

УНИВЕРЗИТЕТ У БЕОГРАДУ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

Никола Пејић

ПОБОЉШАЊЕ СИСТЕМА
ЗА РЕЦЕНЗИЈУ СОФТВЕРСКИХ ИЗМЕНА
НА ОСНОВУ ИНФОРМАЦИЈА ГЕНЕРИСАНИХ
ИЗ ИСТОРИЈЕ РЕЦЕНЗИЈА

докторска дисертација

Београд, 2025

UNIVERSITY OF BELGRADE
SCHOOL OF ELECTRICAL ENGINEERING

Nikola Pejić

IMPROVING CODE REVIEW SYSTEMS
BY GENERATING INSIGHTS FROM
THE REVIEW HISTORY

Doctoral Dissertation

Belgrade, 2025

Ментор:

Др Милош Цветановић, ванредни професор
Универзитет у Београду – Електротехнички факултет

Чланови комисије:

Др Захарије Радивојевић, ванредни професор
Универзитет у Београду – Електротехнички факултет

Др Синиша Влајић, редовни професор
Универзитет у Београду – Факултет организационих наука

Др Зоран Чича, редовни професор
Универзитет у Београду – Електротехнички факултет

Датум одбране:

_____ године.

Резиме

Наслов: Побољшање система за рецензију софтверских измена на основу информација генерисаних из историје рецензија

Са порастом броја софтверских инжењера расте и потреба за аутоматизацијом одређених процеса. Једна од погодних области за аутоматизацију је процес рецензије софтверских измена, услед чињенице да су доступне велике количине података из историје рецензија сваког софтверског репозиторијума. Ова дисертација истражује неколико приступа за генерисање информација из историје рецензија и њихово коришћење за различите видове аутоматизације. Прво је представљена анализа утицаја COVID-19 пандемије на телеметрију сервиса GitHub из које је закључено да је већина догађаја наставила са претходним трендовима, али да су догађаји који се везују за активност open source заједнице и рада на документацији били у паду, док су догађаји који се везују за развој кода и комуникацију о коду имали пораст. Затим је описан систем за аутоматско генерисање експертизе инжењера на основу описа њихових претходних измена. Прототип система је имплементиран и учињен доступним групи од око 300 инжењера током периода од десет месеци, и његове користи су разматране анализом телеметрије интеракције са системом, на основу чега је закључено да већина корисника није у значајној мери интераговала са системом, уз изузетак неколико појединаца који су га активно користили. Затим је представљен систем за подсећање инжењера на рецензије софтверског кода, на који је (након имплементације) било пријављено око 40 инжењера током периода од пет месеци. Утицај система анализиран је прегледом телеметрије и анкетирањем корисника, и закључено је да је систем имао већински позитиван утицај на продуктивност инжењера. Након тога описани су неки од постојећих приступа за препоруку рецензената софтверског кода, и представљена је техника за побољшање перформанси постојећих система. Утицај технике експериментално

је утврђен на основу историјских података неколико репозиторијума из индустрије, и закључено је да већина приступа може имати користи од примене предложене технике.

Кључне речи: софтверско инжењерство; модерна рецензија софтвера; аутоматизација

Научна област: Електротехника и рачунарство

Ужа научна област: Рачунарска техника и информатика

УДК број: 621.3

Abstract

Title: Improving code review systems by generating insights from the review history

With the increase of the number of software engineers there is a growing need to automate certain processes. One convenient area that could benefit from automation is the code review process, due to the large amount of historical data available for each software repository. This thesis explores several approaches for generating insights from the code review history and their utilization for automating different parts of the workflow. First, an analysis of the impact of the COVID-19 pandemic on GitHub telemetry revealed that most events maintained their previous trends, but that events related to the open source community and documentation work experienced a decline, while those associated with code development and communication saw an increase. Next, a system is introduced that is used for generating developer expertise based on the description of the changes they made. A prototype of the system was implemented and made available to a group of around 300 developers during a period of ten months, and it's benefits were evaluated by analyzing the usage telemetry, with the conclusion being that the majority of users did not interact with the system in a meaningful way, apart from a couple of individuals who actively used it. Next, a system for automatically reminding developers about their pending code reviews is introduced, and (after implementing it) around 40 developers were subscribed to it during a period of five months. The impact of the system was analyzed based on the telemetry as well as user feedback, and the majority of users reported a positive impact on their productivity. After that, an overview of several existing reviewer recommendation approaches is given, as well as a technique for improving their performance. The effect of the technique was experimentally determined using the historical data from several repositories from the industry, and it showed that the majority of the approaches could have benefits from the proposed technique.

Keywords: software engineering; modern code review; automation

Scientific field: Electrical engineering and Informatics

Scientific subfield: Computer Engineering and Informatics

UDC number: 621.3

Садржај

Резиме	i
Abstract	iii
1 Увод	1
1.1 Ток истраживања	2
1.2 Доприноси дисертације	3
2 Развој софтвера	7
2.1 Историја колаборативног развоја и рецензије софтверског кода	7
2.1.1 Инспекција кроз састанке	7
2.1.2 Тежња ка асинхроној инспекцији и смањењу састанака	11
2.1.3 Појављивање првих алата	12
2.2 Модерни процес развоја и рецензије софтверског кода	14
2.3 Системи за колаборативни развој и рецензију софтверског кода	16
2.3.1 Azure DevOps	16
2.3.2 GitHub	21
2.4 Секундарни извори телеметрије	23
2.4.1 GH Archive	23
2.4.2 GH Torrennt	24
3 Утицај COVID-19 пандемије на GitHub	27
3.1 Постојећа истраживања	28
3.1.1 Истраживање 1	29
3.1.2 Истраживање 2	30
3.1.3 Истраживање 3	31
3.2 Поставка проблема	31
3.3 Предлог истраживања	31

3.4	Детаљи имплементације	32
3.5	Евалуација	34
3.6	Дискусија	38
4	Претраживање експерата и предлога измена	41
4.1	Постојећа решења	42
4.1.1	TF-IDF	42
4.1.2	Постојећи алати у окружењима за колаборативни развој софтвера	42
4.2	Поставка проблема	43
4.3	Предлог решења	44
4.4	Детаљи имплементације	45
4.5	Евалуација	49
4.6	Дискусија	50
5	Подсећање рецензената	53
5.1	Постојећа решења	54
5.1.1	Nudge	54
5.1.2	Briefing emails (Viva Insights)	57
5.2	Поставка проблема	58
5.3	Предлог решења	58
5.4	Евалуација идејног решења	59
5.5	Детаљи имплементације	62
5.6	Евалуација имплементiranог решења	63
5.7	Дискусија	68
6	Системи за препоруку рецензената	71
6.1	Постојећа решења	72
6.1.1	Приступ 1	72
6.1.2	Приступ 2	75
6.1.3	Приступ 3	76
6.1.4	Приступ 4	77
6.1.5	Приступ 5	77
6.1.6	Приступ 6	78
6.1.7	Приступ 7	79
6.2	Поставка проблема	82
6.3	Предлог решења	86

6.4	Детаљи имплементације	87
6.5	Евалуација	88
6.6	Дискусија	96
7	Закључак	101
A	Системи за колаборативни развој и рецензију софтвера	105
A.1	GitLab	105
A.2	BitBucket	108
	Литература	122
	Списак слика	123
	Списак табела	127
	Списак листинга	129
	Биографија аутора	131

