

ИЗВЕШТАЈ О ОЦЕНИ ДОКТОРСKE ДИСЕРТАЦИЈЕ

I ПОДАЦИ О КОМИСИЈИ		
1. Датум и орган који је именовао комисију: Решење декана Факултета техничких наука у Новом Саду, бр. 012-40/526-2025 од 16.07.2026.		
2. Састав комисије у складу са <i>Правилима докторских студија Универзитета у Новом Саду</i> :		
1. Зарић др Мирослав	редовни професор	Примењене рачунарске науке и информатика, 01.04.2024.
презиме и име	звање	ужа научна област и датум избора
Универзитет у Новом Саду, Факултет техничких наука	председник	
установа у којој је запослен-а		функција у комисији
2. Фртунић Глигоријевић др Милена	доцент	Рачунарство и информатика, 24.09.2024.
презиме и име	звање	ужа научна област и датум избора
Универзитет у Нишу, Електронски факултет	члан	
установа у којој је запослен-а		функција у комисији
3. Ракић др Гордана	доцент	Рачунарске науке, 28.03.2025.
презиме и име	звање	ужа научна област и датум избора
Универзитет у Новом Саду, Природно-математички факултет	члан	
установа у којој је запослен-а		функција у комисији
4. Дејановић др Игор	редовни професор	Примењене рачунарске науке и информатика, 16.05.2022.
презиме и име	звање	ужа научна област и датум избора
Универзитет у Новом Саду, Факултет техничких наука	члан	
установа у којој је запослен-а		функција у комисији
5. Ducasse др Stéphane	научни саветник	Рачунарске науке, 01.09.2007.
презиме и име	звање	ужа научна област и датум избора
Inria (национални институт за истраживање дигиталне науке и технологије), Лил, Француска	ментор	
установа у којој је запослен-а		функција у комисији
6. Милосављевић др Гордана	редовни професор	Примењене рачунарске науке и информатика, 21.10.2020.
презиме и име	Звање	ужа научна област и датум избора
Универзитет у Новом Саду, Факултет техничких наука	ментор	
установа у којој је запослен-а		функција у комисији
II ПОДАЦИ О КАНДИДАТУ		
1. Име, име једног родитеља, презиме:		

Балша, Славиша, Шаренац 2. Датум рођења, општина, држава: 07.06.1997, Требиње, Босна и Херцеговина 3. Назив факултета, назив претходно завршеног нивоа студија и стечени стручни/академски назив: Факултет техничких наука, Универзитет у Новом Саду; мастер академске студије, студијски програм Рачунарство и аутоматика; мастер инжењер електротехнике и рачунарства 4. Година уписа на докторске студије и назив студијског програма докторских студија : 2021, Рачунарство и аутоматика (докторске академске студије), Факултет техничких наука, Универзитет у Новом Саду
III НАСЛОВ ДОКТОРСKE ДИСЕРТАЦИЈЕ:
Унапређење рефакторисања у динамички типизираним објектно-оријентисаним језицима (Improving Refactorings in Dynamically Typed Object-Oriented Languages)
IV ПРЕГЛЕД ДОКТОРСKE ДИСЕРТАЦИЈЕ:
Навести кратак садржај са знаком броја страница, поглавља, слика, схема, графикана и сл. Докторска дисертација Балше Шаренца написана је на енглеском језику, у формату Б5. Укупно има 167 страна, од чега је 121 нумерисана страна. Основни текст организован је у осам поглавља на 92 стране, након којих следе списак литературе са 104 референце (стр. 93–102), четири прилога (стр. 103–120) и биографија аутора. Дисертација садржи 9 слика, 26 табела и 29 листинга кода. Уводни део чине насловна страна, кључна документацијска информација на српском и енглеском језику (Образац 5а), посвета, садржај, проширени резиме на српском језику (15 страна), апстракт на енглеском језику, списак слика, списак табела, списак листинга и списак скраћеница. Садржај дисертације по поглављима је следећи: 1. Увод (Introduction) 2. Основе и преглед стања у области (Background and Related Work) 3. Дизајн референтне архитектуре (Reference Architecture Design) 4. Имплементација референтне архитектуре у <i>Pharo</i> окружењу (Implementation of the Reference Architecture in Pharo) 5. Валидација и евалуација архитектуре (Architecture Validation and Evaluation) 6. <i>Refazz</i>: метаморфно тестирање система за рефакторисање (Refazz: Metamorphic Testing of Refactoring Engines) 7. Евалуација <i>Refazz</i> метода (Refazz Evaluation) 8. Закључак (Conclusion) Затим следе списак литературе и четири прилога: А. Мапа интерфејса модела програма (Program Model Interface Map), В. <i>Extract Method</i> у затеченом систему (Extract Method in the Legacy Engine), С. Потврђени дефекти система за рефакторисање (Confirmed Refactoring Engine Defects) и D. Резултати метода <i>Refazz</i> по пројектима (Project-Specific Refazz Outcomes), као и биографија аутора. На крају је приложен план третмана података.
V ВРЕДНОВАЊЕ ПОЈЕДИНИХ ДЕЛОВА ДОКТОРСKE ДИСЕРТАЦИЈЕ:
Дисертација садржи све елементе у складу са важећим прописима. Осам поглавља логички су распоређена и добро структурирана. Прва два поглавља постављају проблем истраживања, основне појмове и стање у области. Поглавља 3 и 4 представљају прву групу доприноса (референтну архитектуру и њену имплементацију у <i>Pharo</i> окружењу), а поглавље 5 њену евалуацију. Поглавља 6 и 7 представљају другу групу доприноса: метод тестирања <i>Refazz</i> и његову емпиријску евалуацију, док поглавље 8 сумира закључке и правце даљег развоја. Поглавља 5 и 7 наводе поступке, резултате и претње валидности заједно са мерама за њихово ублажавање. Прво поглавље уводи контекст истраживања, предмет истраживања, циљеве, хипотезе, истраживачка питања, метод и доприносе. Полази се од запажања да системи за рефакторисање аутоматизују структурне измене кода уз очување понашања, али да је у динамички типизираним објектно-оријентисаним језицима статичка провера очувања понашања суштински ограничена недостатком информација о типовима. Идентификовани су проблеми у постојећим системима за рефакторисање: архитектонска испреплетеност и тешкоће систематског тестирања оваквих система.

