УНИВЕРЗИТЕТ У ПРИШТИНИ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
КОСОВСКА МИТРОВИЦА

УНИВЕРЗИТЕТ У ПРИШТИНИ
Бр. 26-85/2
13 FEB 2026 год.
ПРИШТИНА

УНИВЕРЗИТЕТ У ПРИШТИНИ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
КОСОВСКА МИТРОВИЦА

ПРИМЉЕНО: 13.02.2026			
ОРГ ЈЕДИН.	БРОЈ	ПРИЛОГ	ВРЕДНОСТ
	938/1		

НАСТАВНО-НАУЧНОМ ВЕЋУ
ФАКУЛТЕТА ТЕХНИЧКИХ НАУКА У КОСОВСКОЈ МИТРОВИЦИ

На седници одржаној 26.01.2026. године, наставно-научно веће Факултета техничких наука у Косовској Митровици, на основу чл. 55. ст. 1. тч. 16. Статута Факултета техничких наука и чл. 42. ст. 1. Правилника о докторским студијама Факултета техничких наука, донело је Одлуку бр. 49/3-12 од 26.01.2026. године, којом се утврђује предлог Комисије за оцену научне заснованости теме докторске дисертације „***Одређивање поузданости и верификација софтверских метрика у скрипт програмским језицима***“ и подобности кандидата **Јована Милутиновића** у саставу:

1. проф. др Ненад Јовановић, редовни професор Факултета техничких наука у Косовској Митровици, ужа научна област рачунарска техника и информатика – председник
2. проф. др Петар Милић, ванредни професор Факултета техничких наука у Косовској Митровици, ужа научна област рачунарска техника и информатика – члан-ментор
3. доц. др Алдина Авдић, доцент Државног универзитета у Новом Пазару, ужа научна област рачунарска техника и информатика – члан.

На основу приложене документације уз пријаву докторске дисертације, образложења теме, научних и стручних радова и увидом у целокупну документацију и научно-истраживачку делатност кандидата, наставно-научном већу Факултета техничких наука у Косовској Митровици Комисија подноси:

ИЗВЕШТАЈ О НАУЧНОЈ ЗАСНОВАНОСТИ ТЕМЕ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Кандидат **Јован Милутиновић**, мастер инжењер електротехнике и рачунарства, за израду докторске дисертације предложио је тему „***Одређивање поузданости и верификација софтверских метрика у скрипт програмским језицима***“. Предложена тема докторске дисертације припада научној области Електротехничко и рачунарско инжењерство у оквиру образовно-научног поља техничко-технолошких наука за коју је Факултет техничких наука у Косовској Митровици акредитован.

Биографски подаци

Кандидат **Јован Милутиновић**, рођен је 15.06.1999. године у Косовској Митровици. Основну школу „Вук Караџић“ завршио је у Звечану а средњу техничку школу на смеру „техничар друмског саобраћаја“ завршио је у Косовској Митровици.

Након завршене средње школе, школске 2018/2019 уписао је Факултет техничких наука Универзитета у Приштини са привременим седиштем у Косовској Митровици, на студијском програму Електротехничко и рачунарско инжењерство, модул Рачунарство и информатика. На истом факултету је дипломирао 2022. године, са просечном оценом 8.02 стекавши звање дипломирани инжењер електротехнике и рачунарства. Исте године уписује и мастер академске студије такође на Факултету техничких наука Универзитета у Приштини са привременим седиштем у Косовској Митровици, које завршава 2023. године са просечном оценом 10,00 стекавши звање мастер инжењер електротехнике и рачунарства.

Школске 2023/2024 године уписао је докторске академске студије на Факултету техничких наука Универзитета у Приштини са привременим седиштем у Косовској Митровици на студијском програму Електротехничко и рачунарско инжењерство. На докторским студијама је положио све предмете предвиђене наставним планом и програмом са просечном оценом 10.

У раније периоду радио је као наставник математике у:

- Основној школи „Вук Караџић“ у Звечану и
- Средњој медицинској школи „Медицинска школа са домом ученика“ у Косовској Митровици.

Аутор је и коаутор научних радова објављених у међународним часописима као и на међународним конференцијама. Области интересовања кандидата **Јована Милутиновића** су софтверско инжењерство, блокчејн технологије, веб технологије, машинско учење.

Подобност кандидата

На основу приложене документације о научно-истраживачком раду, биографским и библиографским подацима, Комисија је утврдила да кандидат **Јован Милутиновић**, студент докторских академских студија Електротехничког и рачунарског инжењерства на Факултету техничких наука Универзитета у Приштини са привременим седиштем у Косовској Митровици испуњава све услове за одобравање теме и израду докторске дисертације.

Кандидат Јован Милутиновић је положио све испите предвиђене наставним планом студијског програма докторских студија на одсеку Електротехничко и рачунарско инжењерство, чиме је испунио све услове и стекао право да пријави тему докторске дисертације.

Као аутор и коаутор објавио је више научних радова из уже научне области из које се пријављује тема докторске дисертације. Научно-стручна активност кандидата верификована је кроз објављене радове, како се наводи у наставку:

- 1 рад категорије М21
- 1 рад категорије М53 и
- 1 рад категорије М34.

Списак објављених радова кандидата:

Категорија М21:

1. **Milutinović, J.**, Milić, P., Milenković, V., Mekić, E., Matović, A. (2025). "Application of Blockchain Technologies in Verification of Software Metrics ", Applied Sciences, Vol. 15, No. 10, pp. 5519, DOI= 10.3390/app15105519, ISSN: 2076-3417, IF = 2.5.

Категорија М53:

1. **Milutinović, J.**, Milić, P., Vučetić, S. (2024). "Wireframe modelling of web based open data applications", Scientific Publications of the State University of Novi Pazar Series A: Applied Mathematics, Informatics and Mechanics, Vol. 16, No. 2, pp. 111 - 123. DOI = 10.46793/SPSUNP2402.111M.