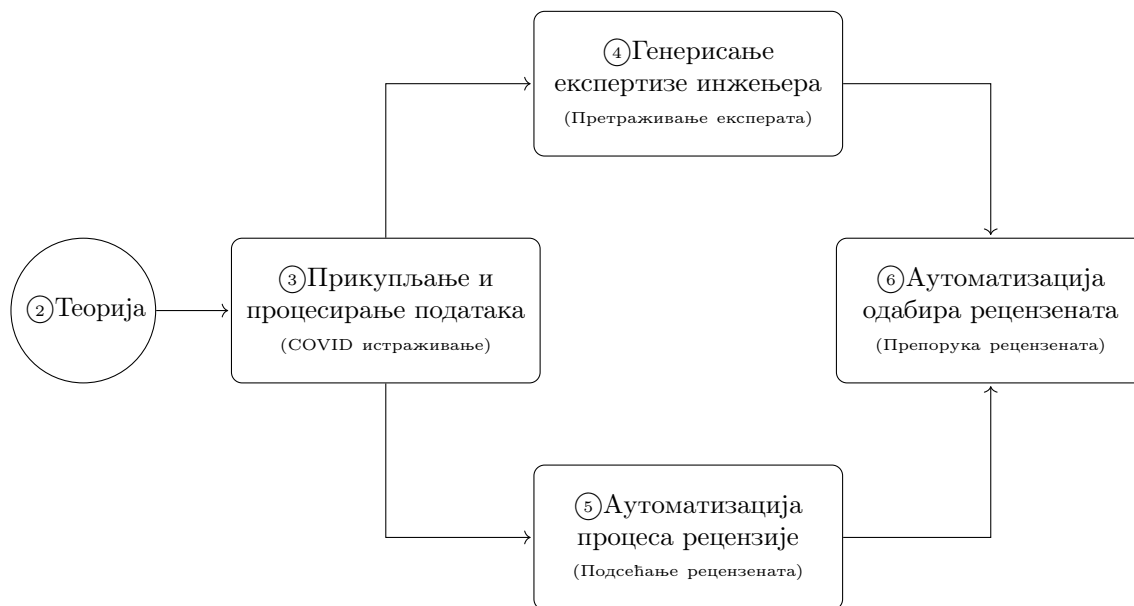
1 Увод

Како број софтверских репозиторијума и број инжењера који раде на њима расте сваке године [1], постаје све битније да се аутоматизују честе и једноставне акције које одузимају време, посебно услед чињенице да се софтверски инжењери осећају све оптерећеније [2]. Једна од области која би могла имати користи од аутоматизације је процес рецензије софтверских измена (*code review*) [3]. Када софтверски инжењер заврши са имплементирањем нове функционалности, он пошаље предлог измена који је обично потребно да одобри макар још један инжењер (рецензент) како би измена била прихваћена. Инжењери који су одабрани да врше рецензију остављају коментаре и предлажу додатна побољшања, и тек када су задовољни квалитетом предложених измена одобравају њихово прихватање.

Процес слања предлога измена и рецензије софтверског кода садржи велику количину података. На пример, одабиром рецензената за дати предлог измена даје се сигнал да су дати рецензенти експерти у делу кода који се мења, што може бити битна информација код великих репозиторијума са многобројним инжењерима, јер у њима рецензенти имају тенденцију да се специјализују за само одређене делове кода [4]. Са друге стране, одабир рецензената може да утиче на квалитет [3, 5, 6], као и на брзину рецензије [7, 8].

Истраживања која су спроведена у оквиру ове докторске дисертације имају за циљ екстракцију података садржаних у рецензијама зарад генерисања информација које могу бити релевантне инжењерима, као и креирање система за аутоматизацију акција које могу олакшати процес развоја и рецензије софтверског кода. Истраживања су спроведена у реалном окружењу софтверске индустрије, или користећи податке из индустрије и open-source заједнице, и зато се очекује да могу бити примењена у било којем реалном окружењу.

1.1 Ток истраживања



Слика 1.1: Ток истраживања дисертације, где бројеви фаза одговарају индексу поглавља.

Истраживање које је обрађено у овој дисертацији вршено је кроз неколико фаза (Слика 1.1), где је свака фаза обрађена у засебном поглављу. Истраживање је кренуло од теоријске основе развоја софтвера које је обрађено у другом поглављу. Ово поглавље садржи преглед историје развоја и рецензије софтвера од својих зачетака до модерног процеса рецензије софтвера. Након тога обрађено је и неколико модерних сервиса за колаборативни развој и рецензије софтвера, а како би се користили подаци из историје рецензија описани су и начини за дохватање података са поменутих платформи.

Затим је приступљено практичним деловима истраживања, где је прво било потребно дохватити скупове података са различитих платформи. Кренуло се од платформе GitHub, где је након дохватања података урађено једноставно процесирање (пребројавање) различитих догађаја. Анализом трендова броја догађаја пре и током COVID-19 пандемије проучаван је њен утицај на open source развој на платформи GitHub. Прикупљање података, њихово процесирање, као и добијени резултати обликовани су у независну истраживачу целину која је обрађена у трећем поглављу.

Након тога се приступило детаљнијој анализи података како би се олакшало претраживање експерата, што је област у којој је примећен простор за допринос тренутним системима за колаборативни развој. Четврто поглавље бавило се имплементацијом система који извлачи кључне речи из описа претходних измена и одређује њихове тежинске факторе, како би

уз њихову помоћ омогућио претраживање инжењера. Систем је учињен доступним групи инжењера из индустрије током периода од десет месеци, а затим је анализирана телеметрија о коришћењу имплементираног система.

Током експеримента из четвртог поглавља приликом интеракције са инжењерима примећен је чест проблем неактивних рецензената. Зато је покренуто независно истраживање које се фокусирало на решавање овог проблема и аутоматизацију процеса рецензије, и то истраживање описано је у петом поглављу. Имплементиран је систем за подсећање рецензената на рецензије које чекају на њих, уз анализу резултата примене таквог система над скупом инжењера/рецензената једног репозиторијума током периода од пет месеци. Анализирани су резултати две анкете, где се једна бавила испитивањем потребе и преференци корисника пре имплементације система, док се друга бавила испитивањем сентимента корисника након периода коришћења, а такође је анализирана и телеметрија о раду система.

Након завршетка истраживања из четвртог и петог поглавља, примећена је област која представља комбинацију претходне две. Наиме, генерисање експертизе инжењера се може користити за препоруку (односно аутоматизацију) одабира рецензената, и неколико постојећих метода је управо то и радило. Зато се даљи ток истраживања фокусирао на системе за препоруку рецензената. Шесто поглавље даје преглед неколико постојећих приступа за препоруку рецензената. Затим је представљена нова техника за побољшање перформанси постојећих приступа која их фокусира само на релевантне рецензенте, где се релевантност одређује на основу више различитих критеријума. Мерења перформанси вршена су над историјским подацима неколико репозиторијума из индустрије. На самом крају је закључак који даје преглед главних резултата дисертације, и даје предлоге за потенцијалне правце даљег рада.

1.2 Доприноси дисертације

Полазне хипотезе истраживања у оквиру ове докторске дисертације су:

1. Постоје дугорочне промене у трендовима догађаја open source заједнице.
2. Постоји потреба за додатним системом за претрагу експерата и предлога измена у системима за колаборативни развој софтвера.

3. Софтверски инжењери желе да примају периодичне нотификације о рецензијама софтверског кода.
4. Периодичне нотификације о рецензијама софтверског кода могу да имају утицај на продуктивност рецензената.
5. Могуће је побољшати перформансе система за препоруку рецензената њиховим фокусирањем само на рецензенте који испуњавају одређене једноставне критеријуме, уместо разматрања свих рецензената који су радили на датом софтверском репозиторијуму.
6. Могуће је побољшати перформансе система за препоруку рецензената њиховим фокусирањем само на рецензенте који испуњавају макар један једноставан критеријум из комбинације једноставних критеријума, уместо разматрања свих рецензената који су радили на датом софтверском репозиторијуму.
7. Критеријуми који ће донети највише побољшања система за препоруку рецензената се могу разликовати од система до система.
8. Критеријуми који ће донети највише побољшања система за препоруку рецензената се могу разликовати у зависности од репозиторијума над којим се врши мерење.