На основу тога формулисани су циљеви, хипотезе X1–X4, које су груписане у две теме (архитектонска модуларност и интеракција; поузданост система и аутоматизовано тестирање), и одговарајућа истраживачка питања RQ1–RQ4. Свако истраживачко питање је повезано са истраживачком активношћу, главним доказима и обухватом евалуације. Поглавље се завршава прегледом доприноса и структуре дисертације.

Друго поглавље дефинише основне појмове (рефакторисање, очување понашања, предуслови, трансформације) и прати еволуцију области од Опдајкове дисертације и првог Refactoring Browser-а до савремених окружења. Анализирана је анатомија система за рефакторисање (модел програма, модел измена, предуслови, извршавање) и специфичности динамички типизираних објектно-оријентисаних језика. Преглед стања у области обухвата начин на који су постојећи системи за рефакторисање грађени (архитектура, представљање програма, поновна употреба и композиција операција), начин на који се користе (интеракција, скриптовање и спецификација предуслова) и начин на који се проверавају (приступи тестирању, укључујући генераторе синтетичких програма, SafeRefactor и новију емпиријску студију дефеката). Посебна пажња посвећена је затеченом *Pharo* систему за рефакторисање као конкретној полазној основи за анализу проблема. Документовани су његови механизми, паралелне хијерархије и интеракција са корисницима. Поглавље се завршава синтезом која утврђује истраживачки јаз којим се дисертација бави. Преглед литературе је актуелан, релевантан и критички вреднован.

Треће поглавље представља централни допринос дисертације: референтну архитектуру која обједињује трансформације и рефакторисања у поново употребљиве, композитне градивне блокове. Архитектура је описана језички независним моделом улога и уговора. Дефинисани су предуслови применљивости, предуслови очувања понашања, одлучивање о наставку извршавања после упозорења и композиција над међустањима програма, чиме су утврђене одговорности улога и тачке посматрања на које се ослања метод тестирања из шестог поглавља. Излагање је прецизно и добро илустровано дијаграмима и током извршавања.

Четврто поглавље пресликава улоге референтне архитектуре на конкретне класе у *Pharo* окружењу и документује миграцију затеченог система на нови дизајн. Описан је полиморфни програмски интерфејс, реализација декоратора, реификовани предуслови, хијерархија интеракционих драјвера и интерфејс модела програма. Миграција је постепена, па затечене и мигриране класе постоје упоредо. Опис је технички прецизан, поткрепљен листинзима кода и довољан за разумевање евалуације која следи.

Пето поглавље евалуира архитектуру кроз два истраживачка питања. RQ1: „Како објединити трансформације и рефакторисања у градивне блокове погодне за поновну употребу и композицију?“ RQ2: „Како изоловати корисничку интеракцију од основне логике операције тако да иста логика подржи и интерактивни рад и неинтерактивно извршавање без графичког интерфејса?“ У склопу миграције за девет рефакторисања направљени су декоратори над шест трансформационих класа, направљено је 35 интеракционих драјвера, а 35 директних употреба апликативног програмског интерфејса (енгл. *Application Programming Interface* – API) модела програма замењено је поновном употребом трансформација. Поновна употреба трансформације *Add Method* порасла је са 5 на 28 клијената, а *Remove Method* са 3 на 14. Удео дуплираних линија смањен је са 6,02% на 1,52%. Студије случаја показују да се композицијом резултујућих операција општа операција *Extract Method* поједностављује, а доменски специфичне варијанте граде од постојећих корака, које замењују или проширују. За RQ2, директни позиви за упозорења и грешке уклоњени су из мигрираних класа рефакторисања, а иста логика операција извршава се и интерактивно и из скрипти.

Шесто поглавље уводи Refazz, метод за метаморфно тестирање система за рефакторисање. Тестови реалних пројеката се користе као делимичне извршиве спецификације понашања, а избор тестова вођен је алатом MuTalk. Метод разликује четири режима извршавања: тестирање исправним улазима (MR1), истраживање параметара укључујући неисправне и граничне конфигурације (MR2), принудно извршавање после одбијеног предуслова применљивости (MR3) и принудно извршавање после упозорења о очувању понашања (MR4). Режији MR3 и MR4 испитују границе предуслова, а не очување понашања. Поглавље описује стратегије генерисања параметара рефакторисања, имплементацију метода и поступак анализе којим се записи о покушајима свде на групе аномалија, прегледане репрезентативне покушаје и потврђене јединствене дефекте. Позиционирање према ранијим приступима јасно образлаже новину методе. Излагање је систематично и повезано са моделом из трећег поглавља.

