/src/views/Home.js, frontend/src/views/UserView.js, frontend/src/views/RepoView.
```

Слика 27. Сумарни преглед статуса анализираних захтева

После сумарног приказа следи детаљна евалуација сваког захтева. Захтеви су груписани у категорије које одговарају структури пројектног задатка: обавезни захтеви, захтеви за вишу оцену, основни захтеви, *frontend* захтеви и *backend* захтеви. За сваки критеријум извештај садржи ознаку и назив захтева, додељени статус, идентификоване доказе, образложење процене и препоруку за даље поступање (Слика 28).

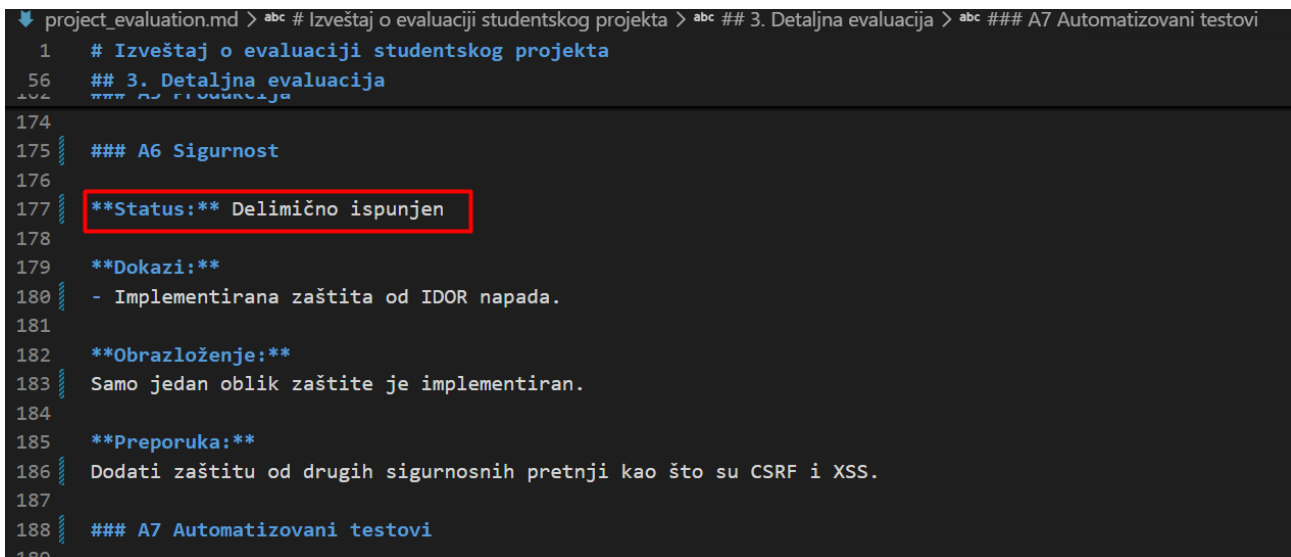
```
137 ### A3 Grane
138
139 **Status:** Ispunjen
140
141 **Dokazi:**
142 - develop, main, feature/activity-refactor, feature/auth-refactor, feature/schemas-cleanup, feature/watchlist-cleanup
143
144 **Образложење:**
145 Repozitorijum sadrži glavnu, develop i više od dve dodatne grane.
146
147 **Preporuka:**
148 Nema potrebnih korekcija.
149
150 ### A4 CI/CD
151
152 **Status:** Nije ispunjen
153
154 **Dokazi:**
155 Nema dokaza o CI/CD pipeline-u.
156
157 **Образложење:**
158 Nema informacija o automatizovanim testovima ili CI/CD procesu.
159
160 **Preporuka:**
161 Preporučuje se implementacija CI/CD pipeline-a koji uključuje testiranje i izgradnju Docker slika.
```

Слика 28. Структура детаљне евалуације једног пројектног захтева

Оваква структура омогућава да резултат не буде сведен само на бинарну оцену испуњености. Студент добија објашњење због чега је одређени статус додељен и које активности треба спровести да би се постојеће решење унапредило.

Уколико је реализован само један део сложеног захтева, агент идентификује постојећи елемент и истовремено указује на оно што недостаје. На пример, захтев може бити делимично испуњен када постоји *CI* конфигурација за покретање тестова, али није дефинисана изградња *Docker*

image-а, или када документација садржи опис апликације, али не и потпуно упутство за њено покретање. Овакав начин формулисања резултата погодан је за формативну примену, јер студенту пружа информацију о постојећем напретку и преосталим активностима. Пример делимично испуњеног захтева приказан је на Слици 29.



```
project_evaluation.md > abc # Izveštaj o evaluaciji studentskog projekta > abc ## 3. Detaljna evaluacija > abc ### A7 Automatizovani testovi
1 # Izveštaj o evaluaciji studentskog projekta
56 ## 3. Detaljna evaluacija
102 ### A7 Automatizovani testovi
174
175 ### A6 Sigurnost
176
177 **Status:** Delimično ispunjen
178
179 **Dokazi:**
180 - Implementirana zaštita od IDOR napada.
181
182 **Obrazloženje:**
183 Samo jedan oblik zaštite je implementiran.
184
185 **Preporuka:**
186 Dodati zaštitu od drugih sigurnosnih pretnji kao što su CSRF i XSS.
187
188 ### A7 Automatizovani testovi
189
```

Слика 29. Приказ делимично испуњеног захтева

Статус није могуће утврдити користи се када анализа репозиторијума није довољна за доношење поузданог закључка. Овај статус може бити додељен када је одређену функционалност могуће проверити само покретањем апликације, када продукцијска адреса није наведена, када је документација доступна у формату чији садржај није обухваћен анализом или када није могуће недвосмислено утврдити квалитативну карактеристику на основу расположивих података. Уместо неоснованог закључка, агент наводи који додатни доказ треба обезбедити.

Након појединачне анализе свих критеријума, агент формира листу приоритетних препорука (Слика 30). Препоруке се уређују према значају уочених недостатака, а не само према редоследу захтева у спецификацији. Предност имају:

- неиспуњени обавезни захтеви;
- неиспуњени основни захтеви;
- безбедносни недостаци и проблеми у аутентификацији и ауторизацији;
- недостаци у тестирању и *CI/CD* поступку;
- остала унапређења структуре, документације и квалитета решења.

Овај део извештаја студентима обезбеђује јасан редослед наредних активности, што је посебно значајно када је идентификован већи број недостатака.

```

project_evaluation.md > abc # Izveštaj o evaluaciji studentskog projekta > abc ## 5. Ograničenja analize
1 # Izveštaj o evaluaciji studentskog projekta
285 Nema dokaza o podršavajućoj dokumentaciji.
542 **Preporuka:**
543 Preporučuje se implementacija zaštite pristupa zaštićenim rutama.
544
545 ## 4. Prioritetne preporuke
546
547 1. Implementirati Swagger ili OpenAPI specifikaciju za API.
548 2. Uključiti API specifikaciju u dokumentaciju.
549 3. Implementirati CI/CD pipeline.
550 4. Postaviti aplikaciju na cloud platformu.
551 5. Implementirati zaštitu od uobičajenih sigurnosnih pretnji.
552 6. Povezati repozitorijum sa Elab GitHub organizacijom.
553 7. Dodati podržavajuću dokumentaciju.
554 8. Razmotriti dodavanje vizualizacije podataka ili interaktivne mape.
555 9. Razmotriti korišćenje Angular-a ili ukloniti taj zahtev.
556 10. Implementirati zaštitu pristupa zaštićenim rutama.

```

Слика 30. Приоритетне препоруке генерисане од стране AI агента

Последњу целину извештаја чине ограничења анализе (Слика 31). Агент наводи који аспекти нису могли бити проверени са довољном поузданошћу и због чега. Ограничења могу бити последица анализе ограниченог броја фајлова, скраћивања веома великих фајлова, немогућности покретања студентске апликације, одсуства продукцијске URL адресе, недоступности садржаја бинарних докумената или немогућности да се квалитет одређених елемената процени само на основу статичких података. Експлицитно навођење ограничења доприноси транспарентности резултата и указује на аспекте који захтевају ручни преглед наставника.

```

project_evaluation.md > abc # Izveštaj o evaluaciji studentskog projekta > abc ## 5. Ograničenja analize
1 # Izveštaj o evaluaciji studentskog projekta
545 ## 4. Prioritetne preporuke
556 10. Implementirati zaštitu pristupa zaštićenim rutama.
557
558 ## 5. Ograničenja analize
559
560 Nije moguće utvrditi zahteve koji se odnose na CI/CD pipeline, produkcionu verziju aplikacije na cloud platformi, zaštitu od sigurnosnih pretnji, kao i povezanost sa Elab GitHub organizacijom, jer nema dovoljno dokaza u dostupnim fajlovima.
561
562

```

Слика 31. Ограничења аутоматизоване анализе наведена у извештају

На основу генерисане повратне информације студенти могу допунити или изменити свој пројекат, након чега се анализа може поново покренути над ажурираном верзијом репозиторијума. Поновљена примена алата омогућава праћење промене статуса захтева и процену напретка између две итерације. Иако тренутна верзија не генерише аутоматски упоредни извештај, сачувани *Markdown* резултати могу се накнадно упоредити.

7 Примена и евалуација развијеног методолошког оквира у наставном процесу

У овом поглављу приказана је примена и евалуација развијеног методолошког оквира у наставном процесу, кроз две итерације истраживања спроведене у различитим фазама развоја софтвера.

У поглављу 7.1 је представљена примена оквира у фазама анализе корисничких захтева и управљања софтверским пројектом, са фокусом на дефинисање и структурирање захтева, планирање обима пројекта, одређивање приоритета и избор функционалности за реализацију.

У поглављу 7.2 је приказана примена оквира у фазама пројектовања, имплементације и тестирања софтвера, при чему су анализирани задаци моделовања система, генерисања и унапређивања програмског кода и провере квалитета развијеног решења.

За обе итерације истраживања представљени су контекст и процедура истраживања, учесници, инструменти, анализа и дискусија резултата, док су у завршном делу поглавља обједињено размотрене практичне импликације и ограничења спроведених истраживања.

7.1 Примена методолошког оквира у фазама анализе корисничких захтева и управљања софтверским пројектом

7.1.1 Контекст истраживања и хипотезе

Студенти који су учествовали у овом истраживању су студенти завршне године основних академских студија студијског програма Информациони системи и технологије на Факултету организационих наука Универзитета у Београду. Током претходних година студија стекли су знања из програмирања, база података, развоја информационих система, електронског пословања и других сродних области, што им је омогућило да активности анализе захтева и управљања софтверским пројектом разматрају у контексту развоја конкретних софтверских решења.

Истраживање је спроведено у оквиру предмета **Управљање ризиком у развоју апликација електронског пословања**. Наставне активности на предмету усмерене су на разумевање и примену поступака идентификовања, анализе и управљања ризицима који се јављају током развоја апликација електронског пословања. У оквиру пројектних активности студенти су дефинисали захтеве, организовали и специфицирали функционалности система, одређивали њихове приоритете и доносили одлуке релевантне за планирање обима софтверског пројекта.

Описани наставни контекст омогућио је примену развијеног методолошког оквира у две повезане целине: **фази анализе корисничких захтева и фази управљања софтверским пројектом**. У складу са областима знања дефинисаним у *SWEBOK*-у, анализа корисничких захтева повезана је са облашћу *Software Requirements*, док је управљање софтверским пројектом повезано са облашћу *Software Engineering Management*.

Фаза анализе корисничких захтева обухватала је активности идентификовања, дефинисања, анализе, структурирања и спецификације захтева за развој софтверског решења. Студенти су применом *GenAI* алата идентификовали функционалне захтеве, дефинисали нефункционалне захтеве и организовали захтеве у структуру погодну за даље планирање и развој система. Посебна пажња била је усмерена на јасноћу, потпуност и конзистентност дефинисаних захтева, као и на потребу за провером и мануелним прецизирањем садржаја генерисаног помоћу *GenAI* алата.

Фаза управљања софтверским пројектом у овом истраживању није обухватала целокупну област управљања софтверским инжењерством, већ активности релевантне за планирање обима софтверског пројекта. Оне су обухватале процену вредности предложених функционалности, одређивање њихових приоритета и избор функционалности предвиђених за реализацију. *GenAI* алати коришћени су као подршка студентима при разматрању алтернативних функционалности и доношењу одлука о томе које функционалности највише доприносе остваривању циљева пројекта.

Ове две фазе су садржински повезане, али представљају различите врсте активности. У фази анализе захтева фокус је на утврђивању и формулисању онога што систем треба да омогући, док је у фази управљања софтверским пројектом фокус на одлучивању о томе које ће функционалности бити обухваћене пројектом и којим редоследом ће бити реализоване. На тај начин, резултати анализе захтева представљају основу за планирање обима и приоритизацију функционалности. Детаљна процедура реализације наставних активности и прикупљања података приказана је у наредном поглављу.

Полазећи од интегрисаног *TAM-TTF* истраживачког модела представљеног у поглављу 6.3, за потребе евалуације примене развијеног методолошког оквира у фазама анализе корисничких захтева и управљања софтверским пројектом формулисано је 11 истраживачких хипотеза. Њима су операционализовани односи између конструката истраживачког модела и омогућено је испитивање усклађености *GenAI* алата са наставним задацима, фактора који утичу на задовољство студената и њихове намере да ове алате користе у будућим активностима у области развоја софтвера.

Истраживање приказано у овом поглављу доприноси испитивању полазних хипотеза X.1 и X.2 докторске дисертације. Испитивањем односа између карактеристика задатака (у фазама анализе корисничких захтева и управљања софтверским пројектом), карактеристика технологије (коришћених *GenAI* алата), индивидуалних карактеристика студената и усклађености задатка и технологије разматрају се услови ефективне и системски организоване интеграције *GenAI* алата у наставне активности, чиме се доприноси провери полазне хипотезе X.1. Анализом перципиране корисности и једноставности употребе алата, задовољства студената и њихове намере да прихвате и наставе да користе *GenAI* алате доприноси се испитивању полазне хипотезе X.2.

За фазу анализе корисничких захтева инструмент је обухватио све конструкте интегрисаног *TAM-TTF* модела. За фазу управљања софтверским пројектом нису биле примењене посебне ставке за перципирану лакоћу учења употребе алата и директно мерење усклађености задатка

и технологије. Због тога за ову фазу нису формулисане хипотезе које укључују наведене конструкте. Карактеристике задатка и карактеристике технологије у фази управљања софтверским пројектом анализиране су дескриптивно, као показатељи сложености задатка и погодности *GenAI* алата за подршку избору и приоритизацији функционалности.

Прва група истраживачких хипотеза заснована је на *TTF* моделу и усмерена је на испитивање усклађености карактеристика *GenAI* алата са задацима које студенти обављају у фази анализе корисничких захтева. Полази се од претпоставке да усклађеност задатка и технологије зависи од карактеристика наставног задатка, карактеристика *GenAI* алата и индивидуалних карактеристика студената:

X1a: Карактеристике задатака позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење анализе корисничких захтева.

X2a: Карактеристике технологије позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење анализе корисничких захтева.

X3a: Индивидуалне карактеристике студената позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење анализе корисничких захтева.

У фази управљања софтверским пројектом карактеристике задатка односиле су се на сложеност одређивања приоритета и процену вредности функционалности, док су карактеристике технологије обухватале релевантност информација и корисност препорука које *GenAI* алати пружају. Пошто директан конструкт усклађености задатка и технологије није био обухваћен инструментом за ову фазу, наведени показатељи нису укључени у тестирање структурних односа *TTF* модела, већ су анализирани дескриптивно.

Друга група хипотеза заснована је на *TAM* моделу и усмерена на идентификовање фактора који утичу на задовољство студената применом *GenAI* алата. Као непосредни фактори задовољства посматране су перципирана корисност и перципирана лакоћа коришћења. Перципирана лакоћа учења употребе алата испитивана је само у фази анализе корисничких захтева. Хипотезе X4a, X5a и X6a односе се на фазу анализе корисничких захтева, док се хипотезе X4b и X5b односе на фазу управљања софтверским пројектом:

X4a: Перципирана корисност позитивно утиче на задовољство студената *GenAI* алатима за учење анализе корисничких захтева.

X4b: Перципирана корисност позитивно утиче на задовољство студената *GenAI* алатима за учење управљања софтверским пројектом.

X5a: Перципирана лакоћа коришћења позитивно утиче на задовољство студената *GenAI* алатима за учење анализе корисничких захтева.

X5b: Перципирана лакоћа коришћења позитивно утиче на задовољство студената *GenAI* алатима за учење управљања софтверским пројектом.

X6a: Перципирана лакоћа учења употребе алата позитивно утиче на задовољство студената *GenAI* алатима за учење анализе корисничких захтева.

Трећа група хипотеза односи се на намеру студената да прихвате и убудуће користе *GenAI* алате. У фази анализе корисничких захтева испитују се утицаји усклађености задатка и

технологије и задовољства на намеру будућег коришћења (хипотезе X7a и X8a). У фази управљања софтверским пројектом испитује се утицај задовољства (хипотеза X8b):

X7a: Усклађеност задатка и технологије позитивно утиче на намеру студената да користе *GenAI* алате за учење анализе корисничких захтева.

X8a: Задовољство позитивно утиче на намеру студената да користе *GenAI* алате за учење анализе корисничких захтева.

X8b: Задовољство позитивно утиче на намеру студената да користе *GenAI* алате за учење управљања софтверским пројектом.

Евалуација примене методолошког оквира у овом делу истраживања била је усмерена на испитивање ставова студената о коришћењу *GenAI* алата, карактеристика наставних задатака и примењених *GenAI* алата, задовољства студената и њихове намере будућег коришћења. Добијени резултати тумаче се као показатељи прихватања технологије и њене перципиране погодности за подршку активностима анализе корисничких захтева и управљања софтверским пројектом.

7.1.2 Процедура истраживања

Примена предложеног методолошког оквира у овом истраживању обухватила је две фазе развоја софтвера: **анализа корисничких захтева и управљање софтверским пројектом**.

Истраживање је реализовано током наставе на предмету Управљање ризиком у развоју апликација електронског пословања, која је трајала 13 недеља. Настава је била организована блоковски, тако да је сваки тематски термин обухватао кратак теоријски увод у обрађивану тему, практични приказ примене одговарајућих метода и алата и радионицу током које су студенти решавали задатке уз подршку *GenAI* алата. Оваква организација омогућила је да студенти непосредно након усвајања теоријских основа примене стечена знања на конкретним задацима из области анализе корисничких захтева и управљања софтверским пројектом.

У реализацији наставних активности примењивани су *GenAI* алати опште намене, као што су *ChatGPT*, *Google Gemini* и *Claude*, као и специјализовани алати *ChatPRD*, *DesignCraft AI* и *REX AI*. Избор алата био је условљен врстом активности у фазама анализе корисничких захтева и управљања софтверским пројектом, при чему су студенти добијене предлоге критички проверавали и прилагођавали контексту конкретног пројекта (Табела 6).

Табела 6. Преглед *GenAI* алата у фазама анализе корисничких захтева и управљања софтверским пројектом

Алат	Активност
Анализа корисничких захтева	
<i>ChatGPT</i>; <i>Gemini</i>	Идентификовање корисничких потреба; генерисање и прецизирање функционалних и нефункционалних захтева; формулисање корисничких прича и критеријума прихватања
<i>Claude</i>	Анализа, груписање и структурирање захтева; унапређивање њихових формулација; идентификовање нејасних, непотпуних или противречних захтева
<i>DesignCraft AI</i>	Структурирање спецификације софтверских захтева и повезивање захтева са почетним предлозима структуре софтверског решења
<i>REX AI</i>	Провера квалитета појединачних захтева; уочавање недостатака и неусаглашености; предлагање унапређених формулација и критеријума прихватања

Управљање софтверским пројектом

ChatGPT; Gemini	Анализа пословне вредности функционалности; разматрање критеријума за избор; предлагање приоритета и подршка формирању <i>product backlog</i> -а
ChatPRD	Дефинисање проблема, циљних група корисника, њихових потреба и кључних функционалности; одређивање обухвата производа и структурирање почетног описа производа
Claude	Поређење функционалности према пословној вредности, сложености и значају за кориснике; образлагање предложених приоритета и разматрање алтернативних одлука

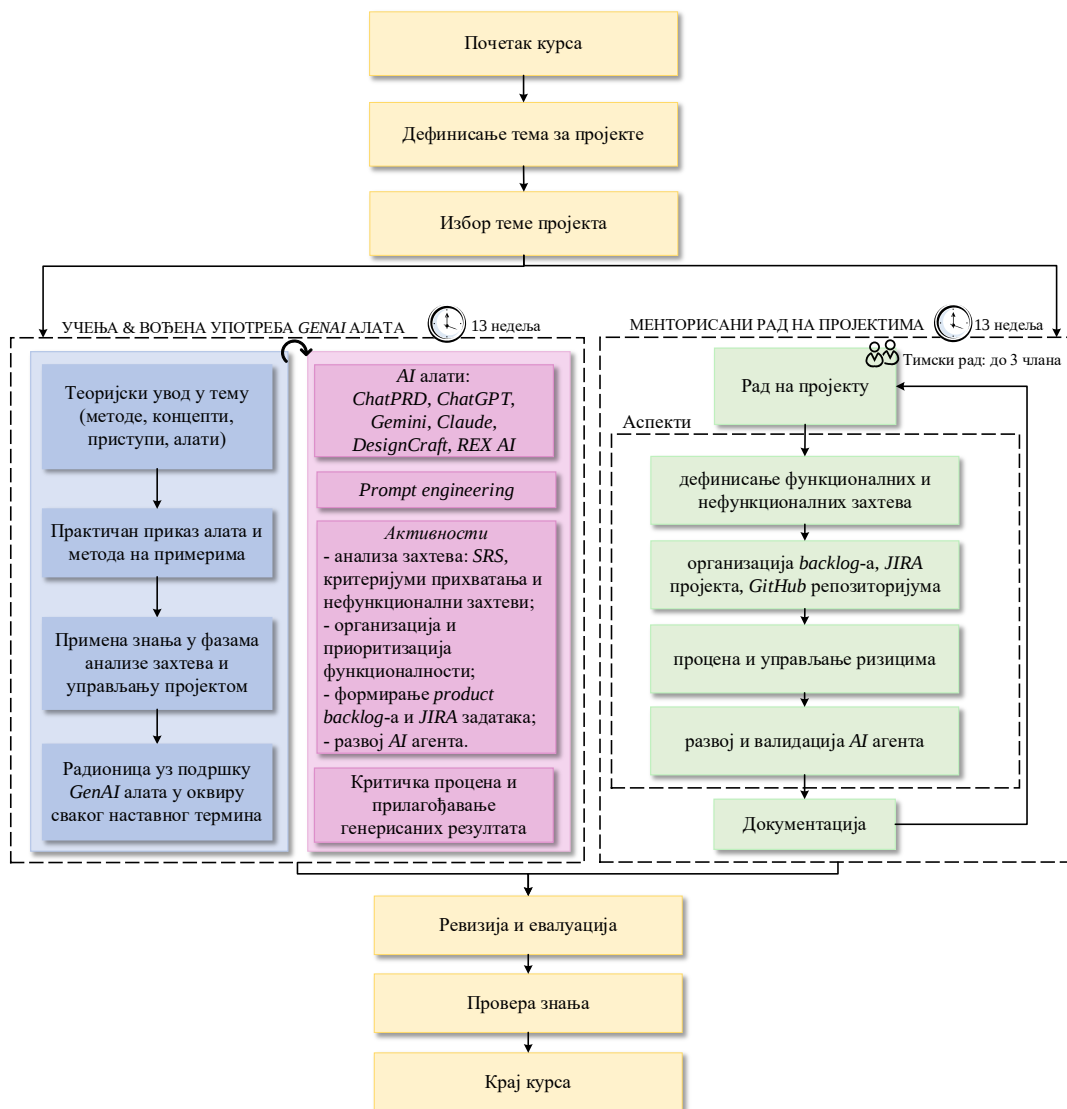
У почетном делу наставе студенти су упознати са основама управљања ИТ пројектима, агилним приступима и *Scrum* оквиром, као и са алатима који се користе за организацију и праћење пројектних активности, укључујући *JIRA*, *Notion* и *Microsoft Planner*. У практичном делу наставе разматрани су начини структурирања *backlog*-а, дефинисања епика, корисничких прича, задатака, одређивања приоритета и праћења статуса активности. Посебна пажња била је посвећена повезивању *JIRA* окружења са *GitHub* репозиторијумом и успостављању везе између планираних активности и развојног рада.

Активности повезане са **анализом корисничких захтева** реализоване су кроз рад на идентификовању корисничких потреба, дефинисању функционалних и нефункционалних захтева, формулисању корисничких прича и критеријума прихватања, као и изради спецификације софтверских захтева. Студенти су користили *SRS* (енг. *Software Requirements Specification*) стандард за структурирање пројектне документације и *GenAI* алате за генерисање почетних предлога захтева, њихово груписање и прецизирање. Добијене резултате требало је критички проверити и прилагодити конкретном софтверском решењу, уз образложење усвојених измена и допуна. Пројектни задаци су предвиђали и дефинисање нефункционалних захтева који се односе на безбедност, перформансе, поузданост, доступност, скалабилност, одрживост и употребљивост система. У оквиру активности **управљања софтверским пројектом** студенти су разматрали пословну вредност софтверског производа, потребе циљних корисника и критеријуме за избор функционалности које ће бити обухваћене пројектом. *GenAI* алати коришћени су као подршка приликом анализе предложених функционалности, процене њихове релативне вредности, одређивања приоритета и формирања *product backlog*-а. Након тога, студенти су у *JIRA* окружењу организовали функционалности кроз епике, корисничке приче, задатке, дефинисали приоритете, статусе и задужења и приказивали ток рада применом *Scrum* или *Kanban* табле. Пројектни захтеви обухватали су и повезивање *JIRA* задатака са *GitHub* активностима, као што су *commit*-ови и *pull request*-ови. Током наставе студенти су се упознали и са техникама формулисања упита за *GenAI* алате. У оквиру радионица анализирани су примери упита различитог нивоа сложености, начини прецизирања контекста, додељивања улоге моделу и итеративног унапређивања генерисаних резултата. Од студената се очекивало да документују примењене упите и добијене резултате, као и да критички процене њихову релевантност, потпуност и применљивост.

Поред тема које су се непосредно односиле на анализу захтева и управљање пројектом, наставом су обухваћене и теме: примена вештачке интелигенције у развоју и тестирању софтвера, управљање ризицима, континуитет пословања и развој *AI* агената. У оквиру теме посвећене развоју *AI* агената студенти су упознати са *LangChain* и *LangGraph* оквирима и

примером развоја агента са дефинисаним улазним подацима, вишекорачним током обраде, *LLM* компонентом и структурисаним излазом.

У завршном делу наставе студенти су били организовани у тимове до три члана и радили су на пројектним задацима уз менторску подршку наставника и сарадника. Пројекти су били груписани у три шире области: **планирање и праћење развоја софтверског производа уз примену *GenAI* алата, тестирање и управљање квалитетом софтвера уз примену *GenAI* алата и развој *AI* агента**. Независно од изабране области, студенти су били у обавези да дефинишу проблем и циљ пројекта, кориснике и њихове потребе, функционалне и нефункционалне захтеве, припреме одговарајућу пројектну документацију и организују рад у *JIRA* окружењу. Пројектни задаци су подразумевали примену *GenAI* алата, документовање коришћених упита и резултата и критичко разматрање прихваћених, измењених или одбачених *AI* предлога. Процедура истраживања, која обухвата блоковски организовану наставу, практичне радионице уз примену *GenAI* алата и менторски рад на тимским пројектима је приказана на Слици 32.



Слика 32. Примена методолошког оквира у пројектно-заснованом учењу у области развоја софтвера (фаза анализа корисничких захтева и управљање софтверским пројектом)

7.1.3 Учесници

У истраживању примене и прихватања *GenAI* алата у фазама анализе корисничких захтева и управљања софтверским пројектом учествовали су студенти завршне године основних академских студија студијског програма Информациони системи и технологије, који су слушали предмет Управљање ризиком у развоју апликација електронског пословања. Учесници су изабрани применом узорковања на основу доступности испитаника (енг. *convenience sampling method*) [281]. Имајући у виду њихово претходно образовање и искуство у развоју информационих система и апликација електронског пословања, студенти представљају релевантну групу за евалуацију примене *GenAI* алата у посматраним активностима.

Подаци су прикупљени у оквиру три итерације реализоване током наставног процеса. Укупно су прикупљена 74 одговора, од којих је у појединачним итерацијама прикупљено 29, 21 и 24 одговора. Будући да је анкетирање било анонимно и да није коришћен јединствени идентификатор испитаника, није било могуће утврдити да ли су поједини студенти учествовали у више итерација. Због тога укупан број од 74 представља број прикупљених одговора, а не број јединствених учесника.

Ради обезбеђивања доследности поступка, у свакој итерацији упитник је примењиван под истим условима и након реализације одговарајућих наставних активности. Сви испитаници су користили исту петостепену Ликертову скалу, док су формулације појединих ставки биле прилагођене активностима анализе корисничких захтева и управљања софтверским пројектом. Основна демографска структура одговора је приказана у Табели 7.

Табела 7. Демографска структура одговора
(фаза анализа корисничких захтева и управљање софтверским пројектом)

Пол	Број одговара
Мушки	16
Женски	58
Просечна оцена [6-10]	Број одговара
7 - 8	42
8 - 9	31
> 9	1

Истраживање је спроведено у складу са институционалним етичким стандардима и уз одобрење надлежне етичке комисије катедре. Учешће у истраживању било је у потпуности добровољно, а студенти су пре прикупљања података били упознати са циљем истраживања. Сви одговори прикупљени су анонимно, без прикупљања личних или осетљивих података. Студентима је наглашено да њихова одлука да учествују или не учествују у истраживању ни на који начин неће утицати на њихово оцењивање.

7.1.4 Инструменти

За прикупљање података и евалуацију примене развијеног методолошког оквира коришћен је упитник, развијен ради испитивања ставова, задовољства и утисака студената у вези са применом *GenAI* алата у фазама анализе корисничких захтева и управљања софтверским

пројектом. Инструмент је заснован на интегрисаном *TTF-TAM* истраживачком моделу представљеном у поглављу 6.3, при чему су формулације ставки прилагођене специфичним задацима које су студенти обављали у оквиру сваке од анализираних фаза.

Део упитника који се односио на анализу корисничких захтева обухватао је девет група тврдњи: перципирану корисност, перципирану лакоћу коришћења, перципирану лакоћу учења, задовољство, намеру будућег коришћења *GenAI* алата, карактеристике задатка, карактеристике технологије (*GenAI* алата), индивидуалне карактеристике студената и усклађеност задатка и технологије. Ставке су се односиле на примену *GenAI* алата приликом идентификовања функционалних захтева, дефинисања нефункционалних захтева, структурирања и организовања захтева, као и процене јасноће, потпуности и конзистентности генерисаних резултата.

Део упитника који се односио на управљање софтверским пројектом обухватао је шест група тврдњи: перципирану корисност, перципирану лакоћу коришћења, задовољство, намеру будућег коришћења *GenAI* алата, карактеристике задатка и карактеристике технологије (*GenAI* алата). Ставке су биле усмерене на примену *GenAI* алата приликом процене вредности функционалности, одређивања њихових приоритета и доношења одлука о избору функционалности за реализацију. У овој фази нису примењене посебне ставке за мерење перципиране лакоће учења и директне усклађености задатка и технологије. Индивидуалне карактеристике испитаника мерене су општим ставкама које су се односиле на способност коришћења *GenAI* алата за академске сврхе, перципирану академску продуктивност и мотивацију, па су разматране као заједничке карактеристике студената, а не као показатељи специфични за појединачну фазу.