Научни и стручни доприноси ове дисертације су:

1. Систематичан преглед литературе у области рецензије софтверских измена.
2. Анализа трендова догађаја у системима за рецензију софтверских измена на примеру платформе GitHub.
3. Пројектовање и евалуација система за аутоматско генерисање експертизе инжењера на основу описа њихових претходних измена.
4. Пројектовање и евалуација система за подсећање рецензената на измене које захтевају њихову пажњу.
5. Систематичан преглед литературе у области препоруке рецензената софтверских измена.
6. Имплементација и евалуација метода за препоруку рецензената софтверских измена описаних у доступној литератури.

7. Предлог нових техника за побољшање перформанси система за препоруку рецензената софтверског кода.
8. Експериментално утврђивање обима очекиваних побољшања применом предложених техника над различитим репозиторијумима.
9. Експериментално утврђивање обима очекиваних побољшања применом предложених техника над различитим методама за препоруку рецензената.

Делови резултата истраживања описаних у овој дисертацији објављени су у следећим публикацијама:

M22 Истакнути међународни часопис

- **N. Pejić**, Z. Radivojević and M. Cvetanović, "Helping Pull Request Reviewer Recommendation Systems to Focus,"IEEE Access, св. 11, стр. 71013-71025, 2023, doi: <https://doi.org/10.1109/ACCESS.2023.3292056>
- **N. Pejić**, Z. Radivojević and M. Cvetanović, "Analyzing the Impact of COVID-19 on GitHub Event Trends. Sustainability", Sustainability 15, св. 19, 2023, doi: <https://doi.org/10.3390/su151914622>

2 Развој софтвера

Софтверско инжењерство се доста променило од првих програма писаних у асемблеру до данас. У оквиру овог поглавља укратко је описан модерни процес развоја и рецензије софтверског кода (2.2). Након тога представљено је неколико популарних система за колаборативни развој софтвера (2.3) уз преглед података који су доступни о свакој рецензији. Представљени су начини на који се ти подаци могу дохватити директно из система за колаборативни развој, као и из неколико познатијих секундарних извора (2.4).

2.1 Историја колаборативног развоја и рецензије софтверског кода

Како би се боље разумео модерни процес развоја и рецензије софтвера, у наставку је дат кратак историјски преглед различитих приступа овом процесу. Представљене су прве формалније дефиниције процеса рецензије софтверског кода са мноштвом састанака, као и неколико покушаја да се процес оптимизује смањењем времена које се проводи на састанцима. На крају је представљена фаза у којој су кренули да се користе посебни алати којима је омогућено дистрибуирано и асинхроно вршење рецензија, тако да сваки рецензент може да прегледа код независно од осталих рецензената у времену када њему одговара.

2.1.1 Инспекција кроз састанке

Једна од првих формалнијих дефиниција процеса развоја и рецензије софтверског кода дата је 1976. године [9], и по њој се развој софтвера врши у следећим фазама:

1. Дефинисање циљева
2. Дизајн:
 - (а) архитектуре
 - (б) екстерних спецификација
 - (в) интерних спецификација
 - і. I_0 инспекција
 - (г) спецификација логике
 - і. I_1 инспекција целог дизајна
3. Кодирање:
 - (а) кодирање/имплементација
 - і. I_1 инспекција кода
 - (б) јединични тестови
4. Тестирање:
 - (а) тестови функција
 - (б) тестови компоненти
 - (в) тестови система
5. Објављивање измене

Процес развоја креће се од прве до последње фазе, осим када инспекције пронађу проблем са резултатом неке од претходних фаза. Тада се враћа на прву фазу која треба да изврши неку од исправки, и наставља се са секвенцијалним проласком кроз фазе. Итерирање кроз инспекције и исправке се врши док се не задовољи квалитет резултата, ма колико итерација да је потребно. Разлог зашто се велики акценат ставља на инспекције и рану детекцију проблема је зато што је закључено да је касније откривање и исправљање грешака јако скупо, и да је зато пожељно наћи грешке ближе фази у којој су настале.

Корист инспекција током дизајнирања и имплементације решења видљив је у Табели 2.1, која приказује број грешака по 1000 линија кода (без коментара) које су детектоване током одговарајућих фаза, у експерименту који је спроведен у оригиналном раду 1976. године. Највећи број грешака пронађен је током процеса креирања решења, док је остатак детектован

Табела 2.1: Број грешака који су пронађени током различитих фаза развоја. [9]

Фаза	број грешака по 1000 линија кода без коментара	проценат од укупних грешака
Дизајн		
I_1 инспекција	38	82
кодирање		
I_2 инспекција		
Јединични тестови	8	18
Припрема за тестове прихватања		
Тестови прихватања	0	0
Коришћење (6 месеци)	0	0
Укупно	46	100

током јединичних тестова и припрема за његово тестирање. Током шестомесечног периода његовог коришћења није пронађена ниједна грешка након објављивања измена/производа.

„Инспекције су формални, ефикасан, и економичан метод проналажења грешака у дизајну и коду.”[9] Инспекције се врше у тимовима које чине следећи чланови:

1. **Модератор** - особа која води тим за инспекцију, заказује састанке, и пише извештаје инспекције. Обично је искуснији програмер, али је пожељно да не ради директно на пројекту над којим се врши рецензија како би био објективан.
2. **Дизајнер** - програмер који је одговоран за дизајн решења/измене. Уколико је иста особа радила и на дизајну и на кодирању (и на тестирању), та особа ће имати улогу искључиво дизајнера у тиму за инспекцију, а неко други ће преузети улогу кодера (а неко трећи улогу тестера) зарад спровођења објективне инспекције.
3. **Кодер** - програмер који је одговоран за претварање дизајна у код (кодирање).
4. **Тестер** - програмер који је одговоран за писање и извршавање тестова.

По оригиналном раду, укупно време које је потребно да се изврше сви делови I_1 и I_2 инспекција четворочланог тима за инспекције износи око 100 особа-сати (или мање, за једноставније системе). Препоручује се да се састанци за инспекције заказују у трајању не дужем од 2 сата (јер је примећено да у супротном ефикасност инспекције опада), и да је прихватљиво да се (по потреби) закажу 2 двочасовне сесије у току дана. Препорука је да се

на инспекције гледа са истим приоритетом као и на друге битне задатке на пројекту, јер у супротном постоји тенденција да се инспекције одлажу зарад другог посла, што може негативно утицати на планове и време израде целог пројекта.

Свака инспекција се састоји од следећих делова:

1. **Преглед** - учествује цео тим. Дизајнер описује област и детаље измене коју је дизајнирао. Пратећа документација о дизајну се дели са свим члановима инспекцијског тима.
2. **Припрема** - врши сваки члан инспекцијског тима за себе. Свако „ради домаћи“, читајући и анализирајући дизајн документацију.
3. **Инспекција** - учествује цео тим. Неко од чланова инспекцијског тима (обично **кодер**) описује како би вршио имплементирање решења описаног дизајн документом. Сваки део логике и свака грана условног извршавања треба да је обрађена макар једном, и у њима се траже пропусти и грешке. Циљ ове фазе је да се само препознају и класификују све грешке - не дискутује се о начинима за њихово отклањање. Очекује се да након завршетка ове фазе, модератор у року од једног дана напише извештај о инспекцији и грешкама које су пронађене, како би се осигурало да сви проблеми буду адресирани током додатне прераде дизајна и имплементације.
4. **Прерада** - врше дизајнер и кодер. Они отклањају све грешке које су примећене током инспекције.
5. **Наставак** - модератор. Модератор проверава да ли су све грешке адресиране. Уколико је више од 5% дизајна или имплементације промењено, од модератора се очекује да поново сазове инспекцијски тим и понови целу процедуру инспекције целе измене од прве фазе. Уколико је промењено мање од 5%, модератор може да одлучи да сам верификује квалитет корекција, или да сазове инспекцијски тим ради провере целе промене или само корекција.