Седмо поглавље емпиријски евалуира Refazz на *Pharo* систему за рефакторисање, кроз два истраживачка питања. RQ3: „У којој мери се постојећи тестови реалних пројеката могу користити као делимичне извршиве спецификације које откривају промене понашања изазване рефакторисањем?“ RQ4: „На који начин испитивање простора параметара и принудно извршавање после одбијања предуслова применљивости или упозорења о очувању понашања откривају недостајуће или претерано конзервативне предуслове у односу на тестирање само исправних улаза?“ Пројекти су изабрани наменским узорковањем: пет реалних пројеката из различитих домена и различитих величина, а испитано је седам рефакторисања кроз укупно 255.699 покушаја. Потврђено је 19 јединствених дефеката система: 12 везаних за трансформације и 7 везаних за предуслове (3 недостајућа, 2 претерано конзервативна и 2 погрешно смештена предуслова). Тестови пројеката који нису прошли открили су седам дефеката (37%), односно измене понашања које синтаксне и структурне провере не откривају, што подржава хипотезу X3 у испитаном обиму. Од 19 дефеката, 13 је приписано тестирању исправним улазима (MR1), пет истраживању параметара (MR2), а један принудном извршавању после одбијања (MR3). Режији MR2 и MR3 заједно су открили 6 од 19 дефеката (32%), све везане за предуслове, што подржава хипотезу X4 за испитане режими и пројекте. Режим MR4 није донео нове дефекте: његови покушаји који су одступили од очекиваног исхода поновили су налазе других режима, а успешни покушаји нису могли да искључе ризике на које су упозорења указивала. Сваки потврђени дефект повезан је са пријавом у јавно доступном систему за праћење грешака, а шест исправки интегрисано је у развојну грану *Pharo* система. Претње валидности наведене су заједно са мерама предузетим да се ограничи њихов утицај, а закључци важе само за испитане пројекте и рефакторисања.

Завршно поглавље сумира оба доприноса и даје експлицитне одговоре на сва четири истраживачка питања, уз оцену сваке хипотезе у испитаном обиму. Разматрана је преносивост решења на друга динамички типизирана окружења, на основу пресликавања архитектонских улога и захтева које циљни систем мора да испуни. Правци даљег развоја укључују довршетак миграције каталога рефакторисања, документовање понављајућих одлука о миграцији у виду каталога за ауторе сличних система, употребу метода *Refazz* као чувара квалитета нових верзија система за рефакторисање и преносе на друге језике. Закључци су јасно образложени и усклађени са представљеним резултатима.

Прилози, а посебно каталог свих 19 потврђених дефеката са везама ка пријавама у јавно доступном систему за праћење грешака (прилог C) и резултати метода *Refazz* по пројектима (прилог D), обезбеђују проверљивост и поновљивост резултата.

Списак литературе (104 референце) релевантан је за тему дисертације и обухвата класичне и савремене изворе из области рефакторисања, архитектуре софтверских алата и тестирања софтвера. Проширени резиме на српском језику верно одражава садржај дисертације.

VI СПИСАК НАУЧНИХ И СТРУЧНИХ РАДОВА КОЈИ СУ ОБЈАВЉЕНИ ИЛИ ПРИХВАЋЕНИ ЗА ОБЈАВЉИВАЊЕ НА ОСНОВУ РЕЗУЛТАТА ИСТРАЖИВАЊА У ОКВИРУ РАДА НА ДОКТОРСКОЈ ДИСЕРТАЦИЈИ:

Таксативно навести називе радова, где и када су објављени. Прво навести најмање један рад објављен или прихваћен за објављивање у складу са *Правилима докторских студија Универзитета у Новом Саду* који је повезан са садржајем докторске дисертације. У случају радова прихваћених за објављивање, таксативно навести називе радова, где и када ће бити објављени и приложити потврду уредника часописа о томе.

Рад у истакнутом међународном часопису (M22):

1. Šarenac, B., Anquetil, N., Ducasse, S., Tesone, P.: A New Architecture Reconciling Refactorings and Transformations. *Journal of Computer Languages*, 80 (2024), 101273, pp. 1–18. <https://doi.org/10.1016/j.col.2024.101273>

Саопштења са међународних скупова штампана у целини (M33):

1. Šarenac, B., Ducasse, S., Polito, G., Rakić, G.: Modular and Extensible Extract Method. In: *International Workshop on Smalltalk Technologies (IWST'24)*, 2024, pp. 1–14. <https://ceur-ws.org/Vol-3893/Paper10.pdf>
2. Šarenac, B., Ducasse, S., Polito, G., Rakić, G.: Building Blocks for Object-Oriented Refactoring Engines. In: *International Conference on Information Society and Technology (ICIST 2025)*, 2025, pp. 1–17. <https://inria.hal.science/PHARO/hal-05022531v1>