Категорија М34:

1. **Milutinović, J.**, Milić, P., Vučetić, S. (2024). Modelling open data based web applications, In the Book of abstracts of the VIII International Scientific Conference "Contemporary Problems of Mathematics, Mechanics and Informatics", Novi Pazar, 55, 03 - 05. June 2024.

На основу наведених чињеница Комисија закључује да кандидат испуњава све услове и да је ПОДОБАН да настави израду докторске дисертације.

Научна заснованост предложене теме

Предмет и циљ докторске дисертације:

Софтверске метрике представљају квантитативне показатеље за процену квалитета кода, његове сложености, одрживости и потенцијала за настанак дефеката. Иако је развијен велики број софтверских метрика у претходним деценијама, остаје нерешено питање њихове поузданости и међуалатне конзистентности када се примењују на скрипт програмске језике. Пошто су скрипт програмски језици динамичке природе, то

може довести до варијација у вредностима софтверских метрика у зависности од алата који их израчунава, начина анализе, као и разлика у имплементацији алгоритама за израчунавање софтверских метрика и начина на који поједини алати интерпретирају структуру програма. За разлику од статички типизираних програмских језика, код скрипт програмских језика структурне информације о програму често нису потпуне у моменту статичке анализе. Као пример може се узети динамичка додела типова, имплицитно проширење објеката, динамичко креирање функција као и варијације у организацији програма које зависе од начина на који се код пише и организује. На основу претходно наведених својстава скрипт програмских језика, доводи се у питање поузданост и тачност мерења софтверских метрика, као и степен сагласности резултата између различитих алата за израчунавање софтверских метрика у скрипт програмским језицима.

На основу наведених проблема, предмет истраживања ове докторске дисертације биће фокусиран на свеобухватну формалну и емпиријску анализу поузданости софтверских метрика у скрипт програмским језицима. Ово истраживање обухватиће формалну и емпиријску анализу поузданости софтверских метрика у скрипт програмским језицима кроз испитивање њихових аксиоматских својстава, логичке доследности, као и ограничења која произилазе из динамичке природе скрипт програмских језика. Из претходно наведног следи да је централни проблем ове докторске дисертације формална верификација поузданости софтверских метрика у условима непотпуне и динамички одређене структуре програмског кода у скрипт програмским језицима.

Истраживање које ће се спровести у оквиру ове докторске дисертације, има за циљ да развије формални верификациони модел који ће на систематичан, математички и логички начин да дефинише услове, поступак и критеријуме за процену исправности и поузданости софтверских метрика у скрипт програмским језицима. Поред развоја формалног верификационог модела, ово истраживање обухвата и емпиријску евалуацију поузданости софтверских метрика кроз примену различитих алата за анализу софтверског кода на скупу пројеката (програма) писаних у скрипт програмским језицима. Применом емпиријске евалуације омогућиће се директно поређење вредности софтверских метрика добијених применом различитих алата, а све то са једним циљем да се испита степен међуалатне конзистентности, стабилност резултата као и осетљивост на различите структурне карактеристике посматраног кода.

Комбинацијом формалног верификационог модела и емпиријске евалуације, дефинисаће се научно утемељена методологија за процену поузданости софтверских метрика као и могућност њихове примене у скрипт програмским језицима.

Основне хипотезе:

На основу постављеног предмета и циља истраживања у оквиру ове докторске дисертације, односно ограничења у мерењу квалитета софтверског кода у скрипт програмским језицима, дефинисан је скуп хипотеза. Дефинисани скуп хипотеза је

одређен тако да директно произилази из предложеног формалног верификационог модела. Он се састоји од следећих хипотеза:

- Формални верификациони модел омогућава класификацију софтверских метрика на поуздане и непоуздане – Ова хипотеза се заснива на формалном дефинисању скупа аксиоматских својстава која софтверске метрике морају да задовоље да би се сматрале поузданим. Софтверске метрике ће бити класификоване као поуздане уколико задовоље сва дефинисана аксиоматска својства као што су дефинисаност, стабилност, међуалатна конзистентност и друге, док ће софтверске метрике које не задовољавају једно или више својстава бити класификоване као непоуздане.
- Софтверске метрике морају да задовоље формални верификациони модел у циљу постизања конзистентности, прецизности, стабилности и фалсификабилности – Ова хипотеза ће бити тестирана кроз проверу задовољености појединачних аксиоматских својстава дефинисаних формалним верификационим моделом. За сваку анализирану софтверску метрику биће утврђено да ли испуњава дефинисане формалне услове у погледу конзистентности, прецизности, стабилности и фалсификабилности. Испуњеност ових услова биће формално проверена, а резултати ће бити повезани са емпиријским утврђеним одступањима у измереним вредностима, чиме ће се утврдити у којој мери испуњеност формалног верификациног модела представља предуслов за поуздану и фалсификабилну примену софтверских метрика.
- Применом различитих алата за анализу софтверског кода добијају се вредности софтверских метрика која показују мерљива одступања – Ова хипотеза ће се проверити спровођењем емпиријске анализе резултата софтверских метрика добијених применом више алата за анализу истог скупа софтверски пројеката написаних у скрипт програмским језицима. За сваку софтверску метрику ће бити прикупљење вредности израчунате различитим алатима, након чега ће бити извршено њихово систематско поређење. Мерљива одступања ће бити квантитативно оцењена применом одговарајућих статистичких мера варијабилност и корелације, чиме ће се утврдити степен међуалатне конзистентности.

Методе истраживања:

Истраживање које ће бити спроведено у оквиру ове докторске дисертације је засновано на примени формалних и емпиријских метода а све са циљем систематске верификације поузданости и валидности софтверских метрика у скрипт програмским језицима.