Како је наведено у оригиналном раду, инспекцијама програмер добија брзи увид у квалитет свог рада, и може да ради на унапређивању својих вештина. Такође је наглашено да ни под којим условима процес инспекције не треба да се користи како би се процењивале перформансе програмера.

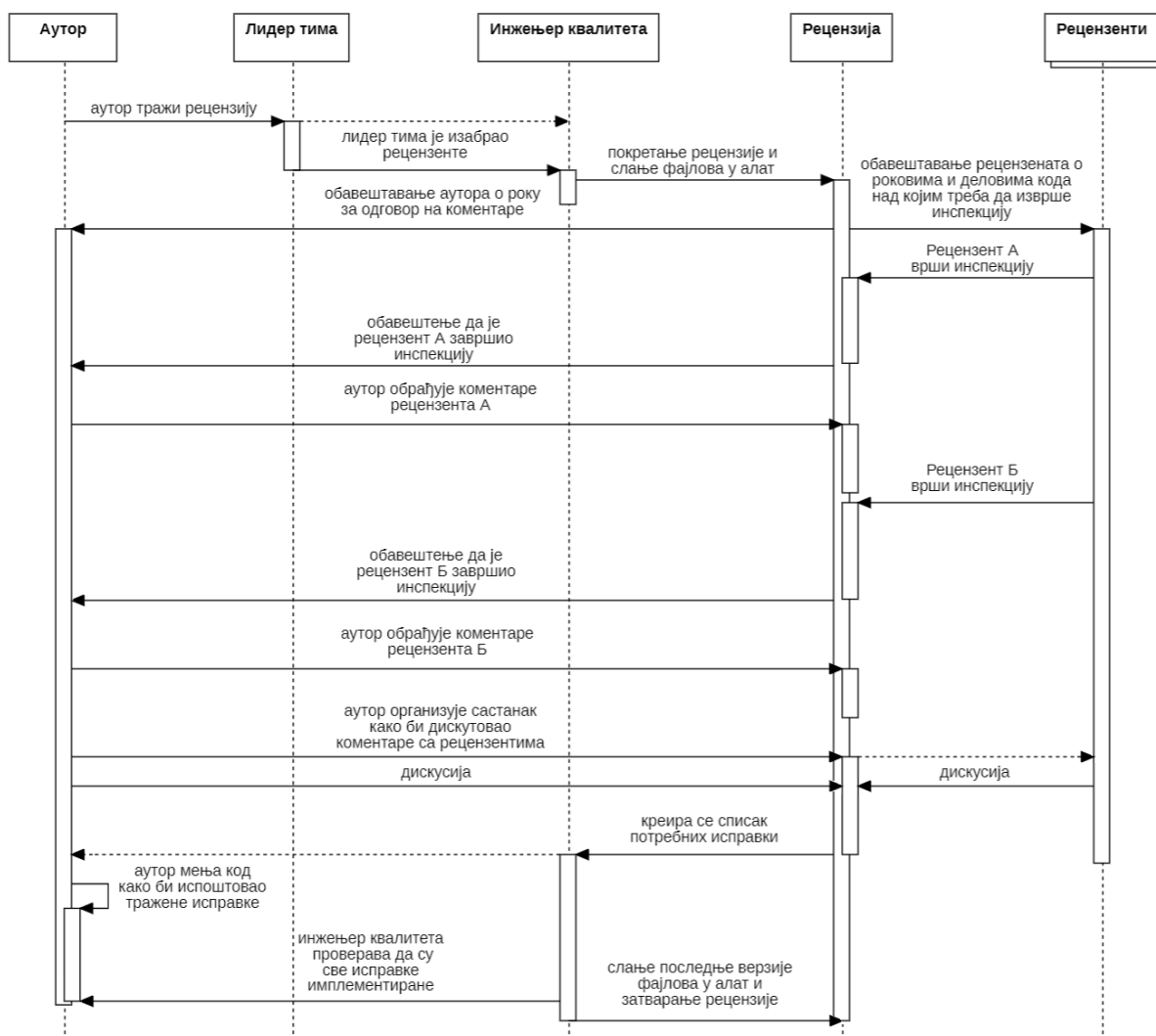
2.1.2 Тежња ка асинхроној инспекцији и смањењу састанака

Након прве верзије дефинисане 1976. године, било је неколико варијација процеса рецензије софтверског кода, као што су тимови за рецензију са мање или више чланова [10–12], или мање модификације фаза и структура тимова за инспекцију кода [13–15]. Али, поменуте варијације се нису суштински разликовале од оригиналног процеса све до 90-их година, када је полако кренуло са разматрањем неопходности и ефикасности инспекцијских састанака.

У 1993. години објављен је рад у којем је анализирана ефикасност инспекцијских састанака [16]. Како се показало, може се потрошити доста времена у чекању на термин за састанак када су сви чланови инспекцијског тима слободни. Инспекцијски састанци имају јасне користи, као што су:

- обучавање мање искусних рецензената посматрањем и сарадњом са искуснијим колегама
- синергијом и такмичарским окружењем на инспекцијским састанцима (као месту где може да се покаже експертиза пред колегама) могу се пронаћи проблеми који се иначе не би нашли у појединачној инспекцији
- заказивањем инспекцијских састанака дају се јасни рокови

Са друге стране, на основу посматрања више инспекцијских састанака закључено је да, иако постоје наведене предности, време које се улаже у састанке можда није најбоље искоришћено. На пример, иако су инспекцијски састанци корисни за обучавање мање искусних колега, исто се може постићи за мање уложеног времена, на пример, писањем квалитетне документације или прављењем курса који би помогли у обуци запослених. Тачно је да се услед синергије на састанцима пронађу додатни дефекти, али на основу истраживања они у просеку представљају само око 4% од укупних пронађених дефеката. Обично је потребно неколико дана како би се инспекцијски тим окупио и извршио инспекцију. Такође, анализом инспекцијских састанака примећено је да обично само 2 члана инспекцијског тима активно дискутују у било којем тренутку, док остали чланови за то време слушају и повремено се укључују у конверзацију. Предлагане оптимизације времена укључивале су различите формате састанака у мањим саставима [16, 17], али се као опција (иако не превише атрактивна у том тренутку) јавља и могућност вршења рецензије у вербалној или писаној форми, без састанака [16, 18].



Слика 2.1: Дијаграм секвенци који упрошћено приказује процес рецензије кода 2000-их година (на основу описа из [19]).

2.1.3 Појављивање првих алата

Тенденција да се смањи број инспекцијских састанака и да се у већој мери пређе на асинхрони вид рецензије софтверског кода настављена је и наредних година, али тек са појавом и ширењем интернет технологија долази до значајнијих промена у процесу рецензије. ICICLE (*Intelligent Code Inspection Environment in a C Language Environment*) из 1990. године [20] је један од првих алата који је дизајниран како би се олакшао и убрзао процес инспекције софтверских измена. Корисник ICICLE алата може да врши инспекцију над свим фајловима из измене кроз саму апликацију, одакле такође може да остави коментар на било којој линији, као и да види све остале коментаре. Алат има опцију и да

аутоматски препозна неке честе грешке и да остави коментар којим скреће на погрешна места у коду. Сви коментари и резултати дискусије током инспекцијских састанака бележе се у самом алату, и тада се коментари прихватају или одбијају (означавају са „*accepted*” или „*rejected*”). Алат такође садржи и различите функционалности које олакшавају инспекцијски састанак, као што је синхронизација приказа свих инстанци алата са оним којим управља вођа инспекцијског састанка (како би се осигурало да сви ученици могу да виде релевантне делове кода), могућност отварања додатних прозора како би се детаљније анализирао код, преглед само коментара који нису обрађени (они који нису прихваћени или одбачени), могућност „показивања” на одређени део кода од стране једног корисника и приказивање показивача у прозорима осталих корисника, итд. Аутори ICICLE рада су разматрали могућност инспекцијског састанка на даљину, али је нису реализовали.