VII ЗАКЉУЧЦИ ОДНОСНО РЕЗУЛТАТИ ИСТРАЖИВАЊА:
Докторска дисертација бави се унапређењем поузданости и поновне употребе унутар система за рефакторисање у динамички типизираним објектно-оријентисаним језицима, кроз два комплементарна доприноса: референтну архитектуру система за рефакторисање и метод тестирања <i>Refazz</i>. Истраживање је довело до следећих закључака и резултата: Методолошки резултати 1. Дефинисана је референтна архитектура система за рефакторисање, описана моделом улога и уговора независним од конкретног језика и система, која се ослања на постојеће моделе програма и измена циљног система. 2. Дефинисан је <i>Refazz</i>, метаморфни метод тестирања система за рефакторисање који тестове реалних пројеката користи као делимичну извршиву спецификацију понашања и који, поред исправних улаза, систематски испитује границе предуслова кроз четири режима извршавања (MR1–MR4). Технички резултати 1. Затечени <i>Pharo</i> систем за рефакторисање мигриран је на нову архитектуру у обиму који обухвата девет декоратора над шест трансформационих класа, 35 интеракционих драјвера и замену 35 директних употреба API-ја модела програма поновном употребом трансформација. Композитна имплементација <i>Extract Method</i> и два доменски специфична рефакторисања показују да се резултујуће операције компонују и да се композитни низ може заменити или проширити. Резултати миграције део су развојне гране <i>Pharo</i> система. 2. Ограничени доказ концепта у независно развијеном <i>Python</i> систему за рефакторисање под називом <i>rope</i> показао је да се основне поставке архитектуре могу имплементирати на још једном динамички типизираним језику, поред <i>Pharo</i>-а. Реализоване су две операције, уз 103 нова теста, док свих 137 постојећих тестова тих операција и даље пролази. Решење је јавно доступно. Емпиријски резултати 1. У делу система обухваћеном миграцијом, удео дуплираних линија у анализираним пакетима смањен је са 6,02% на 1,52% (релативно смањење 74,7%), број парова клонова са 104 на 32, а парова између класе рефакторисања и класе трансформације са 79 на 19. Поновна употреба кључних трансформација порасла је са 5 на 28 клијената (<i>Add Method</i>) и са 3 на 14 клијената (<i>Remove Method</i>). Ови резултати подржавају хипотезу X1. 2. Исто конфигурирано језгро извршава се и интерактивно и из скрипти, а број директних позива за упозорења и за грешке у систему смањен је са 28 на 15, односно са 163 на 98. Ови резултати подржавају хипотезу X2 у обиму имплементације. 3. Применом метода <i>Refazz</i> на пет реалних пројеката и седам рефакторисања извршено је 255.699 покушаја и потврђено 19 јединствених дефеката (12 везаних за трансформације и 7 везаних за предуслове). Тестови који нису прошли открили су седам дефеката који се не испољавају кроз синтаксне и структурне провере. Ови резултати подржавају хипотезу X3 у испитаном обиму. 4. Истраживање параметара и принудно извршавање открили су шест дефеката везаних за предуслове (32% каталога) које тестирање исправним улазима није нашло, што подржава хипотезу X4 за испитане режиме и пројекте. Исправке за шест пријављених дефеката интегрисане су у развојну грану <i>Pharo</i> система. Узете заједно, подржане хипотезе X1–X4 поткрепљују оба доприноса дисертације наведена изнад.
VIII ОЦЕНА НАЧИНА ПРИКАЗА И ТУМАЧЕЊА РЕЗУЛТАТА ИСТРАЖИВАЊА:
Експлицитно навести позитивну или негативну оцену начина приказа и тумачења резултата истраживања.
Докторска дисертација урађена је систематично, уз примену ригорозног научног приступа. Истраживачки проблем, хипотезе, циљеви и методе јасно су формулисани у уводном поглављу, а наредна поглавља следе логичан ток од дефинисања основних појмова и стања у области, преко дефинисања референтне архитектуре и њеног модела, до имплементације, евалуације и метода тестирања. Свако истраживачко питање експлицитно је повезано са истраживачком активношћу, главним доказима и обухватом евалуације. Архитектура и имплементација илустроване су јасним дијаграмима и листинзима кода, а резултати истраживања представљени су нумерички и табеларно и објективно су протумачени. Свака претња валидности наведена је заједно са мерама које ограничавају њен утицај, а тврдње су изнете само у оквиру испитаног обима. Потврђени дефекти повезани су са пријавама у јавно доступном систему за праћење грешака, а прилози и софтверска решења отвореног кода омогућавају проверу и поновљивост резултата. Закључци представљају концизну и кохерентну

синтезу резултата и непосредно су поткрепљени изнетим доказима, што потврђује да су хипотезе критички проверене и да су циљеви дисертације остварени. Рад кандидата у потпуности одговара дефинисаној теми и наслову, како по обиму, тако и по методологији, а резултати пружају солидну основу за даља истраживања у овој области. Дисертација је проверена у софтверу за детекцију плагијаризма iThenticate. На основу наведеног, комисија позитивно оцењује начин приказа и тумачења резултата истраживања.
IX КОНАЧНА ОЦЕНА ДОКТОРСКЕ ДИСЕРТАЦИЈЕ: Експлицитно навести да ли дисертација јесте или није написана у складу са наведеним образложењем, као и да ли она садржи или не садржи све битне елементе. Дати јасне, прецизне и концизне одговоре на 3. и 4. питање. Навести нумеричке податке о резултатима провере оригиналности рада и дати текстуално образложење.
1. Да ли је дисертација написана у складу са образложењем наведеним у пријави теме? Докторска дисертација је написана у складу са образложењем наведеним у пријави теме. Наслов дисертације усклађен је са корекцијом коју је извршила Комисија за оцену подобности кандидата, теме и ментора, што је усвојено приликом прихватања теме (20.11.2025).
2. Да ли дисертација садржи све битне елементе? Дисертација садржи све битне елементе.
3. По чему је дисертација оригиналан допринос науци? Докторска дисертација Балше Шаренца представља оригиналан научни допринос области софтверског инжењерства, посебно архитектури и тестирању система за рефакторисање у динамички типизираним објектно-оријентисаним језицима. Њени доприноси су: • Референтна архитектура која обједињује трансформације и рефакторисања као језички независне улоге и уговоре над постојећим моделима програма и измена циљног система . • Интеракциони драјвери који одвајају интеракцију са корисником од основне логике операције, тако да иста логика подржава интерактивни рад и извршавање скрипти. • Refazz, метаморфни метод тестирања система за рефакторисање који тестове реалних пројеката користи као делимичну извршиву спецификацију понашања и који систематски испитује предуслове истраживањем параметара и принудним извршавањем после одбијања или добијања упозорења. • Емпиријски резултати: миграција затеченог <i>Pharo</i> система са мерљивим смањењем дуплирања и порастом поновне употребе, каталог од 19 потврђених дефеката из 255.699 покушаја, са шест исправки интегрисаних у развојну грану <i>Pharo</i> система и ограничени доказ концепта у <i>Python</i> систему под називом <i>rope</i> који показује да архитектонске улоге нису везане за <i>Smalltalk</i>. Поред научног доприноса, ови резултати имају и непосредну практичну вредност за <i>Pharo</i> екосистем и ауторе сличних система, јер су доступни као решења отвореног кода .
4. Који су недостаци дисертације и какав је њихов утицај на резултат истраживања? Комисија није уочила недостатке дисертације који би утицали на резултате истраживања.
5. Образложење резултата провере оригиналности рада (нумерички и наративно): Провера оригиналности спроведена је софтвером iThenticate у библиотеци Факултета техничких наука, над основним текстом дисертације (26.769 речи). Индекс сличности износи 3%, уз 41 поклапање на 16 извора, при чему ниједан извор не прелази 1%. Највећа поклапања потичу са локација на којима су доступни радови кандидата објављени у оквиру овог истраживања, а преостала се односе на стандардну терминологију области и називе цитираних радова. Комисија стога закључује да је поднета докторска дисертација оригинално дело кандидата Балше Шаренца.