Први корак у оквиру овог истраживања је формална анализа постојећих софтверских метрика, при чему се софтверске метрике посматрају као математичке функције дефинисане над структуром програмског кода. На основу тога, развија се формални верификациони модел који омогућава прецизно дефинисање критеријума исправности софтверских метрика кроз аксиоматска својства, као што су дефинисаност,

инваријантност, стабилност, монотоност, међуалатна конзистентност и фалсификабилност. У оквиру истраживања, појмови дефинисаност, стабилност, монотоност, инваријантност, фалсификабилност и међуалатна конзистентност софтверских метрика биће додатно дефинисани и операционализовани. Дефинисаност софтверске метрике означава да је метрика формално, недвосмислено и потпуно дефинисана за све елементе домена над којима се примењује, односно да за сваки програмски артефакт постоји јединствено одређена вредност метрика, без неодређених, недефинисаних случајева. Поузданост софтверске метрике се дефинише као степен до ког метрика доследно мери оно што је намењено, независност од алата. Стабилност представља способност софтверске метрике да даје конзистентне резултате при структурно еквивалентним трансформацијама програмског кода. Међуалатна конзистентност означава степен сагласности резултата исте софтверске метрика када се примењује на исти скуп програма различитим алатима за анализу кода. Инваријантност се дефинише као напреднији облик стабилности, при чему се захтева да вредност метрике остане потпуно непромењена у присуству строго дефинисаних ирелевантних трансформација кода. Монотоност описује својство метрика да на контролисано повећање или смањење структурне сложености програма реагује у очекиваном и усмереном правцу, без контрадикторних или нелогичних промена вредности. Фалсификабилност се дефинише као могућност да се на основу експерименталних података и формално дефинисаних сценарија, идентификују конкретни случајеви у којима метрика не испуњава једно или више аксиоматских својстава. Формалном анализом омогућава се систематско упоређивање и категоризација софтверских метрика, без ослањања на специфичне имплементације у алатима за анализу кода.

У другом кораку спровешће се емпиријска анализа која представља централни део истраживања у овом раду. Циљ емпиријске анализе је да испита у којој мери се формално идентификована својства софтверских метрика реализују у пракси. У том контексту, емпиријска анализа се не ограничава на један алат или програмски код, већ обухвата више алата за анализу кода и различите софтверске пројекте написане у скрипт програмским језицима. У емпиријској анализи, резултати софтверских метрика се прикупљају и упоређују систематски.

У трећем кораку, вршиће се статистичка обрада резултата емпиријске анализе која омогућава објективно мерење стабилности, монотоности, дефинисаности, поузданости, инваријантности и међуалатне конзистентности софтверских метрика. У складу са постављеним формалним верификационим моделом, аксиоматска својства софтверских метрика су директно повезана са одговарајућим аналитичким и статистичким методама, чиме се обезбеђује јасна, проверљива и методолошки строга операционализација сваког аксиома. Статистичке методе које ће се користити у оквиру овог истраживања су:

- Pearson-ова корелација
- Interclass Correlation Coefficient
- Bland-Altman анализа
- Standard Error of Measurement (SEM)
- Coefficient of Variation (CV)

У складу са постављеним верификационим моделом, аксиоматска својства софтверских метрика су директно повезана са одговарајућим аналитичким и статистичким методама, чиме се обезбеђује јасна, проверљива и методолошки строга операционализација сваког аксиома. Аксиом дефинисаности се верификује применом формалне анализе, у оквиру које се софтверка метрика третира као математичка функција са прецизно дефинисаним доменом и кодоменом, чиме се утврђује постојање јединствене и недвосмислене вредности за сваки валидан улаз. Инваријантност, монотоност и фалсификабилност испитују се анализом осетљивости, која омогућава систематско праћење реакције софтверских метрика на констролисане и структурно еквивалнтне трансформације програмског кода, чиме се формално утврђује да ли метрика реагује искључиво на семантички релеентне промене и дали је подложна емпиријском оповргавању. Аксиом стабилност се додатно квантификује применом кеофицијента варијације и стандардне грешке мерења, што омогућава раздвајање системтских и случајних одступања у измереним вредностима и објективну процену. Међуалатна конзистентност се испитује применом метода ICC, Bland-Altman, Pearson и CV, чиме се обезбеђује вишедимензионална процена степена сагласности резултата добијених различитим алатима. На крају, класификација софтверских метрика на поуздане и непоуздане се спроводи синтезом резултата добијених провером свих аксиома, чиме се обезбеђује формално утемељен, репродуктиван и научно верификован критеријум за процену њихове применљивости у скрипт програмским језицима.

У оквиру овог истраживања, експериментални оквир представља практичну реализацију формалног верификационог модела и омогућава систематско испитивање поузданости софтверских метрика у скрипт програмским језицима. Експериментални оквир је дизајниран тако да обухвати све аспекте анализе, почевши од избора софтверских метрика, алата за анализу софтверског кода, програма чији ће се код анализирати, до примене статистичких метода валидације.

Дефинисани скуп софтверских метрика које ће бити предмет анализе:

- Цикломатска комплексност
- Дубина наслеђивања
- Индекс одрживости
- Халштедове метрике

Пре саме селекције софтверских метрика чија ће се дефинисаност, монотоност, инваријантност, фалсификабилност, стабилност и међуалатна конзистентност испитивати, извршена је идентификација постојећих софтверских метрика кроз анализу радова који су базирани на софтверским метрикама. Након идентификације, одабир софтверских метрика је базиран на њиховој способности да измере различите аспекте квалитета софтверског кода у скрипт програмским језицима, као и на практичној применљивости у више алата за анализу кода.

Применљивост одабраних софтверских метрика на скрипт програмске језике захтева посебно прилагођавање због специфичних динамичких особина скрипт програмских језика. Тако на пример, цикломатска комплексност је ограничена у смислу

детекције грана и услова који се креирају динамички, што може довести до погрешне процене сложености контролних токова, дубина наслеђивања, као софтверска метрика која мери хијерархју класа, може бити непотпуна јер статичка анализа не препознаје динамички додате наследне релације или имплицитно проширене објекте. Индекс одрживости који се ослања на број линија кода, сложеност функција и дубину наслеђивања, може погрешно проценити одрживост јер не узима у обзир динамичке модификације кода које утичу на његову стварну сложеност и одрживост. Халштедове метрике, које су засноване на броју оператора и операнада, могу бити погрешно процењене уколико се током извршавања генеришу нови елементи који статичком анализатору нису видљиви.