Све затворене тврдње оцењиване су на петостепеној Ликертовој скали, од потпуног неслагања до потпуног слагања. Поред затворених ставки, упитник је садржао и отворена питања намењена прикупљању додатних коментара о функционалностима *GenAI* алата, лакоћи њиховог коришћења, прецизности генерисаних резултата, интеграцији са другим алатима, уоченим предностима и изазовима, као и начину реализације наставних активности.

Структура и садржај упитника прилагођени су на основу претходно валидираних инструмената заснованих на *TAM* и *TTF* моделима [282], [283], чиме је обезбеђена теоријска повезаност примењених конструката са истраживачким моделом. Комплетна структура упитника, са приказом свих ставки разврстаних према фазама и конструктима, дата је у прилогу дисертације (Прилог 1 и Прилог 2).

7.1.5 Анализа резултата

За анализу резултата и испитивање сложених односа између конструката истраживачког модела примењено је моделирање структурним једначинама (енг. *Structural Equation Modeling* - *SEM*), како би се проверила статистичка значајност постављених истраживачких хипотеза [284]. Конкретно, коришћен је *SEM* приступ заснован на коваријанси, који омогућава истовремено испитивање односа између више латентних конструката и њихових мерних показатеља [285]. Анализа је спроведена применом пакета *semopy* за програмски језик *Python*,

намењеног моделирању структурним једначинама [286]. Поред *SEM* анализе, примењене су и дескриптивна статистика и корелациона анализа, ради детаљнијег сагледавања карактеристика прикупљених података и повезаности између посматраних променљивих. Примењени аналитички поступци омогућили су добијање садржајно смислених и интерпретабилних резултата. Ипак, приликом њиховог тумачења потребно је узети у обзир специфичности узорка. Подаци су прикупљени у три итерације, при чему је укупно добијено 74 одговора. Будући да је анкетирање било анонимно и да није коришћен јединствени идентификатор испитаника, није могуће утврдити да ли су поједини студенти учествовали у више итерација. Због тога се резултати односе на структуру прикупљених одговора, а не нужно на исти број јединствених и међусобно независних испитаника. Наведено ограничење треба имати у виду приликом тумачења статистичке значајности утврђених односа и могућности генерализације резултата. У наставку је приказана анализа резултата.

7.1.5.1 Анализа усклађености *GenAI* алата са задацима и захтевима учења студената у развоју софтвера (фаза анализе корисничких захтева и управљање софтверским пројектом)

У оквиру ове секције приказано је испитивање усклађености *GenAI* алата са задацима које студенти обављају и њиховим потребама у процесу учења. Резултати који се односе на примену ових алата у фазама анализе корисничких захтева и управљања софтверским пројектом приказани су у Табели 8.

Табела 8. Усклађеност *GenAI* алата са задацима и захтевима студената у процесу учења развоја софтвера (фаза анализе корисничких захтева и управљања софтверским пројектом)

Питања	Анализа захтева		Управљање пројектом	
	СВ	СД	СВ	СД
Карактеристике задатка (енг. <i>task characteristics</i>)				
<i>GenAI</i> алати ми помажу у идентификовању функционалних захтева.	4,23	0,56	-	-
<i>GenAI</i> алати ми помажу у дефинисању нефункционалних захтева.	4,24	0,59	-	-
<i>GenAI</i> алати ми помажу у структурирању и организовању захтева.	4,23	0,59	-	-
Доношење одлука о приоритетима функционалности представља сложен задатак.	-	-	4,19	0,72
Процена вредности функционалности захтева додатну подршку.	-	-	4,07	0,65
Карактеристике технологије (енг. <i>technology characteristics</i>)				
Време генерисања одговара на упит код <i>GenAI</i> алата је одговарајуће.	4,32	0,80	-	-
<i>GenAI</i> алати пружају релевантне информације за доношење одлука о функционалностима.	-	-	4,08	0,68
<i>GenAI</i> алати генеришу корисне препоруке за избор и приоритизацију функционалности.	-	-	4,09	0,69
Индивидуалне карактеристике (енг. <i>individual characteristics</i>)				
Сигуран/на сам у своју способност да ефикасно користим <i>GenAI</i> алате за академске сврхе.	4,23	0,79	-	-
Верујем да ће коришћење <i>GenAI</i> алата побољшати моју академску продуктивност и успех.	4,30	0,70	-	-
Када користим <i>GenAI</i> алате, осећам се мотивисаније да завршим свој академски рад.	4,11	0,82	-	-

Резултати приказани у Табели 8 показују да су студенти *GenAI* алате оценили као корисне у свим посматраним активностима у оквиру фазе анализе корисничких захтева. Средње вредности за идентификовање функционалних захтева, дефинисање нефункционалних захтева и структурирање и организовање захтева биле су готово идентичне и кретале су се од 4,23 до 4,24. Овакви резултати указују на то да студенти препознају могућност примене *GenAI* алата у различитим активностима анализе захтева, без изражених разлика у процени њихове корисности.

У фази управљања софтверским пројектом студенти су доношење одлука о приоритетима функционалности оценили као сложен задатак, са средњом вредношћу 4,19, док су потребу за додатном подршком при процени вредности функционалности оценили средњом вредношћу 4,07. Истовремено, позитивно су оценили релевантност информација и корисност препорука које *GenAI* алати пружају за избор и приоритизацију функционалности, са средњим вредностима 4,08 и 4,09. Ови резултати указују на подударност између потребе студената за подршком у сложеним активностима управљања пројектом и могућности *GenAI* алата да такву подршку пруже.

Подаци из Табеле 8 такође указују на високе средње вредности индивидуалних карактеристика студената. Студенти су исказали висок ниво сигурности у сопствену способност коришћења *GenAI* алата у академске сврхе, са средњом вредношћу 4,23, и сматрали су да њихова примена може да побољша академску продуктивност и успешност, са средњом вредношћу 4,30. Нешто нижа, али и даље висока вредност забележена је за мотивацију за завршавање академских задатака уз примену *GenAI* алата и износила је 4,11.

Тестирање хипотеза применом *SEM* анализе показало је да индивидуалне карактеристике студената имају статистички значајан позитиван утицај на усклађеност задатка и технологије у фази анализе корисничких захтева. Израчуната p вредност износи 0,000619, чиме је потврђена хипотеза $H3a$.

Карактеристике задатка имале су позитиван, али статистички незнатан утицај на усклађеност задатка и технологије, са p вредношћу 0,233968. Због тога хипотеза $H1a$ није потврђена. Карактеристике *GenAI* алата такође нису показале статистички значајан утицај на усклађеност задатка и алата, при чему је p вредност 0,530172, па хипотеза $H2a$ није потврђена.

Добијени резултати указују на то да перципирана усклађеност *GenAI* алата са задацима анализе корисничких захтева у највећој мери зависи од сигурности студената у сопствену способност коришћења алата, њихове перцепције академске продуктивности и мотивације. Насупрот томе, саме карактеристике задатка и време генерисања одговора алата се нису показали као значајни предиктори усклађености задатка и технологије.

7.1.5.2 Анализа фактора који утичу на задовољство студената *GenAI* алатима за учење развоја софтвера (фазе анализе корисничких захтева и управљање софтверским пројектом)

У оквиру ове секције приказана је анализа фактора који утичу на задовољство студената *GenAI* алатима у контексту учења развоја софтвера. Фактори релевантни за фазу анализе корисничких захтева и фазу управљања софтверским пројектом приказани су у Табели 9.

Табела 9. Фактори који утичу на задовољство студената *GenAI* алатима у процесу учења развоја софтвера (фазе анализе корисничких захтева и управљања софтверским пројектом)

Питања	Анализа захтева		Управљање пројектом	
	СВ	СД	СВ	СД
Перципирана корисност (енг. <i>Perceived Usefulness</i>)				
<i>GenAI</i> алати су значајно допринели повећању продуктивности током дефинисања захтева.	4,11	0,80	-	-
<i>GenAI</i> алати су убрзали процес идентификовања и дефинисања захтева.	4,31	0,66	-	-
<i>GenAI</i> алати су допринели квалитету дефинисаних захтева (јасноћа, потпуност, конзистентност).	4,03	0,89	-	-
Резултати које су генерисали <i>GenAI</i> алати су захтевали мануелне поправке и дораде (нпр. прецизирање захтева).	4,08	0,75	-	-
<i>GenAI</i> алати су ми помогли да ефикасније одредим приоритетне функционалности.	-	-	4,20	0,74
<i>GenAI</i> алати су ми помогли да идентификујем функционалности које имају највећу вредност за реализацију софтверског пројекта.	-	-	4,01	0,69
<i>GenAI</i> алати су допринели квалитету доношења одлука о избору функционалности.	-	-	4,23	0,59
Перципирана лакоћа коришћења (енг. <i>Perceived Ease of Use</i>)				
Била су ми потребна додатна упутства/смернице од наставника како бих користио/ла <i>GenAI</i> алате у контексту анализе захтева.	3,43	1,30	-	-
Било ми је лако да разумем које су поправке/дораде биле потребне код резултата које су генерисали <i>GenAI</i> алати.	4,00	0,79	4,05	0,72
<i>GenAI</i> алати су интуитивни за коришћење.	4,07	0,71	4,12	0,78
Перципирана лакоћа учења (енг. <i>Perceived Ease of Learning</i>)				
Било ми је лако да постављам <i>prompt</i> -ове који дају квалитетан излаз у контексту анализе захтева.	4,16	0,70	-	-

Студенти су позитивно оценили корисност *GenAI* алата у обе посматране фазе. У фази анализе корисничких захтева највиша средња вредност забележена је за убрзавање процеса идентификовања и дефинисања захтева и износила је 4,31. Студенти су такође сматрали да *GenAI* алати доприносе повећању продуктивности и квалитету дефинисаних захтева, са средњим вредностима 4,11 и 4,03. Ипак, висока средња вредност од 4,08 показује да су генерисани резултати често захтевали мануелне поправке и додатна прецизирања. Овај резултат указује на то да студенти *GenAI* алате препознају као средство за убрзавање и подршку процесу анализе захтева, али не и као замену за критичку проверу и самостално доношење коначних одлука.

У фази управљања софтверским пројектом студенти су највише оценили допринос *GenAI* алата квалитету одлука о избору функционалности, са средњом вредношћу 4,23. Високо је оцењена

и њихова примена за ефикасније одређивање приоритетних функционалности, са средњом вредношћу 4,20, док је нешто нижа, али и даље позитивна вредност добијена за идентификовање функционалности које имају највећу вредност за реализацију пројекта и износила је 4,01. Резултати указују на то да су студенти *GenAI* алате препознали као корисну подршку за разматрање алтернатива и унапређење квалитета одлука у процесу планирања обима пројекта.

Студенти су *GenAI* алате оценили као једноставне и интуитивне за коришћење у обе фазе. Средње вредности за једноставност употребе током обе фазе износиле су 4,07 и 4,12. Такође су навели да су лако разумели потребне дораде генерисаних захтева и препоруке у вези са избором и приоритизацијом функционалности, са средњим вредностима 4,00 и 4,05.

Ипак, резултати показују да је студентима у одређеној мери била потребна додатна подршка наставника у фази анализе захтева. Средња вредност за ову тврдњу износила је 3,43, уз релативно високу стандардну девијацију од 1,30, што указује на изражене разлике међу студентима. Док су поједини студенти могли самостално да користе *GenAI* алате, другима су била потребна додатна упутства и смернице. Насупрот томе, студенти су у целини исказали висок ниво сигурности у сопствену способност формулисања упита који доводе до квалитетних резултата у контексту анализе захтева, са средњом вредношћу 4,16.

Тестирање хипотеза применом *SEM* анализе показало је да су се фактори који утичу на задовољство студената делимично разликовали у зависности од посматране фазе. У фази анализе корисничких захтева, перципирана корисност *GenAI* алата имала је статистички значајан позитиван утицај на задовољство студената, са p вредношћу 0,020016, чиме је потврђена хипотеза H_{4a} . Перципирана лакоћа учења употребе алата такође је имала статистички значајан позитиван утицај на задовољство, са p вредношћу 0,003257, чиме је потврђена хипотеза H_{6a} . Насупрот томе, перципирана лакоћа коришћења није показала статистички значајан утицај на задовољство студената, па хипотеза H_{5a} није потврђена.

У фази управљања софтверским пројектом није утврђен статистички значајан утицај перципиране корисности нити перципиране лакоће коришћења на задовољство студената. P вредност за утицај перципиране корисности износила је 0,664748, те хипотеза H_{4b} није потврђена. Такође, p вредност за утицај перципиране лакоће коришћења износила је 0,332431, због чега ни хипотеза H_{5b} није потврђена.

Добијени резултати указују на то да је у фази анализе корисничких захтева задовољство студената преваходно повезано са перципираним доприносом *GenAI* алата у извршавању задатака и њиховом способношћу да науче како да формулишу одговарајуће упите. Насупрот томе, у фази управљања софтверским пројектом није потврђено да перципирана корисност и перципирана лакоћа коришћења остварују засебан статистички значајан утицај на задовољство студената приликом избора и приоритизације функционалности.

7.1.5.3 Анализа спремности студената за усвајање GenAI алата за учење развоја софтвера (фаза анализе корисничких захтева и управљање софтверским пројектом)

У оквиру ове секције испитани су фактори који одређују спремност студената да усвоје GenAI алате за учење развоја софтвера. Резултати за фазу анализе корисничких захтева и фазу управљања софтверским пројектом приказани су у Табели 10.

Табела 10. Спремност студената за усвајање GenAI алата за учење развоја софтвера (фаза анализе корисничких захтева и управљања софтверским пројектом)

Питања	Анализа захтева		Управљање пројектом	
	СВ	СД	СВ	СД
Задовољство (енг. <i>satisfaction</i>)				
Одговори и препоруке које GenAI алати дају испуњавају моја очекивања у фази анализе захтева.	4,12	0,74	-	-
Препоручио/ла бих коришћење GenAI алата колегама током анализе захтева.	4,30	0,54	-	-
Повратна информација од наставника је била задовољавајућа за задатке које сам урадио/ла у фази анализе захтева коришћењем GenAI алата.	4,27	0,60	-	-
Задовољан/на сам резултатима које GenAI алати.	4,12	0,62	4,00	0,68
Усклађеност задатка и технологије (енг. <i>task-technology fit</i>)				
Довољно ми је да користим GenAI алате за дефинисање захтева без употребе додатних алата.	3,73	1,10	-	-
GenAI алати се добро уклапају у процес дефинисања и анализе захтева.	4,15	0,59	-	-
GenAI алати се лако интегришу са другим алатима које користим у фази анализе захтева.	4,22	0,65	-	-
Намера усвајања (енг. <i>Intention to adopt</i>)				
Намеравам да наставим да користим GenAI алате у будућности.	4,32	0,58	4,27	0,63

Резултати приказани у Табели 10 указују на висок ниво задовољства студената и њихову изражену спремност за даље коришћење GenAI алата у обе посматране фазе. У фази анализе корисничких захтева студенти су оценили да одговори и препоруке које GenAI алати пружају испуњавају њихова очекивања и исказали су задовољство добијеним резултатима, при чему је средња вредност износила 4,12. Посебно високе вредности забележене су за препоруку коришћења GenAI алата другим студентима и задовољство повратним информацијама наставника, са средњим вредностима 4,30 и 4,27.

Резултати који се односе на усклађеност задатка и технологије показују да су студенти сматрали да се GenAI алати добро уклапају у процес дефинисања и анализе захтева, са средњом вредношћу 4,15, као и да се лако интегришу са другим алатима који се користе у овој фази, са средњом вредношћу 4,22. Насупрот томе, тврдња да је за дефинисање захтева довољно користити GenAI алате без примене додатних алата оцењена је нижом средњом вредношћу од 3,73. Истовремено, код ове тврдње забележена је највећа стандардна девијација од 1,10, што указује на израженије разлике у ставовима студената. Добијени резултат показује да студенти препознају добру уклопљеност GenAI алата у процес рада, али их углавном не посматрају као самосталну замену за друге алате и изворе подршке у анализи захтева.

Намера будућег коришћења *GenAI* алата високо је оцењена у обе фазе. Средња вредност намере студената да наставе да користе *GenAI* алате у анализи корисничких захтева износила је 4,32, док је за њихово будуће коришћење приликом доношења одлука о функционалностима у фази управљања софтверским пројектом износила 4,27. Иако је задовољство препорукама *GenAI* алата за избор и приоритизацију функционалности оцењено нешто нижом средњом вредношћу од 4,00, резултати у целини указују на позитиван став студената према усвајању и наставку примене *GenAI* алата у активностима анализе захтева и управљања софтверским пројектом.

Тестирање хипотеза применом *SEM* анализе показало је да у фази анализе корисничких захтева није утврђен статистички значајан утицај ни усклађености задатка и технологије ни задовољства студената на намеру будућег коришћења *GenAI* алата. *P* вредност за утицај усклађености задатка и технологије износила је 0,290339, те хипотеза *X7a* није потврђена. Такође, *p* вредност за утицај задовољства студената износила је 0,964947, због чега ни хипотеза *X8a* није потврђена.

У фази управљања софтверским пројектом, задовољство студената препорукама које *GenAI* алати пружају за избор и приоритизацију функционалности имало је статистички значајан позитиван утицај на намеру њиховог будућег коришћења. *P* вредност износила је 0,0000003, чиме је потврђена хипотеза *X8b*.

Добијени резултати указују на то да у фази анализе корисничких захтева усклађеност алата са задацима и задовољство студената нису остварили статистички значајан утицај на намеру будућег коришћења *GenAI* алата. Насупрот томе, у фази управљања софтверским пројектом задовољство студената потврђено је као статистички значајан фактор намере будућег коришћења, што указује на то да се механизми прихватања *GenAI* алата могу разликовати у зависности од фазе развоја софтвера и природе активности које студенти извршавају.

7.1.6 Дискусија

Резултати прве итерације истраживања показују да *GenAI* алати могу да пруже значајну подршку студентима у активностима које претходе пројектовању и имплементацији софтверског решења, односно у фазама анализе корисничких захтева и управљања софтверским пројектом. Студенти су алате користили за идентификовање, дефинисање и структурирање функционалних и нефункционалних захтева, као и за избор и приоритизацију функционалности у складу са њиховом очекиваном вредношћу. На тај начин, примена *GenAI* алата није била ограничена на генерисање програмског кода, већ је обухватила и активности које захтевају анализу проблема, разматрање потреба корисника и доношење одлука у раним фазама развоја софтвера.

Дескриптивни резултати показују да су студенти позитивно оценили корисност *GenAI* алата у обе посматране фазе. У фази анализе корисничких захтева посебно су истакнути допринос алата бржем идентификовању и дефинисању захтева, повећању продуктивности и унапређењу квалитета формулисаних захтева. Истовремено, резултати показују да генерисани садржај није могао да се примени без провере и прилагођавања, будући да су студенти указали на потребу

за мануелним дорадама. Овај резултат потврђује да *GenAI* алате у овој фази треба посматрати као подршку анализи и структурирању информација, а не као замену за самостално расуђивање и критичку процену захтева.

У фази управљања софтверским пројектом студенти су позитивно оценили подршку *GenAI* алата приликом одређивања приоритетних функционалности, идентификовања функционалности највеће вредности и доношења квалитетнијих одлука о њиховом избору. Ови резултати указују на потенцијал алата да студентима олакшају разматрање већег броја критеријума и алтернатива приликом планирања пројекта. Ипак, висок ниво перципиране корисности није нужно подразумевао и статистички значајан непосредан утицај на задовољство, што показује да позитивна процена појединачних функционалности алата не мора аутоматски довести до позитивне оцене целокупног искуства њихове употребе.

Резултати такође указују на значај индивидуалних карактеристика студената. Студенти су исказали висок ниво самопоуздања у погледу коришћења *GenAI* алата у академском окружењу, позитивно су оценили њихов допринос продуктивности и показали мотивацију за примену у наставним активностима. Потврђен утицај индивидуалних карактеристика на усклађеност задатка и технологије показује да процена прикладности *GenAI* алата не зависи искључиво од објективних карактеристика задатка или брзине генерисања резултата. Студенти који су сигурнији у сопствене способности, мотивисанији за употребу алата и уверенији у њихов допринос академском раду вероватније препознају *GenAI* као технологију која одговара активностима анализе захтева.

Анализа задовољства у фази анализе корисничких захтева показала је да су перципирана корисност и перципирана лакоћа учења употребе алата имале статистички значајан позитиван утицај на задовољство студената. Овај резултат указује на то да је студентима било важно да алат пружа конкретан допринос извршавању задатка, али и да релативно брзо могу да науче како да формулишу одговарајуће упите. У фази анализе корисничких захтева потребно је прецизно описивање проблема, идентификовање актера и њихових потреба, као и разликовање функционалних и нефункционалних захтева. Због тога се задовољство повећавало када су студенти били у стању да кроз итеративно формулисање упита добију употребљиве предлоге и прилагоде их контексту конкретног пројекта. Насупрот томе, перципирана лакоћа коришћења није показала статистички значајан утицај на задовољство у фази анализе захтева. Овај резултат показује да једноставност комуникације са *GenAI* алатом сама по себи није била довољна за „формирање“ задовољства. Иако су студенти могли релативно лако да унесу упит и добију одговор, квалитет резултата зависио је од прецизности контекста, описа пословног проблема и накнадне провере генерисаних захтева. Студенти су више вредновали применљивост и квалитет одговора него техничку једноставност интеракције са алатом.

У фази управљања софтверским пројектом нису потврђени претпостављени утицаји перципиране корисности и перципиране лакоће коришћења на задовољство. Иако су студенти позитивно оценили подршку алата приликом избора и приоритизације функционалности, ове перцепције нису оствариле засебне статистички значајне ефекте у структурном моделу. Једно од могућих објашњења односи се на међусобну повезаност корисности и лакоће коришћења, али и на сложеност одлука које су студенти доносили. Приоритизација функционалности не

зависи само од квалитета препоруке коју генерише алат, већ и од разумевања пословне вредности, расположивих ресурса, зависности између функционалности и ограничења пројекта. Због тога задовољство може бити условљено ширим скупом фактора који нису у потпуности обухваћени посматраним конструктима.

Посебно је значајан резултат према којем је задовољство имало статистички значајан позитиван утицај на намеру будућег коришћења *GenAI* алата у фази управљања софтверским пројектом. Овај резултат показује да је позитивно искуство употребе алата приликом избора и приоритизације функционалности представљало важан предуслов њиховог даљег прихватања. Када су студенти били задовољни добијеним препорукама и подршком приликом доношења одлука, били су спремнији да *GenAI* алате користе и у будућим активностима управљања софтверским пројектима. У фази анализе корисничких захтева, међутим, нису потврђени непосредни утицаји усклађености задатка и технологије и задовољства на намеру будућег коришћења. Иако су оба конструкта показала позитивну повезаност са намером на нивоу дескриптивних резултата, њихови засебни ефекти у структурном моделу нису били статистички значајни. Овакав резултат може се објаснити њиховом високом међусобном повезаношћу, због које је било отежано раздвајање јединственог доприноса сваког конструкта. Такође је могуће да намера будуће употребе зависи и од фактора као што су претходно искуство са алатима, захтеви конкретног задатка, расположивост алата, поверење у генерисани садржај и потреба за додатном провером резултата.

Резултати *SEM* анализе показују да се односи између конструката разликују у зависности од врсте активности. Од укупно једанаест тестираних хипотеза потврђене су четири: утицај индивидуалних карактеристика студената на усклађеност задатка и *GenAI* алата, утицај перципиране корисности и лакоће учења употребе алата на задовољство у анализи корисничких захтева, као и утицај задовољства на намеру будућег коришћења у управљању софтверским пројектом. Преосталих седам хипотеза није добило статистичку подршку. Ради јаснијег тумачења, могући разлози њиховог непотврђивања разматрају се у наставку према групама односа обухваћених истраживачким моделом.

1. Утицај карактеристика задатка, технологије и студената на усклађеност задатка и технологије

X1a - Утицај карактеристика задатка на усклађеност задатка и технологије у фази анализе корисничких захтева

Хипотеза X1a није потврђена, јер карактеристике задатка нису показале статистички значајан утицај на усклађеност задатка и технологије. Иако су идентификовање функционалних и нефункционалних захтева и њихово структурирање оцењени као активности у којима је примена *GenAI* алата корисна, задаци су били релативно уједначени за све студенте. Учесници су радили у оквиру истог наставног поступка, примењивали упоредиве технике анализе и решавали сличне типове задатака. Ограничена варијабилност карактеристика задатака могла је да умањи њихову способност да објасне разлике у процени усклађености са *GenAI* алатима.

X2a - Утицај карактеристика технологије на усклађеност задатка и технологије у фази анализе корисничких захтева

Хипотеза X2a није потврђена, што показује да брзина генерисања одговора, као посматрана карактеристика технологије, није била довољна да објасни процену усклађености *GenAI*

алата са задацима анализе захтева. Брз одговор може да олакша итеративни рад, али не гарантује да ће генерисани захтеви бити тачни, потпуни, недвосмислени и усклађени са потребама корисника. Студенти су морали да проверавају резултате, допуњују контекст и коригују предлоге, због чега је квалитет садржаја вероватно био значајнији од саме брзине његовог генерисања.

2. Утицај перцепција *GenAI* алата на задовољство студената

X5a - Утицај перципиране лакоће коришћења на задовољство у фази анализе корисничких захтева

Хипотеза X5a није потврђена. Иако су студенти *GenAI* алате углавном сматрали интуитивним и могли су лако да разумеју потребне измене генерисаних резултата, једноставност употребе није остварила статистички значајан утицај на задовољство. Могуће је да су студенти више вредновали способност алата да произведе садржај који се може применити у спецификацији захтева него саму лакоћу интеракције. Уколико је генерисани резултат захтевао додатно прецизирање, проверу и мануелну дораду, једноставан начин постављања упита није био довољан за формирање позитивног укупног искуства.

X4b - Утицај перципиране корисности на задовољство у фази управљања софтверским пројектом

Хипотеза X4b није потврђена, иако су студенти позитивно оценили корисност *GenAI* алата за одређивање приоритета и идентификовање функционалности највеће вредности. Овај резултат указује на то да перципирана корисност није била довољна за формирање задовољства. Одлуке о приоритетима захтевају разматрање више критеријума и компромиса, па задовољство може зависити и од транспарентности препорука, могућности њиховог образлагања и усклађености са циљевима конкретног пројекта.

X5b - Утицај перципиране лакоће коришћења на задовољство у фази управљања софтверским пројектом

Хипотеза X5b такође није потврђена. Иако су студенти позитивно оценили једноставност употребе алата и разумљивост добијених препорука, лакоћа коришћења није имала засебан статистички значајан утицај на задовољство. Могуће је да су студенти приликом управљања пројектом више вредновали оправданост и применљивост предложених приоритета него саму једноставност добијања одговора. Поред тога, перципирана корисност и лакоћа коришћења биле су међусобно повезане, што је могло да отежа издвајање њихових појединачних ефеката у структурном моделу.

3. Утицај усклађености задатка и технологије и задовољства на намеру усвајања *GenAI* алата

X7a - Утицај усклађености задатка и технологије на намеру будућег коришћења у фази анализе корисничких захтева

Хипотеза X7a није потврђена. Иако су студенти позитивно проценили уклапање *GenAI* алата у процес анализе захтева, ова процена није остварила засебан статистички значајан утицај на намеру будућег коришћења. Једно од могућих објашњења јесте да функционално уклапање алата у задатак није довољно уколико студенти нису сигурни у исправност, потпуност и доследност генерисаних захтева. Будућа употреба може зависити и од поверења у резултате, расположивости алата и могућности њихове интеграције у постојеће поступке документовања захтева.

X8a - Утицај задовољства на намеру будућег коришћења у фази анализе корисничких захтева

Хипотеза X8a није потврђена, иако су студенти исказали висок ниво задовољства и намере да наставе са употребом *GenAI* алата. Овај резултат показује да високе просечне оцене два конструкта не подразумевају нужно и статистички значајан структурни однос између њих. Задовољство и усклађеност задатка и технологије били су високо међусобно повезани, па је њихове јединствене ефекте на намеру било тешко раздвојити. Намера будућег коришћења могла је бити условљена и ширим позитивним ставом према *GenAI* технологији, претходним искуством студената или очекивањем да ће ови алати бити све присутнији у професионалном окружењу.

Резултати прве итерације истраживања показују да прихватање *GenAI* алата у раним фазама развоја софтвера не следи у потпуности једноставан линеаран процес у којем позитивна оцена технологије нужно доводи до задовољства и намере будућег коришћења. У анализи корисничких захтева посебно су значајни били индивидуалне карактеристике студената, перципирана корисност и способност учења начина употребе алата. У управљању софтверским пројектом кључан фактор намере будућег коришћења било је задовољство оствареним резултатима.

Ови резултати указују на потребу да се интеграција *GenAI* алата у пројектно-засновану наставу прилагоди конкретној фази развоја софтвера. У анализи захтева наставна подршка треба да буде усмерена на формулисање прецизних упита, проверу потпуности и недвосмислености захтева и критичку процену генерисаних резултата. У управљању пројектом потребно је нагласити образлагање предложених приоритета, примену критеријума пословне вредности и упоређивање препорука *GenAI* алата са одлукама које доносе студенти. На тај начин, *GenAI* алати могу да буду интегрисани као подршка развоју аналитичких и управљачких компетенција, а не као средство за аутоматско преузимање генерисаних одговора.

7.2 Примена методолошког оквира у фазама пројектовања, имплементације и тестирања софтвера

7.2.1 Контекст истраживања и хипотезе

Као и у истраживању приказаном у поглављу 7.1, у овом делу истраживања учествовали су студенти завршне године основних академских студија студијског програма Информациони системи и технологије на Факултету организационих наука Универзитета у Београду. За потребе овог дела истраживања посебно су била значајна знања која су студенти током претходних година стекли из програмирања, структура података и алгоритама, база података, рачунарских мрежа и развоја апликација електронског пословања.

Истраживање је спроведено у оквиру **предмета Интернет технологије**. Циљ предмета је стицање теоријских знања из области веб програмирања и практичних вештина потребних за развој веб апликација које обухватају клијентску и серверску страну система. Настава се реализује током 13 недеља и обухвата изучавање технологија и поступака релевантних за пројектовање, имплементацију и тестирање веб апликација. Предиспитне и испитне обавезе обухватају развој софтверског пројекта и његову усмену одбрану. На почетку семестра студенти бирају тему пројекта и добијају ментора који прати њихов рад и напредовање током целокупне реализације пројектног задатка. Детаљна процедура истраживања приказана је у поглављу 7.2.2.

Описани наставни контекст омогућио је примену развијеног методолошког оквира **у фазама пројектовања, имплементације и тестирања софтвера**. Фаза пројектовања обухватала је активности дефинисања структуре и динамичког понашања система, укључујући креирање концептуалних модела и *UML* дијаграма. Фаза имплементације била је усмерена на развој клијентске и серверске стране апликације, генерисање и унапређивање програмског кода и решавање проблема који су се јављали током реализације. Фаза тестирања обухватала је проверу исправности реализованих функционалности и генерисање и унапређивање тестова. У складу са областима знања дефинисаним у *SWEBOK*-у, пројектовање софтвера је повезано са облашћу *Software Design*, имплементација софтвера са облашћу *Software Construction*, а тестирање софтвера са *Software Testing* облашћу.