X	ПРЕДЛОГ:
	На основу наведеног, комисија предлаже:
	<u>а) да се докторска дисертација прихвати, а кандидату одобри одбрана;</u> б) да се докторска дисертација врати кандидату на дораду (да се допуни односно измени); в) да се докторска дисертација одбије.

Место и датум: Нови Сад, 14.09.2026.

1. др Мирослав Зарић, ред. проф.

_____, председник

2. др Милена Фртунић Глигоријевић, доцент

_____, члан

3. др Гордана Ракић, доцент

_____, члан

4. др Игор Дејановић, ред. проф.

_____, члан

5. др Stéphane Ducasse, научни саветник

_____, ментор

6. др Гордана Милосављевић, ред. проф.

_____, ментор

НАПОМЕНА: Члан комисије који не жели да потпише извештај јер се не слаже са мишљењем већине чланова комисије, дужан је да унесе у извештај образложење односно разлоге због којих не жели да потпише извештај и да исти потпише.

DOCTORAL DISSERTATION EVALUATION REPORT

I COMMITTEE INFORMATION		
1. Date and body that assigned the committee: Decision of the Dean of the Faculty of Technical Sciences in Novi Sad, no. 012-40/526-2025 dated 16.07.2026.		
2. Committee members' information in accordance with the <i>Rules of Doctoral Studies of the University of Novi Sad</i> :		
1. Zarić Miroslav, PhD	full professor	Applied Computer Science and Informatics, 01.04.2024.
Surname and name	Title	Scientific subfield and date of election
University of Novi Sad, Faculty of Technical Sciences		chair
Institution of employment		Committee role
2. Frtunić Gligorjević Milena, PhD	assistant professor	Computer Science and Informatics, 24.09.2024.
Surname and name	Title	Scientific subfield and date of election
University of Niš, Faculty of Electronic Engineering		member
Institution of employment		Committee role
3. Rakić Gordana, PhD	assistant professor	Computer Science, 28.03.2025.
Surname and name	Title	Scientific subfield and date of election
University of Novi Sad, Faculty of Sciences		member
Institution of employment		Committee role
4. Dejanović Igor, PhD	full professor	Applied Computer Science and Informatics, 16.05.2022.
Surname and name	Title	Scientific subfield and date of election
University of Novi Sad, Faculty of Technical Sciences		member
Institution of employment		Committee role
5. Ducasse Stéphane, PhD	research director	Computer Science, 01.09.2007.
Surname and name	Title	Scientific subfield and date of election
Inria (French National Institute for Research in Digital Science and Technology), Lille, France		mentor
Institution of employment		Committee role
6. Milosavljević Gordana, PhD	full professor	Applied Computer Science and Informatics, 21.10.2020.
Surname and name	Title	Scientific subfield and date of election
University of Novi Sad, Faculty of Technical Sciences		mentor
Institution of employment		Committee role
II INFORMATION ABOUT THE CANDIDATE		