Након избора софтверских метрика, следи избор алата за анализу софтверског кода. За ово истраживање биће коришћено више алата који подржавају анализу скрипт програмских језика и омогућавају израчунавање свих наведених софтверских метрика. Већи број алата за анализу софтверског кода омогућава идентификацију потенцијалних разлика које могу настати услед различитих алгоритама мерења, интерпретација синтаксе и обраде структурних елемената кода. Са аспекта избора програмских језика, акценат је стављен на најпопуларније скрипт програмске језике са широм применом у индустрији и академским истраживањима, који пружају подршку у алатима за статичку и анализу софтверског кода. За потребе овог истраживања, користиће се следећи програмски језици:

- *Python*
- *JavaScript*
- *Ruby*
- *PHP*

Укључивање више скрипт програмских језика у истраживачки оквир представља кључни методолошки корак за проверу применљивости предложеног формалног верификационог модела и његову генерализованост. Анализом софтверских метрика у језицима као што су *Python*, *JavaScript*, *Ruby* и *PHP* омогућава се испитивање утицаја језичких специфичности, укључујући динамичку типизацију, имплицитно проширење објеката, динамичко генерисање програмских структура на стабилност, поузданост и међуалатну конзистентност.

Након избора скрипт програмских језика, у оквиру овог истраживања је прво извршена идентификација постојећих алата за статичку анализу софтверског кода. Избор алата за статичку анализу кода је извршен на основу њихове распрострањености у пракси, подршке скрипт језицима који ће се користити у оквиру овог истраживања и способности да израчунају упоредиве и формално дефинисане софтверске метрике. Посебан акценат је стављен на алате који су широко прихваћени у индустријском и истраживачком окружењу. За потребе овог истраживања, одабрани су следећи алати:

- *SonarQube (Python, JavaScript, Ruby)*
- *Escomplex (JavaScript)*

- *MetricFu (Ruby)*
- *Radon (Python)*
- *Lizard (Python, JavaScript)*
- *Understand (SciTools) (Python)*
- *JSHint (JavaScript)*
- *PHPdepend (PHP)*
- *PhpMetrics (PHP)*
- *SourceMeterп (Python, JavaScript)*
- *Pyreverse (Python)*

Очекивани резултати и научни допринос:

Након спровођења овог истраживања, очекују се резултати који ће омогућити дубље и прецизније разумевање поузданости софтверских метрика када се примењују у скрипт програмским језицима. Тако добијени резултати ће омогућити лако разграничавање теоријских и практичних узрока варијација у добијеним вредностима. Пошто је ово истраживање усмерено ка испитивању поузданости софтверских метрика односно међуалатне конзистентности, очекује се да резултати покажу да непоузданост и међуалатна неконзистентност нису искључиво последица различитих имплементационих приступа у алатима већ да у значајној мери произилази из формалних ограничења и недовољно прецизно дефинисаних својстава самих софтверских метрика. Осим тога, анализом постојећих алата, уочено је да они пружају непотпун скуп софтверских метрика које се могу анализирати. Са ограниченим скупом софтверских метрика, формални верификациони модел је такође ограничен на мали број софтверских метрика које може анализирати. Како би формални верификациони модел пружио што дубљи увид у поузданост постојећих софтверских метрика, развијен је софтверски алат који омогућава ширу примену верификационог модела, односно алат који обезбеђује добијање резултата за широк спектар софтверских метрика у скрипт програмским језицима. Важно је нагласити да овај алат представља инструмент за спровођење емпиријске валидације. Овај алат имплементира блокчејн технологије у складиштењу и верификацији резултата анализе софтверских метрика софтвера писаним у скриптним језицима чиме се смањује ризик од манипулације подацима и повећава поверење у извршене анализе. Прецизније, како би се обезбедила додатна сигурност и поверење у резултате анализираног кода, развијени софтверски алат имплементира блокчејн технологије искључиво као механизам за складиштење и верификацију израчунатих вредности софтверских метрика.

Поред претходно наведеног, очекује се да најзначајнији резултат овог истраживања буде формални верификациони модел који ће омогућити систематску

анализу софтверских метрика независно од конкретних алата и скрипт програмских језика. Овај модел ће формално дефинисати шта се сматра валидним улазом за софтверску метрику, који су неопходни предуслови за њено конкретно израчунавање, као и која аксиоматска својства софтверска метрика мора да задовољи да би се могла сматрати поузданом. Оригинални научни допринос ове докторске дисертације огледа се у развоју формалног верификационог модела за процену поузданости софтверских метрика у скрипт програмским језицима, који омогућава систематску, логички доследну и фалсификабилну анализу софтверских метрика. Очекује се да ће развијен формални верификациони модел омогућити идентификацију ситуација у којима софтверска метрика не испуњава основне услове логичке доследности или показује осетљивост на структурне варијације које нису релевантне за оно што софтверска метрика настоји да мери. Како би се формални верификациони модел утемељио, важно је успостављање јасне везе између аксиоматских својстава софтверских метрика и њихове емпиријске утврђене стабилности и међуалатне конзистентности.

Оквирни опис садржаја дисертације:

Докторска дисертација ће бити структурирана у седам повезаних поглавља, где ће свако поглавље имати специфичну улогу у развоју предложене методологије. Након седмог поглавља, биће наведене референце које ће се користити у дисертацији.

У уводном поглављу ће бити представљен значај софтверских метрика као кватитавних показатеља квалитета софтвера, проблем поузданости, утицај динамичке природе скрипт језика на вредности софтверских метрика, предмет и циљ истраживања, као и хипотезе.