Полазећи од интегрисаног *TAM-TTF* истраживачког модела представљеног у поглављу 6.3, за потребе евалуације примене развијеног методолошког оквира у фазама пројектовања, имплементације и тестирања софтвера формулисано је 16 истраживачких хипотеза. Њима су операционализовани односи између конструктора истраживачког модела и омогућено је испитивање усклађености *GenAI* алата са наставним задацима, фактора који утичу на задовољство студената и њихове намере да ове алате користе у будућности.

Истраживање приказано у овом поглављу доприноси испитивању полазних хипотеза *X.1*, *X.2* и *X.3* ове докторске дисертације. Испитивањем односа између карактеристика задатака, карактеристика технологије (*GenAI* алата), индивидуалних карактеристика студената и усклађености задатка и технологије разматрају се услови ефективне и системски организоване интеграције *GenAI* алата у наставни процес, чиме се доприноси провери полазне хипотезе *X.1*.

Анализом задовољства студената и њихове намере да прихвате и наставе да користе *GenAI* алате испитује се полазна хипотеза X.2. Анализом оцена који су студенти добили се разматра полазна хипотеза X.3, према којој примена генеративне вештачке интелигенције у настави доприноси побољшању исхода учења.

Истраживачке хипотезе формулисане су за три фазе животног циклуса развоја софтвера: пројектовање, имплементацију и тестирање софтвера. У контексту наставног предмета на којем је истраживање реализовано, фазе имплементације и тестирања разматране су као јединствена целина. Оваква одлука донета је јер су активности тестирања биле реализоване у мањем броју часова и биле су непосредно повезане са имплементацијом програмског решења. Фокус студената био је превасходно на имплементацији софтвера, док је тестирање представљало пратећу активност провере исправности и квалитета развијених функционалности.

Обједињавање фаза имплементације и тестирања не представља ограничење развијеног методолошког оквира. На наставним предметима на којима тестирање софтвера има већи обим, посебне исходе учења и самосталну улогу, ове фазе могу се засебно реализовати и евалуирати. На тај начин, предложени методолошки оквир задржава могућност прилагођавања структури, садржају и специфичностима различитих наставних предмета.

Прва група истраживачких хипотеза заснована је на *TTF* моделу и усмерена је на испитивање усклађености карактеристика *GenAI* алата са задацима које студенти обављају. Полази се од претпоставке да усклађеност задатка и технологије зависи од карактеристика наставног задатка, карактеристика *GenAI* алата и индивидуалних карактеристика студената. Хипотезе X1a, X2a и X3a односе се на фазу пројектовања софтвера, док се хипотезе X1b, X2b и X3b односе на обједињене фазе имплементације и тестирања софтвера:

X1a: Карактеристике задатака позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење пројектовања софтвера.

X1b: Карактеристике задатака позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење имплементације и тестирања софтвера.

X2a: Карактеристике технологије позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење пројектовања софтвера.

X2b: Карактеристике технологије позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење имплементације и тестирања софтвера.

X3a: Индивидуалне карактеристике студената позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење пројектовања софтвера.

X3b: Индивидуалне карактеристике студената позитивно утичу на усклађеност задатка и технологије при примени *GenAI* алата за учење имплементације и тестирања софтвера.

Друга група истраживачких хипотеза заснована је на *TAM* моделу и усмерена је на идентификовање фактора који утичу на задовољство студената *GenAI* алатима. Као непосредни фактори задовољства посматрани су перципирана корисност, перципирана једноставност употребе и перципирана лакоћа учења употребе алата. Хипотезе X4a, X5a и X6a

односе се на фазу пројектовања софтвера, док се хипотезе X4б, X5б и X6б односе на обједињене фазе имплементације и тестирања софтвера:

X4а: Перципирана корисност позитивно утиче на задовољство студената *GenAI* алатима за учење пројектовања софтвера.

X4б: Перципирана корисност позитивно утиче на задовољство студената *GenAI* алатима за учење имплементације и тестирања софтвера.

X5а: Перципирана једноставност употребе позитивно утиче на задовољство студената *GenAI* алатима за учење пројектовања софтвера.

X5б: Перципирана једноставност употребе позитивно утиче на задовољство студената *GenAI* алатима за учење имплементације и тестирања софтвера.

X6а: Перципирана лакоћа учења употребе алата позитивно утиче на задовољство студената *GenAI* алатима за учење пројектовања софтвера.

X6б: Перципирана лакоћа учења употребе алата позитивно утиче на задовољство студената *GenAI* алатима за учење имплементације и тестирања софтвера.

Трећа група истраживачких хипотеза односи се на намеру студената да прихвате и убудуће користе *GenAI* алате у учењу развоја софтвера. У оквиру истраживачког модела испитује се утицај усклађености задатка и технологије, као и задовољства студената, на намеру будуће употребе *GenAI* алата. Хипотезе X7а и X8а односе се на фазу пројектовања софтвера, док се хипотезе X7б и X8б односе на обједињене фазе имплементације и тестирања софтвера:

X7а: Усклађеност задатка и технологије позитивно утиче на намеру студената да користе *GenAI* алате за учење пројектовања софтвера.

X7б: Усклађеност задатка и технологије позитивно утиче на намеру студената да користе *GenAI* алате за учење имплементације и тестирања софтвера.

X8а: Задовољство позитивно утиче на намеру студената да користе *GenAI* алате за учење пројектовања софтвера.

X8б: Задовољство позитивно утиче на намеру студената да користе *GenAI* алате за учење имплементације и тестирања софтвера.

Поред испитивања наведених односа између конструката интегрисаног *TAM-TTF* модела, анализирани су и исходи учења код студената. Исходи учења процењени су на основу оцена које су студенти остварили на усменом испиту, при чему су анализирани резултати студената који су *GenAI* алате користили у складу са развијеним методолошким оквиром. На овај начин, евалуација је, поред ставова, задовољства и намере прихватања технологије, обухватила и остварене резултате учења.

7.2.2 Процедура истраживања

Примена предложеног методолошког оквира у овом истраживању, која се у овом делу рада детаљно разматра, обухватила је три фазе животног циклуса развоја софтвера: **пројектовање, имплементацију и тестирање** софтвера.

Током прве две недеље наставе студенти су похађали уводна предавања на теме као што су животни циклус развоја софтвера, интернет технологије и други основни концепти развоја софтвера. У истом периоду наставници и сарадници дефинисали су теме пројектних задатака усмерене на решавање конкретних пословних проблема.

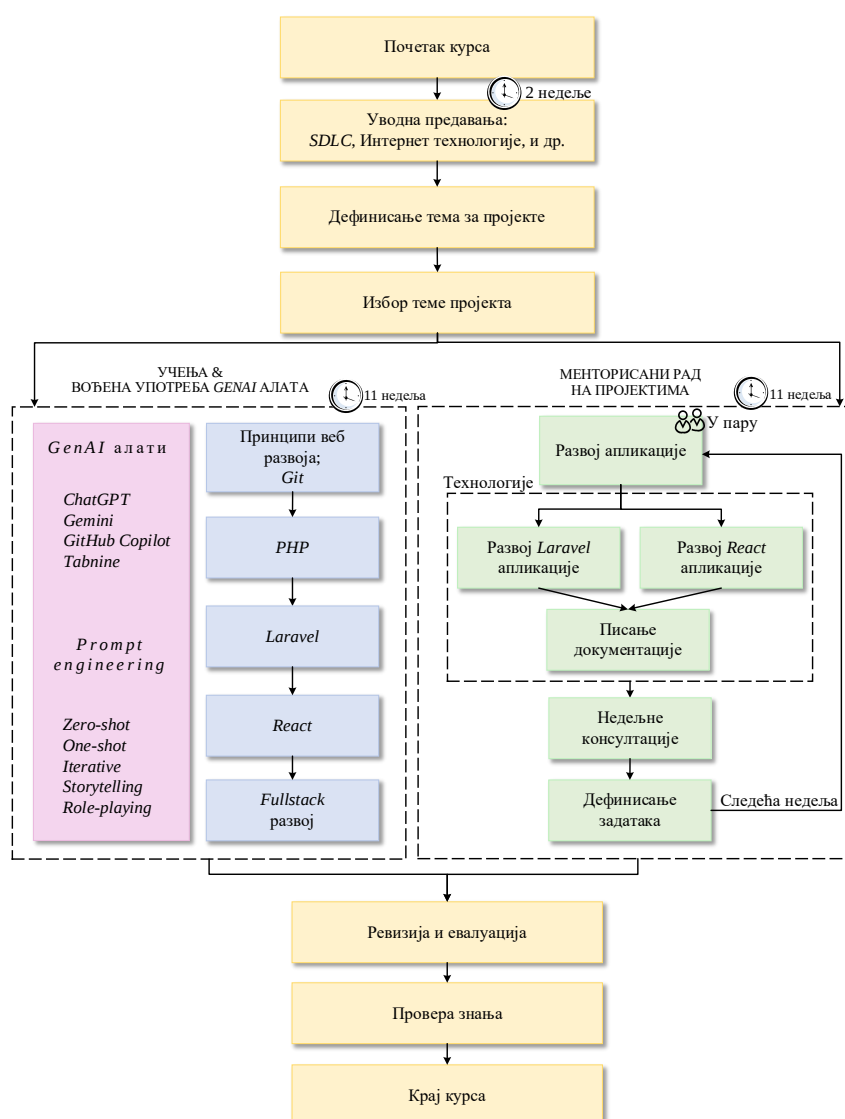
Након уводних недеља наставе, наредних 11 недеља било је посвећено двома паралелним активностима: учењу и менторисаном раду на пројектима. Фаза учења је обухватила практична и теоријска предавања о основама програмског језика *PHP*, *Laravel* оквира за развој *backend* система, *React* библиотеку за *frontend* развој и *Git* система за контролу верзија. Током фазе учења студенти су се упознали са различитим техникама креирања упита, укључујући *zero-shot*, *one-shot*, *few-shot*, итеративне и упите засноване на улогама (енг. *role-based prompts*). Током недељних консултација студентима су од ментора добијали смернице за унапређење структуре и јасноће упита, као и за критичку процену релевантности и исправности садржаја генерисаног применом *GenAI* алата. Како је рад на пројектима напредовао, студенти су постајали самосталнији у коришћењу ових алата. Већина студената се постепено усмеравала ка примени итеративних и улогом заснованих упита, јер су ове стратегије омогућавале добијање прецизнијих резултата који су у већој мери били прилагођени контексту задатка. Током консултација разматрани су и етички аспекти примене *GenAI* алата, укључујући заштиту података, ризик од плагијаризма и значај провере генерисаних резултата. Избор *GenAI* алата био је прилагођен активностима које су студенти реализовали у појединим фазама развоја софтвера, те је преглед алата по активностима дат у Табели 11.

Табела 11. Преглед *GenAI* алата у фазама пројектовања, имплементације и тестирања софтвера

Алат	Активност
Пројектовање софтвера	
<i>Software Architect GPT</i> модул са <i>PlantUML</i>	Генерисање почетних предлога архитектуре софтверског система; моделовање структуре и динамике система; генерисање <i>PlantUML</i> кода за приказ дијаграма; итеративно унапређивање модела на основу захтева пројекта
<i>Eraser.io</i>	Визуелизација архитектуре система и односа између компоненти; израда и унапређивање <i>UML</i> дијаграма
Имплементација и тестирање софтвера	
<i>GitHub Copilot</i>	Генерисање и допуњавање програмског кода; предлагање имплементације функција и компоненти; рефакторисање и отклањање грешака; генерисање почетних предлога тестова
<i>Google Gemini</i>	Генерисање, објашњавање и анализа програмског кода; предлагање алтернативних решења; идентификовање потенцијалних грешака; предлагање тест сценарија, улазних података и граничних случајева; тумачење резултата тестирања
<i>Tabnine</i>	Аутоматско допуњавање програмског кода; предлагање наредних линија и програмских конструкција; убрзавање понављајућих имплементационих активности; подршка при изради аутоматизованих тестова

У оквиру пројектно-заснованог менторског рада студенти су били организовани у групе са задатком да развију веб апликацију на изабрану тему, применом технологија обрађених на настави. Свака група је једном недељно имала консултације са ментором у вези са реализацијом пројектног задатка. Консултације су обухватале процену активности реализованих током претходне недеље, разговор са студентима о искуствима у примени *GenAI* алата, избор и препоруку одговарајућих алата за поједине пројектне активности, пројектовање архитектуре апликације, имплементацију и тестирање програмског решења, као и дефинисање задатака за наредни период.

Након једанаест недеља рада уз подршку ментора, студенти су приступили усменој провери знања, током које су представљали развијене веб апликације и пројектну документацију, као коначне резултате пројектних активности. Ментори су процењивали знање студената постављањем питања која су се односила на конкретне делове програмског кода и донете одлуке у фазама пројектовања, имплементације и тестирања софтвера. Процедура истраживања је приказана на Слици 33.



Слика 33. Примена методолошког оквира у пројектно-заснованом учењу у области развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)

7.2.3 Учесници

У истраживању примене и прихватања *GenAI* алата у фазама пројектовања, имплементације и тестирања софтвера је учествовао 21 студент четврте године који је слушао предмет Интернет технологије. Учесници, старости од 21 до 23 године, изабрани су применом узорковања на основу доступности испитаника (енг. *convenience sampling method*) [281]. Имајући у виду да су сви учесници били студенти завршне године студијског програма из области софтверског инжењерства, они представљају релевантну групу у контексту спроведеног истраживања. Ради обезбеђивања доследности поступка, сви студенти су под истим условима попуњавали упитник. Основни демографски подаци учесника су приказани у Табели 12.

Табела 12. Демографски подаци учесника (фазе пројектовања, имплементације и тестирања софтвера)

Пол	Број учесника
Мушки	7
Женски	14
Просечна оцена [6-10]	Број учесника
7 - 8	7
8 - 9	12
> 9	2

Истраживање је спроведено у складу са институционалним етичким стандардима и уз одобрење надлежне етичке комисије катедре. Учешће у истраживању било је у потпуности добровољно, а студенти су пре прикупљања података били упознати са циљем истраживања. Сви одговори прикупљени су анонимно, без прикупљања личних или осетљивих података. Студентима је наглашено да њихова одлука да учествују или не учествују у истраживању ни на који начин неће утицати на њихово оцењивање.

7.2.4 Инструменти

За прикупљање података и евалуацију истраживања коришћена су два инструмента:

1. **Упитник** - развијен ради испитивања ставова, задовољства и утисака студената у вези са применом *GenAI* алата. Заснован је на интегрисаном *TTF-TAM* моделу и обухватио је девет група тврдњи оцењиваних на петостепеној Ликертовој скали: перципирану корисност, перципирану једноставност употребе, перципирану лакоћу учења, задовољство, намеру прихватања *GenAI* алата за учење развоја софтвера, карактеристике задатка, карактеристике технологије, индивидуалне карактеристике и усклађеност задатка и технологије. Поред затворених питања, упитник је садржао и отворена питања намењена опису проблема и давању предлога. Репрезентативне ставке из сваке групе приказане су и анализиране у наредном поглављу. Структура и садржај упитника прилагођени су на основу претходно валидираних инструмената заснованих на *TAM* и *TTF* моделима [282], [283], чиме је обезбеђена теоријска доследност конструката у посматраном образовном контексту. У прилогу дисертације се налазе упитници за посматране фазе (Прилог 3 и Прилог 4).
Поузданост инструмента проверена је применом Кронбаховог алфа коефицијента. Добијена вредност указује на одличну унутрашњу конзистентност скале и потврђује да ставке поуздано мере исти основни конструкт (Кронбахова алфа = 0,923).

2. **Усмени испит** - коришћен је за процену знања студената и обухватао је питања у вези са развијеним пројектима. Оцене су додељиване засебно за фазе пројектовања, имплементације и тестирања софтвера, као и за презентацију пројектног решења. Испит је заснован на стандардизованом скупу питања који се већ 15 година примењује за проверу знања студената на предмету Интернет технологије.

7.2.5 Анализа резултата

За анализу резултата и испитивање сложених односа између конструктора истраживачког модела примењено је моделирање структурним једначинама (енг. *Structural Equation Modeling* - *SEM*), како би се проверила статистичка значајност постављених истраживачких хипотеза [284]. Конкретно, коришћен је *SEM* приступ заснован на коваријанси, који омогућава истовремено испитивање односа између више латентних конструктора и њихових мерних показатеља [285]. Анализа је спроведена применом програмског пакета *semopy* за програмски језик *Python*, намењеног моделирању структурним једначинама [286]. Поред *SEM* анализе, примењене су и дескриптивна статистика и корелациона анализа, ради детаљнијег сагледавања карактеристика прикупљених података и повезаности између посматраних променљивих. Примењени аналитички поступци омогућили су добијање садржајно смислених и интерпретабилних резултата, али приликом њиховог тумачења треба имати у виду ограничење које се односи на релативно мали узорак. У наставку је приказана анализа резултата.

7.2.5.1 Анализа усклађености *GenAI* алата са задацима и захтевима учења студената у развоју софтвера (фазе пројектовања, имплементације и тестирања софтвера)

У оквиру ове секције приказано је испитивање усклађености *GenAI* алата са задацима које студенти обављају и њиховим потребама у процесу учења. Резултати који се односе на примену ових алата у фазама пројектовања, имплементације и тестирања софтвера приказани су у Табели 13.

Табела 13. Усклађеност *GenAI* алата са задацима и захтевима студената у процесу учења развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)

Питања	Пројектовање		Имплементација и тестирање	
	СВ	СД	СВ	СД
Карактеристике задатка (енг. <i>task characteristics</i>)				
<i>GenAI</i> алати су ми корисни за израду структуре система (концептуалног модела) у фази пројектовања софтвера.	4,02	0,78	-	-
<i>GenAI</i> алати су ми корисни за моделовање динамичког понашања система (<i>UML</i> дијаграми секвенци и случајева коришћења) у фази пројектовања софтвера.	3,67	0,95	-	-
<i>GenAI</i> алати су ми корисни за генерисање програмског кода у фази имплементације софтвера.	-	-	4,46	0,52
<i>GenAI</i> алати су ми корисни за тестирање апликације.	-	-	4,00	0,58
Карактеристике технологије (енг. <i>technology characteristics</i>)				
<i>GenAI</i> алати су поуздани у фази пројектовања.	3,64	0,79	-	-
<i>GenAI</i> алати су поуздани при генерисању програмског кода.	-	-	4,04	0,66
<i>GenAI</i> алати су поуздани за тестирање апликације.	-	-	3,50	0,64
Време генерисања одговара на упит код <i>GenAI</i> алата је одговарајуће.	4,45	0,67	4,04	0,59

Индивидуалне карактеристике (енг. <i>individual characteristics</i>)				
Сигуран/на сам у своју способност да ефикасно користим <i>GenAI</i> алате за академске сврхе.	4,60	0,66	4,35	0,55
Верујем да ће коришћење <i>GenAI</i> алата побољшати моју академску продуктивност и успех.	4,33	0,78	4,08	0,95
Када користим <i>GenAI</i> алате, осећам се мотивисаније да завршим свој академски рад.	4,31	0,68	3,73	0,83

Резултати приказани у Табели 13 показују да су студенти исказали веће поверење у примену *GenAI* алата за генерисање структуре система него за моделовање динамике система, при чему су средње вредности износиле 4,02 и 3,67. У фазама имплементације и тестирања софтвера студенти су показали висок ниво сигурности у коришћење *GenAI* алата за генерисање програмског кода и оценили су их као корисне у обе фазе, са средњим вредностима 4,46 и 4,00.

Када је реч о поузданости *GenAI* алата, студенти су имали више поверења у њихову примену за генерисање програмског кода него за пројектовање система и тестирање софтвера. Средње вредности износиле су 4,04 за имплементацију и 3,50 за тестирање, док су у фази пројектовања оцене биле неуједначене: 11 студената позитивно је оценило поузданост алата, осам је било неодлучно, а два студента су је оценила негативно.

Подаци из Табеле 13 такође указују на високе средње вредности конструкта индивидуалних карактеристика студената. И у фази пројектовања и у обједињеним фазама имплементације и тестирања студенти су исказали висок ниво сигурности у сопствену способност да ефикасно користе *GenAI* алате у академске сврхе. Средње вредности износиле су 4,60 у фази пројектовања и 4,35 у фазама имплементације и тестирања. Студенти су такође сматрали да *GenAI* алати доприносе унапређењу њихове академске продуктивности и успешности, при чему је средња вредност у фази пројектовања износила 4,33, а слично високе вредности забележене су и у фазама имплементације и тестирања. Ипак, док су у фази пројектовања студенти показали висок ниво мотивације за наставак академског рада уз подршку *GenAI* алата, са средњом вредношћу 4,31, у фазама имплементације и тестирања та мотивација била је умеренија, са средњом вредношћу 3,73.

Тестирање хипотеза применом *SEM* анализе показало је различите резултате у зависности од фазе развоја софтвера. У фази пројектовања, карактеристике задатка и индивидуалне карактеристике студената имале су статистички значајан позитиван утицај на усклађеност задатка и технологије, са p вредностима 0,00007 и 0,0148, чиме су потврђене хипотезе $H1a$ и $H3a$. Насупрот томе, карактеристике технологије нису имале статистички значајан утицај, због чега је хипотеза $H2a$ одбачена.

У обједињеним фазама имплементације и тестирања, карактеристике технологије имале су статистички значајан умерено позитиван утицај на усклађеност задатка и технологије, са p вредношћу 0,030740, чиме је потврђена хипотеза $H2b$. Карактеристике задатка и индивидуалне карактеристике студената нису показале статистички значајан утицај, па су хипотезе $H1b$ и $H3b$ одбачене.

7.2.5.2 Анализа фактора који утичу на задовољство студената *GenAI* алатима за учење развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)

У оквиру ове секције приказана је анализа фактора који утичу на задовољство студената *GenAI* алатима у контексту учења развоја софтвера. Фактори релевантни за фазе пројектовања, имплементације и тестирања софтвера приказани су у Табели 14.

Табела 14. Фактори који утичу на задовољство студената *GenAI* алатима у процесу учења развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)

Питања	Пројектовање		Имплементација и тестирање	
	СВ	СД	СВ	СД
Перципирана корисност (енг. <i>Perceived Usefulness</i>)				
<i>GenAI</i> алати су значајно допринели повећању продуктивности.	4,29	0,54	4,00	0,58
<i>GenAI</i> алати су убрзали процес доношења одлука.	4,17	0,70	3,73	0,93
<i>GenAI</i> алати су допринели квалитету крајњег производа (дијаграми, описи) током фазе пројектовања.	3,90	0,77	-	-
<i>GenAI</i> алати су допринели квалитету програмског кода током имплементације.	-	-	4,19	0,63
<i>GenAI</i> алати су допринели квалитету тестова.	-	-	3,54	0,78
Резултати које су генерисали <i>GenAI</i> алати су захтевали мануелне поправке/дораде и сл.	4,36	0,65	3,85	0,69
Перципирана лакоћа коришћења (енг. <i>Perceived Ease of Use</i>)				
<i>GenAI</i> алати су интуитивни за коришћење.	3,81	0,58	4,54	0,52
Било ми је лако да разумем које су поправке/дораде биле потребне код резултата које су генерисали <i>GenAI</i> алати.	3,93	0,75	3,89	0,65
Била су ми потребна додатна упутства/смернице од наставника како бих користио <i>GenAI</i> алате.	2,26	1,26	2,31	1,18
Перципирана лакоћа учења (енг. <i>Perceived Ease of Learning</i>)				
Било ми је лако да постављам <i>prompt</i> -ове које дају квалитетан излаз.	3,95	0,57	3,92	0,64

Студенти су оценили да *GenAI* алати доприносе повећању продуктивности у задацима развоја софтвера, пружају подршку у доношењу одлука и унапређују квалитет коначних резултата у фазама пројектовања и имплементације софтвера. Средње вредности за наведене тврдње износиле су 4,29, 4,17, 3,90 и 4,19. Такође су се сложили да *GenAI* алати могу да убрзају процес доношења одлука током имплементације, при чему је средња вредност износила 3,73. Ипак, већина студената навела је да резултати које генеришу *GenAI* алати захтевају ручне исправке или додатно унапређивање, што потврђује висока средња вредност од 4,36.

Студенти су *GenAI* алате оценили као интуитивне за коришћење, а потребне исправке као лаке за разумевање и примену и у фази пројектовања и у фази имплементације. Средње вредности за ове тврдње износиле су 3,81, 3,93 и 3,89. Ставови су били нешто неуједначенији када је реч о доприносу *GenAI* алата унапређењу тестова програмског кода, где је средња вредност износила 3,54. Поред тога, два од укупно 21 студента навела су да им је била потребна додатна подршка ментора при коришћењу *GenAI* алата. Ипак, студенти су у целини исказали висок ниво сигурности у сопствену способност формулисања упита који доводе до квалитетних резултата, са средњом вредношћу 3,95.

Тестирање хипотеза применом *SEM* анализе показало је да перципирана корисност има статистички значајан позитиван утицај на задовољство студената у фази пројектовања софтвера ($p = 0,0087$). Насупрот томе, перципирана лакоћа учења употребе алата и перципирана једноставност употребе нису показале статистички значајан утицај на задовољство. Због тога је хипотеза *X4a* прихваћена, док су хипотезе *X5a* и *X6a* одбачене.

У фази имплементације, перципирана лакоћа учења употребе алата показала се као статистички значајан предиктор задовољства ($p = 0,000083$), док перципирана корисност и перципирана једноставност употребе нису имале статистички значајан утицај. Сходно томе, хипотеза *X6b* је прихваћена, док су хипотезе *X4b* и *X5b* одбачене.

7.2.5.3 Анализа спремности студената за усвајање *GenAI* алата за учење развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)

У оквиру ове секције испитани су фактори који одређују спремност студената да усвоје *GenAI* алате за учење развоја софтвера. Резултати за фазе пројектовања, имплементације и тестирања софтвера приказани су у Табели 15.

Табела 15. Спремност студената за усвајање *GenAI* алата за учење развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)

Питања	Пројектовање		Имплементација и тестирање	
	СВ	СД	СВ	СД
Задовољство (енг. <i>satisfaction</i>)				
Одговори и препоруке које <i>GenAI</i> алати дају испуњавају моја очекивања.	3,88	0,59	3,96	0,59
Задовољан/на сам резултатима које <i>GenAI</i> алати пружају.	4,14	0,65	4,19	0,48
Препоручио/ла бих коришћење <i>GenAI</i> алата колегама.	4,60	0,62	4,23	0,72
Повратна информација од наставника је била задовољавајућа за задатке које сам урадио/ла коришћењем <i>GenAI</i> алата.	4,52	0,58	4,54	0,52
Усклађеност задатка и технологије (енг. <i>task-technology fit</i>)				
Довољно ми је да користим <i>GenAI</i> алате за генерисање структуре система (концептуални модел) у фази пројектовања.	3,38	0,97	-	-
Довољно ми је да користим <i>GenAI</i> алате за генерисање динамике система (<i>UML</i> дијаграми) у фази пројектовања.	3,29	0,94	-	-
Довољно ми је да користим <i>GenAI</i> алате за генерисање програмског кода у фази имплементације.	-	-	2,96	1,09
Довољно ми је да користим <i>GenAI</i> алате за тестирање апликације.	-	-	2,85	0,92
<i>GenAI</i> алати се лако интегришу са другим алатима које користим.	4,02	0,51	4,04	0,66
Намера усвајања (енг. <i>Intention to adopt</i>)				
Намеравам да наставим да користим <i>GenAI</i> алате у будућности.	4,48	0,58	4,27	0,66

Студенти су оценили да *GenAI* алати у целини испуњавају њихова очекивања и да пружају задовољавајуће резултате и у фази пројектовања и у фази имплементације софтвера, при чему су средње вредности износиле 3,88, 4,14, 3,96 и 4,19. Такође су исказали висок ниво спремности да препоруче употребу *GenAI* алата, са средњим вредностима 4,60 и 4,23, док су

повратне информације ментора оцењене као веома значајне током читавог процеса, са средњим вредностима 4,52 и 4,54.

Студенти су препознали да *GenAI* алати могу да пруже подршку при генерисању структуре система и извршавању задатака програмирања, али су истовремено нагласили да је и даље неопходно добро разумевање програмерских концепата, нарочито у фази имплементације. Иако су оценили да се *GenAI* алати релативно лако интегришу са другим алатима за развој софтвера у различитим фазама, са средњим вредностима 4,02 и 4,04, ставови у погледу тога да су резултати које *GenAI* алати довољни сами по себи били су неуједначени. Упркос наведеним резервама, студенти су показали изражену намеру да наставе да користе *GenAI* алате и у фази пројектовања и у фази имплементације софтвера, са средњим вредностима 4,48 и 4,27.

Тестирање хипотеза применом *SEM* анализе показало је да задовољство има статистички значајан позитиван утицај на намеру студената да усвоје *GenAI* алате и у фази пројектовања и у фази имплементације софтвера. На основу тога, хипотезе *X8a* и *X8b* су прихваћене. Насупрот томе, усклађеност задатка и технологије није показала статистички значајан утицај ни у једној од посматраних фаза, због чега су хипотезе *X7a* и *X7b* одбачене.

7.2.5.4 Анализа исхода учења студената који користе *GenAI* алате за развој софтвера (фазе пројектовања, имплементације и тестирања софтвера)

У Табели 16 приказане су оцене којима је процењено знање студената. Наставници су оцењивали студенте у четири категорије, које одговарају фазама животног циклуса развоја софтвера обухваћеним предложеним методолошким оквиром. Сваки студент је појединачно оцењен на основу показаног знања у областима пројектовања, имплементације и тестирања софтвера, као и презентације пројекта.

Процена знања спроведена је кроз структурисани усмени испит заснован на стандардизованом формату. За сваку фазу студентима су постављана циљана питања у вези са развијеним пројектом и примењеним технолошким окружењем, које је обухватало *Laravel* и *React* технологије. Циљ испита није био само да се провери познавање чињеница, већ и да се процени разумевање одлука донетих током развоја софтвера.

У фази пројектовања студенти су образлагали архитектонска решења, укључујући структуру *Laravel* компоненти, начин моделовања релација у бази података применом *Eloquent ORM* механизма и интеграцију *React* корисничког интерфејса са *API* сервисима. У фази имплементације разматрани су детаљи реализованог решења, као што су аутентификација корисника, примена функционалности *Laravel* оквира, управљање стањем у *React* апликацији и употреба *GenAI* алата. Посебна пажња била је посвећена начину на који су студенти прегледали, проверавали и прилагођавали програмски код генерисан применом *GenAI* алата.

У фази тестирања студенти су описивали примењене стратегије тестирања, коришћене алате, као што су *Jest* и *React Testing Library*, значајније грешке које су уочили и отклонили, као и евентуалну употребу *GenAI* алата за генерисање тестова или откривање грешака. У оквиру презентације пројекта процењивана је њихова способност да објасне значајан технички изазов, образложе изабрано решење и предложи могућа будућа унапређења.

У зависности од почетних одговора студената, постављана су и додатна питања ради детаљнијег испитивања начина размишљања и потврђивања разумевања. Овакав динамичан, али стандардизован приступ усменом оцењивању примењиван је доследно код свих студената. Ради обезбеђивања правичности и поузданости процене, наставници су користили унапред дефинисане рубрике и заједничке критеријуме оцењивања. Коначна оцена сваког студента израчуната је као просечна вредност четири појединачне оцене.

Табела 16. Резултати процене знања студената
(фазе пројектовања, имплементације и тестирања софтвера)

Фаза	СВ	СД
Пројектовање софтвера	8,21	1,46
Имплементација софтвера	8,20	1,57
Тестирање софтвера	8,76	1,08
Презентација	8,71	1,25
Финална оцена	8,47	1,12

На основу резултата приказаних у Табели 16 може се закључити да су студенти показали висок ниво знања из области обухваћених предметом Интернет технологије. Средње оцене за пројектовање, имплементацију и тестирање софтвера износиле су 8,21, 8,20 и 8,76 од максималних 10 поена. Високе просечне оцене за презентацију пројекта и коначну оцену, које су износиле 8,71 и 8,47, додатно потврђују добро усвојено знање студената.