Затим, било је и неколико алата који су сакупљали коментаре рецензената на једном уређају, али који су захтевали ручно слање и дељење са осталим рецензентима (InspeQ [21]), као и алата који су омогућили дистрибуирано сакупљање коментара инспектора који су на различитим географским локацијама (Scrutiny [22], CSI - Collaborative Software Inspection[23]), што је било једно од ограничења првих алата као што је ICICLE.

Алати за инспекцију кода су наставили да се развијају током 1990-их и 2000-их година [24], и 2007. године објављен је рад који даје добар преглед процеса инспекције коришћењем тадашњих алата, и који је свео број и обим састанака на минимум [19]. Алат је развијен користећи програмски језик *PHP*, *MySQL* базу података, и *Apache Server*.

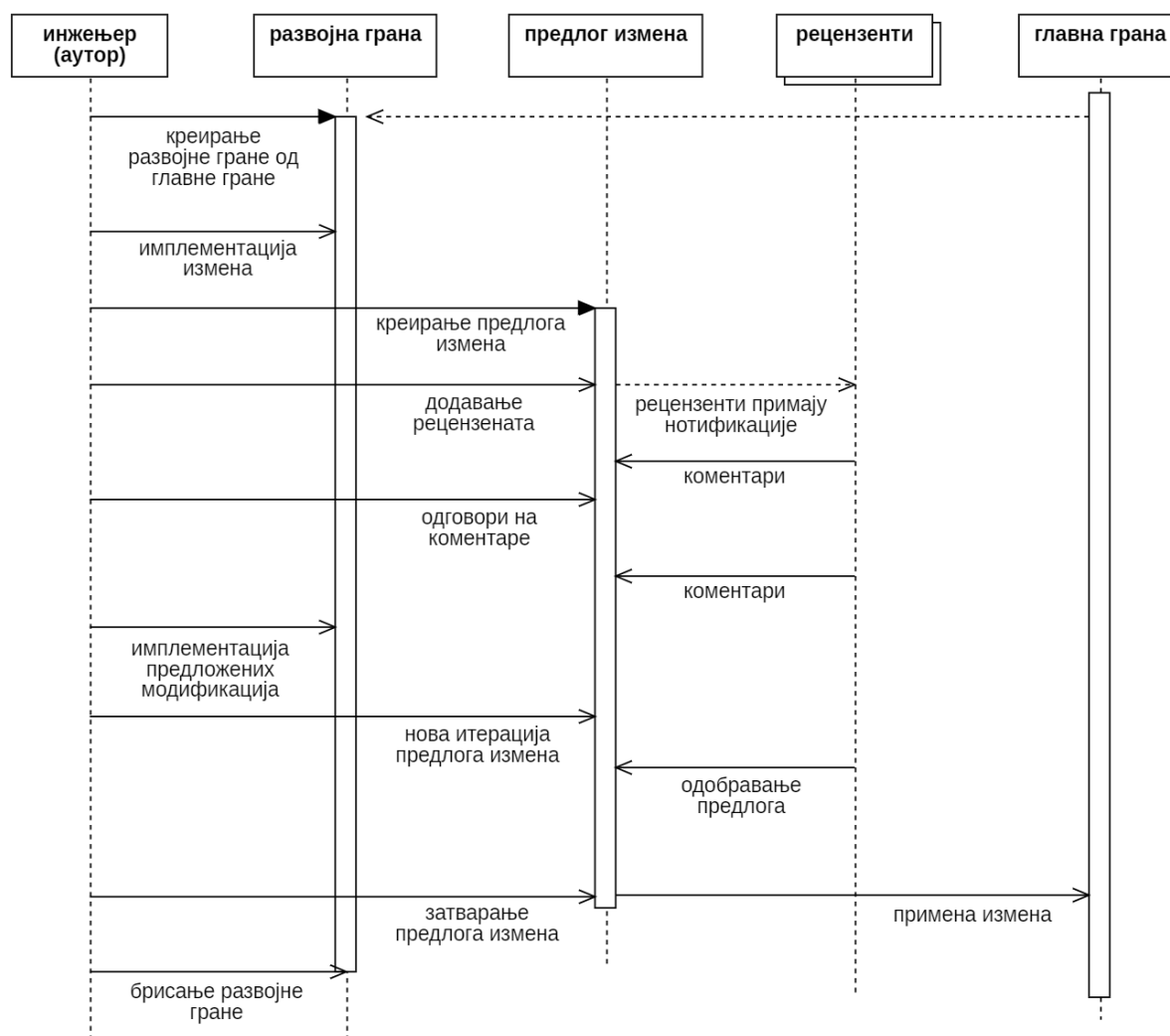
По овом систему (приказаном на Слици 2.1), аутор и инжењер квалитета софтвера (*SQE - Software Quality Engineer*) управљају процесом рецензије и деле измену на функционалне целине. Идеја је да сваки рецензент врши рецензију само одређеног дела кода који је њима додељен. Тиме се олакшава рецензија великих измена, јер није неопходно да се рецензенти упознају са свим детаљима, што може да захтева велике количине времена и пажње самих рецензената. Сваки рецензент пролази кроз списак ставки које су генерисане од стране алата, проверава их, и бележи коментаре у самом алату. Алат омогућава селекцију делова кода и остављање таргетираних коментара. Коментари које оставе рецензенти похрањују се у базу података, којој могу приступити само аутор и инжењер квалитета. Кроз алат, аутор може да одговори на сваки коментар, као и да постави статус коментара на „слажем се”, „не слажем се”, и „дискусија” („*agree*”, „*disagree*”, и „*discuss*”, респективно). Сваки рецензент може да види одговоре аутора на своје коментаре.

Иако се коришћењем ових алата доста олакшава асинхрона рецензија и смањује потреба за синхронизацијом чланова инспекцијског тима, и даље се организује један краћи састанак аутора и свих рецензената како би се продискутовало о свим остављеним коментарима и акцијама које су потребне да би се они адресирали, посебно у случају коментара које је аутор означио са „не слажем се” или „дискусија”.

Предности предложеног процеса су да се највећи део инспекције врши асинхронно, тако да аутор и рецензенти могу да процесуирају своје делове инспекције у време када њима одговара. Након што сваки појединачни рецензент заврши са својом инспекцијом аутор одмах добије повратне информације, и јако мало времена се троши на саму тријажу коментара, услед ограниченог броја статуса (слагање и неслагање, као и додатна дискусија), и на једином састанку између аутора и рецензената фокус је само на коментарима где постоји неслагање или које је потребно додатно дискутовати. Коришћењем онлајн алата олакшан је процес остављања и прикупљања коментара, и време рецензије је скраћено. Како су сви подаци о рецензијама доступни на једном месту у структурираном формату, они релативно једноставно могу да се дохвате и анализирају ради унапређивања естимација и процеса.