Кроз друго поглавље ће бити представљене теоријске основне софтверских метрика и карактеристике скрипт програмских језика. Биће разматрани појам, класификација и историјски развој софтверских метрика. Поред тога, биће представљен и начин израчунавања. Такође, ово поглавље укључује и опис основних особина скрипт програмских језика (као што су динамичка типизација, имплицитно проширење објеката, динамички креирање структура), као и утицај наведених особина на тачност, стабилност и интерпретацију софтверских метрика.

У трећем поглављу ће бити дефинисана поузданост софтверских метрика са акцентом на међуалатну конзистентност, стабилност и осетљивост софтверских метрика, као и аксиоматска својства софтверских метрика. Поред тога, већи део поглавља биће посвећен методолошком оквиру који чине формални верификациони модел, емпиријска евалуација и статистичка анализа резултата.

У четвртом поглављу ће детаљно бити представљен централни методолошки део докторске дисертације. Другим речима, биће детаљно описан формални верификациони модел, аксиоматска својства и услови које софтверске метрике морају да задовоље, емпиријска евалуација кроз примену различитих лата, као и статистичке методе.

У петом поглављу ће бити извршена детаљна анализа добијених резултата која има за циљ да идентификује поуздане и непоуздане софтверске метрике, да

идентификује обрасце у одступањима између алата. Поред тога, ово поглавље обухватиће и дискусију о ограничењима постојећих алата и софтверских метрика.

У шестом поглављу биће представљен развијени софтверски систем и концепт примене блокчејн технологије у контексту верификације резултата софтверских метрика. Биће детаљно описана архитектура система, његова функционалност, подршка скрипт програмским језицима и различитим софтверским метрикама, као и начин интеграције блокчејн технологије.

Седмо поглавље је закључак. Ово поглавље ће обухватити резиме кључних резултата истраживања, теоријске и практичне доприносе и ограничења предложеног приступа.

Оквирна литература:

1. Fenton, N. E., & Neil, M. (2000, May). Software metrics: roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 357-370). DOI:https://doi.org/10.1145/336512.336588.
2. Lincke, R., Lundberg, J., & Löwe, W. (2008, July). Comparing software metrics tools. In Proceedings of the 2008 international symposium on Software testing and analysis (pp. 131-142). DOI:https://doi.org/10.1145/1390630.1390648.
3. Honglei, T., Wei, S., & Yanan, Z. (2009, December). The research on software metrics and software complexity metrics. In 2009 International Forum on Computer Science-Technology and Applications (Vol. 1, pp. 131-136). IEEE. DOI:https://doi.org/10.1109/IFCSTA.2009.39.
4. Singh, G., Singh, D., & Singh, V. (2011). A study of software metrics. IJCEM International Journal of Computational Engineering & Management, 11(2011), 22-27.
5. Bhardwaj, M., & Rana, A. (2016). Key software metrics and its impact on each other for software development projects. ACM SIGSOFT Software Engineering Notes, 41(1), 1-4. DOI:http://dx.doi.org/10.11591/ijece.v6i1.pp242-248.
6. Ferreira, K. A., Bigonha, M. A., Bigonha, R. S., Mendes, L. F., & Almeida, H. C. (2012). Identifying thresholds for object-oriented software metrics. Journal of Systems and Software, 85(2), 244-257. DOI:https://doi.org/10.1016/j.jss.2011.05.044.
7. Belachew, E. B., Gobena, F. A., & Nigatu, S. T. (2018). Analysis of software quality using software metrics. International Journal on Computational Science & Applications (IJCSA), 8(4/5). DOI:http://dx.doi.org/10.5121/ijcsa.2018.8502.
8. Filó, T. G., Bigonha, M. A., & Ferreira, K. A. (2024). Evaluating thresholds for object-oriented software metrics. Journal of the Brazilian Computer Society, 30(1), 315-346. DOI:http://dx.doi.org/10.5753/jbcs.2024.3373.
9. Imoize, A. L., Idowu, D., & Bolaji, T. (2019). A brief overview of software reuse and metrics in software engineering. World Scientific News, (122), 56-70.

10. Lee, M. C. (2014). Software quality factors and software quality metrics to enhance software quality assurance. British Journal of Applied Science & Technology, 4(21), 3069-3095. DOI:http://dx.doi.org/10.9734/BJAST/2014/10548.

11. Ronchieri, E., & Canaparo, M. (2017). METRICS FOR SOFTWARE RELIABILITY: AN INITIAL SYSTEMATIC MAPPING STUDY. DOI:https://doi.org/10.3233/jid-2018-0008.

12. Hlybovets, A., & Shapoval, O. (2020). Investigation of the Relationship Between Software Metrics Measurements and its Maintainability Degree. Наукові записки НаУКМА. Комп'ютерні науки, 3, 12-16. DOI:https://doi.org/10.18523/2617-3808.2020.3.12-16.

13. Kopyltsov, A. V. (2020, April). Selection of metrics in software quality evaluation. In Journal of Physics: Conference Series (Vol. 1515, No. 3, p. 032018). IOP Publishing. DOI:http://dx.doi.org/10.1088/1742-6596/1515/3/032018.

14. Choraś, M., Kozik, R., Pawlicki, M., Hołubowicz, W., & Franch, X. (2019, August). Software development metrics prediction using time series methods. In IFIP International Conference on Computer Information Systems and Industrial Management (pp. 311-323). Cham: Springer International Publishing. DOI:https://doi.org/10.1007/978-3-030-28957-7_26.

15. Pilipović, D., & Simeunović, D. (2021). The Quality of Software Metrics. JITA-Journal of Information Technology and Applications, 21(1), 61-65. DOI:https://doi.org/10.7251/JIT2101061P.