Варијабилност оцена приказана је стандардном девијацијом и била је умерено ниска. Највећа одступања забележена су у областима пројектовања и имплементације софтвера, где су вредности стандардне девијације износиле 1,46 и 1,57. Ово се може објаснити тиме што је мањи број студената показао недовољно теоријско знање у наведеним областима. Тај недостатак утицао је на мање успешно ослањање на *GenAI* алате и, последично, на слабији учинак током процене знања.

7.2.6 Дискусија

Резултати спроведеног истраживања показују да интеграција *GenAI* алата у пројектно-засновану наставу може да пружи значајну подршку студентима у различитим фазама развоја софтвера. Добијени налази су у складу са резултатима претходних истраживања, према којима *GenAI* алати могу да олакшају усвајање програмерских концепата [287], понуде више начина за решавање истог проблема и пруже подршку приликом прегледања, исправљања и отклањања грешака у програмском коду [288]. Међутим, резултати овог истраживања показују да њихова улога није била ограничена на генерисање кода и решавање појединачних програмерских задатака. Студенти су их примењивали и у сложенијим активностима, као што су моделовање структуре и динамике софтверског система, аутоматизација понављајућих имплементационих задатака и сл. Посебно је уочена њихова корисност приликом генерисања *JSON* структура за тестирање програмских интерфејса.

Један од значајних закључака се односи на чињеницу да током истраживања нису уочени проблеми у вези са плагијаризмом, што се разликује од забринутости истакнутих у ранијим

истраживањима о могућој злоупотреби *GenAI* алата у академском окружењу [18], [289]. Студенти су развили међусобно различита софтверска решења и током усмене одбране показали разумевање програмског кода, архитектуре система и одлука донетих током имплементације. Њихова способност да образложе примењена решења указује на то да нису некритички преузимали генерисани садржај. Ови резултати упућују на закључак да *GenAI* алати, када се користе у оквиру структурисаног наставног поступка и уз континуирани менторски надзор, могу унапредити учење без нарушавања оригиналности студентског рада.

Резултати додатно потврђују значај претходног знања студената и систематичног наставног усмеравања за ефективну примену *GenAI* алата. Претходна истраживања показују да студенти који улажу више напора у формулисање јасних и структурираних упита и који добро познају основне концепте области остварују веће користи од употребе ових алата [290], [291]. У овом истраживању учествовали су студенти завршне године, који су већ поседовали знања из програмирања, база података, алгоритама, рачунарских мрежа и сродних области. То им је омогућило да критички процењују добијене резултате, препознају грешке и прилагођавају генерисана решења захтевима пројекта.

Значајну улогу у овом процесу имали су континуирано праћење напретка, примена различитих *GenAI* алата у зависности од фазе развоја софтвера и недељне менторске консултације. На почетку реализације пројекта студентима је била потребна већа подршка у формулисању упита и процени генерисаних резултата. Како је пројекат напредовао, они су постајали самосталнији и све чешће примењивали итеративне и улогом засноване упите. Овакав начин рада омогућио је постепено развијање компетенција потребних за критичку и одговорну употребу *GenAI* алата. Резултати указују и на потенцијал *GenAI* алата за подршку персонализованом учењу, што је у складу са налазима претходних истраживања о њиховој способности прилагођавања различитим потребама студената [292]. Студенти који су активно унапређивали своје упите и критички процењивали добијене одговоре могли су да прилагоде употребу *GenAI* алата сопственим циљевима учења и конкретним проблемима у пројекту. Ипак, студенти са развијенијим основним знањима остваривали су веће користи од примене ових алата. Овај налаз указује на то да *GenAI* треба посматрати као допуну традиционалним облицима учења и наставе, а не као замену за стицање фундаменталних знања и развијање способности самосталног решавања проблема.

Анализа резултата у фази пројектовања софтвера показала је да перципирана корисност *GenAI* алата има статистички значајан позитиван утицај на задовољство студената. Студенти су ове алате посебно вредновали због подршке у моделовању структуре и динамике система, које су препознали као сложене активности. Овај резултат може се објаснити чињеницом да су пре почетка имплементације имали ограничено искуство у пројектовању сложенијих веб-апликација. *GenAI* алати помогли су им да јасније сагледају структуру будућег решења и лакше пређу са концептуалног модела на имплементацију. Када су студенти процењивали да им алат конкретно помаже у реализацији ових активности, повећавало се и њихово задовољство његовом употребом.

У фазама имплементације и тестирања, статистички значајан утицај на задовољство имала је перципирана лакоћа учења употребе *GenAI* алата, док су карактеристике технологије значајно

утицале на усклађеност задатка и технологије. Ови налази указују на важност брзог усвајања начина употребе алата, њихове интеграције са развојним окружењем и подршке за генерисање и анализу програмског кода. Истовремено, ниже оцене поузданости резултата у тестирању показују да су студенти били свесни ограничења ових алата. Генерисани тестови и предлози за отклањање грешака често су захтевали проверу, допуну или прилагођавање конкретном програмском решењу. Због тога су студенти *GenAI* алате превасходно доживљавали као подршку сопственом раду и учењу, а не као замену за самостално расуђивање и решавање проблема.

Резултати *SEM* анализе показали су да се односи између конструктора разликују у зависности од фазе животног циклуса развоја софтвера. Поред потврђених односа, више истраживачких хипотеза није добило статистичку подршку. Ради јаснијег тумачења, могући разлози њиховог непотврђивања разматрани су у наставку према групама односа обухваћених истраживачким моделом.

1. Утицај карактеристика задатка, технологије и студената на усклађеност задатка и технологије

X16 - Утицај карактеристика задатка на усклађеност задатка и технологије у фазама имплементације и тестирања

Хипотеза X16 није потврђена, јер карактеристике задатка нису показале статистички значајан утицај на усклађеност задатка и технологије у фазама имплементације и тестирања. Једно од могућих објашњења односи се на релативну уједначеност пројектних активности. Све студентске групе развијале су веб-апликације применом истог или сличног технолошког окружења и реализовале упоредиве функционалности. Због ограничене варијабилности задатака, њихове карактеристике нису у довољној мери објашњавале разлике у процени усклађености са *GenAI* алатима.

X2a - Утицај карактеристика технологије на усклађеност задатка и технологије у фази пројектовања

Хипотеза X2a није потврђена, будући да карактеристике технологије нису имале статистички значајан утицај на усклађеност задатка и технологије у фази пројектовања. Иако су *GenAI* алати омогућавали брзо генерисање резултата, њихова поузданост и подршка разликовале су се у зависности од врсте модела. Алати су пружали квалитетнију подршку при генерисању концептуалних модела и структуре система него при моделовању динамике система, нарочито *UML* дијаграма секвенци и случајева коришћења. Због неуједначеног квалитета резултата, саме карактеристике алата нису биле довољне да објасне усклађеност са свим задацима пројектовања.

X3b - Утицај индивидуалних карактеристика студената на усклађеност задатка и технологије у фазама имплементације и тестирања

Хипотеза X3b није потврђена, јер индивидуалне карактеристике студената нису показале статистички значајан утицај на усклађеност задатка и технологије у фазама имплементације и тестирања. Иако су студенти исказали висок ниво самопоуздања у коришћењу *GenAI* алата и поседовали претходна знања из програмирања, ове разлике нису значајно утицале на процену уклапања алата у конкретне развојне задатке. Могуће је да су заједничко технолошко окружење, структурисане наставне активности и континуирана менторска подршка смањили разлике које би иначе произилазиле из индивидуалних карактеристика студената.

2. Утицај перцепција *GenAI* алата на задовољство студената

X46 - Утицај перципиране корисности на задовољство у фазама имплементације и тестирања

Хипотеза X46 није потврђена, иако су студенти *GenAI* алате оценили као корисне за генерисање програмског кода, решавање развојних проблема и подршку при тестирању. Овај резултат указује на то да сама корисност није била довољна за формирање укупног задовољства у овим фазама. Задовољство је вероватно зависило и од других фактора, као што су лакоћа учења употребе алата, квалитет генерисаног кода, потреба за мануелним исправкама и могућност интеграције резултата у постојећу архитектуру апликације.

X5a - Утицај перципиране једноставности употребе на задовољство у фази пројектовања

Хипотеза X5a није потврђена, што показује да интуитивност коришћења чет-ботова није била довољна да значајно утиче на задовољство студената у фази пројектовања. Иако је комуникација са алатима била релативно једноставна, генерисани модели често су захтевали додатна прецизирања, мануелне исправке и смернице ментора. Због тога су студенти вероватно више вредновали квалитет, исправност и применљивост генерисаног резултата него једноставност интеракције са алатом.

X5b - Утицај перципиране једноставности употребе на задовољство у фазама имплементације и тестирања

Хипотеза X5b такође није потврђена. Иако су студенти алате за генерисање програмског кода углавном сматрали интуитивним, једноставност употребе није била довољна да повећа задовољство када су се јавили проблеми у вези са исправљањем кода, интеграцијом генерисаних решења и тестирањем апликације. Посебан изазов представљали су генерисање одговарајућих тестова, обухватање граничних случајева и откривање сложенијих грешака, што је захтевало додатну проверу и интервенцију студената.

X6a - Утицај перципиране лакоће учења употребе алата на задовољство у фази пројектовања

Хипотеза X6a није потврђена, што указује на то да лакоћа савладавања техника формулисања упита није непосредно утицала на задовољство студената у фази пројектовања. Студенти су у овој фази вероватно више вредновали квалитет и тачност генерисаних модела него брзину којом су научили да користе алат. Чак и када су успешно усвојили технике формулисања упита, задовољство је зависило од тога да ли су добијени резултати били употребљиви и усклађени са захтевима конкретног пројекта.

3. Утицај усклађености задатка и технологије на намеру усвајања *GenAI* алата

X7a - Утицај усклађености задатка и технологије на намеру усвајања у фази пројектовања

Хипотеза X7a није потврђена. Иако су студенти у одређеној мери препознали усклађеност *GenAI* алата са задацима пројектовања, та процена није довела до израженије намере да наставе да користе алате. Резултати показују да је задовољство било значајнији фактор намере будуће употребе. То значи да функционално уклапање алата у задатак није било довољно уколико употреба алата није довела до позитивног укупног искуства и задовољавајућег квалитета резултата.

X7b - Утицај усклађености задатка и технологије на намеру усвајања у фазама имплементације и тестирања

Ни хипотеза X7b није потврђена. Усклађеност *GenAI* алата са задацима имплементације и тестирања није имала статистички значајан утицај на намеру студената да их користе у будућности. Могуће је да су студенти будућу употребу више повезивали са задовољством

добијеним резултатима, подршком ментора и личним искуством током развоја пројекта него са општом проценом усклађености технологије и задатка.

Посматрано у целини, непотврђивање појединих хипотеза указује на то да прихватање технологије у структурисаном образовном окружењу не мора у потпуности следити класичне претпоставке *TAM* и *TTF* модела. У овом истраживању студенти су *GenAI* алате користили у оквиру педагошки вођеног процеса, који је подразумевао недељне консултације, континуирано праћење напретка и редовне повратне информације. Због тога су менторска подршка, квалитет искуства и задовољство могли да имају значајнију улогу у формирању намере будуће употребе од саме усклађености задатка и технологије.

7.3 Импликације и ограничења истраживања

7.3.1 Практичне импликације

Добијени резултати пружају смернице студентима, наставницима и високошколским установама за интеграцију *GenAI* алата на начин који доприноси унапређењу учења, подржава њихову етички одговорну употребу и омогућава усклађивање образовне праксе са савременим потребама индустрије. Кључне импликације приказане су у Табели 17.

Табела 17. Импликације примене *GenAI* алата у високом образовању

Учесник	Импликације
Студенти	- Одговорна употреба <i>GenAI</i> алата - обезбеђивање доброг познавања основних концепата како би се препознале и избегле нетачне или обмањујуће препоруке алата. - Развој вештина формулисања упита - учење начина креирања квалитетних, јасних и довољно детаљних упита ради добијања прецизнијих и релевантнијих одговора. - Пажљива интеграција <i>GenAI</i> алата у процес учења - коришћење алата за добијање илустративних примера, разматрање различитих приступа решавању проблема и подршку самосталном учењу. - Уважавање све веће примене <i>GenAI</i> алата у индустрији - развијање знања и вештина усклађених са савременим трендовима и потребама тржишта рада у области софтверског инжењерства.
Наставници	- Укључивање <i>GenAI</i> алата у наставне програме - унапређивање искуства учења и проширивање знања студената. - Пажљиво праћење напретка студената током израде задатака - спречавање прекомерног ослањања на <i>GenAI</i> алате. - Усмеравање употребе <i>GenAI</i> алата кроз менторство - пружање подршке студентима у остваривању циљева предмета и трајнијем усвајању знања.
Високошколске установе	- Пружање подршке за укључивање <i>GenAI</i> алата у наставне програме - обезбеђивање њихове делотворне интеграције и унапређивање исхода учења. - Управљање процесом увођења <i>GenAI</i> алата - дефинисање одговарајућих политика, етичких стандарда и правила употребе. - Праћење тржишних трендова у примени <i>GenAI</i> алата кроз сарадњу са професионалном заједницом - развијање компетенција студената прилагођавањем наставе савременим и променљивим потребама индустрије.

Надовезујући се на претходно представљене импликације, у наставку су дате препоруке за њихову практичну примену:

1. Студенти

- Проверавати резултате које генеришу *GenAI* алати и анализирати уочене грешке ради продубљивања разумевања основних концепата.
- Примењивати технике формулисања упита како би се побољшале релевантност и тачност генерисаних одговора.
- Користити *GenAI* као допунско средство за учење и разматрање различитих приступа решавању проблема.
- Развијати свест о примени *GenAI* алата у индустрији како би се циљеви учења ускладили са професионалним очекивањима.

2. Наставници

- Интегрисати *GenAI* алате у наставне програме кроз пројектно-засноване и истраживачки усмерене активности.
- Осмишљавати постепено структурисане задатке који успостављају равнотежу између рада уз подршку *GenAI* алата и самосталног рада студената.
- Континуирано пратити напредак студената како би се спречило прекомерно ослањање на генерисана решења.
- Кроз менторски рад усмеравати студенте ка етичкој, критичкој и одговорној употреби *GenAI* алата.

3. Високошколске установе

- Обезбедити обуку и подршку наставницима за педагошки сврсисходну примену *GenAI* алата.
- Развити институционалне смернице, етичке стандарде и правила оцењивања у вези са употребом *GenAI* алата у образовању.
- Успоставити сарадњу са професионалном заједницом како би наставни програми пратили нове праксе примене *GenAI* алата.
- Улагати у инфраструктуру и ресурсе неопходне за одрживу интеграцију *GenAI* алата у наставу и учење.

Наведене импликације и препоруке јачају практични значај истраживања јер нуде конкретне мере које студенти, наставници и високошколске установе могу применити ради одговорне и ефективне интеграције *GenAI* алата у високо образовање.

7.3.2 Ограничења

Упркос доприносима и практичним импликацијама размотреним у претходном делу, спроведено истраживање има одређена ограничења која је потребно узети у обзир приликом тумачења резултата. Њихово разматрање омогућава прецизније сагледавање добијених резултата и указује на могућности за унапређење будућих истраживања. Најзначајнија ограничења и могуће стратегије за њихово превазилажење приказани су у Табели 18.

Табела 18. Ограничења истраживања и стратегије за њихово превазилажење

Ограничење	Стратегија превазилажења
Мали узорак који ограничава могућност генерализације налаза.	Будућа истраживања треба да обухвате веће и разноврсније узорке студената како би се повећале поузданост и спољашња валидност резултата.
Истраживање је спроведено у оквиру једног пројектно-заснованог предмета са интензивном менторском подршком, што може ограничити применљивост налаза на друге образовне контексте.	Поновити и прилагодити приступ у различитим институционалним окружењима и облицима наставе, укључујући и оне са ограниченијим менторским ресурсима, како би се испитала његова шири применљивост.
Није извршена процена претходног знања или искуства студената у коришћењу <i>GenAI</i> алата.	Укључити инструменте за почетну процену како би се стратегије интеграције прилагодиле почетном нивоу компетенција студената.
Није испитана спремност млађих или мање искусних студената за примену <i>GenAI</i> алата.	Спровести циљана истраживања усмерена на студенте почетнике како би се утврдили механизми подршке који олакшавају усвајање <i>GenAI</i> алата.
Није систематски мерен почетни ниво познавања и вештина студената у коришћењу <i>GenAI</i> алата.	Укључити процену почетног нивоа знања и вештина како би се наставни процес прецизније прилагодио студентима и унапредили исходи учења.

Иако наведена ограничења утичу на обухват и могућност генерализације резултата, она не умањују значај основних налаза. Напротив, указују на контекстуалну природу истраживања и услове у којима предложени методолошки оквир даје најбоље резултате, истовремено пружајући основу за његово даље унапређење.

8 Научни и стручни допринос

Најзначајнији допринос овог рада је развој модела пројектно-заснованог учења у области развоја софтвера који омогућава систематску и ефективну интеграцију генеративне вештачке интелигенције. Оваквим приступом очекује се унапређење наставне праксе, повећање ангажованости студената, подршка индивидуалном учењу и приближавање образовног процеса захтевима савременог тржишта рада у ИТ сектору.

Научни доприноси ове докторске дисертације подразумевају:

- развој методолошког оквира за интеграцију генеративне вештачке интелигенције у пројектно-засновано учење;
- формализацију процеса примене генеративне вештачке интелигенције у настави у области развоја софтвера кроз све фазе животног циклуса софтвера;
- анализу и систематизацију улоге генеративне вештачке интелигенције у подршци студентима и наставницима током реализације наставних активности;
- развој модела за праћење ефеката примене генеративне вештачке интелигенције на исходе учења, развој компетенција и самосталност студената;
- развој приступа за ко-креирање наставних материјала и задатака уз примену генеративне вештачке интелигенције, уз активно учешће наставника и студената;
- развој модела за примену генеративне вештачке интелигенције у процесима оцењивања, укључујући подршку аутоматизованом давању повратних информација и вршњачком оцењивању;
- идентификацију ризика, ограничења и етичких изазова везаних за примену генеративне вештачке интелигенције у образовању;
- допринос академској дебати о педагошким импликацијама и трансформацијама наставног процеса у ери вештачке интелигенције.

Стручни доприноси истраживања обухватају:

- дефинисање сценарија примене алата генеративне вештачке интелигенције у настави ИТ дисциплина и развоју професионалних компетенција;
- систематизацију метода и алата генеративне вештачке интелигенције који се примењују у пракси у развоју софтвера, са циљем њихове педагошке интеграције;
- израда образовних ресурса и смерница за примену генеративне вештачке интелигенције у настави;
- преглед и анализа алата и технологија које подржавају имплементацију генеративне вештачке интелигенције у образовне процесе.

Друштвени доприноси се огледају у:

- подршци наставницима у бољој адаптацији наставних метода у складу са технолошким трендовима;
- повећању спремности студената за рад у реалним условима ИТ индустрије и њиховом бољем позиционирању на тржишту рада;
- омогућавању другим образовним институцијама да примене резултате истраживања у циљу унапређења наставне праксе;
- подстицању етичке и критичке употребе вештачке интелигенције у образовању, уз развој дигиталне писмености и одговорности код студената.

Друштвени допринос истраживања може се посматрати и у контексту циљева одрживог развоја Уједињених нација (енг. *Sustainable Development Goals - SDGs*) [293]. Истраживање пре свега доприноси Циљу 4 - Квалитетно образовање (енг. *Quality Education*), кроз развој и евалуацију методолошког оквира усмереног на унапређење образовања у области развоја софтвера применом генеративне вештачке интелигенције и пројектно-заснованог учења. Допринос Циљу 8 - Достојанствен рад и економски раст (енг. *Decent Work and Economic Growth*) огледа се у развоју савремених дигиталних и професионалних компетенција студената, релевантних за њихово будуће укључивање у тржиште рада. Истовремено, интеграција савремених AI технологија у образовање у области развоја софтвера доприноси Циљу 9 - Индустрија, иновације и инфраструктура (енг. *Industry, Innovation and Infrastructure*), кроз подстицање иновација, дигиталне трансформације и развоја компетенција неопходних за примену савремених технологија у софтверском инжењерству.

Резултати истраживања ове докторске дисертацији објављени су у часописима међународног значаја и саопштени на научним скуповима у земљи и иностранству:

Simić, M., Jolović, M., Naumović, T., Lukovac, P., & Despotović-Zrakić, M. (2026). Integrating GenAI Tools Into Software Engineering Education. *Computer Applications In Engineering Education*, 34(2). <https://doi.org/10.1002/cae.70177>, IF(2025)=3.0, M21

Simić, M., Marković, R., & Bogdanović, Z. (2026). Design and Implementation of an AI Agent for Automated Analysis of GitHub Repositories Using LangChain. *Resilient and Intelligent Organizations in Times of Permanent Disruption*, 20th International Symposium SymOrg. University of Belgrade - Faculty of Organizational Sciences, M33

Muljajić, M., Simić, M., Barać, D., Despotović-Zrakić, M., Bogdanović, Z. (2026). Developing Entrepreneurial Competencies and Digital Skills in High School Students Through Project-Based Learning. In: Reis, J.L., Bigne, E., Bogdanović, Z., Santos, J.P.M.d. (eds) *Marketing and Smart Technologies. ICMarTech 2025. Smart Innovation, Systems and Technologies*, vol 473. Springer, Cham. https://doi.org/10.1007/978-3-032-16284-7_23, M33

Miličević, A., Simić, M., Bogdanović, Z., Despotović-Zrakić, M., & Suvajdžić, M. (2025). Entrepreneurship Education Approaches: Hackathon and Project-Based Learning. *The Journal of East European Management Studies, special issue Entrepreneurship Education in Central and East Europe*, pp. 83-111. Nomos Verlagsgesellschaft mbH & Co. KG. <http://doi.org/10.5771/9783748949602-83>, IF(2023)=0.5, M23

Mujezinović, A., Joksimović, A., Kostić, D., Jolović, M., Simić, M. (2025). Blockchain-Based Platform for Tracking Students' Achievements. Paper presented at *Sinteza 2025 - International Scientific Conference on Information Technology, Computer Science, and Data Science*. doi:10.15308/Sinteza-2025-267-273, M33

Simić, M., Šikman, K., Vrljanac, M., Despotović, V., Despotović-Zrakić, M. (2024). Methodology of creating NFT fashion projects. *Facta Universitatis, Series: Electronics and Energetics* 37(1), pp. 29-52. ISSN: 0353-3670, IF(2024)=0.7, M23

Jafarzadeh, H., Simić, M., & Barać, D. (2024). The Role of AI in B2B E-business. In *Proceedings of the International Conference on Marketing and Technologies*, ICMarTech 2024, 5 – 7 December, São Miguel, Azores, M33

Tomić P. & Simić, M. (2024). Development of a Hybrid App based on a REST API. In: M. Kostić-Stanković, I. Mijatović, J. Krivokapić (eds.) *Proceedings of the 19th International Symposium SymOrg*, 12 – 15 June, Zlatibor, Serbia. Retrieved from: <https://symorg.fon.bg.ac.rs/wp-content/uploads/2024/06/Zbornik-SymOrg-2024.pdf>, M33

Kalimančević A., Simić, M., & Radenković, M. (2024). Blockchain Infrastructure for Managing Competences in Higher Education. In: M. Kostić-Stanković, I. Mijatović, J. Krivokapić (eds.) *Proceedings of the 19th International Symposium SymOrg*, 12 – 15 June, Zlatibor, Serbia. Retrieved from: <https://symorg.fon.bg.ac.rs/wp-content/uploads/2024/06/Zbornik-SymOrg-2024.pdf>, M33

Kostić, D., Simić, M., & Marković, R. (2025). Software testing leveraging AI. *11th User Conference on Advanced Automated Testing - UCAAT*, 1-3 April 2025, Thessaloniki, Greece, M34

Antonić, N., Jolović, M., Lukovac, P., Simić, M., & Bogdanović, Z. (2025). Project-Based Learning in Lifelong Education. *European Project Management Journal*, 15(2), 69-85. DOI: 10.56889/eqwg6335, M51

Simić, M., Naumović, T., Lukovac, P., Kostić, D., & Barać, D. (2024). Application of Generative AI tools in Software Development. *Artificial Intelligence Conference* (pp. 124-125), Belgrade, Serbia, December 2024. Retrieved from: https://www.mi.sanu.ac.rs/~ai_conf/previous_editions/2024/AI_Conference_Book_of_Abstracts.pdf, M34

Miličević A., Simić M., Bogdanović Z., Despotović-Zrakić M. & Suvajdžić M (2023). Exploring Pedagogical Pathways: Hackathon and Project-Based Learning (PBL) in Entrepreneurship Education. *2nd International Danube Cup Conference on Entrepreneurship Education*, 24-25 November, Belgrade, Serbia, M34

9 Будућа истраживања

Будућа истраживања треба усмерити ка даљој провери, унапређењу и широкој примени предложеног методолошког оквира за интеграцију алата генеративне вештачке интелигенције у пројектно-засновано образовање у области развоја софтвера. Полазећи од уочених ограничења спроведеног истраживања, наредна истраживања би требало да обухвате веће и разноврсније узорке студената, укључујући студенте са различитих студијских програма, нивоа студија и високошколских установа. На тај начин било би могуће повећати поузданост добијених резултата, унапредити њихову спољашњу валидност и утврдити у којој мери се закључци истраживања могу генерализовати на различите образовне контексте.

Предложени приступ потребно је поновити и прилагодити различитим институционалним окружењима, организационим моделима наставе и облицима реализације пројектно-заснованог учења. Посебно је значајно испитати његову применљивост у условима са ограниченим наставним и менторским ресурсима, већег броја студената, потпуно онлајн или хибридне наставе, као и у институцијама које располажу различитим техничким и инфраструктурним могућностима. Таква истраживања омогућила би идентификовање минималних предуслова за успешну примену оквира и формулисање препорука за његово прилагођавање специфичностима различитих наставних окружења.

Један од важних праваца будућих истраживања односи се на увођење систематичне почетне процене знања, вештина и искуства студената. Примена одговарајућих инструмената пре почетка наставних активности омогућила би прецизније утврђивање почетног нивоа компетенција у области развоја софтвера, употребе *GenAI* алата, формулисања *prompt*-ова и критичке процене генерисаних резултата. На основу добијених података стратегије интеграције алата могле би се прилагодити потребама различитих група студената, чиме би се обезбедила примеренија наставна подршка и потенцијално унапредили исходи учења.

Посебну пажњу потребно је посветити студентима прве године, који услед ограниченог доменског знања могу имати веће потешкоће приликом процене исправности, потпуности и применљивости резултата које генеришу *GenAI* алати. Такође, поред испитивања ставова и перцепција студената, будућа истраживања би требало у већој мери да обухвате објективне показатеље успешности, као што су квалитет софтверских пројеката, тачност и потпуност захтева, квалитет програмског кода, покривеност тестовима, број и врста грешака, време потребно за извршавање задатака и ниво самосталности студената. Комбиновањем субјективних и објективних мера могао би се добити потпунији увид у стварни допринос *GenAI* алата продуктивности, квалитету рада и усвајању знања. Корисно би било укључити и контролне или упоредне групе, како би се резултати студената који користе предложени оквир могли упоредити са резултатима студената који наставне активности реализују без *GenAI* подршке или уз мање структурисан начин њене примене.

Будућа истраживања би могла пратити студенте током више предмета, семестара или након завршетка студија. На тај начин могло би се испитати да ли се позитивни ефекти употребе *GenAI* алата задржавају током времена, у којој мери студенти преносе стечене компетенције на нове и сложеније задатке и да ли примена ових алата утиче на њихову професионалну

спремност. Посебно је важно утврдити да ли дуготрајна употреба *GenAI* подршке доприноси развоју стручних компетенција или може довести до прекомерног ослањања на генерисане предлоге и смањења способности самосталног решавања проблема.

С обзиром на брз развој генеративних модела и алата, потребно је испитивати и утицај карактеристика различитих технологија на исходе примене предложеног оквира. Будуће студије могу обухватити поређење различитих језичких модела, интегрисаних развојних асистената и *AI* агената, као и анализу утицаја њихове тачности, објашњивости, доступности, цене и начина интеграције у развојно окружење. Тиме би се могло утврдити да ли успешност примене зависи понајвише од методолошког приступа или и од конкретних технолошких решења која се користе.

Реализацијом наведених праваца будућих истраживања било би могуће додатно проверити робусност предложеног методолошког оквира, прецизније утврдити услове његове успешне примене и развити конкретне препоруке за његово прилагођавање различитим групама студената, наставним окружењима и фазама животног циклуса развоја софтвера.

10 Закључак

У овој докторској дисертацији разматране су могућности системске и педагошки оправдане интеграције генеративне вештачке интелигенције у пројектно-засновано образовање у области развоја софтвера. Истраживање је засновано на полазишту да примена *GenAI* алата у настави не треба да буде сведена на њихову повремену и неструктурирану употребу, већ да захтева методолошки утемељен приступ којим се њихове функционалности повезују са циљевима наставе, активностима студената, фазама животног циклуса развоја софтвера и очекиваним исходима учења.

У теоријском делу дисертације извршена је анализа концепата генеративне вештачке интелигенције, великих језичких модела, *AI* асистената и *AI* агената, као и њихове примене у образовању и развоју софтвера. Посебна пажња посвећена је пројектно-заснованом учењу као педагошком приступу који омогућава повезивање теоријских знања са решавањем практичних проблема. Размотрене су и промене које употреба генеративне вештачке интелигенције уноси у поједине фазе развоја софтвера, укључујући анализу захтева, управљање софтверским пројектом, пројектовање, имплементацију и тестирање софтвера.

Анализом постојећих педагошких и технолошких оквира утврђено је да они пружају значајна полазишта за интеграцију генеративне вештачке интелигенције у наставу, али да најчешће нису непосредно прилагођени специфичностима образовања у области развоја софтвера. Посебно је уочена потреба за приступом који истовремено обухвата фазе софтверског процеса, пројектне активности студената, избор одговарајућих *GenAI* алата, развој компетенција, улогу наставника, критичку проверу генерисаних резултата и етички одговорну употребу технологије.

На основу теоријске анализе и резултата емпиријских истраживања развијен је методолошки оквир за унапређење образовања у области развоја софтвера интеграцијом алата генеративне вештачке интелигенције у пројектно-засновано учење. Предложени оквир представља доменски специфичан приступ којим се *GenAI* алати повезују са различитим фазама развоја софтвера и одговарајућим наставним активностима. Његова сврха није да замени наставника или самосталан рад студената, већ да обезбеди структуру за планирану употребу технологије, развој критичког мишљења и подстицање активне улоге студената у процесу учења.

Применљивост предложеног оквира испитана је у реалном високошколском окружењу, у оквиру наставних активности које су обухватиле анализу корисничких захтева, управљање софтверским пројектом, пројектовање, имплементацију и тестирање софтвера. На тај начин обухваћене су различите врсте задатака и различити облици употребе генеративне вештачке интелигенције, од генерисања и унапређивања захтева и корисничких прича, преко подршке при доношењу одлука и планирању пројекта, до генерисања програмског кода, тестирања и анализе резултата.