2.2 Модерни процес развоја и рецензије софтверског кода

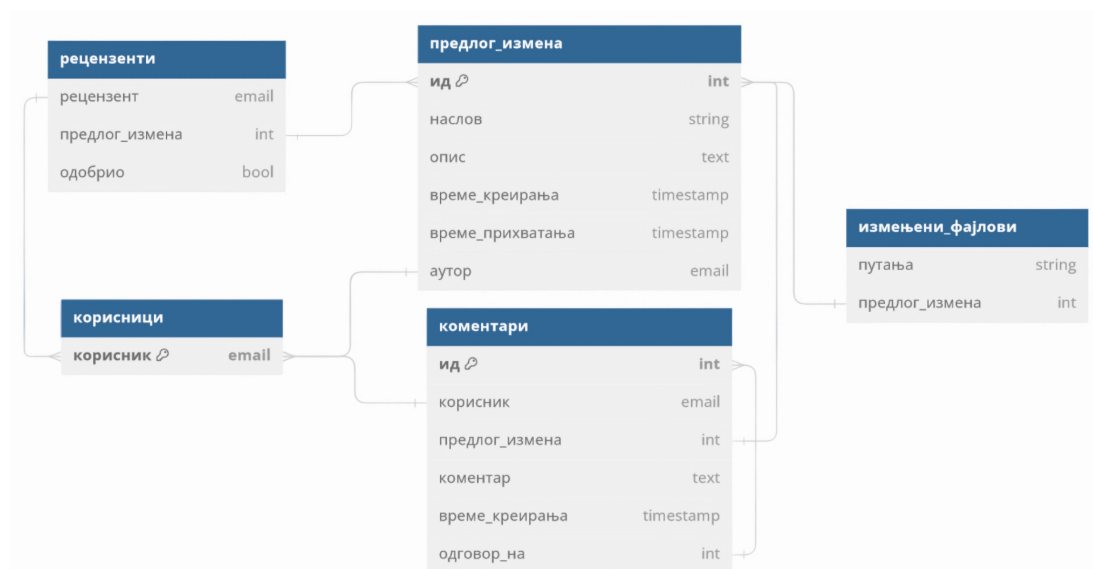
Модерни процес развоја и рецензије једне измене илустрован је на Слици 2.2 [25]. Када софтверски инжењер треба да имплементира неку нову измену, прво креира развојну грану на основу последњих измена са главне гране (*main* или *master* гране). Затим приступа имплементацији и тестирању планираних измена. Када је задовољан њиховим квалитетом, он креира предлог измена који, осим самих измена, садржи и њихов кратак текстуални опис. Како би се побољшао квалитет кода у главној грани и онемогућило насумично мењање кода, често је неопходно да макар један рецензент одобри измене да би се оне примениле над главном граном. Инжењер може сам да дода одговарајуће рецензенте, или они могу бити аутоматски додати уколико постоји полиса или систем за аутоматски додавање рецензената. Рецензенти који се додају на рецензију добију нотификације и у неком тренутку (када имају довољно времена) прегледају предложене измене и могу оставити коментаре у којима траже појашњења неких делова измена или дају савете како се може додатно побољшати њихов квалитет. Аутор измена затим одговара на коментаре рецензената и/или имплементира предложена побољшања и објављује нову итерацију измена. Процес остављања коментара и одговарања на њих уз нове итерације промена се понавља док рецензенти не одлуче да је постигнут задовољавајући квалитет измена, и тада их они одобравају. Након тога



Слика 2.2: Дијаграм секвенци који упрошћено приказује процес развоја и рецензије кода (© 2023 IEEE). [25]

предлог измена се затвара и означава као прихваћен, и врши се аутоматска примењивање предложених измена над кодом из главне гране.

Осим претходно поменутих наслова и описа измена, предлози измена описани су и подацима као што су време креирања, време прихватања, и имејл адресом аутора, као што се може видети на Слици 2.3. Даље, предлог измена садржи и списак измењених фајлова, као и листу рецензената са информацијом да ли су одобрили измене. Коначно, рецензенти и аутор могу да оставе нове коментаре или да одговоре на постојеће коментаре, где коментари осим текста садрже и метаподатке о времену креирања.



Слика 2.3: Упрошћени приказ структуре података који су доступни за сваки предлог измена.

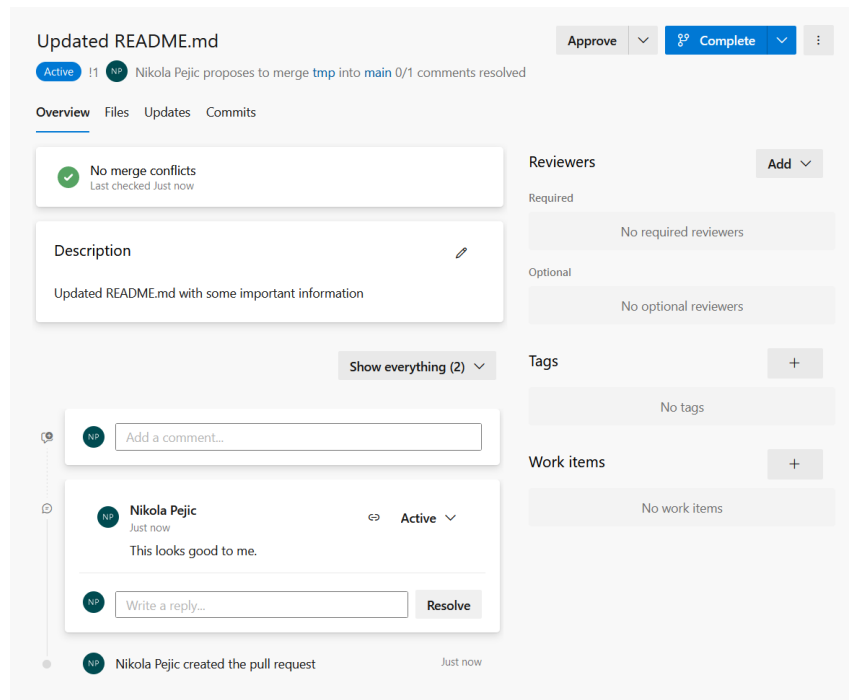
2.3 Системи за колаборативни развој и рецензију софтверског кода

На тржишту данас постоји неколико система за дистрибуирани развој и рецензију софтверског кода [26]. Уобичајено је да, осим графичких интерфејса, системи пружају и неку врсту програмских интерфејса за једноставан аутоматски приступ подацима и интеракцију са рецензијама. У овој секцији биће представљена два система (*Azure DevOps* и *GitHub*), док су у Прилогу А дати примери још два система (*GitLab* и *BitBucket*).

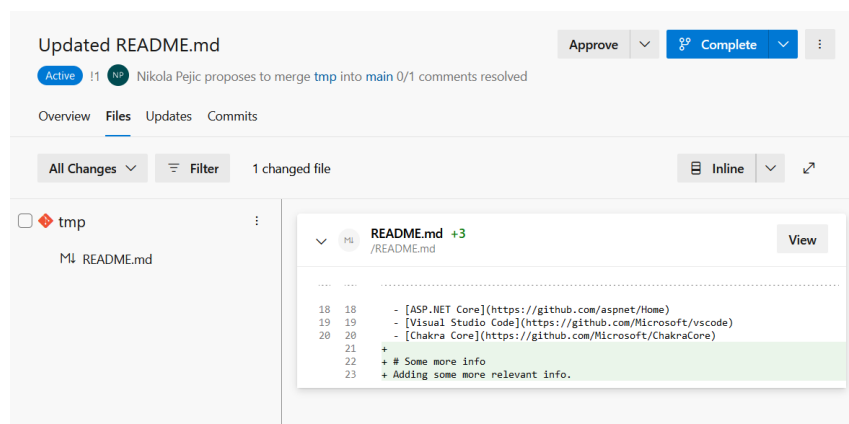
2.3.1 Azure DevOps

Azure DevOps (ADO) је сервис за колаборативни развој софтвера компаније Microsoft [27]. Осим система за верзионисање кода (*git* [28] и *Team Foundation Version Control* [29]), ADO нуди и сервисе за организацију и праћење послова (boards [30]), као и системе за аутоматско компајлирање, тестирање и објављивање софтвера (pipelines [31]) [32].