16. Omri, Safa, Pascal Montag, and Carsten Sinz. "Static analysis and code complexity metrics as early indicators of software defects." Journal of software engineering and applications 11.04 (2018): 153. DOI:http://dx.doi.org/10.4236/jsea.2018.114010.

17. Alakus, T. B., Das, R., & Turkoglu, I. (2019, September). An overview of quality metrics used in estimating software faults. In 2019 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-6). IEEE. DOI:http://dx.doi.org/10.1109/IDAP.2019.8875925.

18. Trautsch, A., Herbold, S., & Grabowski, J. (2020, September). Static source code metrics and static analysis warnings for fine-grained just-in-time defect prediction. In 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 127-138). IEEE. DOI:https://dx.doi.org/10.1109/ICSME46990.2020.00022.

19. Masmali, O., & Badreddin, O. (2020). Code quality metrics derived from software design. ICSEA 2020, 151.

20. Paz, F., & Pow-Sang, J. A. (2016). A systematic mapping review of usability evaluation methods for software development process. International Journal of Software Engineering and Its Applications, 10(1), 165-178. DOI:http://dx.doi.org/10.14257/ijseia.2016.10.1.16.

21. Sabharwal, S., Kaur, P., & Sibal, R. (2017). Empirical and Theoretical Validation of a Use Case Diagram Complexity Metric. International Journal of Information Technology and Computer Science, 9(11), 35-47. DOI:http://dx.doi.org/10.5815/ijitcs.2017.11.04.

22. Srinivasan, K. P., & Devi, T. (2014). Software metrics validation methodologies in software engineering. International Journal of Software Engineering & Applications, 5(6), 87. DOI:http://dx.doi.org/10.5121/ijsea.2014.5606.

23. Selvarani, R., Nair, T. G., Ramachandran, M., & Prasad, K. (2010). Software metrics evaluation based on entropy. In Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization (pp. 139-151). IGI Global Scientific Publishing. DOI:http://dx.doi.org/10.4018/978-1-60566-731-7.ch011.

24. Amara, D., Fatnassi, E., & Ben Arfa Rabai, L. (2021). An empirical assessment and validation of redundancy metrics using defect density as reliability indicator. Scientific Programming, 2021(1), 8325417. DOI:https://doi.org/10.1155/2021/8325417.

25. Hota, C., Kumar, L., & Neti, L. B. M. (2019, July). An Empirical Analysis on Effectiveness of Source Code Metrics for Aging Related Bug Prediction. In DMSVIVA (pp. 83-94). DOI:https://doi.org/10.18293/jvlc2019-n2-022.

26. Deshpande, M. B. S., Kumar, B., & Kumar, A. (2020). Object oriented design metrics for software defect prediction: An empirical study. TEST Eng. Manage., 83(1), 10501-10518.

27. Tao, H., Chen, Y., & Wu, H. (2022). Theoretical and empirical validation of software trustworthiness measure based on the decomposition of attributes. Connection Science, 34(1), 1181-1200. DOI:https://doi.org/10.1080/09540091.2022.2061424.

28. Alphonce, M. (2024). Enhancing Software Quality through Early-Phase of Software Verification and Validation Techniques. Available at SSRN 4611404. DOI:http://dx.doi.org/10.47604/ijts.2268.

29. Vyas, G., & Sharma, A. (2016). Empirical Evaluation of Metrics to Assess Software Product Line Feature Model Usability. International Journal of Science, Engineering and Computer Technology, 6(2), 82.

30. Mahmoud, A. T., Mohammed, A. A., Ayman, M., Medhat , W., Selim , S., Zayed, H., Yousef, A. H., & Elaraby , N. (2024). Formal Verification of Code Conversion: A Comprehensive Survey. Technologies, 12(12), 244. DOI:https://doi.org/10.3390/technologies12120244

31. of Research in Science, W. A. R. S. E. T. W. A., & E. (2020). Analysis and Assessment of Existing Software Quality Models to Predict the Reliability of Component-Based Software. International Journal of Emerging Trends in Engineering Research. DOI:https://doi.org/10.30534/IJETER/2020/96862020

32. Ghareb, M. I., & Allen, G. (2021). Quality Metrics measurement for Hybrid Systems (Aspect Oriented Programming--Object Oriented Programming). Technium, 3(3). DOI:https://doi.org/10.47577/technium.v3i3.3261

33. Malhotra, R., & Chug, A. (2016). Software maintainability: Systematic literature review and current trends. International Journal of Software Engineering and Knowledge Engineering, 26(08), 1221-1253. DOI:https://doi.org/10.1142/S0218194016500431

34. Suri, P. R., & Singhani, H. (2015). Object oriented software testability (OOSTE) metrics analysis. Int. J. Comput. Appl. Technol. Res, 4(5), 359-367. DOI:10.7753/IJCATR0405.1006

35. Aggarwal, J., & Kumar, M. (2021). Software metrics for reusability of component based software system: a review. Int. Arab J. Inf. Technol., 18(3), 319-325. DOI:https://dx.doi.org/10.34028/iajit/18/3/8.

36. Gerlec, Č., Rakić, G., Budimac, Z., & Heričko, M. (2012). A programming language independent framework for metrics-based software evolution and analysis. Computer science and information systems, 9(3), 1155-1186. DOI:https://dx.doi.org/10.2298/CSIS120104026G.

37. Nguyen, A. T. P., & Hoang, V. D. (2024). Development of code evaluation system based on abstract syntax tree. Journal of Technical Education Science, 19(Special Issue 01), 15-24. DOI:https://doi.org/10.54644/jte.2024.1514.

38. Antal, G., Tóth, Z., Hegedűs, P., & Ferenc, R. (2021). Enhanced Bug Prediction in JavaScript Programs with Hybrid Call-Graph Based Invocation Metrics. Technologies, 9(1), 3. DOI:https://doi.org/10.3390/technologies9010003.

39. Colakoglu, F. N., Yazici, A., & Mishra, A. (2021). Software product quality metrics: A systematic mapping study. IEEE access, 9, 44647-44670.