Резултати истраживања показали су да студенти *GenAI* алате углавном доживљавају као корисну и релативно једноставну подршку у реализацији пројектних активности. Најизраженија корисност уочена је у задацима генерисања програмског кода, убрзавања

појединих активности и подршке при формулисању и унапређивању пројектних артефаката. Истовремено, резултати су показали да генерисани садржаји захтевају критичку проверу, ручне исправке и прилагођавање конкретном контексту пројекта. Ово потврђује да ефекти употребе генеративне вештачке интелигенције не зависе само од способности алата, већ и од доменског знања студената, квалитета постављених *prompt*-ова, наставничког усмеравања и способности корисника да процене исправност и применљивост добијених резултата.

Поред ставова студената, анализирани су њихово задовољство, усклађеност технологије са задацима и намера да наставе са употребом *GenAI* алата. Добијени резултати указују на позитиван однос студената према интеграцији ових алата у наставу развоја софтвера. Студенти су исказали спремност да *GenAI* алате користе и у будућим пројектним и професионалним активностима, нарочито када су алати непосредно повезани са задацима које треба да реше и када доприносе продуктивности, квалитету резултата и лакшем разумевању сложених активности.

У наставку се разматрају резултати истраживања у односу на хипотезе постављене у дисертацији.

Хипотеза Х.1, према којој је могуће обезбедити ефективну и етички одговорну примену генеративне вештачке интелигенције у развоју софтвера уколико се њена интеграција спроводи на системски и плански начин, уз примену одговарајућег методолошког оквира који подржава развој релевантних дигиталних и професионалних компетенција студената, потврђена је резултатима теоријског и емпиријског дела истраживања.

Развијени методолошки оквир омогућио је планско повезивање наставних циљева, пројектних активности, фаза развоја софтвера и одговарајућих *GenAI* алата. Његовом применом дефинисане су активности у којима се генеративна вештачка интелигенција користи као подршка, али и механизми који захтевају проверу, критичко преиспитивање и документовање генерисаних резултата. На тај начин употреба технологије није посматрана само са становишта ефикасности, већ и са становишта одговорности, транспарентности, академског интегритета и развоја самосталности студената. Емпиријска примена показала је да се предложени приступ може реализовати у реалном наставном окружењу и прилагодити различитим фазама развоја софтвера.

Хипотеза Х.2, према којој су студенти спремни да прихвате употребу генеративне вештачке интелигенције као подршку у пројектно-заснованом учењу и настави у области развоја софтвера, потврђена је резултатима спроведених истраживања.

Високе оцене перципиране корисности, задовољства и намере за будућу употребу показале су да студенти имају позитиван став према примени *GenAI* алата. Спремност за прихватање била је присутна у различитим фазама софтверског процеса, иако су уочене разлике у зависности од врсте задатка и нивоа поверења у генерисане резултате. Студенти су били спремнији да користе алате у активностима у којима су могли непосредно да провере, исправе и прилагоде резултат, док је већи опрез уочен код задатака који захтевају процену тачности, поузданости и потпуности генерисаних предлога. Резултати стога показују да прихватање технологије не

подразумева некритичко ослањање на њу, већ да је повезано са перципираном корисношћу, лакоћом употребе и усклађеношћу алата са конкретним задатком.

Хипотеза Х.3, према којој примена генеративне вештачке интелигенције у настави доприноси побољшању исхода учења студената, подржана је резултатима истраживања у оквиру обухваћеног наставног контекста.

Студенти су остварили високе резултате приликом процене знања у областима пројектовања, имплементације, тестирања и представљања развијених решења. Поред тога, њихови одговори указали су на позитиван допринос *GenAI* алата продуктивности, квалитету пројектних резултата, разумевању потребних исправки и подршци приликом доношења одлука. Ипак, имајући у виду да истраживањем није обухваћена контролна група и да није у свим случајевима спроведено непосредно поређење почетног и завршног нивоа знања, резултати се не могу тумачити као коначан доказ искључивог узрочног утицаја *GenAI* алата на исходе учења. Они, међутим, пружају довољну основу за закључак да примена ових алата, када је методолошки структурисана и праћена наставничким усмеравањем, може позитивно допринети процесу учења и квалитету студентског рада.

Хипотеза Х.4, према којој је генеративну вештачку интелигенцију могуће применити као алат за ко-креирање курикулума и наставних садржаја, уз активно учешће наставника и студената, што доводи до веће релевантности и прилагођености образовних материјала савременим потребама ИТ сектора, није непосредно емпиријски испитана у оквиру реализованих истраживања.

Ипак, анализа научне литературе, постојећих оквира и примера примене показала је да *GenAI* алати могу подржати анализу потреба, формулисање исхода учења, генерисање и прилагођавање наставних материјала, развој пројектних задатака и усклађивање садржаја са технолошким променама. Њихова употреба може омогућити брже ажурирање наставних садржаја и укључивање студената у предлагање тема, задатака и облика рада. Међутим, таква примена захтева активну улогу наставника, проверу тачности и релевантности генерисаног садржаја и јасно дефинисане механизме одговорности. Због тога се може закључити да је хипотеза теоријски утемељена и подржана резултатима из литературе, али да њена потпуна потврда захтева посебно емпиријско истраживање усмерено на процес ко-креирања курикулума.

Хипотеза Х.5, према којој се генеративна вештачка интелигенција може ефикасно користити као подршка у формативном и вршњачком оцењивању, чиме се унапређује квалитет повратне информације и омогућава правовремено праћење напредовања студената, потврђена је са становишта техничке изводљивости и потенцијала примене развијеног решења.

У оквиру дисертације развијен је *AI* агент за анализу и евалуацију студентских софтверских пројеката. Агент омогућава прикупљање и анализу релевантних пројектних артефаката, њихово поређење са унапред дефинисаним захтевима и генерисање структурисаних повратних информација. На тај начин показано је да генеративна вештачка интелигенција може обезбедити правовремену, доследну и контекстуално прилагођену подршку студентима током рада на пројекту. Оваква подршка може бити коришћена у формативном оцењивању, као

допуна наставничкој повратној информацији, али и као основа за структурисање вршњачког оцењивања. Ипак, коначну процену квалитета рада и одлуке које утичу на оцену треба да доноси наставник, док *AI* агент има улогу помоћног инструмента. Даља евалуација на већем броју пројеката потребна је како би се додатно испитале тачност, доследност и педагошка вредност генерисаних повратних информација.

На основу разматрања постављених хипотеза може се закључити да су хипотезе X.1, X.2 и X.3 потврђене резултатима теоријског и емпиријског истраживања. Хипотеза X.4 подржана је анализом литературе, али није непосредно емпиријски проверена, док је хипотеза X.5 потврђена у погледу могућности развоја и примене *AI* агента за генерисање формативних повратних информација, уз потребу за његовом додатном евалуацијом у различитим наставним условима.

Главни научни допринос дисертације огледа се у развоју методолошког оквира који повезује генеративну вештачку интелигенцију, пројектно-засновано учење и образовање у области развоја софтвера. За разлику од општих оквира за примену генеративне вештачке интелигенције у образовању, предложени оквир полази од специфичних фаза, активности и артефаката софтверског процеса. Он обухвата избор и примену *GenAI* алата, развој *prompt*-ова, критичку проверу резултата, наставничко усмеравање, праћење исхода и етичке принципе примене.

Резултати дисертације истовремено указују да генеративна вештачка интелигенција не представља самостално решење за унапређење наставе. Њена вредност зависи од начина интеграције, квалитета наставног дизајна, претходног знања студената и улоге наставника. Уколико се употребљава без јасно дефинисаних циљева и механизма провере, она може довести до површног учења, некритичког прихватања резултата и прекомерног ослањања на технологију. Насупрот томе, када је њена употреба структурисана, транспарентна и повезана са пројектним задацима, *GenAI* може представљати значајну подршку развоју дигиталних, техничких и професионалних компетенција студената.

Генеративна вештачка интелигенција се може успешно интегрисати у пројектно-засновано образовање у области развоја софтвера, али њени позитивни ефекти нису последица доступности технологије. За њихово остваривање неопходни су системски приступ, јасно дефинисани наставни циљеви, одговарајући задаци, активно учешће наставника, критичка процена генерисаних резултата и континуирано праћење исхода примене. Предложени методолошки оквир представља основу за такву интеграцију и може допринети модернизацији наставе, унапређењу квалитета пројектног рада и припреми студената за развој софтвера у окружењу у којем сарадња човека и система заснованих на вештачкој интелигенцији постаје све значајнија.

Литература

- [1] M. Prince, “Does Active Learning Work? A Review of the Research,” *J. Eng. Educ.*, vol. 93, no. 3, pp. 223–231, Jul. 2004, doi: 10.1002/J.2168-9830.2004.TB00809.X.
- [2] J. W. Thomas, “A REVIEW OF RESEARCH ON PROJECT-BASED LEARNING,” 2000, Accessed: Sep. 02, 2025. [Online]. Available: <http://www.autodesk.com/foundation>
- [3] R. Pucher and M. Lehner, “Project Based Learning in Computer Science – A Review of More than 500 Projects,” *Procedia - Soc. Behav. Sci.*, vol. 29, pp. 1561–1566, Jan. 2011, doi: 10.1016/J.SBSPRO.2011.11.398.
- [4] D. Kokotsaki, V. Menzies, and A. Wiggins, “Project-based learning: A review of the literature,” *Improv. Sch.*, vol. 19, no. 3, pp. 267–277, Nov. 2016, doi: 10.1177/1365480216659733.
- [5] R. Luckin and W. Holmes, “Intelligence Unleashed: An argument for AI in Education,” *UCL Knowl. Lab London, UK.*, pp. 849–851, Feb. 2016, Accessed: Jul. 21, 2025. [Online]. Available: <https://www.pearson.com/content/dam/corporate/global/pearson-dot-com/files/innovation/Intelligence-Unleashed-Publication.pdf>
- [6] R. Annapureddy, A. Fornaroli, and D. Gatica-Perez, “Generative AI Literacy: Twelve Defining Competencies,” *Digit. Gov. Res. Pract.*, vol. 6, no. 1, Feb. 2025, doi: 10.1145/3685680/ASSET/DB045752-BB32-4D32-AD45-F17038C9B2BB/ASSETS/IMAGES/LARGE/DGOV-2024-0031-F01.JPG.
- [7] D. K. Kanbach, L. Heiduk, G. Blueher, M. Schreiter, and A. Lahmann, “The GenAI is out of the bottle: generative artificial intelligence from a business model innovation perspective,” *Rev. Manag. Sci.*, vol. 18, no. 4, pp. 1189–1220, Apr. 2024, doi: 10.1007/S11846-023-00696-Z/FIGURES/3.
- [8] M. Y. Mohammed and M. J. Skibniewski, “Law and Business The Role of Generative AI in Managing Industry Projects: Transforming Industry 4.0 Into Industry 5.0 Driven Economy,” 2023, doi: 10.2478/law-2023-0006.
- [9] UNESCO, *AI competency framework for teachers*. United Nations Educational, Scientific and Cultural Organization, 2024. doi: 10.54675/zjte2084.
- [10] UNESCO, *AI competency framework for students*. United Nations Educational, Scientific and Cultural Organization, 2024. doi: 10.54675/jkjb9835.
- [11] European Commission, “Ethical guidelines for educators on using artificial intelligence - European Education Area.” <https://education.ec.europa.eu/focus-topics/digital-education/action-plan/ethical-guidelines-for-educators-on-using-ai> (accessed Aug. 26, 2025).
- [12] S. Lau and P. Guo, “From ‘Ban It Till We Understand It’ to ‘Resistance is Futile’: How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot,” in *Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER '23)*, K. Fisler, P. Denny, D. Franklin, and M. Hamilton, Eds., Association for Computing Machinery, New York, NY, USA, Aug. 2023, pp. 106–121. doi: 10.1145/3568813.3600138.
- [13] J. Luo, “A critical review of GenAI policies in higher education assessment: a call to reconsider the ‘originality’ of students’ work,” *Assess. Eval. High. Educ.*, vol. 49, no. 5, pp. 651–664, 2024, doi: 10.1080/02602938.2024.2309963.
- [14] Z. Chen, S. Cross, and B. Rienties, “Empowering Assessors in Providing Quality Feedback with GenAI Assistance: A Preliminary Exploration,” *Commun. Comput. Inf. Sci.*, vol. 2330 CCIS, pp. 134–148, 2024, doi: 10.1007/978-981-96-0205-6_10.
- [15] O. Robayo-Pinzon, S. Rojas-Berrio, J. Rincon-Novoa, and A. Ramirez-Barrera, “Artificial Intelligence and the Value Co-Creation Process in Higher Education Institutions,” *Int. J. Human-Computer Interact.*, vol. 40, no. 20, pp. 6659–6675, Oct. 2024, doi: 10.1080/10447318.2023.2259722.
- [16] C. Bird et al., “Taking Flight with Copilot,” *Queue*, vol. 20, no. 6, pp. 35–57, Dec. 2022, doi: 10.1145/3582083.
- [17] K. B. Ooi et al., “The Potential of Generative Artificial Intelligence Across Disciplines: Perspectives and Future Directions,” *J. Comput. Inf. Syst.*, vol. 65, no. 1, pp. 76–107, Jan. 2023, doi: 10.1080/08874417.2023.2261010.
- [18] L. Chen, P. Chen, and Z. Lin, “Artificial Intelligence in Education: A Review,” *IEEE Access*, vol. 8, pp. 75264–75278, 2020, doi: 10.1109/ACCESS.2020.2988510.
- [19] K. Z. Gajos and L. Mamykina, “Do People Engage Cognitively with AI? Impact of AI Assistance on

- Incidental Learning,” *Int. Conf. Intell. User Interfaces, Proc. IUI*, vol. 22, pp. 794–806, Mar. 2022, doi: 10.1145/3490099.3511138.
- [20] I. Celik, M. Dindar, H. Muukkonen, and S. Järvelä, “The Promises and Challenges of Artificial Intelligence for Teachers: a Systematic Review of Research,” *TechTrends*, vol. 66, no. 4, pp. 616–630, Jul. 2022, doi: 10.1007/s11528-022-00715-y.
- [21] M. Neumann, M. Rauschenberger, and E. M. Schon, “‘We Need To Talk About ChatGPT’: The Future of AI and Higher Education,” in *Proceedings - 2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation, SEENG 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 29–32. doi: 10.1109/SEENG59157.2023.00010.
- [22] D. Ravšelj *et al.*, “Higher education students’ perceptions of ChatGPT: A global study of early reactions,” *PLoS One*, vol. 20, no. 2, p. e0315011, Feb. 2025, doi: 10.1371/JOURNAL.PONE.0315011.
- [23] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, “The Impact of AI on Developer Productivity: Evidence from GitHub Copilot,” Feb. 2023, Accessed: Feb. 05, 2025. [Online]. Available: <https://arxiv.org/abs/2302.06590v1>
- [24] A. E. Hassan, G. A. Oliva, D. Lin, B. Chen, Z. Ming, and Jiang, “Towards AI-Native Software Engineering (SE 3.0): A Vision and a Challenge Roadmap,” Oct. 2024, Accessed: Feb. 10, 2025. [Online]. Available: <https://arxiv.org/abs/2410.06107v1>
- [25] M. Barenkamp, J. Rebstadt, and O. Thomas, “Applications of AI in classical software engineering,” *AI Perspect.* 2020 21, vol. 2, no. 1, pp. 1–15, Jul. 2020, doi: 10.1186/S42467-020-00005-4.
- [26] J. M. Ali, “AI-driven software engineering,” *Adv. Eng. Innov.*, vol. 3, no. 1, pp. 17–21, Oct. 2023, doi: 10.54254/2977-3903/3/2023030.
- [27] V. Kozov, G. Ivanova, and D. Atanasova, “Practical Application of AI and Large Language Models in Software Engineering Education,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 1, pp. 690–696, 2024.
- [28] S. Sarsa, P. Denny, A. Hellas, and J. Leinonen, “Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models,” in *Proceedings of the 2022 ACM Conference on International Computing Education Research (ICER ’22)*, J. Vahrenhold, K. Fisler, M. Hauswirth, and D. Franklin, Eds., ACM, New York, NY, USA, Aug. 2022, pp. 27–43. doi: 10.1145/3501385.3543957.
- [29] R. Chugh *et al.*, “The Promise and Pitfalls: A Literature Review of Generative Artificial Intelligence as a Learning Assistant in ICT Education,” *Comput. Appl. Eng. Educ.*, vol. 33, no. 2, p. e70002, Mar. 2025, doi: 10.1002/CAE.70002.
- [30] N. Forsgren, J. Humble, and G. Kim, *Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations*. IT Revolution, 2018.
- [31] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, “DevOps,” *IEEE Softw.*, vol. 33, no. 3, pp. 94–100, May 2016, doi: 10.1109/MS.2016.68.
- [32] L. Corral and I. Fronza, “Design thinking and agile practices for software engineering an opportunity for innovation,” *SIGITE 2018 - Proc. 19th Annu. SIG Conf. Inf. Technol. Educ.*, pp. 26–31, Sep. 2018, doi: 10.1145/3241815.3241864.
- [33] J. C. Pereira and R. de F. S. M. Russo, “Design Thinking Integrated in Agile Software Development: A Systematic Literature Review,” *Procedia Comput. Sci.*, vol. 138, pp. 775–782, Jan. 2018, doi: 10.1016/J.PROCS.2018.10.101.
- [34] H. Washizaki, Ed., *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*, Version 4. IEEE Computer Society, 2024. [Online]. Available: www.swebok.org
- [35] M. J. Page *et al.*, “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews,” *BMJ*, vol. 372, Mar. 2021, doi: 10.1136/BMJ.N71.
- [36] M. Towhidnejad, O. Ochoa, and A. Kiselev, “An Analysis of the Software Engineering Curriculum Using the Guideline Models,” *ASEE Southeast. Sect. Conf.*, 2020.
- [37] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, and W. Visser, “SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering,” *Computer (Long. Beach. Calif.)*, vol. 48, no. 11, pp. 106–109, Nov. 2015, doi: 10.1109/MC.2015.345.
- [38] M. Barr, O. Andrei, A. Morrison, and S. W. Nabi, “The Development of Students’ Professional Competencies on a Work-Based Software Engineering Program,” *SIGCSE 2024 - Proc. 55th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 1, pp. 81–87, Mar. 2024, doi: 10.1145/3626252.3630944.
- [39] Y. Afshar, M. Moshirpour, E. A. Marasco, J. Kawash, L. Behjat, and M. Moussavi, “An Integrated Software Engineering Curriculum Through Project-Based Learning (PBL),” *ASEE Annu. Conf. Expo. Conf. Proc.*, Aug. 2022, doi: 10.18260/1-2--40924.

- [40] S. E. Gallagher and T. Savage, "Challenge-based learning in higher education: an exploratory literature review," *Teach. High. Educ.*, vol. 28, no. 6, pp. 1135–1157, Aug. 2023, doi: 10.1080/13562517.2020.1863354.
- [41] C. E. Hmelo-Silver, "Problem-based learning: What and how do students learn?," *Educ. Psychol. Rev.*, vol. 16, no. 3, pp. 235–266, Sep. 2004, doi: 10.1023/B:EDPR.0000034022.16470.F3/METRICS.
- [42] W. Hung, D. H. Jonassen, and R. Liu, "Problem-Based Learning," *Handb. Res. Educ. Commun. Technol. Third Ed.*, pp. 485–506, 2008, Accessed: Jul. 20, 2026. [Online]. Available: <https://www.taylorfrancis.com/chapters/edit/10.4324/9780203880869-42/problem-based-learning-woei-hung-david-jonassen-rude-liu>
- [43] M. Leijon, P. Gudmundsson, P. Staaf, and C. Christersson, "Challenge based learning in higher education– A systematic literature review," *Innov. Educ. Teach. Int.*, vol. 59, no. 5, pp. 609–618, Sep. 2022, doi: 10.1080/14703297.2021.1892503.
- [44] A. Kolmos, "Characteristics of Problem-Based Learning," *Int. J. Mech. Eng. Educ.*, vol. 19, no. 5, pp. 657–662, 2003, Accessed: Jul. 21, 2026. [Online]. Available: <https://vbn.aau.dk/en/publications/characteristics-of-problem-based-learning/>
- [45] D. F. Wood, "Problem based learning," *BMJ*, vol. 326, no. 7384, pp. 328–330, Feb. 2003, doi: 10.1136/BMJ.326.7384.328.
- [46] D. E. Allen, R. S. Donham, and S. A. Bernhardt, "Problem-based learning," *New Dir. Teach. Learn.*, vol. 2011, no. 128, pp. 21–29, Dec. 2011, doi: 10.1002/TL.465.
- [47] I. Richardson and Y. Delaney, "Problem based learning in the software engineering classroom," *Proc. - 22nd Conf. Softw. Eng. Educ. Training, CSEET 2009*, pp. 174–181, 2009, doi: 10.1109/CSEET.2009.34.
- [48] P. Dolog, L. L. Thomsen, and B. Thomsen, "Assessing Problem-Based Learning in a Software Engineering Curriculum Using Bloom's Taxonomy and the IEEE Software Engineering Body of Knowledge," *ACM Trans. Comput. Educ.*, vol. 16, no. 3, May 2016, doi: 10.1145/2845091.
- [49] D. Kokotsaki, V. Menzies, and A. Wiggins, "Project-based learning: A review of the literature," *Improv. Sch.*, vol. 19, no. 3, pp. 267–277, Nov. 2016, doi: 10.1177/1365480216659733.
- [50] B. Perez and A. L. Rubio, "A project-based learning approach for enhancing learning skills and motivation in software engineering," *SIGCSE 2020 - Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, pp. 309–315, Feb. 2020, doi: 10.1145/3328778.3366891.
- [51] A. Aliyah, A. Rozi, M. Adhit, and D. Yuda, "BRIDGING ACADEMIA AND INDUSTRY: A SYSTEMATIC LITERATURE REVIEW OF SOFTWARE ENGINEERING EDUCATION APPROACHES AND THEIR," 2026, Accessed: Jul. 21, 2026. [Online]. Available: https://www.researchgate.net/profile/Asro-Asro/publication/405569311_Internationalization_of_the_Tridharma_of_Higher_Education_through_Academic_Industrial_and_Global_Empowerment_Collaboration_BRIDGING_ACADEMIA_AND_INDUSTRY_A_SYSTEMATIC_LITERATURE_REVIEW_O
- [52] L. F. Johnson, R. S. Smith, J. T. Smythe, and R. K. Varon, *Challenge-Based Learning: An Approach for Our Time*. 2009. Accessed: Jul. 21, 2026. [Online]. Available: <https://www.learntechlib.org/p/182083/>
- [53] K. Doulougeri, J. D. Vermunt, G. Bombaerts, and M. Bots, "Challenge-based learning implementation in engineering education: A systematic literature review," *J. Eng. Educ.*, vol. 113, no. 4, pp. 1076–1106, Oct. 2024, doi: 10.1002/JEE.20588.
- [54] M. Detoni, A. Sales, R. Chanin, L. H. Villwock, and A. R. Santos, "Using challenge based learning to create an engaging classroom environment to teach software startups," *ACM Int. Conf. Proceeding Ser.*, pp. 547–552, Sep. 2019, doi: 10.1145/3350768.3353821.
- [55] F. V. Binder, M. Nichols, S. Reinehr, and A. Malucelli, "Challenge Based Learning Applied to Mobile Software Development Teaching," *Proc. - 30th IEEE Conf. Softw. Eng. Educ. Training, CSEET 2017*, vol. 2017-Janua, pp. 57–64, Dec. 2017, doi: 10.1109/CSEET.2017.19.
- [56] K. Helker, M. Bruns, I. M. M. J. Reymen, and J. D. Vermunt, "A framework for capturing student learning in challenge-based learning," *Act. Learn. High. Educ.*, vol. 26, no. 1, pp. 213–229, Mar. 2025, doi: 10.1177/14697874241230459/ASSET/0BAA1BEA-6DA2-441C-8DD1-C07C1E77AFCD/ASSETS/IMAGES/LARGE/10.1177_14697874241230459-FIG1.JPG.
- [57] S. Willis, G. Byrd, and B. D. Johnson, "Challenge-based learning," *Computer (Long. Beach. Calif.)*, vol. 50, no. 7, pp. 13–16, 2017, doi: 10.1109/MC.2017.216.

- [58] A. C. Bock and U. Frank, "Low-Code Platform," *Bus. Inf. Syst. Eng.*, vol. 63, no. 6, pp. 733–740, Dec. 2021, doi: 10.1007/S12599-021-00726-8/FIGURES/1.
- [59] X. Hou *et al.*, "Large Language Models for Software Engineering: A Systematic Literature Review," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 8, Dec. 2024, doi: 10.1145/3695988.
- [60] J. Liu *et al.*, "Large Language Model-Based Agents for Software Engineering: A Survey," *ACM Trans. Softw. Eng. Methodol.*, Sep. 2024, doi: 10.1145/3796507.
- [61] M. O. Ajimati, N. Carroll, and M. Maher, "Adoption of low-code and no-code development: A systematic literature review and future research agenda," *J. Syst. Softw.*, vol. 222, p. 112300, Apr. 2025, doi: 10.1016/J.JSS.2024.112300.
- [62] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, "Supporting the understanding and comparison of low-code development platforms," *2020 46th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA). IEEE*, pp. 171–178, Aug. 2020, Accessed: Jul. 22, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9226356/>
- [63] N. Upadhyaya, "Low-Code/No-Code platforms and their impact on traditional software development: A literature review," *No-Code Platforms Their Impact Tradit. Softw. Dev.*, 2023, Accessed: Jul. 22, 2026. [Online]. Available: https://www.academia.edu/download/119551328/Low_Code_No_Code.pdf
- [64] U. Frank, P. Maier, and A. Bock, "Low code platforms: Promises, concepts and prospects. A comparative study of ten systems," 2021, doi: 10.17185/DUEPUBLICO/75244.
- [65] J. X. Silva and G. Avelino, "Evaluation of low code and no code platforms as a strategy to increase productivity in software development," *ACM Int. Conf. Proceeding Ser.*, pp. 92–102, Dec. 2024, doi: 10.1145/3701625.3701649/ASSET/D28E5BFB-F152-41E3-A3D1-C5C5DA77ACD0/ASSETS/GRAPHIC/SBQS2024-24-IMG11.SVG.
- [66] S. Shridhar and S. Bose, "Analysis of low code-no code development platforms in comparison with traditional development methodologies," *Int. J. Res. Appl. Sci. Eng.*, vol. 9, no. 12, pp. 508–513, 2021, Accessed: Jul. 22, 2026. [Online]. Available: https://www.academia.edu/download/78542364/Analysis_of_Low_Code_No_Code_Development_Platforms_in_comparison_with_Traditional.pdf
- [67] S. K. K. Parimi, "Impact of Low-Code/No-Code Platforms," Feb. 2025, doi: 10.36227/TECHRXIV.174076815.55327992/V1.
- [68] A. Fan *et al.*, "Large language models for software engineering: Survey and open problems," *2023 IEEE/ACM Int. Conf. Softw. Eng. Futur. Softw. Eng.*, pp. 31–53, May 2023, Accessed: Jul. 22, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10449667/>
- [69] P. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models," *Conf. Hum. Factors Comput. Syst. - Proc.*, Apr. 2022, doi: 10.1145/3491101.3519665/SUPPL_FILE/3491101.3519665-TALK-VIDEO.MP4.
- [70] Q. Zhang *et al.*, "A survey on large language models for software engineering," *Sci. China Inf. Sci.*, vol. 69, no. 4, pp. 141102–, Apr. 2026, doi: 10.1007/S11432-025-4670-0/METRICS.
- [71] J. White *et al.*, "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," 2023.
- [72] M. Borg, "Requirements Engineering and Large Language Models: Insights From a Panel," *IEEE Softw.*, vol. 41, no. 2, pp. 6–10, Mar. 2024, doi: 10.1109/MS.2023.3339934.
- [73] A. Sergeyuk, Y. Golubev, T. Bryksin, and I. Ahmed, "Using AI-based coding assistants in practice: State of affairs, perceptions, and ways forward," *Inf. Softw. Technol.*, vol. 178, p. 107610, Feb. 2025, doi: 10.1016/J.INFSOF.2024.107610.
- [74] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software Testing With Large Language Models: Survey, Landscape, and Vision," *IEEE Trans. Softw. Eng.*, vol. 50, no. 4, pp. 911–936, Apr. 2024, doi: 10.1109/TSE.2024.3368208.
- [75] P. Chandrashekar and M. Kari, "A Study on Artificial Intelligence in Software Engineering with Methodologies, Applications, and Effects on SDLC," *TIJER-Int. Res. J.*, vol. 11, no. 12, pp. 932–937, 2024, Accessed: Jul. 22, 2026. [Online]. Available: https://www.researchgate.net/profile/Pooja-Chandrashekar-2/publication/398285037_A_Study_on_Artificial_Intelligence_in_Software_Engineering_with_Methodologies_Applications_and_Effects_on_SDLC/links/69304d39ade0272d5bba81fb/A-Study-on-Artificial-Intelligen
- [76] S. Barke, M. B. James, and N. Polikarpova, "Grounded Copilot: How Programmers Interact with Code-

- Generating Models,” *Proc. ACM Program. Lang.*, vol. 7, no. OOPSLA1, Apr. 2023, doi: 10.1145/3586030.
- [77] J. Becker, N. Rush, E. Barnes, and D. Rein, “Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity,” Jul. 2025, Accessed: Jul. 22, 2026. [Online]. Available: <https://arxiv.org/abs/2507.09089v2>
- [78] J. T. Liang, C. Yang, and B. A. Myers, “A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and Challenges,” *Proc. - Int. Conf. Softw. Eng.*, May 2024, doi: 10.1145/3597503.3608128.
- [79] “Lovable AI App Builder.” <https://lovable.dev/> (accessed Jul. 22, 2026).
- [80] “Hercules | The Best AI App Builder for Business.” <https://hercules.app/> (accessed Jul. 22, 2026).
- [81] “Vibe Coding Platform for Building Apps and Websites | Base44.” <https://base44.com/> (accessed Jul. 22, 2026).
- [82] “Replit | The Next Evolution in AI App Building Tools | Agent 4.” <https://replit.com/agent4> (accessed Jul. 22, 2026).
- [83] A. E. Hassan *et al.*, “Agentic Software Engineering: Foundational Pillars and a Research Roadmap,” vol. 1, no. 1, p. 30, Sep. 2025, doi: 10.48550/arXiv.2509.06216.
- [84] L. Applis, Y. Zhang, S. Liang, N. Jiang, L. Tan, and A. Roychoudhury, “Unified Software Engineering Agent as AI Software Engineer,” *2026 IEEE/ACM 48th Int. Conf. Softw. Eng. (ICSE '26)*, vol. 1, 2026, doi: 10.1145/3744916.3773202.
- [85] L. Wang *et al.*, “A survey on large language model based autonomous agents,” *Front. Comput. Sci.*, vol. 18, no. 6, pp. 186345–, Dec. 2024, doi: 10.1007/S11704-024-40231-1/METRICS.
- [86] A. Roychoudhury, C. Păsăreanu, M. Pradel, and B. Ray, “Agentic AI Software Engineers: Programming with Trust,” *Commun. ACM*, vol. 69, no. 5, pp. 56–58, May 2026, doi: 10.1145/3769314.
- [87] N. Otoum and N. Elkhaili, “Methods and Techniques of Agentic Software Engineering: A Systematic Literature Review,” *IEEE Access*, vol. 14, pp. 7443–7465, Jan. 2026, doi: 10.1109/ACCESS.2026.3652325.
- [88] M. Simić, M. Jolović, T. Naumović, P. Lukovac, and M. Despotović-Zrakić, “Integrating GenAI Tools Into Software Engineering Education,” *Comput. Appl. Eng. Educ.*, vol. 34, no. 2, p. e70177, Mar. 2026, doi: 10.1002/CAE.70177.
- [89] Y. Dong *et al.*, “A Survey on Code Generation with LLM-based Agents,” 2025, doi: 10.48550/arXiv.2508.00083.