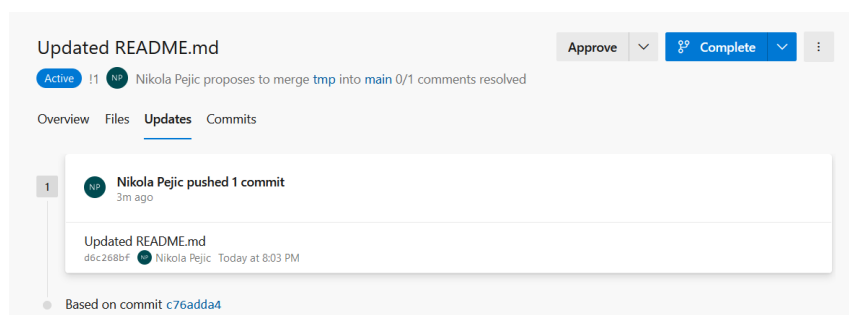
Предлагање измена и рецензија је слично на већини модерних система, и пример једног предлога измена на ADO-у може се видети на Слици 2.4. Наслов овог предлога измена је „*Updated README.md*”, тренутни статус је активан (*Active*), аутор измена је аутор тезе (*Nikola Pejić*), и опис измена (*Description*) је „*Updated README.md with some important information*”. Са десне стране се могу видети додељени рецензенти (*Reviewers*) који припадају



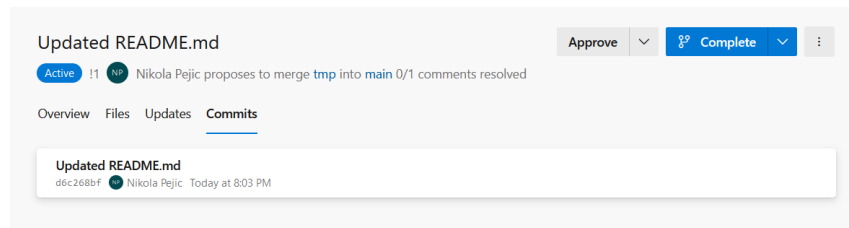
Слика 2.4: Пример предлога измена на сервису Azure DevOps - преглед.



Слика 2.5: Пример предлога измена на сервису Azure DevOps - фајлови.



Слика 2.6: Пример предлога измена на сервису Azure DevOps - итерације.



Слика 2.7: Пример предлога измена на сервису Azure DevOps - измене.

двема групама: обавезним (*Required*) и опционим (*Optional*), и у овом фиктивном примеру једини рецензент је аутор тезе. Пример измене има само један коментар који је оставио аутор тезе, чији је текст „*This looks good to me.*”, и чије је стање активно (*Active*), што означава да коментар није обрађен.

Затим, у табу *Files* на Слици 2.5 може се видети преглед измењених фајлова (у случају овог примера то је само фајл „*README.md*”), као и измењене линије у датим фајловима (у овом примеру додате су линије 21-23). У табу *Updates* (Слика 2.6) могу се видети итерације које дати предлог измена садржи, где једна итерација обухвата све измене које су направљене над локалном копијом кода пре првог слања на сервис (у случају прве итерације), односно између било која два слања на сервис (у случају друге и свих наредних итерација). У овом примеру постоји само једна итерација, која је садржала само једну измену (*commit*). Сличан приказ може се видети и у табу *Commits* (Слика 2.7), само што овде појединачне измене нису груписане у итерације.

-
- 1 GET `https://dev.azure.com/{organization}/{project}/_apis/git/repositories/{repositoryId}/pullrequests?api-version=7.1`
-

Листинг 2.1: HTTP GET захтев за дохватање података о предлозима измена и рецензијама са сервиса Azure DevOps

Зарад лакшег приступа подацима о предлозима измена и рецензија, ADO нуди и REST [33] интерфејс [34]. Слањем HTTP GET захтева који је дат у Листингу 2.1, уз спецификацију одговарајућих параметара (идентификатора организације, пројекта, и репозиторијума, као и уз одговарајућу аутентикацију) [35], може се итеративно дохватити преглед целокупне историје предлога измена и рецензија једног репозиторијума. Пример одговора који се може добити слањем претходног захтева дат је у Листингу 2.3.

```

1 {
2   "value": [
3     {
4       "repository": {
5         "id": "3411ebc1-d5aa-464f-9615-0b527bc66719",
6         "name": "2016_10_31",
7         "url": "https://dev.azure.com/fabrikam/_apis/git/repositories/3411ebc1-d5aa-464f-9615-0b527bc66719",
8         "project": {
9           "id": "a7573007-bbb3-4341-b726-0c4148a07853",
10          "name": "2016_10_31",
11          "state": "unchanged"
12        }
13      },
14      "pullRequestId": 20,
15      "codeReviewId": 20,
16      "status": "completed",
17      "createdBy": {
18        "id": "d6245f20-2af8-44f4-9451-8107cb2767db",
19        "displayName": "Normal Paulk",
20        "uniqueName": "fabrikamfiber16@hotmail.com",
21        "url": "https://dev.azure.com/fabrikam/_apis/Identities/d6245f20-2af8-44f4-9451-8107cb2767db",
22        "imageUrl": "https://dev.azure.com/fabrikam/_api/_common/identityImage?id=d6245f20-2af8-44f4-9451-8107cb2767db"
23      },
24      "creationDate": "2016-11-01T16:28:08.8900118Z",
25      "closedDate": "2016-11-01T16:30:02.0413914Z",
26      "title": "Updated pull request title",
27      "description": "Updated pull request description",
28      "sourceRefName": "refs/heads/npaulk/my_work",
29      "targetRefName": "refs/heads/new_feature",
30      "mergeStatus": "succeeded",
31      "mergeId": "1b5f1536-565e-46f4-ace2-e8cbf50c676b",
32      "lastMergeSourceCommit": {
33        "commitId": "ae875f124fe5c625664fae12af26cc294ed1575f",
34        "url": "https://dev.azure.com/fabrikam/_apis/git/repositories/3411ebc1-d5aa-464f-9615-0b527bc66719/commits/ae875f124fe5c625664fae12af26cc294ed1575f"
35      },
36      "lastMergeTargetCommit": {
37        "commitId": "f47bbc106853afe3c1b07a81754bce5f4b8dbf62",
38        "url": "https://dev.azure.com/fabrikam/_apis/git/repositories/3411ebc1-d5aa-464f-9615-0b527bc66719/commits/f47bbc106853afe3c1b07a81754bce5f4b8dbf62"
39      },
40      "lastMergeCommit": {
41        "commitId": "1e924c570d532d92c12e955b9f8948935f54944e",
42        "url": "https://dev.azure.com/fabrikam/_apis/git/repositories/3411ebc1-d5aa-464f-9615-0b527bc66719/commits/1e924c570d532d92c12e955b9f8948935f54944e"
43      },
44      "reviewers": [
45        {
46          "reviewerUrl": "https://dev.azure.com/fabrikam/_apis/git/repositories/3411ebc1-d5aa-464f-9615-0b527bc66719/pullRequests/20/reviewers/d6245f20-2af8-44f4-9451-8107cb2767db",
47          "vote": 10,
48          "id": "d6245f20-2af8-44f4-9451-8107cb2767db",
49          "displayName": "Normal Paulk",
50          "uniqueName": "fabrikamfiber16@hotmail.com",
51          "url": "https://dev.azure.com/fabrikam/_apis/Identities/d6245f20-2af8-44f4-9451-8107cb2767db",
52          "imageUrl": "https://dev.azure.com/fabrikam/_api/_common/identityImage?id=d6245f20-2af8-44f4-9451-8107cb2767db"
53        }
54      ],
55      "url": "https://dev.azure.com/fabrikam/_apis/git/repositories/3411ebc1-d5aa-464f-9615-0b527bc66719/pullRequests/20",
56      "supportsIterations": true,
57      "completionQueueTime": "2016-11-01T16:30:00.6380112Z"
58    }
59  ],
60  "count": 1
61 }