40. Colakoglu, F. N., Yazici, A., & Mishra, A. (2021). Software product quality metrics: A systematic mapping study. IEEE access, 9, 44647-44670. DOI:https://dx.doi.org/10.1109/ACCESS.2021.3054730.

41. Galin, D. (2018). Software process quality metrics. DOI:https://doi.org/10.1002/9781119134527.ch21.

42. Abran, A. (2010). Software Metrics and Software Metrology (1st ed.). Wiley. Retrieved from https://www.perlego.com/book/1010334/software-metrics-and-software-metrology-pdf (Original work published 2010).

43. Ludwig, J., Xu, S., & Webber, F. (2017, October). Compiling static software metrics for reliability and maintainability from GitHub repositories. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 5-9). IEEE. DOI:https://doi.org/10.1109/SMC.2017.8122569.

44. Jin, S., Li, Z., Chen, B., Zhu, B., & Xia, Y. (2023, October). Software code quality measurement: Implications from metric distributions. In 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS) (pp. 488-496). IEEE. DOI:https://doi.org/10.1109/QRS60937.2023.00054.

45. Nilson, M., Antinyan, V., & Gren, L. (2019, November). Do internal software quality tools measure validated metrics?. In International Conference on Product-Focused Software Process Improvement (pp. 637-648). Cham: Springer International Publishing. DOI:https://doi.org/10.48550/arXiv.1909.09682.

46. Misra, S., Akman, I., & Colomo-Palacios, R. (2014). A Framework for Evaluation and Validation of Software Complexity Measures. DOI:https://doi.org/10.1049/iet-sen.2011.0206.

47. Olague, H. M., Etzkorn, L. H., Gholston, S., & Quattlebaum, S. (2007). Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes. IEEE Transactions on software Engineering, 33(6), 402-419. DOI:https://doi.org/10.1109/TSE.2007.1015.

48. Zhong, Y., Song, K., Lv, S., & He, P. (2021). An Empirical Study of Software Metrics Diversity for Cross-Project Defect Prediction. Mathematical Problems in Engineering, 2021(1), 3135702. DOI:https://doi.org/10.1155/2021/3135702.

49. Tahir, A., & MacDonell, S. G. (2012, September). A systematic mapping study on dynamic metrics and software quality. In 2012 28th IEEE International Conference on Software Maintenance (ICSM) (pp. 326-335). IEEE. DOI:https://doi.org/10.1109/ICSM.2012.6405289.

50. Ardito, L., Coppola, R., Barbato, L., & Verga, D. (2020). A Tool-Based Perspective on Software Code Maintainability Metrics: A Systematic Literature Review. Scientific Programming, 2020(1), 8840389. DOI:https://doi.org/10.1155/2020/8840389.

51. Nuñez-Varela, A. S., Pérez-Gonzalez, H. G., Martínez-Perez, F. E., & Soubervielle-Montalvo, C. (2017). Source code metrics: A systematic mapping study. Journal of Systems and Software, 128, 164-197. DOI:https://doi.org/10.1016/j.jss.2017.03.044.

52. Chawla, M. K., & Chhabra, I. (2012). Implementing source code metrics for software quality analysis. International Journal of Engineering Research & Technology (IJERT), 1(5). 10.17577/IJERTV1IS5099.

53. Lenarduzzi, V., Lomio, F., Huttunen, H., & Taibi, D. (2020, February). Are sonarqube rules inducing bugs?. In 2020 IEEE 27th international conference on software analysis, evolution and reengineering (SANER) (pp. 501-511). IEEE. DOI:https://doi.org/10.48550/arXiv.1907.00376.

54. Yeboah, J., & Popoola, S. (2023, December). Efficacy of static analysis tools for software defect detection on open-source projects. In 2023 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1588-1593). IEEE. DOI:https://doi.org/10.48550/arXiv.2405.12333.

55. Bhutani, V., Toosi, F. G., & Buckley, J. (2024). Analysing the Analysers: An Investigation of Source Code Analysis Tools. Applied Computer Systems, 29(1), 98-111. DOI:https://doi.org/10.2478/acss-2024-0013.

56. Hlomani, H., & Stacey, D. (2014). Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. Semantic Web Journal, 1(5), 1-11.

57. Skogeby, J. (2022). Exploring Subjectivity in ad hoc Assessment of Open Source Software. LU-CS-EX.

58. Meneely, A., Smith, B., & Williams, L. (2013). Validating software metrics: A spectrum of philosophies. ACM Transactions on Software Engineering and Methodology (TOSEM), 21(4), 1-28. DOI:https://doi.org/10.1145/2377656.2377661.

59. Peitek, N., Apel, S., Parnin, C., Brechmann, A., & Siegmund, J. (2021, May). Program comprehension and code complexity metrics: An fmri study. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE) (pp. 524-536). IEEE.

60. Niswar, M. (2022). Application of JavaScript Code Similarity Detection for Assessment of Web Programming Assignment. EPI International Journal of Engineering, 5(2), 81-85. DOI:https://doi.org/10.25042/epi-ije.082022.01.

61. Pang, A., Anslow, C., & Noble, J. (2018, October). What programming languages do developers use? A theory of static vs dynamic language choice. In 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) (pp. 239-247). IEEE. DOI:https://doi.org/10.1109/VLHCC.2018.8506534.

62. Harper, R. (2016). Practical foundations for programming languages. Cambridge University Press.

63. Ortin, F., Garcia, M., Perez-Schofield, B. G., & Quiroga, J. (2022). The StaDyn programming language. SoftwareX, 20, 101211. DOI:https://doi.org/10.1016/j.softx.2022.101211.

64. Alphonce, M. (2024). Enhancing Software Quality through Early-Phase of Software Verification and Validation Techniques. Available at SSRN 4611404. DOI:https://dx.doi.org/10.2139/ssrn.4611404.