- [90] D. Lee, N. He, P. Kamalaruban, and V. Cevher, “Optimization for Reinforcement Learning: From a single agent to cooperative agents,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 123–135, May 2020, doi: 10.1109/MSP.2020.2976000.
- [91] C. Qian *et al.*, “ChatDev: Communicative Agents for Software Development,” *Proc. Annu. Meet. Assoc. Comput. Linguist.*, vol. 1, pp. 15174–15186, 2024, doi: 10.18653/V1/2024.ACL-LONG.810.
- [92] J. He, C. Treude, and D. Lo, “LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead,” *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 5, pp. 1–30, May 2025, doi: 10.1145/3712003.
- [93] M. Azeem Akbar *et al.*, “Agentic AI in Software Engineering: Practitioner Perspectives Across the Software Development Life Cycle,” vol. 1, 2018, doi: 10.2139/ssrn.5520159.
- [94] K. Zhang, J. Li, G. Li, X. Shi, and Z. Jin, “Codeagent: Enhancing Code Generation with Tool-Integrated Agent Systems for Real-World Repo-level Coding Challenges,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024, pp. 13643–13658.
- [95] J. Eguzkitza, “Evaluating Tool-Augmented ReAct Language Agents,” *Univ. BARCELONA*, 2025.
- [96] W. Gan, Z. Ning, Z. Qi, and P. S. Yu, “Mixture of experts (MoE): A big data perspective,” *Inf. Fusion*, vol. 127, p. 103664, Mar. 2026, doi: 10.1016/J.INFFUS.2025.103664.
- [97] V. Garousi, G. Giray, and E. Tuzun, “Understanding the Knowledge Gaps of Software Engineers,” *ACM Trans. Comput. Educ.*, vol. 20, no. 1, Nov. 2019, doi: 10.1145/3360497.
- [98] M. A. Talib, A. Khelifi, A. Abran, and O. Ormandjieva, “Techniques for quantitative analysis of software quality throughout the SDLC: The SWEBOK guide coverage,” *8th ACIS Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2010*, pp. 321–328, 2010, doi: 10.1109/SERA.2010.47.
- [99] A. Alarifi, M. Zarour, N. Alomar, Z. Alshaikh, and M. Alsaleh, “SECDEP: Software engineering curricula development and evaluation process using SWEBOK,” *Inf. Softw. Technol.*, vol. 74, pp. 114–126, Jun. 2016, doi: 10.1016/J.INFSOF.2016.01.013.

- [100] B. S. Bloom, *Taxonomy of educational objectives : The classification of educational goals*, vol. 1. McKay, 1956.
- [101] S. Aldabbus, "Project-based learning: Implementation & challenges," *Int. J. Educ. Learn. Dev.*, vol. 6, no. 3, pp. 71–79, 2018, Accessed: Sep. 02, 2025. [Online]. Available: https://www.researchgate.net/profile/Shaban-Aldabbus/publication/328368222_PROJECT-BASED_LEARNING_IMPLEMENTATION_CHALLENGES/links/5bc8cd20a6fdcc03c79095e0/PROJECT-BASED-LEARNING-IMPLEMENTATION-CHALLENGES.pdf
- [102] J. Larmer and J. R. Mergendoller, "Seven Essentials for Project-Based Learning", Accessed: Sep. 02, 2025. [Online]. Available: http://www.ascd.org/publications/educational_leadership/sept10/vol68/num01
- [103] P. Guo, N. Saab, L. S. Post, and W. Admiraal, "A review of project-based learning in higher education: Student outcomes and measures," *Int. J. Educ. Res.*, vol. 102, p. 101586, Jan. 2020, doi: 10.1016/J.IJER.2020.101586.
- [104] J. E. Hinostroza, S. Armstrong-Gallegos, P. Soto-Valenzuela, and M. Villafaena, "Phases and Activities of Technology-Integrated Project-Based Learning in K-12: Findings from a Systematic Literature Review," *Educ. Sci.* 2025, Vol. 15, Page 1021, vol. 15, no. 8, p. 1021, Aug. 2025, doi: 10.3390/EDUCSCI15081021.
- [105] M. Mihic and I. Zavrski, "Professors' and students' perception of the advantages and disadvantages of project based learning," *Int. J. Eng. Educ.*, pp. 1737–1750, 2017, Accessed: Sep. 02, 2025. [Online]. Available: https://www.researchgate.net/profile/Matej-Mihic/publication/329754724_Professors'_and_Students'_Perception_of_the_Advantages_and_Disadvantages_of_Project_Based_Learning/links/5d66488e458515b5b41def02/Professors-and-Students-Perception-of-the-Advantages-and-Disadvantages-of-Project-Based-Learning.pdf
- [106] S. Palmer and W. Hall, "An evaluation of a project-based learning initiative in engineering education," *Eur. J. Eng. Educ.*, vol. 36, no. 4, pp. 357–365, Aug. 2011, doi: 10.1080/03043797.2011.593095.
- [107] I. D. L. Ríos, A. Cazorla, J. M. Díaz-Puente, and J. L. Yagüe, "Project-based learning in engineering higher education: two decades of teaching competences in real environments," *Procedia - Soc. Behav. Sci.*, vol. 2, no. 2, pp. 1368–1378, Jan. 2010, doi: 10.1016/J.SBSPRO.2010.03.202.
- [108] M. L. Fioravanti *et al.*, "Integrating project based learning and project management for software engineering teaching: An experience report," *SIGCSE 2018 - Proc. 49th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 2018-January, pp. 806–811, Feb. 2018, doi: 10.1145/3159450.3159599.
- [109] G. Sindre, M. Giannakos, B. R. Krogstie, R. Munkvold, and T. Aalberg, "Project-Based Learning in IT Education: Definitions and Qualities," *Uniped*, vol. 41, no. 2, pp. 147–163, Jun. 2018, doi: 10.18261/ISSN1893-8981-2018-02-06.
- [110] S. S. Yadav and J. Xiahou, "Integrated project based learning in software engineering education," *ICENT 2010 - 2010 Int. Conf. Educ. Netw. Technol.*, pp. 34–36, 2010, doi: 10.1109/ICENT.2010.5532120.
- [111] S. & Guo, Y. Yang, S. Guo, and Y. Yang, "Project-Based Learning: an Effective Approach to Link Teacher Professional Development and Students Learning," *J. Educ. Technol. Dev. Exch.*, vol. 5, no. 2, p. 5, Jun. 2012, doi: 10.18785/jetde.0502.04.
- [112] B. Pasi, V. Shinde, and M. C. Education, "Teacher's perception towards their role in course level project-based learning environment," *J. Eng. Educ. Transform.*, no. 1, pp. 2394–1707, 2019, Accessed: Sep. 02, 2025. [Online]. Available: <https://journaleet.in/index.php/jeet/article/view/1550>
- [113] A. Trauth-Nare and G. Buck, "Assessment for learning: Using formative assessment in problem-and project-based learning," *Sci. Teach.*, 2011, Accessed: Sep. 02, 2025. [Online]. Available: <https://scholarworks.iu.edu/dspace/bitstreams/f26f0768-373b-47b7-b3d9-4478d942f1db/download>
- [114] A. M. Mahasneh and A. F. Alwan, "The Effect of Project-Based Learning on Student Teacher Self-Efficacy and Achievement.," *Int. J. Instr.*, vol. 11, no. 3, pp. 511–524, Jul. 2018, doi: 10.12973/iji.2018.11335a.
- [115] S. fong Lam, R. W. yi Cheng, and W. Y. K. Ma, "Teacher and student intrinsic motivation in project-based learning," *Instr. Sci.*, vol. 37, no. 6, pp. 565–578, Sep. 2009, doi: 10.1007/S11251-008-9070-9/FIGURES/1.
- [116] M.-H. Shin, "Effects of Project-based Learning on Students' Motivation and Self-efficacy," *English Teach.*, vol. 73, no. 1, 2018, doi: 10.15858/engtea.73.1.201803.95.
- [117] J. Morrison, J. Frost, C. Gotch, A. R. McDuffie, B. Austin, and B. French, "Teachers' Role in Students'

- Learning at a Project-Based STEM High School: Implications for Teacher Education,” *Int. J. Sci. Math. Educ.*, vol. 19, no. 6, pp. 1103–1123, Aug. 2021, doi: 10.1007/S10763-020-10108-3/METRICS.
- [118] J. H. Fetzer, “What is Artificial Intelligence?,” *Artif. Intell. Its scope limits*, pp. 3–27, 1990, doi: 10.1007/978-94-009-1900-6_1.
- [119] J. McCarthy, “WHAT IS ARTIFICIAL INTELLIGENCE?,” 2007, Accessed: Aug. 24, 2025. [Online]. Available: <http://www-formal.stanford.edu/jmc/>
- [120] R. Raman, R. Kowalski, K. Achuthan, A. Iyer, and P. Nedungadi, “Navigating artificial general intelligence development: societal, technological, ethical, and brain-inspired pathways,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–22, 2025, doi: 10.1038/s41598-025-92190-7.
- [121] M. Damar, A. Özen, Ü. E. Çakmak, E. Özoğuz, and F. S. Erenay, “Super AI, Generative AI, Narrow AI and Chatbots: An Assessment of Artificial Intelligence Technologies for The Public Sector and Public Administration,” *J. AI*, vol. 8, no. 1, pp. 83–106, 2024, doi: 10.61969/jai.1512906.
- [122] B. Goertzel, “Artificial general intelligence: Concept, state of the art, and future prospects,” *J. Artif. Gen. Intell.*, vol. 5, no. 1, 2014, Accessed: Aug. 26, 2025. [Online]. Available: <https://sciencido.com/2/v2/download/article/10.2478/jagi-2014-0001.pdf>
- [123] H. Kim *et al.*, “The Road to Artificial SuperIntelligence: A Comprehensive Survey of Superalignment,” 2024.
- [124] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, “Generative AI,” *Bus. Inf. Syst. Eng.*, vol. 66, no. 1, pp. 111–126, Feb. 2024, doi: 10.1007/s12599-023-00834-7.
- [125] G. K. W. Wong, H.-Y. Cheung, and F. Kalota, “A Primer on Generative Artificial Intelligence,” *Educ. Sci. 2024, Vol. 14, Page 172*, vol. 14, no. 2, p. 172, Feb. 2024, doi: 10.3390/EDUCSCI14020172.
- [126] S. S. Sengar, A. Bin Hasan, S. Kumar, and F. Carroll, “Generative artificial intelligence: a systematic review and applications,” *Multimed. Tools Appl.*, vol. 84, no. 21, pp. 23661–23700, Jun. 2025, doi: 10.1007/S11042-024-20016-1/TABLES/9.
- [127] I. El Naqa and M. J. Murphy, “What Is Machine Learning?,” *Mach. Learn. Radiat. Oncol.*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3_1.
- [128] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science (80-.)*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/SCIENCE.AAA8415.
- [129] Z.-H. Zhou, *Machine Learning*. Springer nature, 2021.
- [130] D. Pandey, K. Niwaria, and B. Chourasia, “Machine Learning Algorithms: A Review,” *Int. Res. J. Eng. Technol.*, 2008, Accessed: Aug. 26, 2025. [Online]. Available: www.irjet.net
- [131] T. Oladipupo Ayodele, “Types of Machine Learning Algorithms,” in *New Advances in Machine Learning*, 2010, pp. 19–49. Accessed: Aug. 26, 2025. [Online]. Available: www.intechopen.com
- [132] N. K. Chauhan and K. Singh, “A review on conventional machine learning vs deep learning,” *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 347–352, Mar. 2019, doi: 10.1109/GUCON.2018.8675097.
- [133] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nat.* 2015 5217553, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [134] Y. Bengio, I. Goodfellow, and A. Courville, “Deep Learning,” 2015.
- [135] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/S12525-021-00475-2/TABLES/2.
- [136] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/J.NEUNET.2014.09.003.
- [137] A. Shrestha and A. Mahmood, “Review of deep learning algorithms and architectures,” *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
- [138] A. Vaswani *et al.*, “Attention Is All You Need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 5999–6009, Jun. 2017, Accessed: Jan. 28, 2025. [Online]. Available: <https://arxiv.org/abs/1706.03762v7>
- [139] S. Islam *et al.*, “A comprehensive survey on applications of transformers for deep learning tasks,” *Expert Syst. Appl.*, vol. 241, p. 122666, May 2024, doi: 10.1016/J.ESWA.2023.122666.
- [140] E. D. Liddy, “Natural Language Processing,” 2001, Accessed: Aug. 29, 2025. [Online]. Available: <https://surface.syr.edu/istpub>
- [141] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, “Natural language processing: An introduction,” *J. Am. Med. Informatics Assoc.*, vol. 18, no. 5, pp. 544–551, Sep. 2011, doi: 10.1136/AMIAJNL-2011-000464/3/AMIAJNL-2011-000464FIG4.JPEG.

- [142] K. R. Chowdhary, "Natural Language Processing," *Fundam. Artif. Intell.*, pp. 603–649, 2020, doi: 10.1007/978-81-322-3972-7_19.
- [143] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/S11042-022-13428-4/FIGURES/3.
- [144] E. Liddy, "Enhanced Text Retrieval Using Natural Language Processing | Enhanced Reader," in *Bulletin of the American Society for Information Science*, 1998.
- [145] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science (80-.)*, vol. 349, no. 6245, pp. 261–266, Jul. 2015, doi: 10.1126/SCIENCE.AAA8685.
- [146] M. Shanahan, "Talking about Large Language Models," *Commun. ACM*, vol. 67, no. 2, pp. 68–79, Jan. 2024, doi: 10.1145/3624724.
- [147] H. Naveed *et al.*, "A Comprehensive Overview of Large Language Models," *ACM Trans. Intell. Syst. Technol.*, Aug. 2025, doi: 10.1145/3744746.
- [148] Y. Chang *et al.*, "A Survey on Evaluation of Large Language Models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, p. 39, Mar. 2024, doi: 10.1145/3641289.
- [149] G. Yenduri *et al.*, "GPT (Generative Pre-Trained Transformer) - A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions," *IEEE Access*, vol. 12, pp. 54608–54649, 2024, doi: 10.1109/ACCESS.2024.3389497.
- [150] M. A. Haque, "Brief Analysis of 'ChatGPT'-A Revolutionary Tool Designed by OpenAI," 2023, doi: 10.4108/airo.v1i1.2983.
- [151] A. Sai, "Explaining GPT-3. Architecture and Working," 2023. <https://medium.com/@tsaiabhi.cool/explaining-gpt-3-architecture-and-working-d0219c79202c> (accessed Sep. 01, 2025).
- [152] A. Kargwal, "Everything you should know about GPT-5," 2025. <https://botpress.com/blog/everything-you-should-know-about-gpt-5> (accessed Aug. 29, 2025).
- [153] S. Wang, M. Hu, Q. Li, M. Safari, and X. Yang, "Capabilities of GPT-5 on Multimodal Medical Reasoning," Aug. 2025, Accessed: Aug. 29, 2025. [Online]. Available: <https://arxiv.org/abs/2508.08224v2>
- [154] Z. Zhang *et al.*, "LLM Hallucinations in Practical Code Generation: Phenomena, Mechanism, and Mitigation," *Proc. ACM Softw. Eng.*, vol. 2, no. ISSTA, pp. 481–503, Jun. 2025, doi: 10.1145/3728894.
- [155] K. I. Roumeliotis and N. D. Tselikas, "ChatGPT and Open-AI Models: A Preliminary Review," *Futur. Internet 2023, Vol. 15, Page 192*, vol. 15, no. 6, p. 192, May 2023, doi: 10.3390/FI15060192.
- [156] J. Deng, Y. L.- Benefits, and undefined 2023, "The benefits and challenges of ChatGPT: An overview," *pdfs.semanticscholar.orgJ Deng, Y LinBenefits, 2023•pdfs.semanticscholar.org*, Accessed: Aug. 30, 2025. [Online]. Available: <https://pdfs.semanticscholar.org/2bf0/2911cbc3bc24e43a910477a90363225bed95.pdf>
- [157] B. D. Lund *et al.*, "ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing," *Wiley Online Libr. Lund, T Wang, NR Mannuru, B Nie, S Shimray, Z WangJournal Assoc. Inf. Sci. Technol. 2023•Wiley Online Libr.*, vol. 74, no. 5, pp. 570–581, May 2023, doi: 10.1002/asi.24750.
- [158] OpenAI, "Introducing GPT-5," 2025. <https://openai.com/index/introducing-gpt-5/> (accessed Aug. 30, 2025).
- [159] A. Nazir and Z. Wang, "A comprehensive survey of ChatGPT: Advancements, applications, prospects, and challenges," *Meta-Radiology*, vol. 1, no. 2, p. 100022, Sep. 2023, doi: 10.1016/J.METRAD.2023.100022.
- [160] F. Fui-Hoon Nah, R. Zheng, J. Cai, K. Siau, and L. Chen, "Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration," *J. Inf. Technol. Case Appl. Res.*, vol. 25, no. 3, pp. 277–304, Jul. 2023, doi: 10.1080/15228053.2023.2233814.
- [161] A. Casheekar, A. Lahiri, K. Rath, K. S. Prabhakar, and K. Srinivasan, "A contemporary review on chatbots, AI-powered virtual conversational agents, ChatGPT: Applications, open challenges and future research directions," *Comput. Sci. Rev.*, vol. 52, p. 100632, May 2024, doi: 10.1016/J.COSREV.2024.100632.
- [162] M. Abdullah, A. Madain, and Y. Jararweh, "ChatGPT: Fundamentals, Applications and Social Impacts," 2022 *9th Int. Conf. Soc. Networks Anal. Manag. Secur. SNAMS 2022*, 2022, doi: 10.1109/SNAMS58071.2022.10062688.

- [163] A. Bahrini *et al.*, “ChatGPT: Applications, Opportunities, and Threats,” *2023 Syst. Inf. Eng. Des. Symp. SIEDS 2023*, pp. 274–279, 2023, doi: 10.1109/SIEDS58326.2023.10137850.
- [164] G. Team *et al.*, “Gemini: A Family of Highly Capable Multimodal Models,” Dec. 2023, Accessed: Aug. 30, 2025. [Online]. Available: <https://arxiv.org/abs/2312.11805v5>
- [165] Z. Bin Akhtar, “From bard to Gemini: An investigative exploration journey through Google’s evolution in conversational AI and generative AI,” *Comput. Artif. Intell.*, vol. 2, no. 1, pp. 1378–1378, Jun. 2024, doi: 10.59400/CAI.V2I1.1378.
- [166] H. R. Saeidnia, “Welcome to the Gemini era: Google DeepMind and the information industry,” *Libr. Hi Tech News*, 2023, doi: 10.1108/LHTN-12-2023-0214/1254678/WELCOME-TO-THE-GEMINI-ERA-GOOGLE-DEEPMIND-AND-THE.
- [167] N. Liladhar Rane, S. P. Choudhary, and J. Rane, “Gemini versus ChatGPT: applications, performance, architecture, capabilities, and implementation,” *J. Appl. Artif. Intell.*, vol. 5, no. 1, pp. 69–93, Mar. 2024, doi: 10.48185/JAAI.V5I1.1052.
- [168] M. M. Carlà *et al.*, “Large language models as assistance for glaucoma surgical cases: a ChatGPT vs. Google Gemini comparison,” *Graefe’s Arch. Clin. Exp. Ophthalmol.*, vol. 262, no. 9, pp. 2945–2959, Sep. 2024, doi: 10.1007/S00417-024-06470-5/FIGURES/3.
- [169] M. Thelwall, “Is Google Gemini better than ChatGPT at evaluating research quality?”, doi: 10.2478/jdis-2025-0014.
- [170] M. Imran and N. Almusharraf, “Google Gemini as a next generation AI educational tool: a review of emerging educational technology,” *Smart Learn. Environ.*, vol. 11, no. 1, pp. 1–8, Dec. 2024, doi: 10.1186/S40561-024-00310-Z/METRICS.
- [171] A. Pande, R. Patil, ... R. M.-... of E. &, and undefined 2024, “Comprehensive Study of Google Gemini and Text Generating Models: Understanding Capabilities and Performance,” *Res. Pande, R Patil, R Mukkemwar, R Panchal, S Bhoite Grenze Int. J. Eng. Technol. (GIJET)*, 2024•[researchgate.net](https://www.researchgate.net/profile/Sachin-Bhoite/publication/386101230_Comprehensive_Study_of_Google_Gemini_and_Text_Generating_Models_Understanding_Capabilities_and_Performance/links/674480d26dedd318c89c6989/Comprehensive-Study-of-Google-Gemini-and-Text-Generating-Models-Understanding-Capabilities-and-Performance.pdf), Accessed: Aug. 30, 2025. [Online]. Available: https://www.researchgate.net/profile/Sachin-Bhoite/publication/386101230_Comprehensive_Study_of_Google_Gemini_and_Text_Generating_Models_Understanding_Capabilities_and_Performance/links/674480d26dedd318c89c6989/Comprehensive-Study-of-Google-Gemini-and-Text-Generating-Models-Understanding-Capabilities-and-Performance.pdf
- [172] GitHub, “GitHub Copilot · Your AI pair programmer.” <https://github.com/features/copilot> (accessed Aug. 30, 2025).
- [173] A. Moradi Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. (Jack) Jiang, “GitHub Copilot AI pair programmer: Asset or Liability?,” *J. Syst. Softw.*, vol. 203, p. 111734, Sep. 2023, doi: 10.1016/J.JSS.2023.111734.
- [174] S. L. France, “Navigating software development in the ChatGPT and GitHub Copilot era,” *Bus. Horiz.*, vol. 67, no. 5, pp. 649–661, Sep. 2024, doi: 10.1016/J.BUSHOR.2024.05.009.
- [175] B. Puryear, G. S.-J. of C. S. in Colleges, and undefined 2022, “Github copilot in the classroom: learning to code with AI assistance,” *ccsc.org B Puryear, G Sprint Journal Comput. Sci. Coll. 2022•ccsc.org*, vol. 38, no. 1, 2022, Accessed: Aug. 30, 2025. [Online]. Available: <https://www.ccsc.org/publications/journals/NW2022.pdf#page=37>
- [176] M. Wermelinger, “Using GitHub Copilot to Solve Simple Programming Problems,” *SIGCSE 2023 - Proc. 54th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 1, pp. 172–178, Mar. 2023, doi: 10.1145/3545945.3569830.
- [177] B. Yetistiren, I. Ozsoy, and E. Tuzun, “Assessing the quality of GitHub copilot’s code generation,” *PROMISE 2022 - Proc. 18th Int. Conf. Predict. Model. Data Anal. Softw. Eng. co-located with ESEC/FSE 2022*, vol. 10, no. 22, pp. 62–71, Nov. 2022, doi: 10.1145/3558489.3559072.
- [178] J. Liu, X. Zhao, X. Shang, and Z. Shen, “Dive into Claude Code: The Design Space of Today’s and Future AI Agent Systems,” Apr. 2026, Accessed: Jul. 23, 2026. [Online]. Available: <https://arxiv.org/abs/2604.14228v2>
- [179] H. V. F. dos Santos, V. Costa, J. E. Montandon, and M. T. Valente, “Decoding the Configuration of AI Coding Agents: Insights from Claude Code Projects,” *Proc. 2026 Int. Work. Agentic Eng. AGENT 2026*, vol. 1, pp. 63–67, May 2026, doi: 10.1145/3786167.3788412/ASSET/989B3B52-46B0-47AD-B7F9-3A414FDFAE40/ASSETS/IMAGES/LARGE/AGENT26-9-FIG8.JPG.
- [180] A. Sobo, A. Mubarak, A. Baimagambetov, and N. Polatidis, “Evaluating LLMs for Code Generation in HRI: A Comparative Study of ChatGPT, Gemini, and Claude,” *Appl. Artif. Intell.*, vol. 39, no. 1, Dec.

- 2025, doi: 10.1080/08839514.2024.2439610.
- [181] Y. Alraba'nah, A. Sleit, I. Qaddara, and M. Hiari, "Code generation by large language models: a comparative analysis of ChatGPT, Claude, and DeepSeek," *Int. J. Electr. Electron. Eng. Telecommun.*, vol. 15, no. 1, pp. 19–28, 2026, Accessed: Jul. 23, 2026. [Online]. Available: https://www.researchgate.net/profile/Azzam_Sleit/publication/399804213_Code_Generation_by_Large_Language_Models_A_Comparative_Analysis_of_ChatGPT_Claude_and_DeepSeek/links/69b7b83ca5bf176ab550c5d5/Code-Generation-by-Large-Language-Models-A-Comparative-Ana
- [182] J. J. U. Ricárdez, M. P. Vasconcelos, R. Gómez Domínguez, and F. C. Romero, "Integration of Generative AI with ChatGPT in Software Development," 2024.
- [183] Y. Qianyi, "Systematic Evaluation of AI-Generated Python Code: A Comparative Study across Progressive Programming Tasks," Sep. 2024, doi: 10.21203/RS.3.RS-4955982/V1.
- [184] V. Joshi and I. Band, "Disrupting Test Development with AI Assistants," *2025 IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol.*, pp. 421–425, Jul. 2025, doi: 10.1109/IAICT65714.2025.11101520.
- [185] D. Fischer, F. Heffeter, ... S. G.-I. C., and undefined 2023, "AI in Strategic Foresight–Evaluation of ChatGPT, BARD and Perplexity," *Res. Fischer, F Heffeter, SR Grothe, V Joachim, HH JungISPIM Conf. Proceedings, 2023•researchgate.net*, Accessed: Aug. 30, 2025. [Online]. Available: https://www.researchgate.net/profile/Hans-Jung/publication/381377112_AI_in_Strategic_Foresight_-_Evaluation_of_ChatGPT_BARD_and_Perplexity/links/67b5f0bd461fb56424dfce48/AI-in-Strategic-Foresight-Evaluation-of-ChatGPT-BARD-and-Perplexity.pdf
- [186] M. Shukla, I. Goyal, B. Gupta, and J. Sharma, "A Comparative Study of ChatGPT, Gemini, and Perplexity," *Int. J. Innov. Res. Comput. Sci. Technol.*, 2024, doi: 10.55524/ijrcst.2024.12.4.2.
- [187] T. F.-J. of language intelligence and culture and undefined 2024, "Using chatbot-based artificial intelligence (AI) for writing an english essay: The ability of ChatGPT, perplexity AI, and ChatSonic," *jlic.uinkhas.ac.idTN FitriaJournal Lang. Intell. Cult. 2024•jlic.uinkhas.ac.id*, vol. 6, no. 2, pp. 103–128, 2024, doi: 10.35719/jlic.v6i2.139.
- [188] L. Xiong *et al.*, "DeepSeek: Paradigm Shifts and Technical Evolution in Large AI Models," *IEEE/CAA J. Autom. Sin.*, vol. 12, no. 5, pp. 841–858, 2025, doi: 10.1109/JAS.2025.125495.
- [189] S. Joshi, "Technical Review of DeepSeek AI: Ca Insights from," 2025, Accessed: Aug. 30, 2025. [Online]. Available: https://www.researchgate.net/profile/Satyadhar-Joshi-2/publication/394470493_Technical_Review_of_DeepSeek_AI_Capabilities_and_Comparisons_with_Insights_from_Q1_2025/links/689cffe1daa95834904eea1c/Technical-Review-of-DeepSeek-AI-Capabilities-and-Comparisons-with-Insights-from-Q1-2025.pdf
- [190] M. Motaleb and H. Manik, "ChatGPT vs. DeepSeek: A Comparative Study on AI-Based Code Generation," Jan. 2025, Accessed: Aug. 30, 2025. [Online]. Available: <https://arxiv.org/abs/2502.18467v1>
- [191] Y. Awasthi, T. Garikayi, L. T. Fundisi, and B. Mukhalela, "A Comparative Study: Evaluating ChatGPT and DeepSeek AI Tools in Practice," *Int. J. Open Inf. Technol.*, vol. 13, no. 5, 2025, Accessed: Aug. 30, 2025. [Online]. Available: <https://cyberleninka.ru/article/n/a-comparative-study-evaluating-chatgpt-and-deepseek-ai-tools-in-practice>
- [192] R. N. Winarno, B. T. Adiratna, and A. K. Azzahra, "User Acceptance Analysis of AI GROK on Platform X," *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 3, pp. 2114–2120, Jun. 2025, doi: 10.59934/JAIEA.V4I3.1107.
- [193] M. de C. Souza, L. W.-I. ARTIFICIAL, and U. 2025, "Grok, gemini, chatgpt and deepseek: Comparison and applications in conversational artificial intelligence".
- [194] N. Dsouza, "Comparative Analysis of Leading Generative AI Conversational Systems: ChatGPT, Grok AI, Gemini, and Meta AI," *Authorea Prepr.*, Jul. 2025, doi: 10.36227/TECHRXIV.175393328.84629021/V1.
- [195] A. A. Q. Mohammed, B. A. Mudhsh, W. R. A. Bin-Hady, and A. S. Al-Tamimi, "DeepSeek and Grok in the Spotlight After ChatGPT in English Education: A Review Study," *J. English Stud. Arab. Felix*, vol. 4, no. 1, pp. 13–22, Mar. 2025, doi: 10.56540/JESAF.V4I1.114.
- [196] W. ; Budiardjo, E. K. ; Putra, I. W. Syahputri, E. K. Budiardjo, and P. O. H. Putra, "Unlocking the Potential of the Prompt Engineering Paradigm in Software Engineering: A Systematic Literature Review," *AI 2025, Vol. 6, Page 206*, vol. 6, no. 9, p. 206, Aug. 2025, doi: 10.3390/AI6090206.
- [197] G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende, "Prompt Engineering in Large Language

- Models,” pp. 387–402, 2024, doi: 10.1007/978-981-99-7962-2_30.
- [198] S. Schulhoff *et al.*, “The Prompt Report: A Systematic Survey of Prompt Engineering Techniques,” Jun. 2024, Accessed: Sep. 01, 2025. [Online]. Available: <https://arxiv.org/abs/2406.06608v6>
- [199] Y. Sasaki, H. Washizaki, J. Li, N. Yoshioka, N. Ubayashi, and Y. Fukazawa, “Landscape and Taxonomy of Prompt Engineering Patterns in Software Engineering,” *IT Prof.*, vol. 27, no. 1, pp. 41–49, 2025, doi: 10.1109/MITP.2024.3525458.
- [200] S. Ekin, “Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices,” *Authorea Prepr.*, Oct. 2023, doi: 10.36227/TECHRXIV.22683919.V1.
- [201] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications”.
- [202] Y. Qian, “Prompt Engineering in Education: A Systematic Review of Approaches and Educational Applications,” *J. Educ. Comput. Res.*, 2025, doi: 10.1177/07356331251365189/ASSET/D5073224-F5BC-403F-A173-E649390583A0/ASSETS/IMAGES/LARGE/10.1177_07356331251365189-FIG6.JPG.
- [203] R. Sapkota, R. Shrestha, M. Rijal, and M. Karkee, “LangChain vs. LangGraph vs. LangSmith: Taxonomies of Agentic AI Toolchains for End-to-End Orchestration,” 2025, doi: 10.36227/techrxiv.175695645.52670060/v1.
- [204] M. A. Ferrag, N. Tihanyi, and M. Debbah, “From llm reasoning to autonomous ai agents: A comprehensive review,” *IEEE Access*, 2026, Accessed: Jul. 23, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/11540994/>
- [205] L. Hughes *et al.*, “AI Agents and Agentic Systems: A Multi-Expert Analysis,” *J. Comput. Inf. Syst.*, vol. 65, no. 4, pp. 489–517, Jul. 2025, doi: 10.1080/08874417.2025.2483832.