1. Name, name of one parent, surname: Balša, Slaviša, Šarenac 2. Date, municipality and country of birth: 07.06.1997, Trebinje, Bosnia and Herzegovina 3. Name of the academic institution, previously completed academic program and the title acquired: Faculty of Technical Sciences, University of Novi Sad; Master Academic Studies, study programme Computing and Control Engineering; Master of Science in Electrical and Computer Engineering 4. Year of enrolment into doctoral studies and name of the doctoral studies programme: 2021, Computing and Control Engineering (Doctoral Academic Studies), Faculty of Technical Sciences, University of Novi Sad
III TITLE OF THE DOCTORAL DISSERTATION: Improving Refactorings in Dynamically Typed Object-Oriented Languages (Унапређење рефакторисања у динамички типизираним објектно-оријентисаним језицима)
IV OVERVIEW OF THE DOCTORAL DISSERTATION: Provide a brief overview indicating the number of pages, chapters, figures, diagrams, graphs, etc. The doctoral dissertation of Balša Šarenac is written in English and presented in B5 format. It has 167 pages in total, of which 121 are numbered. The main text is organized into eight chapters across 92 pages, followed by a bibliography with 104 references (pages 93–102), four appendices (pages 103–120), and the author's biography. The dissertation contains 9 figures, 26 tables, and 29 code listings. The front matter comprises the title page, Key Word Documentation in Serbian and English (Form 5a), a dedication, the table of contents, an extended summary in Serbian (15 pages), an English abstract, and the lists of figures, tables, listings, and abbreviations. The dissertation's chapters are: 1. Introduction 2. Background and Related Work 3. Reference Architecture Design 4. Implementation of the Reference Architecture in Pharo 5. Architecture Validation and Evaluation 6. Refazz: Metamorphic Testing of Refactoring Engines 7. Refazz Evaluation 8. Conclusion The main text is followed by the bibliography and four appendices: A. Program Model Interface Map, B. Extract Method in the Legacy Engine, C. Confirmed Refactoring Engine Defects, and D. Project-Specific Refazz Outcomes, and the author's biography. A data management plan is appended at the end.
V EVALUATION OF INDIVIDUAL DISSERTATION CHAPTERS: The dissertation adheres to all prescribed regulations. Its eight chapters are logically arranged and well-structured. The first two chapters establish the research problem, basic concepts, and the state of the art. Chapters 3 and 4 present the first group of contributions (the reference architecture and its implementation in Pharo), and Chapter 5 its evaluation. Chapters 6 and 7 present the second group of contributions: the Refazz testing method and its empirical evaluation, while Chapter 8 summarizes the conclusions and directions for future work. Chapters 5 and 7 report procedures, results, and threats to validity together with the measures that limit their impact. The first chapter introduces the research context, subject, objectives, hypotheses, research questions, method, and contributions. It starts from the observation that refactoring engines automate structural changes while preserving behavior, but that in dynamically typed object-oriented languages the static checking of behavior preservation is fundamentally limited by the absence of type information. Problems were identified in existing refactoring engines: architectural coupling and the difficulty of systematically testing such engines. On this basis, the research is guided by four hypotheses, H1–H4, grouped into two themes (Architectural Modularity and Interaction; Engine Reliability and Automated Testing), and by the corresponding research questions RQ1–RQ4. Each research question is connected to its research activity, principal evidence, and evaluation scope. The chapter concludes with an overview of the contributions and the structure of the dissertation.

The second chapter defines the basic concepts (refactoring, behavior preservation, applicability and behavior-preserving preconditions, transformations) and traces the evolution of refactoring from Opdyke's dissertation and the original Refactoring Browser to modern development environments. It describes the anatomy of a refactoring engine (program model, change model, preconditions, execution) and the specifics of dynamically typed object-oriented languages. The state-of-the-art review covers how existing refactoring engines are built (architecture, program representation, reuse and composition of operations), how they are used (interaction, scripting, and the specification of preconditions), and how they are verified (testing approaches, including synthetic program generators, SafeRefactor, and a recent empirical study of defects). The Pharo legacy engine is examined as the concrete baseline for the architectural work. Its mechanisms, parallel hierarchies, and embedded interaction are documented in detail. The chapter closes with a synthesis that states the research gap. The literature review is current, relevant, and critically evaluated.

The third chapter presents the central contribution of the dissertation: a reference architecture for refactoring engines that unifies transformations and refactorings into reusable, composable building blocks. The architecture is described in terms of roles and contracts, in a language-independent model that defines applicability preconditions, behavior-preserving preconditions, the decision whether to proceed after a warning, and composition over intermediate program states. The model fixes the responsibilities of the roles and the observation points on which the testing method of Chapter 6 relies. The presentation is precise and well illustrated with diagrams and an execution flow.

The fourth chapter describes the implementation of the reference architecture in the Pharo refactoring engine. It focuses on the concrete classes and protocols that realize the architectural roles: a shared execution API, refactoring decorators, reified preconditions, interaction drivers, and the program-model interface. The migration is incremental, so legacy and migrated classes coexist. The description is technically precise, supported by code listings, and sufficient for understanding the evaluation that follows.

The fifth chapter evaluates the architecture through two research questions. RQ1: "How can a refactoring engine be architected to unify behavior-agnostic transformations and behavior-preserving refactorings into reusable, composable building blocks?" RQ2: "How can user interaction be isolated from core refactoring logic while preserving both interactive guidance and headless batch execution?" Within the scope of the migration, nine refactoring wrappers use the decorator pattern and delegate to six distinct transformation classes, 35 interaction drivers have been created for several families of refactorings, and 35 usages of the program model API were replaced with explicit transformation reuse. *Add Method* is reused by 28 clients, up from 5, and *Remove Method* by 14, up from 3. The duplicated-line ratio decreased from 6.02% to 1.52%. The case studies show that the resulting operations compose into a simplified *Extract Method* refactoring and into domain-specific variants that replace or extend composed sequences. For RQ2, the migrated families contain no direct sends to the legacy signaling entry methods in their core operations, and the migrated operations use the same core implementation for interactive and unattended execution.

The sixth chapter presents Refazz, a method for the metamorphic testing of refactoring engines. Project tests serve as partial executable behavioral specifications, and test selection is guided by MuTalk. Four execution modes are distinguished: valid input (MR1), input exploration with invalid and boundary configurations (MR2), forced execution after an applicability rejection (MR3), and forced execution after a behavior-preserving warning (MR4). MR3 and MR4 examine the precondition boundaries, not behavior preservation. The chapter describes the strategies for generating refactoring parameters, the implementation of the method, and the analysis pipeline, which reduces attempt records to anomaly groups, inspected representatives, and confirmed unique defects. The positioning against prior testing approaches clearly explains the novelty of the method. The method is presented systematically and is connected to the model of Chapter 3.