65. Reddy, J. M., & Prasad, S. V. A. V. (2016, March). The role of verification and validation in software testing. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 1298-1301). IEEE.

66. Abdulkareem, S. A., & Abboud, A. J. (2021, February). Evaluating python, c++, javascript and java programming languages based on software complexity calculator (halstead metrics). In IOP conference series: Materials science and engineering (Vol. 1076, No. 1, p. 012046). IOP Publishing.

67. Demyanova, Y., Pani, T., Veith, H., & Zuleger, F. (2017). Empirical software metrics for benchmarking of verification tools. Formal Methods in System Design, 50(2), 289-316.

68. Chapetta, W. A., das Neves, J. S., & Machado, R. C. S. (2021). Quantitative Metrics for Performance Monitoring of Software Code Analysis Accredited Testing Laboratories. Sensors, 21(11), 3660. https://doi.org/10.3390/s21113660.

69. Briand, L., El Emam, K., & Morasca, S. (1995). Theoretical and empirical validation of software product measures. International Software Engineering Research Network, Technical Report ISERN-95-03, 31-46.

70. Schneidewind, N. F. (1992). Methodology for validating software metrics.

Компетенције ментора

На основу детаљне анализе приложене документације, Комисија предлаже да се менторство за израду ове докторске дисертације повери проф. др Петру Милићу, ванредном професору Факултета техничких наука Универзитета у Приштини са привременим седиштем у Косовској Митровици, чија је ужа научна област рачунарска техника и информатика.

У складу са стандардом 9 Правилника о стандардима и поступку за акредитацију студијских програма, проф. др Петар Милић има најмање 5 научних радова категорије М20 из уже научне области рачунарска техника и информатика објављених у научним часописима у претходних 10 година и не води више од 5 докторанада истовремено. Конкретно, проф. др Петар Милић у протеклих 10 година има публикованих 13 радова категорије М20, његова ужа научна област је рачунарска техника и информатика, а кандидат Јован Милутиновић је један од 5 докторанада које предложени ментор води. Према томе, проф. др Петар Милић испуњава се прописане услове по основу компетенција за ментора ове докторске дисертације. Пет репрезентативних референци проф. др Петра Милића из области теме докторске дисертације су:

1. Milutinović, J., **Milić, P.**, Milenković, V., Mekić, E., Matović, A. (2025). "Application of Blockchain Technologies in Verification of Software Metrics ", Applied Sciences, Vol. 15, No. 10, pp. 5519, DOI= 10.3390/app15105519, ISSN: 2076-3417, IF = 2.5. **M21**
2. Katipoglu, G., Utku, S., Mijailović, I., Mekić, E., Avdić, Dž., **Milić, P.** (2024). "Action Research Approach to Analysis of Teaching of Blockchain Web 3.0 Application Based on MACH Architecture", Applied Sciences, Vol. 24, No. 13, pp. 11158, DOI=10.3390/app142311158, ISSN: 2076-3417, IF = 2.5. **M21**
3. Lnenicka, M., Nikiforova, A., Luterek, M., **Milić P.**, Rudmark, D., Neumaier, S., Kević, K., Zuiderwijk, A., Rodriguez Bolivar, M. P., (2024). „ Understanding the development of public data ecosystems: From a conceptual model to a six-generation model of the evolution of public data ecosystems", Telematics and Informatics, Vol. 94., No. 1., pp. 1 – 23, 2024, DOI = 10.1016/j.tele.2024.102190, ISSN: 0736-5853, IF = 7.6. **M21**
4. Lnenicka, M., Nikiforova, A., Luterek, M., **Milić P.**, Rudmark, D., Neumaier, S., Santoro, C., Casiano Flores, C., Janssen, M., Rodriguez Bolivar, M. P., (2024). „Identifying patterns and recommendations of and for sustainable open data initiatives: A benchmarking-driven analysis of open government data initiatives among European countries", Government Information Quarterly, Vol. 41., No. 1., pp. 1 – 24, 2024, DOI = 10.1016/j.giq.2023.101898, ISSN: 0740-624X, IF = 7.8. **M21**
5. Spalević, Ž., Kaljević, J., Vučetić, S., **Milić, P.**, (2023). „Enhancing Legally-Based E-Government Services in Education Through Artificial Intelligence", International Journal of Cognitive Research in Science, Engineering and Education (IJCRSEE), Vol. 11., No. 3., pp. 511 – 518, 2023, DOI = 10.23947/2334-8496-2023-11-3-511-518, ISSN: 2334-8496, IF = 0.7. **M23**

Закључак

На основу анализе постављеног проблема и увида у податке о научној и стручној делатности кандидата, Комисија закључује следеће:

- Предложена истраживања, хипотезе, циљеви, методологија и очекивани резултати истраживања су врло добро осмишљени и подобни за израду докторске дисертације,
- Досадашњи научни и научно-истраживачки резултати рада кандидата **Јована Милутиновића**, мастер инжењера електротехнике и рачунарства, показују његову подобност за израду докторске дисертације,
- Кандидат испуњава све услове предвиђене законом и одговарајућим општим актима Факултета техничких наука у Косовској Митровици за израду докторске дисертације.

На основу изнетог, Комисија предлаже наставно-научном већу Факултета техничких наука у Косовској Митровици да прихвати тему за израду докторске дисертације под називом: ***„Одређивање поузданости и верификација софтверских метрика у скрипт програмским језицима“***, кандидата **Јована Милутиновића**, мастер инжењера електротехнике и рачунарства и да се менторство за израду ове докторске дисертације повери др Петру Милићу, ванредном професору Факултета техничких наука у Косовској Митровици.

У Косовској Митровици, Новом Пазару

др Ненад Јовановић, редовни професор
Факултета техничких наука у Косовској Митровици

др Петар Милић, ванредни професор
Факултета техничких наука у Косовској Митровици

др Алдина Авдић, доцент
Државног универзитета у Новом Пазару