- [206] D. B. Acharya, K. Kuppan, and B. Divya, “Agentic AI: Autonomous intelligence for complex goals—A comprehensive survey,” *IEEE Access*, pp. 18912–18936, Jan. 2025, Accessed: Jul. 23, 2026. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10849561/>
- [207] R. Sapkota, K. I. Roumeliotis, and M. Karkee, “AI Agents vs. Agentic AI: A Conceptual taxonomy, applications and challenges,” *Inf. Fusion*, vol. 126, p. 103599, Feb. 2026, doi: 10.1016/J.INFFUS.2025.103599.
- [208] A. Bandi, B. Kongari, R. Naguru, S. Pasnoor, and S. V. Vilipala, “The Rise of Agentic AI: A Review of Definitions, Frameworks, Architectures, Applications, Evaluation Metrics, and Challenges,” *Futur. Internet* 2025, Vol. 17, Page 404, vol. 17, no. 9, p. 404, Sep. 2025, doi: 10.3390/FI17090404.
- [209] M. A. Islam, S. Somu, and F. M. F. Aldaihani, “The Rise of Agentic AI: Synthesis of Current Knowledge and Future Research Agenda,” *Glob. Bus. Organ. Excell.*, vol. 45, no. 3, pp. 402–416, Mar. 2026, doi: 10.1002/JOE.70019.
- [210] S. Hosseini and H. Seilani, “The role of agentic AI in shaping a smart future: A systematic review,” *Array*, vol. 26, p. 100399, Jul. 2025, doi: 10.1016/J.ARRAY.2025.100399.
- [211] Y. K. Dwivedi *et al.*, “Agentic AI Systems: What It Is and Isn’t,” *Glob. Bus. Organ. Excell.*, vol. 45, no. 3, pp. 253–263, Mar. 2026, doi: 10.1002/JOE.70018.
- [212] C. Zhang *et al.*, “ProAgent: Building Proactive Cooperative Agents with Large Language Models,” *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 16, pp. 17591–17599, Mar. 2024, doi: 10.1609/aaai.v38i16.29710.
- [213] G. R. Gonzalez, J. Habel, and G. K. Hunter, “AI agents, agentic AI, and the future of sales,” *J. Bus. Res.*, vol. 202, p. 115799, Jan. 2026, doi: 10.1016/J.JBUSRES.2025.115799.
- [214] S. Kapoor, B. Stroebl, Z. S. Siegel, N. Nadgir, and A. Narayanan, “AI Agents That Matter,” *Trans. Mach. Learn. Res.*, vol. 2025-May, Jul. 2024, Accessed: Jul. 23, 2026. [Online]. Available: <https://arxiv.org/abs/2407.01502v1>
- [215] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24824–24837, Dec. 2022.
- [216] J. Wang and Z. Duan, “Agent AI with LangGraph: A Modular Framework for Enhancing Machine Translation Using Large Language Models,” Dec. 2024, Accessed: Apr. 17, 2026. [Online]. Available: <https://arxiv.org/abs/2412.03801v1>
- [217] P. J. Hyun, “Building Intelligent AI Agents with LangChain: A Complete Guide,” *MetaVision J. Multidiscip. Stud.*, vol. 1, no. 4, pp. 9–15, Oct. 2024, Accessed: Apr. 23, 2026. [Online]. Available: <https://sinojournals.com/index.php/mvjms/article/view/9>
- [218] K. Pandya, B. Vishvakarma Mahavidyalaya, and I. Mehfuza Holia, “Automating Customer Service

- using LangChain: Building custom open-source GPT Chatbot for organizations,” Oct. 2023, Accessed: Apr. 23, 2026. [Online]. Available: <https://arxiv.org/pdf/2310.05421>
- [219] A. Jain, “Components of Langchain,” *medium.com*, Oct. 2025. <https://medium.com/@abhishekjainindore24/langchain-part-2-components-of-langchain-525cd7c7cb65> (accessed Jul. 23, 2026).
- [220] Q. Wu *et al.*, “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations,” in *First conference on language modeling*, 2024.
- [221] S. Shailendra, R. Kadel, and A. Sharma, “Framework for Adoption of Generative Artificial Intelligence (GenAI) in Education,” *IEEE Trans. Educ.*, vol. 67, no. 5, pp. 777–785, 2024, doi: 10.1109/TE.2024.3432101.
- [222] S. Al-Ali, A. Tlili, and A. R. Al-Ali, “LAIK your classroom: A practical framework to integrate generative AI in higher education classrooms,” *J. Appl. Learn. Teach.*, vol. 7, no. 2, pp. 61–76, Aug. 2024, doi: 10.37074/JALT.2024.7.2.30.
- [223] T. D. Memon and P. Kwan, “A Collaborative Model for Integrating Teacher and GenAI into Future Education,” *TechTrends*, pp. 1–15, Jun. 2025, doi: 10.1007/S11528-025-01105-W/TABLES/2.
- [224] M. Belkina *et al.*, “Implementing generative AI (GenAI) in higher education: A systematic review of case studies,” *Comput. Educ. Artif. Intell.*, vol. 8, p. 100407, Jun. 2025, doi: 10.1016/J.CAEAI.2025.100407.
- [225] P. Mishra and M. J. Koehler, “Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge,” *Teach. Coll. Rec.*, vol. 108, no. 6, pp. 1017–1054, 2006, doi: 10.1111/J.1467-9620.2006.00684.X.
- [226] V. Garousi, Z. Jafarov, A. Movsumova, A. Namazov, and H. Mirzayev, “Encouraging Students’ Responsible Use of GenAI in Software Engineering Education: A Causal Model and Two Institutional Applications,” May 2025, Accessed: Aug. 30, 2025. [Online]. Available: <https://arxiv.org/abs/2506.00682v1>
- [227] A. B.-O. Praxis and undefined 2024, “GenAI et al. Cocreation, authorship, ownership, academic ethics and integrity in a time of generative AI,” *search.informit.orgA BozkurtOpen Praxis, 2024•search.informit.org*, doi: 10.55982/openpraxis.16.1.654.
- [228] K. de Fine Licht, “Rethinking the Ethics of GenAI in Higher Education: A Critique of Moral Arguments and Policy Implications,” *J. Appl. Philos.*, vol. 42, no. 4, pp. 1317–1337, Aug. 2025, doi: 10.1111/JAPP.70026.
- [229] J. Laine, M. Minkkinen, and M. Mäntymäki, “Understanding the Ethics of Generative AI: Established and New Ethical Principles,” *Commun. Assoc. Inf. Syst.*, vol. 56, no. 1, pp. 1–25, Mar. 2025, doi: 10.17705/1CAIS.05601.
- [230] D. Mhlanga, “Open AI in Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning,” pp. 387–409, 2023, doi: 10.1007/978-3-031-37776-1_17.
- [231] E. Đerić, D. Frank, and D. Vuković, “Exploring the Ethical Implications of Using Generative AI Tools in Higher Education,” *Informatics 2025, Vol. 12, Page 36*, vol. 12, no. 2, p. 36, Apr. 2025, doi: 10.3390/INFORMATICS12020036.
- [232] G. Lan, X. Feng, S. Du, F. Song, and Q. Xiao, “Integrating ethical knowledge in generative AI education: constructing the GenAI-TPACK framework for university teachers’ professional development,” *Educ. Inf. Technol.*, vol. 30, no. 11, pp. 15621–15644, Jul. 2025, doi: 10.1007/S10639-025-13427-6/FIGURES/4.
- [233] C. C. Outeda, “European Education Area and Digital Education Action Plan (2021–2027): One More Step Towards the Europeanisation of Education Policy,” *Contrib. to Polit. Sci.*, vol. Part F3080, pp. 187–206, 2024, doi: 10.1007/978-3-031-56045-3_13.
- [234] E. W. Wach, Krzysztof Côté, Doanh Dương Ejdyś, Joanna Kazlauskaitė, Rūta Korzyński, Paweł Mazurek, Grzegorz Paliszkiwicz, Joanna Ziemia, “The dark side of generative artificial intelligence: A critical analysis of controversies and risks of ChatGPT,” *Entrep. Bus. Econ. Rev.*, vol. 11, no. 2, pp. 7–30, 2023.
- [235] Univerzitet u Beogradu - Fakultet organizacionih, “Opšte smernice za korišćenje generativne veštačke inteligencije na Fakultetu organizacionih nauka.” 2024.
- [236] P. Samuelson, “Generative AI meets copyright,” *Science (80-.)*, vol. 381, no. 6654, pp. 158–161, Jul. 2023, doi: 10.1126/SCIENCE.ADI0656.
- [237] E. Dickey and A. Bejarano, “GAIDE: A Framework for Using Generative AI to Assist in Course

- Content Development,” *Proc. - Front. Educ. Conf. FIE*, 2024, doi: 10.1109/FIE61694.2024.10893132.
- [238] Y. Chen *et al.*, “A GenAI Competence Framework for Engineering Curriculum Enhancement in Higher Education,” 2025, Accessed: Sep. 02, 2025. [Online]. Available: https://figshare.le.ac.uk/articles/journal_contribution/A_GenAI_Competence_Framework_for_Engineering_Curriculum_Enhancement_in_Higher_Education/29301884/1/files/55328645.pdf
- [239] K. Karpouzis, D. Pantazatos, J. Taouki, and K. Meli, “Tailoring Education with GenAI: A New Horizon in Lesson Planning,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, 2024, doi: 10.1109/EDUCON60312.2024.10578690.
- [240] D. Karadeniz and M. de Candia, “Redesigning how we redesign the curriculum: Cognitive Group Interviews as a qualitative tool for student-staff co-creation in the age of Generative AI,” *J. Educ. Innov. Partnersh. Chang.*, vol. 11, no. 1, p. 2025, Jul. 2025, Accessed: Sep. 02, 2025. [Online]. Available: <https://journals.studentengagement.org.uk/index.php/studentchangeagents/article/view/1341>
- [241] H. Sandhaus, Q. Gu, M. T. Parreira, and W. Ju, “Co-Designing with Transformers: Unpacking the Complex Role of GenAI in Interactive System Design Education,” pp. 1228–1243, Jul. 2025, doi: 10.1145/3715336.3735805/ASSET/48471947-477F-48BF-B7A6-FA9AB76A3CCE/ASSETS/IMAGES/LARGE/DIS25-175-FIG10.JPG.
- [242] S. Pozdniakov, J. Brazil, M. Mohammadi, M. Dollinger, S. Sadiq, and H. Khosravi, “AI-Assisted Co-Creation: Bridging Skill Gaps in Student-Generated Content,” *J. Learn. Anal.*, vol. 12, no. 1, pp. 129–151, 2025, doi: 10.18608/jla.2025.8601.
- [243] G. W. Choi, S. H. Kim, D. Lee, and J. Moon, “Utilizing Generative AI for Instructional Design: Exploring Strengths, Weaknesses, Opportunities, and Threats,” *TechTrends*, vol. 68, no. 4, pp. 832–844, Jul. 2024, doi: 10.1007/S11528-024-00967-W/FIGURES/6.
- [244] Y. Ma *et al.*, “Preparing Students for an AI-Driven World: Generative AI and Curriculum Reform in Higher Education,” *Front. Digit. Educ.* 2025 24, vol. 2, no. 4, pp. 30–, Sep. 2025, doi: 10.1007/S44366-025-0067-6.
- [245] S. Zeivots *et al.*, “Reshaping Higher Education Designs and Futures: Postdigital Co-design with Generative Artificial Intelligence,” *Postdigital Sci. Educ.*, vol. 7, no. 4, pp. 1334–1374, Dec. 2025, doi: 10.1007/S42438-025-00595-4/FIGURES/14.
- [246] T. J. Marlowe, C. S. Ku, J. R. Laracy, V. D. Kirova, and K. G. Herbert, “A Systemic View of a Software Engineering Education Curriculum: Requirements and Guidelines in the Era of Generative AI,” *J. Integr. Des. Process Sci.*, Aug. 2026, doi: 10.1177/10920617251405471/ASSET/26067158-15AA-467B-B5EB-7AFB8B2DF60C/ASSETS/IMAGES/LARGE/10.1177_10920617251405471-FIG2.JPG.
- [247] F. J. Agbo, C. Olivia, G. Oguibe, I. T. Sanusi, and G. Sani, “Computing education using generative artificial intelligence tools: A systematic literature review,” *Comput. Educ. Open*, vol. 9, p. 100266, Dec. 2025, doi: 10.1016/J.CAEO.2025.100266.
- [248] A. Strzelecki, “Education and Information Technologies ChatGPT in higher education: Investigating bachelor and master students’ expectations towards AI tool,” *Educ. Inf. Technol.*, pp. 1–25, 2024, doi: 10.1007/s10639-024-13222-9.
- [249] O. Petrovska, F. Moller, and R. Pearsall, “Incorporating Generative AI into Software Development Education,” 2024, doi: 10.1145/3633053.3633057.
- [250] C. Bull and A. Kharrufa, “Generative AI Assistants in Software Development Education A vision for integrating Generative AI into educational practice, not instinctively defending against it,” 2023.
- [251] M. Giannakos *et al.*, “The promise and challenges of generative AI in education,” *Behav. Inf. Technol.*, vol. 44, no. 11, pp. 2518–2544, Jul. 2025, doi: 10.1080/0144929X.2024.2394886.
- [252] A. Nguyen-Duc *et al.*, “Generative Artificial Intelligence for Software Engineering—A Research Agenda,” *Softw. Pract. Exp.*, 2025, doi: 10.1002/SPE.70005.
- [253] C. Sengul, R. Neykova, and G. Destefanis, “Software engineering education in the era of conversational AI: current trends and future directions,” *Frontiers in Artificial Intelligence*, vol. 7. Frontiers Media SA, 2024. doi: 10.3389/frai.2024.1436350.
- [254] T. C. Yang, Y. C. Hsu, and J. Y. Wu, “The effectiveness of ChatGPT in assisting high school students in programming learning: evidence from a quasi-experimental research,” *Interact. Learn. Environ.*, pp. 1–18, 2025, doi: 10.1080/10494820.2025.2450659.
- [255] A. S. Pothukuchi, L. V. Kota, and V. Mallikarjunaradhya, “Impact of generative AI on the software development lifecycle (SDLC),” *Int. J. Creat. Res. Thoughts*, vol. 11, no. 8, pp. 2320–2882, 2023.

- [256] B. Arslan *et al.*, “Opportunities and challenges of using generative AI to personalize educational assessment,” *Front. Artif. Intell.*, vol. 7, p. 1460651, Oct. 2024, doi: 10.3389/FRAI.2024.1460651/BIBTEX.
- [257] M. S. Ibarra-Sáiz, G. Rodríguez-Gómez, and D. Boud, “Developing student competence through peer assessment: the role of feedback, self-regulation and evaluative judgement,” *High. Educ.*, vol. 80, no. 1, pp. 137–156, Jul. 2020, doi: 10.1007/S10734-019-00469-2/TABLES/8.
- [258] C. Liu, J. Cui, R. Shang, Q. Jia, P. Rashid, and E. Gehringer, “Generative AI for Peer Assessment Helpfulness Evaluation,” pp. 412–419, 2024, doi: 10.5281/ZENODO.12729848.
- [259] K. J. Topping, E. Gehringer, H. Khosravi, S. Gudipati, K. Jadhav, and S. Susarla, “Enhancing peer assessment with artificial intelligence,” *Int. J. Educ. Technol. High. Educ.*, vol. 22, no. 1, pp. 1–33, Dec. 2025, doi: 10.1186/S41239-024-00501-1/TABLES/2.
- [260] M. Hooda, C. Rana, O. Dahiya, A. Rizwan, and M. S. Hossain, “Artificial Intelligence for Assessment and Feedback to Enhance Student Success in Higher Education,” *Math. Probl. Eng.*, vol. 2022, no. 1, p. 5215722, Jan. 2022, doi: 10.1155/2022/5215722.
- [261] V. González-Calatayud, P. Prendes-Espinosa, and R. Roig-Vila, “Artificial Intelligence for Student Assessment: A Systematic Review,” *Appl. Sci. 2021, Vol. 11, Page 5467*, vol. 11, no. 12, p. 5467, Jun. 2021, doi: 10.3390/APP11125467.
- [262] P. Vittorini, S. Menini, and S. Tonelli, “An AI-Based System for Formative and Summative Assessment in Data Science Courses,” *Int. J. Artif. Intell. Educ.*, vol. 31, no. 2, pp. 159–185, Jun. 2021, doi: 10.1007/S40593-020-00230-2/TABLES/5.
- [263] A. Saini, B. Cope, M. Kalantzis, and G. Zapata, “The future of feedback: integrating peer and generative AI reviews to support student work,” 2024, Accessed: Sep. 02, 2025. [Online]. Available: <https://osf.io/x3dct/download>
- [264] A. Darvishi, H. Khosravi, S. Sadiq, and D. Gašević, “Incorporating AI and learning analytics to build trustworthy peer assessment systems,” *Br. J. Educ. Technol.*, vol. 53, no. 4, pp. 844–875, Jul. 2022, doi: 10.1111/BJET.13233.
- [265] M. Usher, “Generative AI vs. instructor vs. peer assessments: a comparison of grading and feedback in higher education,” *Assess. Eval. High. Educ.*, 2025, doi: 10.1080/02602938.2025.2487495.
- [266] A. Strzelecki, “To use or not to use ChatGPT in higher education? A study of students’ acceptance and use of technology,” 2023, doi: 10.1080/10494820.2023.2209881.
- [267] J. Alvarez Ariza, M. Benitez Restrepo, and C. Hernandez Hernandez, “Generative AI in Engineering and Computing Education: A Scoping Review of Empirical Studies and Educational Practices,” *IEEE Access*, vol. 13, pp. 30789–30810, 2025, doi: 10.1109/ACCESS.2025.3541424.
- [268] H. Cornide-Reyes, D. Monsalves, E. Durán, F. Silva-Aravena, and J. Morales, “Generative Artificial Intelligence in Agile Software Development Processes: A Literature Review Focused on User eXperience,” *Lect. Notes Comput. Sci.*, vol. 15787 LNCS, pp. 228–246, 2025, doi: 10.1007/978-3-031-93536-7_16.
- [269] “Popular AI tools among developers worldwide,” *www.statista.com*, 2024. <https://www.statista.com/statistics/1483838/ai-tools-usage-among-developers-use-worldwide/> (accessed Jun. 19, 2025).
- [270] V. Corso, L. Mariani, D. Micucci, and O. Riganelli, “Generating Java Methods: An Empirical Assessment of Four AI-Based Code Assistants,” *IEEE Int. Conf. Progr. Compr.*, pp. 13–23, 2024, doi: 10.1145/3643916.3644402.
- [271] Q. Cai, Y. Lin, and Z. Yu, “Factors Influencing Learner Attitudes Towards ChatGPT-Assisted Language Learning in Higher Education,” *Int. J. Human–Computer Interact.*, Nov. 2024, doi: 10.1080/10447318.2023.2261725.
- [272] N. F. Liu, T. Zhang, and P. Liang, “Evaluating Verifiability in Generative Search Engines,” *Find. Assoc. Comput. Linguist. EMNLP 2023*, pp. 7001–7025, Apr. 2023, doi: 10.18653/v1/2023.findings-emnlp.467.
- [273] K. El Haji, C. Brandt, and A. Zaidman, “Using GitHub Copilot for Test Generation in Python: An Empirical Study,” *Proc. - 2024 IEEE/ACM Int. Conf. Autom. Softw. Test, AST 2024*, pp. 45–55, Apr. 2024, doi: 10.1145/3644032.3644443.
- [274] F. Davis, “Technology acceptance model: TAM,” *Al-Suqri, MN, Al-Aufi, AS Inf. Seek. Behav. Technol. Adopt.*, vol. 205, no. 219, p. 5, 1989.
- [275] B. Furneaux, “Task-Technology Fit Theory: A Survey and Synopsis of the Literature,” *Inf. Syst. Theory*

- Explain. Predict. Our Digit. Soc.*, vol. 1, pp. 87–106, 2012, doi: 10.1007/978-1-4419-6108-2_5.
- [276] M. Simić, T. Naumović, P. Lukovac, D. Kostić, and D. Barać, “Application of Generative AI tools in Software Development,” *ARTIFICIAL Intell. Conf. B. Abstr.*, pp. 124–125, 2024.
- [277] A. Qayyum, A. Albaseer, J. Qadir, A. Al-Fuqaha, and M. Abdallah, “LLM-Driven Multi-Agent Architectures for Intelligent Self-Organizing Networks,” *IEEE Netw.*, 2025, doi: 10.1109/MNET.2025.3605319.
- [278] S. Yao *et al.*, “ReAct: Synergizing Reasoning and Acting in Language Models,” in *The eleventh international conference on learning representations.*, 2022.
- [279] L. Wang *et al.*, “A survey on large language model based autonomous agents,” *Front. Comput. Sci.*, vol. 18, no. 6, pp. 186345–, Dec. 2024, doi: 10.1007/S11704-024-40231-1/METRICS.
- [280] T. Schick *et al.*, “Toolformer: Language Models Can Teach Themselves to Use Tools,” *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 68539–68551, Dec. 2023.
- [281] J. Golzar, S. Noor, and O. Tajik, “Convenience Sampling,” *Int. J. Educ. Lang. Stud.*, vol. 1, no. 2, pp. 72–77, Dec. 2022, doi: 10.22034/IJELS.2022.162981.
- [282] Y. H. Al-Mamary, A. A. Alfalah, M. M. Alshammari, and A. A. Abubakar, “Exploring factors influencing university students’ intentions to use ChatGPT: analysing task-technology fit theory to enhance behavioural intentions in higher education,” *Futur. Bus. J.*, vol. 10, no. 1, p. 119, Dec. 2024, doi: 10.1186/S43093-024-00406-5.
- [283] B. Wu and X. Chen, “Continuance intention to use MOOCs: Integrating the technology acceptance model (TAM) and task technology fit (TTF) model,” *Comput. Human Behav.*, vol. 67, pp. 221–232, Feb. 2017, doi: 10.1016/J.CHB.2016.10.028.
- [284] R. O. Mueller and G. R. Hancock, “Structural Equation Modeling,” *Rev. Guid. to Quant. Methods Soc. Sci.*, pp. 445–456, Jan. 2018.
- [285] J. F. Hair, B. J. Babin, and N. Krey, “Covariance-Based Structural Equation Modeling in the Journal of Advertising: Review and Recommendations,” *J. Advert.*, vol. 46, no. 1, pp. 163–177, Jan. 2017, doi: 10.1080/00913367.2017.1281777.
- [286] A. A. Igolkina and G. Meshcheryakov, “semopy: A Python Package for Structural Equation Modeling,” *Struct. Equ. Model. A Multidiscip. J.*, vol. 27, no. 6, pp. 952–963, Nov. 2020, doi: 10.1080/10705511.2019.1704289.
- [287] L. Roest, H. Keuning, and J. Jeuring, “Next-Step Hint Generation for Introductory Programming Using Large Language Models,” *ACM Int. Conf. Proceeding Ser.*, pp. 144–153, Jan. 2024, doi: 10.1145/3636243.3636259.
- [288] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos, “Programming Is Hard - or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation,” *SIGCSE 2023 - Proc. 54th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 1, pp. 500–506, Mar. 2023, doi: 10.1145/3545945.3569759.
- [289] R. T. Williams, “The ethical implications of using generative chatbots in higher education,” *Front. Educ.*, vol. 8, p. 1331607, Jan. 2024.
- [290] N. Perry, M. Srivastava, D. Kumar, and D. Boneh, “Do Users Write More Insecure Code with AI Assistants?,” *CCS 2023 - Proc. 2023 ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 2785–2799, Nov. 2023, doi: 10.1145/3576915.3623157.
- [291] D. J. Woo, D. Wang, T. Yung, and K. Guo, “Effects of a Prompt Engineering Intervention on Undergraduate Students’ AI Self-Efficacy, AI Knowledge and Prompt Engineering Ability: A Mixed Methods Study,” *Int. J. Artif. Intell. Educ.*, Jul. 2024.
- [292] S. Mirhosseini, A. Z. Henley, and C. Parnin, “What Is Your Biggest Pain Point? An Investigation of CS Instructor Obstacles, Workarounds, and Desires,” *SIGCSE 2023 - Proc. 54th ACM Tech. Symp. Comput. Sci. Educ.*, vol. 1, pp. 291–297, Mar. 2023, doi: 10.1145/3545945.3569816.
- [293] United Nations, “Transforming our world: the 2030 Agenda for Sustainable Development,” 2015. https://sdgs.un.org/sites/default/files/publications/21252030_Agenda_for_Sustainable_Development_web.pdf (accessed Aug. 18, 2026).

Листа слика

Слика 1. Примена вештачке интелигенције у развоју софтвера (прилагођено према [74])	15
Слика 2. Таксономија приступа софтверског инжењерства заснованог на AI агентима (прилагођено према [86])	18
Слика 3. SWEBOK области знања	20
Слика 4. GPT-3 архитектура	30
Слика 5. LangChain компоненте	43
Слика 6. 4E оквир за усвајање генеративне вештачке интелигенције у образовању [219]	49
Слика 7. LAIK оквир за интеграцију генеративне вештачке интелигенције у наставу високог образовања [220]	50
Слика 8. Колаборативни модел интеграције наставника и генеративне вештачке интелигенције [221]	52
Слика 9. GenAI-TRACK оквир са кључним знањима и способностима наставника за примену генеративне вештачке интелигенције [222]	54
Слика 10. Дијаграм тока анализе литературе за примену генеративне вештачке интелигенције у образовању у домену развоја софтвера	60
Слика 11. Процес идентификације потреба и захтева за развој методолошког оквира	66
Слика 12. Методолошки оквир за унапређење образовања у области развоја софтвера интеграцијом генеративне вештачке интелигенције у пројектно-засновано учење	67
Слика 13. Примена методолошког оквира у пројектно-заснованом учењу у области развоја софтвера (највиши ниво апстракције)	69
Слика 14. ТАМ-ТТФ истраживачки модел	70
Слика 15. Архитектура AI агента за анализу и евалуацију студентских пројеката	72
Слика 16. Ток рада AI агента током формативне евалуације студентског пројекта	74
Слика 17. Функција <code>fetch_repository_evidence()</code> - прикупљање и обједињавање података о GitHub репозиторијуму студентског пројекта	77
Слика 18. Функције <code>select_requirement_relevant_paths()</code> и <code>_requirement_path_score()</code> - избор и рангирање фајлова релевантних за евалуацију	78
Слика 19. Фајл <code>project_requirements.py</code> - део спецификације критеријума за евалуацију студентског пројекта	80
Слика 20. Системски <code>prompt</code> - дефинисање статуса испуњености захтева	80
Слика 21. Функција <code>analyzer.py</code> - генерисање формативне повратне информације	81
Слика 22. Функција <code>main()</code> - чување генерисаног извештаја у Markdown формату	82
Слика 23. Интерактиван унос URL адресе након покретања AI агента	83
Слика 24. Приказ прикупљених података из GitHub репозиторијума	84
Слика 25. Чување генерисаног извештаја	84
Слика 26. Основни подаци о репозиторијуму у генерисаном извештају	85
Слика 27. Сумарни преглед статуса анализираних захтева	86
Слика 28. Структура детаљне евалуације једног пројектног захтева	86
Слика 29. Приказ делимично испуњеног захтева	87
Слика 30. Приоритетне препоруке генерисане од стране AI агента	88
Слика 31. Ограничења аутоматизоване анализе наведена у извештају	88
Слика 32. Примена методолошког оквира у пројектно-заснованом учењу у области развоја софтвера (фазе анализа корисничких захтева и управљање софтверским пројектом)	94
Слика 33. Примена методолошког оквира у пројектно-заснованом учењу у области развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)	111

Листа табела

Табела 1. Алати генеративне вештачке интелигенције - кључне карактеристике	37
Табела 2. UNESCO AI компетенције наставника – димензије и нивои прогресије	46
Табела 3. Упоредни приказ оквира за интеграцију генеративне вештачке интелигенције у образовање	56
Табела 4. Преглед ризика и мера у примени генеративне вештачке интелигенције у образовању	59
Табела 5. Преглед GenAI алата по фазама животног циклуса развоја софтвера	68
Табела 6. Преглед GenAI алата у фазама анализе корисничких захтева и управљања софтверским пројектом	92
Табела 7. Демографска структура одговора (фазе анализе корисничких захтева и управљање софтверским пројектом)	95
Табела 8. Усклађеност GenAI алата са задацима и захтевима студената у процесу учења развоја софтвера (фазе анализе корисничких захтева и управљања софтверским пројектом)	97
Табела 9. Фактори који утичу на задовољство студената GenAI алатима у процесу учења развоја софтвера (фазе анализе корисничких захтева и управљања софтверским пројектом)	99
Табела 10. Спремност студената за усвајање GenAI алата за учење развоја софтвера (фазе анализе корисничких захтева и управљања софтверским пројектом)	101
Табела 11. Преглед GenAI алата у фазама пројектовања, имплементације и тестирања софтвера	110
Табела 12. Демографски подаци учесника (фазе пројектовања, имплементације и тестирања софтвера)	112
Табела 13. Усклађеност GenAI алата са задацима и захтевима студената у процесу учења развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)	113
Табела 14. Фактори који утичу на задовољство студената GenAI алатима у процесу учења развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)	115
Табела 15. Спремност студената за усвајање GenAI алата за учење развоја софтвера (фазе пројектовања, имплементације и тестирања софтвера)	116
Табела 16. Резултати процене знања студената (фазе пројектовања, имплементације и тестирања софтвера)	118
Табела 17. Импликације примене GenAI алата у високом образовању	122
Табела 18. Ограничења истраживања и стратегије за њихово превазилажење	124
Табела 19. Структура упитника - фаза анализе корисничких захтева	152
Табела 20. Структура упитника - фаза управљања софтверским пројектом	153
Табела 21. Структура упитника - фаза пројектовања	154
Табела 22. Структура упитника - фаза имплементације и тестирања	155

Листа акронима

AI -	<i>Artificial Intelligence</i>
ASE -	<i>Agentic Software Engineering</i>
AVM -	<i>Academic eValuation Matrix</i>
CBL -	<i>Challenge-Based Learning</i>
CFS -	<i>Competency Framework for Students</i>
CFT -	<i>Competency Framework for Teachers</i>
CI/CD -	<i>Continuous Integration and Continuous Delivery/Deployment</i>
CNN -	<i>Convolutional Neural Network</i>
CoT -	<i>Chain of Thought</i>
DL -	<i>Deep Learning</i>
DT -	<i>Design Thinking</i>
GAIDE -	<i>Generative AI for Developing Education</i>
GAN -	<i>Generative Adversarial Neural Network</i>
GenAI -	<i>Generative Artificial Intelligence</i>
GPT -	<i>Generative Pretrained Transformer</i>
GRPO -	<i>Group Relative Policy Optimization</i>
GRU -	<i>Gated Recurrent Unit</i>
HTTP -	<i>Hypertext Transfer Protocol</i>
IDE -	<i>Integrated Development Environment</i>
IEEE -	<i>Institute of Electrical and Electronics Engineers</i>
JSON -	<i>JavaScript Object Notation</i>
KA -	<i>Knowledge Area (SWEBOK)</i>
LLM -	<i>Large Language Model</i>
LMS -	<i>Learning Management System</i>
LSTM -	<i>Long Short-Term Memory</i>
ML -	<i>Machine Learning</i>
MLA -	<i>Multi-Head Latent Attention</i>
MoE -	<i>Mixture of Experts</i>
MSF -	<i>Microsoft Solutions Framework</i>
MTP -	<i>Multi-Token Prediction</i>
NLP -	<i>Natural Language Processing</i>
ORM -	<i>Object-Relational Mapping</i>
PBL -	<i>Project-Based Learning</i>
RAG -	<i>Retrieval-Augmented Generation</i>
ReAct -	<i>Reasoning and Acting</i>
RNN -	<i>Recurrent Neural Networks</i>
SE -	<i>Software Engineering</i>
SECDEP -	<i>Software Engineering Curricula Development and Evaluation Process</i>
SEM -	<i>Structural Equation Modeling</i>
SRS -	<i>Software Requirements Specification</i>
SWEBOK -	<i>Software Engineering Body of Knowledge</i>
TAM -	<i>Technology Acceptance Model</i>
TCK -	<i>Technological Content Knowledge</i>
TK -	<i>Technological Knowledge</i>
TPACK -	<i>Technological Pedagogical Content Knowledge</i>
TPK -	<i>Technological Pedagogical Knowledge</i>
TTF -	<i>Task Technology Fit</i>
UI -	<i>User Interface</i>
UML -	<i>Unified Modeling Language</i>
XK -	<i>Contextual Knowledge</i>

Прилог 1 - Структура упитника (анализа корисничких захтева)

Табела 19. Структура упитника - фаза анализе корисничких захтева

Перципирана корисност (енг. <i>Perceived Usefulness</i>)	
1	GenAI алати су значајно допринели повећању продуктивности током дефинисања захтева.
2	GenAI алати су убрзали процес идентификовања и дефинисања захтева.
3	GenAI алати су допринели квалитету дефинисаних захтева (јасноћа, потпуност, конзистентност).
4	Резултати које су генерисали GenAI алати су захтевали мануелне поправке и дораде (нпр. прецизирање захтева).