The seventh chapter empirically evaluates Refazz on the Pharo refactoring engine, addressing two research questions. RQ3: "To what extent can existing test suites from real-world projects act as partial executable behavioral specifications that expose behavior changes caused by refactorings?" RQ4: "How do parameter exploration and forced execution past an applicability rejection or a behavior-preserving warning reveal missing or overly conservative refactoring preconditions that valid-input testing does not expose?" The projects were chosen by purposeful sampling, from various domains and of different sizes. Across seven studied refactorings and five projects, 255,699 attempts confirmed 19 unique refactoring engine defects: 12 transformation defects and 7 precondition defects (3 missing, 2 overly conservative, and 2 misplaced checks). Project-test failures exposed seven of the 19 confirmed defects (37%), that is, behavior changes that syntactic and structural checks do not reveal, which supports hypothesis H3 within the observed scope. Valid-input testing surfaced 13 defects. Three missing checks, one overly conservative check, and one misplaced

behavior-preserving check were uniquely attributed to MR2, and one overly conservative check to MR3. Together, these account for 6 of the 19 confirmed defects (32%), all of them precondition defects, which supports hypothesis H4 within the studied modes and projects. MR4 added no confirmed defect: its anomalous runs repeated evidence from other modes, and passing runs could not rule out the behavior risk reported by a warning. Every counted defect is linked to a public tracker issue, and six fixes have been integrated into the Pharo development branch. Threats to validity are stated together with the measures that limit their impact, and the conclusions apply only to the projects and refactorings studied.

The final chapter summarizes both contributions and gives explicit answers to all four research questions, together with an assessment of each hypothesis within the observed scope. The portability of the solution to other dynamic environments is discussed on the basis of the mapping of architectural roles and the requirements a target engine must fulfill. Future work includes completing the migration of the refactoring catalog under the new architecture, compiling the recurring migration decisions into a migration catalog for developers working on similar engines, making Refazz a permanent quality gate before each Pharo release, and transfers to other languages. The conclusions are clearly justified and consistent with the presented results.

The appendices, in particular the catalog of all 19 confirmed defects with links to reports in the public bug tracker (Appendix C) and the per-project Refazz outcomes (Appendix D), make the results verifiable and reproducible.

The bibliography (104 references) is relevant to the topic of the dissertation and covers classic and recent sources in refactoring, software tool architecture, and software testing. The extended summary in Serbian faithfully reflects the content of the dissertation.

VI LIST OF SCIENTIFIC AND PROFESSIONAL MANUSCRIPTS PUBLISHED OR ACCEPTED FOR PUBLICATION, WHICH ARE BASED ON THE RESULTS OF THE DOCTORAL RESEARCH:

List the titles of the manuscripts, where and when they were published. First list at least one manuscript published or accepted for publication in accordance with the *Rules of Doctoral Studies of the University of Novi Sad* which is related to the content of the doctoral dissertation. In the case of manuscripts accepted for publication, list their titles, where and when they will be published, and enclose the confirmation of the journal editor thereof.

Paper in a distinguished international journal (M22):

1. Šarenac, B., Anquetil, N., Ducasse, S., Tesone, P.: A New Architecture Reconciling Refactorings and Transformations. *Journal of Computer Languages*, 80 (2024), 101273, pp. 1–18.
<https://doi.org/10.1016/j.jcola.2024.101273>

Papers presented at international conferences and published in full (M33):

1. Šarenac, B., Ducasse, S., Polito, G., Rakić, G.: Modular and Extensible Extract Method. In: *International Workshop on Smalltalk Technologies (IWST'24)*, 2024, pp. 1–14. <https://ceur-ws.org/Vol-3893/Paper10.pdf>
2. Šarenac, B., Ducasse, S., Polito, G., Rakić, G.: Building Blocks for Object-Oriented Refactoring Engines. In: *International Conference on Information Society and Technology (ICIST 2025)*, 2025, pp. 1–17. <https://inria.hal.science/PHARO/hal-05022531v1>

VII CONCLUSIONS AND RESEARCH RESULTS:

The doctoral dissertation addresses reliability and reuse in refactoring engines in dynamically typed object-oriented languages through two complementary contributions: a reference architecture for refactoring engines and the Refazz testing method. The research led to the following conclusions and results:

Methodological results

1. A reference architecture for refactoring engines was defined, characterized by a model of roles and contracts that is independent of any specific language or engine. This architecture builds on the existing program and change models of the target engine.
2. Refazz, a method for the metamorphic testing of refactoring engines, was defined. It uses real project tests as a partial behavioral oracle and combines parameter exploration with forced execution past rejected applicability checks and behavior-preserving warnings (modes MR1–MR4).

Technical results

1. The legacy Pharo refactoring engine was migrated to the new architecture in a scope covering nine decorators over six transformation classes, 35 interaction drivers, and the replacement of 35 direct usages of