Перципирана лакоћа коришћења (енг. <i>Perceived Ease of Use</i>)	
5	GenAI алати су интуитивни за коришћење током дефинисања захтева.
6	Било ми је лако да разумем које су поправке/дораде биле потребне код резултата које су генерисали GenAI алати.
7	Била су ми потребна додатна упутства/смернице од наставника како бих користио/ла GenAI алате у контексту анализе захтева.
Перципирана лакоћа учења (енг. <i>Perceived Ease of Learning</i>)	
8	Било ми је лако да постављам <i>prompt</i> -ове који дају квалитетан излаз у контексту анализе захтева.
Задовољство (енг. <i>Satisfaction</i>)	
9	Одговори и препоруке које GenAI алати дају испуњавају моја очекивања у фази анализе захтева.
10	Задовољан/на сам резултатима које GenAI алати пружају током анализе захтева.
11	Препоручио/ла бих коришћење GenAI алата колегама током анализе захтева.
12	Повратна информација од наставника је била задовољавајућа за задатке које сам урадио/ла у фази анализе захтева коришћењем GenAI алата.
Намера коришћења (енг. <i>Intention to adopt</i>)	
13	Намеравам да наставим да користим GenAI алате у фази анализе захтева у будућности.
Карактеристике задатка (енг. <i>Task Characteristics</i>)	
14	GenAI алати ми помажу у идентификовању функционалних захтева.
15	GenAI алати ми помажу у дефинисању нефункционалних захтева.
16	GenAI алати ми помажу у структурирању и организовању захтева.
Карактеристике технологије (енг. <i>Technology Characteristics</i>)	
17	Време генерисања одговора на упит код GenAI алата је одговарајуће.
Индивидуалне карактеристике (енг. <i>Individual Characteristics</i>)	
18	Сигуран/на сам у своју способност да ефикасно користим GenAI алате за академске сврхе.
19	Верујем да ће коришћење GenAI алата побољшати моју академску продуктивност и успех.
20	Када користим GenAI алате, осећам се мотивисаније да завршим свој академски рад.
Усклађеност задатка и технологије (енг. <i>Task Technology Fit</i>)	
21	Довољно ми је да користим GenAI алате за дефинисање захтева без употребе додатних алата.
22	GenAI алати се добро уклапају у процес дефинисања и анализе захтева.
23	GenAI алати се лако интегришу са другим алатима које користим у фази анализе захтева.
Отворена питања	
24	Додатни коментари (за алате коришћене у фази анализе захтева) у вези функционалности, лакоће коришћења, прецизности, интеграције са другим алатима, као и евентуални изазови или предности које сте приметили...
25	Додатне сугестије, коментари, утисци у вези са начином рада на настави

Прилог 2 - Структура упитника (управљање софтверским пројектом)

Табела 20. Структура упитника - фаза управљања софтверским пројектом

Перципирана корисност (енг. <i>Perceived Usefulness</i>)	
1	<i>GenAI</i> алати су ми помогли да ефикасније одредим приоритетне функционалности.
2	<i>GenAI</i> алати су ми помогли да идентификујем функционалности које имају највећу вредност за реализацију софтверског пројекта.
3	<i>GenAI</i> алати су допринели квалитету доношења одлука о избору функционалности.
Перципирана лакоћа коришћења (енг. <i>Perceived Ease of Use</i>)	
4	<i>GenAI</i> алати су интуитивни за коришћење приликом доношења одлука о функционалностима.
5	Било ми је лако да разумем препоруке које су <i>GenAI</i> алати генерисали у вези са избором и приоритизацијом функционалности.
Задовољство (енг. <i>Satisfaction</i>)	
6	Задовољан/на сам препорукама које <i>GenAI</i> алати пружају за избор и приоритизацију функционалности.
Намера коришћења (енг. <i>Intention to adopt</i>)	
7	Намеравам да наставим да користим <i>GenAI</i> алате за доношење одлука о функционалностима у будућности.
Карактеристике задатка (енг. <i>Task Characteristics</i>)	
8	Доношење одлука о приоритетима функционалности представља сложен задатак.
9	Процена вредности функционалности захтева додатну подршку.
Карактеристике технологије (енг. <i>Technology Characteristics</i>)	
10	<i>GenAI</i> алати пружају релевантне информације за доношење одлука о функционалностима.
11	<i>GenAI</i> алати генеришу корисне препоруке за избор и приоритизацију функционалности.
Отворена питања	
12	Додатни коментари (за алате коришћене у фази управљања софтверским пројектом) у вези функционалности, лакоће коришћења, прецизности, интеграције са другим алатима, као и евентуални изазови или предности које сте приметили...
13	Додатне сугестије, коментари, утисци у вези са начином рада на настави

Прилог 3 - Структура упитника (пројектовање софтвера)

Табела 21. Структура упитника - фаза пројектовања

Перципирана корисност (енг. <i>Perceived Usefulness</i>)	
1	GenAI алати су значајно допринели повећању продуктивности током пројектовања.
2	GenAI алати су убрзали процес доношења одлука током пројектовања.
3	GenAI алати су допринели квалитету крајњег производа (дијаграми, описи) током фазе пројектовања.
4	Резултати које су генерисали GenAI алати су захтевали мануелне поправке/дораде и сл.
Перципирана лакоћа коришћења (енг. <i>Perceived Ease of Use</i>)	
5	GenAI алати су интуитивни за коришћење током пројектовања.
6	Било ми је лако да разумем које су поправке/дораде биле потребне код резултата које су генерисали GenAI алати.
7	Била су ми потребна додатна упутства/смернице од наставника како бих користио/ла GenAI алате у контексту пројектовања.
Перципирана лакоћа учења (енг. <i>Perceived Ease of Learning</i>)	
8	Било ми је лако да постављам <i>prompt</i> -ове које дају квалитетан излаз у контексту пројектовања.
Задовољство (енг. <i>Satisfaction</i>)	
9	Одговори и препоруке које GenAI алати дају испуњавају моја очекивања у фази пројектовања.
10	Задовољан/на сам резултатима које GenAI алати пружају током пројектовања.
11	Препоручио/ла бих коришћење GenAI алата колегама током пројектовања.
12	Повратна информација од наставника је била задовољавајућа за задатке које сам урадио/ла у фази пројектовања коришћењем GenAI алата.
Намера коришћења (енг. <i>Intention to adopt</i>)	
13	Намеравам да наставим да користим GenAI алате у фази пројектовања у будућности.
Карактеристике задатка (енг. <i>Task Characteristics</i>)	
14	GenAI алати су ми корисни за израду структуре система (концептуалног модела) у фази пројектовања софтвера.
15	GenAI алати су ми корисни за моделовање динамичког понашања система (UML дијаграми секвенци и случајева коришћења) у фази пројектовања софтвера.
Карактеристике технологије (енг. <i>Technology Characteristics</i>)	
16	GenAI алати су поуздани у фази пројектовања.
17	Време генерисања одговара на упит код GenAI алата је одговарајуће.
Индивидуалне карактеристике (енг. <i>Individual Characteristics</i>)	
18	Сигуран/на сам у своју способност да ефикасно користим GenAI алате за академске сврхе.
19	Верујем да ће коришћење GenAI алата побољшати моју академску продуктивност и успех.
20	Када користим GenAI алате, осећам се мотивисаније да завршим свој академски рад.
Усклађеност задатка и технологије (енг. <i>Task Technology Fit</i>)	
21	Довољно ми је да користим GenAI алате за генерисање структуре система (концептуални модел) у фази пројектовања.
22	Довољно ми је да користим GenAI алате за генерисање динамике система (UML дијаграми) у фази пројектовања.
23	GenAI алати се лако интегришу са другим алатима које користим у фази пројектовања софтвера.
Отворена питања	
24	Додатни коментари (за алате коришћене у фази пројектовања) у вези функционалности, лакоће коришћења, прецизности, интеграције са другим алатима, као и евентуални изазови или предности које сте приметили...
25	Додатне сугестије, коментари, утисци у вези са начином рада на пројекту

Прилог 4 - Структура упитника (имплементација и тестирање софтвера)

Табела 22. Структура упитника - фаза имплементације и тестирања

Перципирана корисност (енг. <i>Perceived Usefulness</i>)	
1	GenAI алати су значајно допринели повећању продуктивности током имплементације.
2	GenAI алати су убрзали процес доношења одлука током имплементације.
3	GenAI алати су допринели квалитету програмског кода током имплементације.
4	GenAI алати су допринели квалитету тестова.
5	Резултати које су генерисали GenAI алати су захтевали мануелне поправке/дораде и сл.
Перципирана лакоћа коришћења (енг. <i>Perceived Ease of Use</i>)	
6	GenAI алати су интуитивни за употребу током имплементације.
7	Било ми је лако да разумем које су поправке/дораде биле потребне код резултата које су генерисали GenAI алати.
8	Била су ми потребна додатна упутства/смернице од наставника како бих користио/ла GenAI алате у контексту имплементације.
Перципирана лакоћа учења (енг. <i>Perceived Ease of Learning</i>)	
9	Било ми је лако да постављам <i>prompt</i> -ове које дају квалитетан излаз у контексту имплементације.
Задовољство (енг. <i>Satisfaction</i>)	
10	Одговори и препоруке које GenAI алати дају испуњавају моја очекивања у фази имплементације.
11	Задовољан/на сам резултатима које GenAI алати пружају током имплементације.
12	Препоручио/ла бих коришћење GenAI алата колегама током имплементације.
13	Повратна информација од наставника је била задовољавајућа за задатке које сам урадио/ла у фази имплементације коришћењем GenAI алата.
Намера усвајања (енг. <i>Intention to adopt</i>)	
14	Намеравам да наставим да користим GenAI алате у фази имплементације у будућности.
Карактеристике задатка (енг. <i>Task Characteristics</i>)	
15	GenAI алати су ми корисни за генерисање програмског кода у фази имплементације софтвера.
16	GenAI алати су ми корисни за тестирање апликације.
Карактеристике технологије (енг. <i>Technology Characteristics</i>)	
17	GenAI алати су поуздани при генерисању програмског кода.
18	GenAI алати су поуздани за тестирање апликације.
19	Време генерисања одговара на упит код GenAI алата је одговарајуће.
Индивидуалне карактеристике (енг. <i>Individual Characteristics</i>)	
20	Сигуран/на сам у своју способност да ефикасно користим GenAI алате за академске сврхе.
21	Верујем да ће коришћење GenAI алата побољшати моју академску продуктивност и успех.
22	Када користим GenAI алате, осећам се мотивисаније да завршим свој академски рад.
Усклађеност задатка и технологије (енг. <i>Task Technology Fit</i>)	
23	Довољно ми је да користим GenAI алате за генерисање програмског кода у фази имплементације.
24	Довољно ми је да користим GenAI алате за тестирање апликације.
25	GenAI алати се лако интегришу са другим алатима које користим у фази имплементације.
Отворена питања	
26	Додатни коментари (за алате коришћене у фази имплементације) <i>у вези функционалности, лакоће коришћења, прецизности, интеграције...</i>
27	Додатни коментари (за алате коришћене у фази тестирања) <i>у вези функционалности, лакоће коришћења, прецизности, интеграције са другим алатима, као и евентуални изазови или предности које сте приметили...</i>
28	Додатне сугестије, коментари, утисци у вези са начином рада на пројекту

Биографија

Милица Симић је рођена 31.08.1999. године у Смедереву. Основну школу је завршила у Лозовику као ђак генерације, а средњу школу у Смедереву. Дипломирала је на Факултету организационих наука (смер - Информациони системи и технологије) 2022. године са просечном оценом 9.16. Одбранила је дипломски рад под називом: „Развој *Salesforce* апликације за управљање односима са студентима“, са оценом 10 (десет). Мастер академске студије, студијски програм Електронско пословање, студијска група Технологије електронског пословања, уписала је на Факултету организационих наука 2022. године, а завршни мастер рад одбранила у септембру 2023. године под називом „Модел електронског пословања базиран на блокчејн технологијама“, са просечном оценом 10.

Милица Симић је од 2021. године ангажована као сарадник ван радног односа – демонстратор. Од 2022. године је запослена као сарадник у настави, а затим и као асистент на Факултету организационих наука на Катедри за електронско пословање. Током свог досадашњег ангажмана, Милица је активно учествовала у бројним пројектима и образовним активностима.

Од јуна 2022. године била је ангажована на изради и одржавању сајтова основних, мастер и докторских академских студија Факултета организационих наука. Као предавач, допринела је Летњој школи „Технологије електронског пословања“ 2022, 2023. и 2024. године. У мају 2024. године била је гостујући предавач на мастер академским студијама Универзитета у Флориди, где је представила теме из области блокчејн технологија. Поред тога, активно је укључена у организацију и реализацију конференција и хакатона, као члан организационих тимова конференције *E-business technologies* (2022, 2023, 2025), *Web3 Algorand Hackathon* (2023), *FON Hakaton - Blockchain Challenge* (2024), *Web3 Hackathon* (2025) као и као стручни ментор на ФОН Хакатону и Хакатону за средњошколце у периоду 2023-2025.

Милица је члан *IEEE* и *IEEE Computer Society* и амбасадор је за такмичење *IEEE Xtreme Programming Competition*. Њен допринос образовању и развоју младих кадрова у ИТ сектору препознат је и у студентским организацијама ФОНИС, *BEST Belgrade* и *AIESEC*, у оквиру којих је одржала низ предавања по позиву. Милица је активно укључена у низ пројеката на пољу информационих технологија и дигиталних иновација. Као члан тима учествује у пројекту *Trophy - The Next Generation of Deep Technology Leaders*, који финансира *EIT HEI*, као и у пројекту *Web3 Innovation Hub (W3IH)*, подржаном од стране *European Education and Culture Executive Agency* у оквиру *ERASMUS* програма. Такође, као члан тима допринела је пројекту *Supercluster Excellent3*, финансираном од стране *USAID*-а. У оквиру програма *UNDP Pre-qualification program* учествовала је као предавач, а на пројекту *UF Blockchain Lab*, финансираном од стране *Algorand* фондације (*ACE-W3*), учествовала је од 2022. до 2024. године.

Списак објављених радова

Међународни часописи (M20):

1. **Simić, M.**, Jolović, M., Naumović, T., Lukovac, P., & Despotović-Zrakić, M. (2026). Integrating GenAI Tools Into Software Engineering Education. *Computer Applications In Engineering Education*, 34(2). <https://doi.org/10.1002/cae.70177>, IF(2025)=3.0, M21
2. Naumović, T., Kostić, D., Barać, D., **Simić, M.**, & Despotović-Zrakić, M. (2026). Redesigning learning pathways: recognizing non-formal competencies through micro-credentials in higher education. *Education + Training*, 1–32. <https://doi.org/10.1108/ET-11-2025-0958>. IF(2025)=3.9, M21a
3. Kendrišić, M., Rodić, B., Labus, A., **Simić, M.**, & Despotović, V. (2025). Exploring the potential for introducing crowdsourced e-health services. *Health Informatics Journal*. 2025;31(3). doi:10.1177/14604582251356208, IF(2024)=2.3, M22
4. Miličević, A., **Simić, M.**, Bogdanović, Z., Despotović-Zrakić, M., & Suvajdžić, M. (2025). Entrepreneurship Education Approaches: Hackathon and Project-Based Learning. *The Journal of East European Management Studies, special issue Entrepreneurship Education in Central and East Europe*, pp. 83-111. Nomos Verlagsgesellschaft mbH & Co. KG. <http://doi.org/10.5771/9783748949602-83>, IF(2023)=0.5, M23
5. **Simić, M.**, Šikman, K., Vrljanac, M., Despotović, V., Despotović-Zrakić, M. (2024). Methodology of creating NFT fashion projects. *Facta Universitatis, Series: Electronics and Energetics* 37(1), pp. 29-52. ISSN: 0353-3670, doi:10.2298/FUEE2401029S, IF(2024)=0.7, M23

Монографија међународног значаја (M10):

1. **Simić, M.**, Despotović-Zrakić, M., Labus, A. (2023). A Methodological Approach for the Promotion of a New Fashion Brand Based on a Digital Marketing Strategy. In: Mihić, M., Jednak, S., Savić, G. (eds) *Sustainable Business Management and Digital Transformation: Challenges and Opportunities in the Post-COVID Era*. SymOrg 2022. Lecture Notes in Networks and Systems, vol 562. Springer, Cham. ISBN: 978-3-031-18644-8 https://doi.org/10.1007/978-3-031-18645-5_16, M14

Радови саопштени на скупу међународног значаја штампани у целини (M30):

1. **Simić, M.**, Marković, R., & Bogdanović, Z. (2026). Design and Implementation of an AI Agent for Automated Analysis of GitHub Repositories Using LangChain. Resilient and Intelligent Organizations in Times of Permanent Disruption, 20th International Symposium SymOrg. University of Belgrade – Faculty of Organizational Sciences. M33
2. Deljanin, T., Jevtić, M., Bjelica, A., & **Simić, M.** (2026). Enhancing Blood Donor Transparency and Motivation through Decentralized Digital Platforms: A Case Study of LifeFlow. Resilient and Intelligent Organizations in Times of Permanent Disruption, 20th International Symposium SymOrg. University of Belgrade - Faculty of Organizational Sciences. M33
3. Joksimović, A., Jolović, M., Lukovac, P., **Simić, M.**, & Despotović-Zrakić, M. (2026) HealthOCR-RAG: A Modular, Privacy-First Framework for the Automated Simplification of Scanned Internal Medicine Reports. *Proceedings of 13th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN)*. https://www.etrans.rs/2026/wp-content/uploads/2026/07/IcETRAN_Proceedings_web.pdf, M33
4. Muljajić, M., **Simić, M.**, Barać, D., Despotović-Zrakić, M., Bogdanović, Z. (2026). Developing Entrepreneurial Competencies and Digital Skills in High School Students Through Project-Based Learning. In: Reis, J.L., Bigne, E., Bogdanović, Z., Santos, J.P.M.d. (eds) *Marketing and Smart*

- Technologies*. ICMarkTech 2025. Smart Innovation, Systems and Technologies, vol 473. Springer, Cham. https://doi.org/10.1007/978-3-032-16284-7_23
5. Joksimović, A., Lukovac, P., **Simić, M.**, & Košević, S. (2025). Smart Apiculture Business Model. *Proceedings of the 33rd International Conference on Information Systems Development*. <https://doi.org/10.62036/isd.2025.155>, M33
 6. Lukovac, P., Jafarzadeh, H., **Simić, M.**, Marković, R. & Jovanović, S. (2025). Design and development of smart beehive solution. *E-Business Technologies Conference Proceedings*, <https://ebt.rs/journals/index.php/conf-proc/article/view/265/219>, M33
 7. Naumović, T., **Simić, M.**, Suvajdžić, M., Vulić, M., & Bjelica, A. (2025). Roles, Responsibilities, and Value Flows: A Stakeholder Map of Micro-Credential Processes in Higher Education Ecosystem. *E-Business Technologies Conference Proceedings*, <https://ebt.rs/journals/index.php/conf-proc/article/view/269>, M33
 8. Mujezinović, A., Joksimović, A., Kostić, D., Jolović, M., **Simić, M.** (2025). Blockchain-Based Platform for Tracking Students' Achievements. *International Scientific Conference on Information Technology, Computer Science, and Data Science*, Sinteza 2025. doi:10.15308/Sinteza-2025-267-273, M33
 9. Jafarzadeh, H., **Simić, M.**, & Barać, D. (2024). The Role of AI in B2B E-business. In *Proceedings of the International Conference on Marketing and Technologies*, ICMarkTech 2024, 5 – 7 December, São Miguel, Azores, M33
 10. Labus, A., Rodić, B., **Simić, M.**, Trpkov, A., & Sovtić, D. (2024). E-government business model based on blockchain: Examining citizens' readiness. In *Proceedings of the International Conference on Marketing and Technologies*, ICMarkTech 2024, 5 – 7 December, São Miguel, Azores, M33
 11. Tomić P. & **Simić, M.** (2024). Development of a Hybrid App based on a REST API. In: M. Kostić-Stanković, I. Mijatović, J. Krivokapić (eds.) *Proceedings of the 19th International Symposium SymOrg*, 12 – 15 June, Zlatibor, Serbia. Retrieved from: <https://symorg.fon.bg.ac.rs/wp-content/uploads/2024/06/Zbornik-SymOrg-2024.pdf>, M33
 12. Kalimančević A., **Simić, M.**, & Radenković, M. (2024). Blockchain Infrastructure for Managing Competences in Higher Education. In: M. Kostić-Stanković, I. Mijatović, J. Krivokapić (eds.) *Proceedings of the 19th International Symposium SymOrg*, 12 – 15 June, Zlatibor, Serbia. Retrieved from: <https://symorg.fon.bg.ac.rs/wp-content/uploads/2024/06/Zbornik-SymOrg-2024.pdf>, M33
 13. Veličković, J., Despotović-Zrakić, M., Naumović, T., **Simić, M.**, & Savićević, M. (2023). Razvoj vizuelnog identiteta podkasta visokoškolske obrazovne ustanove. *XIV Skup privrednika i naučnika*, SPIN'23, M33
 14. Lukovac, P., **Simić, M.**, Duranović, M., Radenković, M., & Labus, A. (2023). Blockchain-based C2C Business Models in a Smart City. In *Marketing and Smart Technologies: Proceedings of ICMarkTech 2023*, November 30 – December 02 2023, Prague, Czech Republic, Volume 1, Springer, M33
 15. **Simić, M.**, Despotović, V., Suvajdžić, M., Naumović, T., & Bogdanovic, Z. (2023). An e-business model for the fashion industry based on blockchain technologies and NFTs. In J. L. Reis, M. Peter, L. P. Reis, & Z. Bogdanović (eds.), *Marketing and Smart Technologies: Proceedings of ICMarkTech* (Vol. 2). Springer, M33
 16. Miletić A., Stamenković R., Barać D., Despotović-Zrakić M., & **Simić M.** (2023). Extending Marketplace Business Model in Academia: A Case of FONBoard. In Van Sinderen, M. Hammoudi, S., & Wijnhoven, F. (eds.) *Proceedings of the 20th International Conference on Smart Business Technologies*, ICSBT 2023, Rome, Italy, July 11-13, 2023. (pp. 66-72), ISBN: 978-989-758-667-5, 11-13. July, Italy, <https://www.scitepress.org/Link.aspx?doi=10.5220/00120939000003552>, M33
 17. Labus, A., Barać, D., Lukovac, P., Despotović, V., & **Simić, M.** (2023). Modeling Blockchain System for Fashion Industry. In Rocha, A., Adeli, H., Dzemyda, G., Moreira, F., Colla, V. (eds.)

Information Systems and Technologies. WorldCIST'23. Lecture Notes in Networks and Systems, vol 801. Springer, Cham. ISBN: 978-3-031-45647-3, <https://doi.org/10.1007/978-3-031-45648-0>

18. Lukovac, P., **Simić, M.**, & Radenković, B. (2023). IoT system for smart beekeeping. In: M. Despotović-Zrakić, Z. Bogdanović, A. Labus, D. Barać, B. Radenković (eds.) *E-business technologies conference proceedings* (Vol. 3, No. 1, pp. 159-162), June 2023., Belgrade, Serbia. ISBN: 978-86-7680-427-6. Retrieved from: <https://www.ebt.rs/journals/index.php/conf-proc/article/view/194/137>, M33
19. Vrljanac, M., Šikman, K., **Simić, M.**, Naumović, T., & Despotović-Zrakić, M. (2023). Application of 3D Modeling in the Fashion Industry. In: M. Despotović-Zrakić, Z. Bogdanović, A. Labus, D. Barać, B. Radenković (eds.) *E-business technologies conference proceedings* (Vol. 3, No. 1, pp. 176-184), June 2023., Belgrade, Serbia. ISBN: 978-86-7680-427-6. Retrieved from: <https://www.ebt.rs/journals/index.php/conf-proc/article/view/172/140>, M33
20. **Simić, M.**, Despotović, V., Jović, M., Labus, A. and Despotović-Zrakić, M. (2022). The methodological approach to introducing a new fashion brand to the digital market. In: M. Despotović-Zrakić, Z. Bogdanović, A. Labus, D. Barać, B. Radenković (eds.) *E-business technologies conference proceedings* (Vol. 2, No. 1, pp. 14-20), June 2022., Belgrade, Serbia. ISBN: 978-86-7680-414-6. Retrieved from: <https://www.ebt.rs/journals/index.php/conf-proc/article/view/109/58>, M33

Радови саопштени на скупу међународног значаја штампани у изводу (M34):

1. Kostić, D., **Simić, M.**, & Marković, R. (2025). Software testing leveraging AI. *11th User Conference on Advanced Automated Testing - UCAAT*, 1-3 April 2025., Thessaloniki, Greece
2. **Simić, M.**, Naumović, T., Lukovac, P., Kostić, D., & Barać, D. (2024). Application of Generative AI tools in Software Development. *Artificial Intelligence Conference* (pp. 124-125), Belgrade, Serbia, December 2024. Retrieved from: https://www.mi.sanu.ac.rs/~ai_conf/previous_editions/2024/AI_Conference_Book_of_Abstracts.pdf
3. Miličević A., **Simić M.**, Bogdanović Z., Despotović-Zrakić M. & Suvajdžić M (2023). Exploring Pedagogical Pathways: Hackathon and Project-Based Learning (PBL) in Entrepreneurship Education. *2nd International Danube Cup Conference on Entrepreneurship Education*, 24-25 November, Belgrade, Serbia
4. Šikman, K., Vrljanac, M., **Simić, M.**, Suvajdžić M., & Despotović-Zrakić, M. (2023). Application of Virtual Reality in Fashion Design. In: A. Ionescu & O. Gkounta (eds.) *Atiner23, 4th Annual International Conference on Computer & Software Engineering*, 10-13 July 2023., Athens, Greece. ISBN: 978-960-598-593-6. <https://www.atiner.gr/abstracts/2023ABST-SFW.pdf>
5. **Simić, M.**, Despotović-Zrakić, M., Labus, A. (2022). Digital marketing in the promotion of a new fashion brand. In: M.Mihić, S.Jednak, G.Savić (eds.) *Sustainable Business Management and Digital Transformation: Challenges and Opportunities in the Post-COVID Era*, 18th International Symposium SymOrg (pp. 188-189), ISBN: 978-86-7680-411-5, 11-14 June, Belgrade, Serbia, <https://symorg.fon.bg.ac.rs/wp-content/uploads/2022/07/Book-of-Abstracts-Symorg2022.pdf>

Радови у часописима од националног значаја (M50):

1. Ignjatović, S., Kostić, D., Naumović, T., **Simić, M.**, & Joksimović, A. (2026). Rethinking Project Management for Decentralized Digital Product Development. *European Project Management Journal*, 16(1), 49-66. DOI: 10.56889/qose7846, M51
2. Antić, N., Jolović, M., Lukovac, P., **Simić, M.**, & Bogdanović, Z. (2025). Project-Based Learning in Lifelong Education. *European Project Management Journal*, 15(2), 69-85. DOI: 10.56889/eqwg6335, M51

Радови саопштени на скупу националног значаја штампани у целини (M60):

1. Kostić, D., Naumović, T., **Simić, M.**, & Janković, M. (2026). Integracija EBSI platforme u procese verifikacije neformalnog učenja u visokom obrazovanju. *The 2nd Conference on Mathematics and Computer Science Teaching*, TEMATCOM 2026, Kragujevac, Serbia: https://imi.pmf.kg.ac.rs/TEMATCOM2026/assets/Knjiga_apstrakata_TEMATCOM2026_B5_ELEKTRONSKI.pdf, M61
2. Radosavljević, L., **Simić, M.**, Joksimović, A., Naumović, T., & Despotović-Zrakić, M. (2025). Enhancing e-government services through chatbot development using Azure OpenAI. *Ivannikov Memorial Workshop*.
3. Rodić, B., Labus, A., Despotović-Zrakić, M., Lukovac, P., & **Simić, M.** (2023) An ecosystem for smart cities based on blockchain. In: D. Bećirović, D. Arnaut (eds.) *DIEC 2023, 6th International Scientific Conference on Digital Economy*, 25. August 2023., Tuzla, Bosnia and Herzegovina. ISSN: 2566-4514. <https://ipi-akademija.ba/file/diec-6-online/189>, M61

Уџбеници:

Милица Симић је коаутор уџбеника који је предвиђен за коришћење као основна литература на неколико предмета на основним и мастер академским студијама на Факултету организационих наука:

1. Radenković M., Despotović-Zrakić M., Bogdanović Z., Barać D., Labus A., **Simić M.** (2024). *Razvoj softvera orijentisanog na procese, odabrana poglavlja: DevOps, Mikroservisi, Kontejneri. II Izmenjeno i dopunjeno izdanje*. ISBN: 978-86-7680-458-0. Retrieved from: <https://elab.fon.bg.ac.rs/wp-content/uploads/2024/06/Razvoj-softvera-orijentisanog-na-procese-udzbenik.pdf>

Изјава о коришћењу генеративне вештачке интелигенције

У складу са Општим смерницама за коришћење генеративне вештачке интелигенције на Факултету организационих наука Универзитета у Београду [235], приликом израде ове докторске дисертације коришћени су алати генеративне вештачке интелигенције као подршка у формулисању и језичком унапређењу појединих делова текста, структурирању садржаја и анализи литературе. Сви резултати добијени употребом ових алата критички су размотрени, проверени и кориговани од стране аутора. Аутор у потпуности преузима одговорност за тачност, оригиналност, академски интегритет и коначни садржај дисертације.

Изјава о ауторству

Име и презиме аутора: Милица Симић

Број индекса: 5081/2023

Изјављујем

да је докторска дисертација под насловом

Модел пројектно-заснованог учења у области развоја софтвера заснован на примени генеративне вештачке интелигенције

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, _____

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: **Милица Симић**

Број индекса: **5081/2023**

Студијски програм: **Електронско пословање и софтверско инжењерство**

Наслов рада: **Модел пројектно-заснованог учења у области развоја софтвера заснован на примени генеративне вештачке интелигенције**

Ментор **проф. др Зорица Богдановић**

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањења у **Дигиталном репозиторијуму Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива докторанаука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, _____

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Модел пројектно-заснованог учења у области развоја софтвера заснован на примени генеративне вештачке интелигенције

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (*Creative Commons*) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

③ Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.

Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, _____

1. **Ауторство.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.
2. **Ауторство – некомерцијално.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.
3. **Ауторство – некомерцијално – без прерада.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.
4. **Ауторство – некомерцијално – делити под истим условима.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.
5. **Ауторство – без прерада.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.
6. **Ауторство – делити под истим условима.** Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.