the program model API with transformation reuse. The composite <i>Extract Method</i> implementation and two domain-specific refactorings show that the resulting operations compose and that a composed sequence can be replaced or extended. The migration results are part of the Pharo development branch. 2. A scoped proof of concept in <i>rope</i>, an independently developed Python refactoring engine, shows that the core architectural roles can be realized in another dynamically typed language besides Pharo. Two operations were realized, with 103 new tests, while all 137 pre-existing tests of those operations still pass. The artifact is publicly available. Empirical results 1. Within the scope of the migration, the duplicated-line ratio in the analyzed packages decreased from 6.02% to 1.52% (a relative reduction of 74.7%), the number of clone pairs from 104 to 32, and the pairs between refactoring and transformation classes from 79 to 19. The reuse of the essential transformations increased from 5 to 28 (<i>Add Method</i>) and from 3 to 14 (<i>Remove Method</i>) distinct clients. These results support hypothesis H1. 2. The same configured core runs from both interactive and scripted entry points, and direct warning- and error-signaling sends in the engine decreased from 28 to 15 and from 163 to 98, respectively. These results support hypothesis H2 within the implementation scope. 3. Applying Refazz to five real projects and seven refactorings produced 255,699 attempts and confirmed 19 unique defects (12 transformation and 7 precondition defects). Project-test failures exposed seven defects that syntactic and structural checks do not reveal. These results support hypothesis H3 within the observed scope. 4. Parameter exploration and forced execution exposed six precondition defects (32% of the catalog) that valid-input testing did not find, which supports hypothesis H4 within the studied modes and projects. Fixes for six of the reported defects have been integrated into the Pharo development branch. Taken together, the supported hypotheses H1–H4 substantiate both contributions of the dissertation stated above.	VIII EVALUATION OF THE METHOD OF PRESENTATION AND INTERPRETATION OF RESEARCH RESULTS: Explicitly state a positive or negative evaluation of the method of presentation and interpretation of the research results. The doctoral dissertation has been prepared systematically, applying a rigorous scientific approach. The research problem, hypotheses, objectives, and methods are clearly formulated in the introductory chapter, and the subsequent chapters progress logically from the definition of the basic concepts and the state of the art, through the definition of the reference architecture and its model, to the implementation, the evaluation, and the testing method. Each research question is explicitly connected to its research activity, principal evidence, and evaluation scope. The architecture and the implementation are illustrated with clear diagrams and code listings, and the research results are presented numerically and in tables and are objectively interpreted. Every threat to validity is stated together with the measures that limit its impact, and the claims are made only within the observed scope. The confirmed defects are linked to reports in the public bug tracker, and the appendices and publicly available software solutions enable verification and reproducibility of the results. The conclusions represent a concise and coherent synthesis of the results and are directly supported by the presented evidence, confirming that the hypotheses were critically examined and that the objectives of the dissertation were achieved. The candidate's work fully corresponds to the defined topic and title, in scope as well as in methodology, and the results provide a solid basis for further research in this area. The dissertation was checked using the iThenticate plagiarism detection software. Based on the above, the Committee positively evaluates the method of presentation and interpretation of the research results. IX FINAL EVALUATION OF DOCTORAL DISSERTATION: Explicitly state whether the dissertation was or was not written in accordance with the stated rationale, and whether it does or does not contain all the important elements. Give clear, precise and concise answers to questions 3 and 4. Provide numerical data on the results of the originality check of the work and give a textual explanation. 1. Was the dissertation written in accordance with the previously submitted and approved explanation given in the topic proposal?
--	--

The doctoral dissertation was written in accordance with the explanation given in the dissertation topic proposal. The title of the dissertation reflects the correction made by the Committee for the Evaluation of the Suitability of the Candidate, Topic, and Mentors, which was adopted when the thesis topic was accepted (20.11.2025).
2. Does the dissertation contain all the important elements? The dissertation contains all the important elements.
3. Why does this dissertation provide original contributions to science? The doctoral dissertation of Baša Šarenac represents an original scientific contribution to software engineering, in particular to the architecture and testing of refactoring engines in dynamically typed object-oriented languages. Its contributions are: • A reference architecture unifying transformations and refactorings as language-independent roles and contracts built on a target engine's existing program and change models. • Interaction drivers that isolate parameter collection, warning resolution, and preview from the refactoring core, so the same logic runs under interactive use and headless execution. • Refazz, a metamorphic testing method for refactoring engines that uses project tests as a partial behavioral oracle and combines parameter exploration with forced execution past rejected applicability checks and behavior-preserving warnings. • Empirical results: the migration of the legacy Pharo engine with a measurable reduction in duplication and increase in reuse, a catalog of 19 confirmed defects from 255,699 attempts, with six fixes integrated into the Pharo development branch, and a scoped proof of concept in the Python <i>rope</i> engine showing that the architectural roles are not Smalltalk-specific. Beyond their scientific contribution, these results have direct practical value for the Pharo ecosystem and for authors of similar engines, as they are available as open-source solutions.
4. What are the shortcomings of the dissertation and what is their impact on the research results? The Committee did not identify any shortcomings of the dissertation that would affect the research results.
5. Explanation of the results of the originality check of the work (numerical and narrative): The originality check was performed using the iThenticate software in the library of the Faculty of Technical Sciences, on the main text of the dissertation (26,769 words). The similarity index is 3%, with 41 matches across 16 sources, and no source exceeds 1%. The largest matches come from repositories where the candidate's papers published within this research are available, and the remaining ones concern standard terminology of the field and titles of cited works. The Committee therefore concludes that the submitted doctoral dissertation is an original work of the candidate Baša Šarenac.

X	PROPOSAL:
Based on the indicated information, the committee proposes:	
<u>a) to accept the doctoral dissertation and approve the candidate's defense;</u> b) to return the doctoral dissertation to the candidate for revision (to be supplemented or amended); c) to reject the doctoral dissertation.	

Place and date: Novi Sad, 14.09.2026.

1. Dr Miroslav Zarić, Full Professor
_____, Chair of the Committee
2. Dr Milena Frtunić Gligorićević, Assistant Professor
_____, Member
3. Dr Gordana Rakić, Assistant Professor
_____, Member
4. Dr Igor Dejanović, Full Professor
_____, Member
5. Dr Stéphane Ducasse, Research Director
_____, Mentor
6. Dr Gordana Milosavljević, Full Professor
_____, Mentor

NOTE: A committee member who does not want to sign the report because they disagree with the majority opinion of the committee is obliged to provide an explanation or reasons why they do not want to sign the report and to sign it accordingly.