```

Листинг 2.2: Пример одговора на захтев за дохватање података о предлозима измена и рецензијама са сервиса Azure DevOps (пример је преузет из званичне документације сервиса [35]).

```

1 {
2   "value": [
3     {
4       "pullRequestThreadContext": {
5         "iterationContext": {
6           "firstComparingIteration": 1,
7           "secondComparingIteration": 2
8         },
9         "changeTrackingId": 1
10      },
11      "id": 148,
12      "publishedDate": "2016-11-01T16:30:50.083Z",
13      "lastUpdatedDate": "2016-11-01T16:30:52.48Z",
14      "comments": [
15        {
16          "id": 1,
17          "parentCommentId": 0,
18          "author": {
19            "id": "d6245f20-2af8-44f4-9451-8107cb2767db",
20            "displayName": "Normal Paulk",
21            "uniqueName": "fabrikamfiber16@hotmail.com",
22            "url": "https://dev.azure.com/fabrikam/_apis/Identities/d6245f20-2af8-44f4-9451-8107cb2767db",
23            "imageUrl": "https://dev.azure.com/fabrikam/_api/_common/identityImage?id=d6245f20-2af8-44f4-9451-8107cb2767db"
24          },
25          "content": "Should we add a comment about what this value means?",
26          "publishedDate": "2016-11-01T16:30:50.083Z",
27          "lastUpdatedDate": "2016-11-01T16:30:50.083Z",
28          "commentType": "text",
29          "usersLiked": []
30        }
31      ],
32      "status": "active",
33      "threadContext": {
34        "filePath": "/new_feature.cpp",
35        "rightFileStart": {
36          "line": 5,
37          "offset": 1
38        },
39        "rightFileEnd": {
40          "line": 5,
41          "offset": 13
42        }
43      },
44      "properties": {
45        "Microsoft.TeamFoundation.Discussion.SupportsMarkdown": {
46          "$type": "System.Int32",
47          "$value": 1
48        }
49      },
50      "isDeleted": false
51    }
52  ],
53  "count": 1
54 }

```

Листинг 2.3: Пример одговора на захтев за дохватање података о коментарима на рецензији сасервиса Azure DevOps (пример представља редуковану верзију из званичне документације сервиса [36]).

Одговор на захтев садржи већину релевантних података, али је додатно потребно за сваки предлог измена послати захтев који ће дохватити све коментаре који су остављени на предлогу измена (Листинг 2.4). Пример одговора који се добија слањем овог захтева дат је у Листингу 2.3. У овом примеру се може видети да је остављен само један коментар („*Should we add a comment about what this value means?*”) 01.11.2016. у 16:30. Коментар је оставио *Normal*

Paulk (fabrikamfiber16@hotmail.com) и коментар се односи на фајл *new_feature.cpp* који се налази у корену репозиторијума, конкретно на садржај његове пете линије од првог до тринаестог карактера, што се може видети на основу структура *rightFileStart* и *rightFileEnd*.

```
1 GET https://dev.azure.com/{organization}/{project}/_apis/git/repositories/{
    repositoryId}/pullRequests/{pullRequestId}/threads?api-version=7.1
```

Листинг 2.4: HTTP GET захтев за дохватање података о коментарима на појединачној рецензији са сервиса Azure DevOps

2.3.2 GitHub

GitHub (GH) је још један сервис за колаборативни развој софтвера [37]. Као и код других сервиса, осим система за верзионисање кода (*git* [28]), GitHub нуди и сервисе за организацију и праћење послова (*issues* [38] и *projects* [39]), као и системе за аутоматско компајлирање, тестирање и објављивање софтвера (*actions* [40]). GitHub такође нуди могућност хостовања статичких веб страница на основу кода из репозиторијума (GitHub Pages [41]).

Пример предлога измена на сервису GitHub може се видети на Слици 2.8. Наслов предлога измена је „*Update README.md with some text*”, аутор измене је *nikolapejic*, и предлог измена је још увек активан (*Open*). Опис измене дат је испод и гласи „*Added some more text to the README.md for the example PR.*”. Тренутно стање рецензије укључује само један коментар, и то коментар аутора измене, који гласи „*Looks good to me!*”. У следећем табу *Commits* може се видети списак измена које чине овај предлог измена (Слика 2.9). У овом случају направљена је само једна измена („*Update README.md*”). У последњем табу *Files changed* могу се видети саме измене (Слика 2.10), које у овом примеру представљају додавање једне празне линије и једне линије са текстом „*Adding some more info for the example PR.*”, и модификовање прве линије тако да садржи специјалне карактере за прелазак у нови ред уместо непринтабилног карактер за крај фајла, што је разлог зашто је пријављено да је прва линија модификована иако је наизглед остала иста.

```
1 GET https://api.github.com/repos/OWNER/REPO/pulls
```

Листинг 2.5: HTTP GET захтев за дохватање података о предлозима измена са сервиса GitHub

```

1 [
2 {
3   "url": "https://api.github.com/repos/octocat/Hello-World/pulls/1347",
4   "id": 1,
5   ...
6   "number": 1347,
7   "state": "open",
8   "locked": true,
9   "title": "Amazing new feature",
10  "user": {
11    "login": "octocat",
12    "id": 1,
13    ...
14    "type": "User",
15    "site_admin": false
16  },
17  "body": "Please pull these awesome changes in!",
18  "labels": [
19    {
20      "id": 208045946,
21      "node_id": "MDU6TGFiZWwzMDgwNDU5NDY=",
22      "url": "https://api.github.com/repos/octocat/Hello-World/labels/bug",
23      "name": "bug",
24      "description": "Something isn't working",
25      "color": "#f29513",
26      "default": true
27    }
28  ],
29  ...
30  "active_lock_reason": "too heated",
31  "created_at": "2011-01-26T19:01:12Z",
32  "updated_at": "2011-01-26T19:01:12Z",
33  "closed_at": "2011-01-26T19:01:12Z",
34  "merged_at": "2011-01-26T19:01:12Z",
35  "merge_commit_sha": "e5bd3914e2e596debea16f433f57875b5b90bcd6",
36  "assignee": {
37    "login": "octocat",
38    ...
39    "type": "User",
40    "site_admin": false
41  },
42  "assignees": [
43    {
44      "login": "octocat",
45      ...
46      "type": "User",
47      "site_admin": false
48    },
49    {
50      "login": "hubot",
51      ...
52      "type": "User",
53      "site_admin": true
54    }
55  ],
56  "requested_reviewers": [
57    {
58      "login": "other_user",
59      ...
60      "type": "User",
61      "site_admin": false
62    }
63  ],
64  "requested_teams": [
65    {
66      "id": 1,
67      "node_id": "MDQ6VG9hbTE=",
68      "url": "https://api.github.com/teams/1",
69      "html_url": "https://github.com/orgs/github/teams/justice-league",
70      "name": "Justice League",
71      "slug": "justice-league",
72      "description": "A great team.",
73      ...
74    }
75  ],
76  ...
77 }
78 ]

```

Листинг 2.6: Пример одговора на захтев за дохватање података о предлозима измена сервиса GitHub (пример представља поједностављену верзију из званичне документације [42]).

Зарад лакшег приступа подацима о предлозима измена и рецензија, GitHub нуди и REST [33] интерфејс [43]. Слањем захтева из Листинга 2.5 могу се излистати сви предлози измена датог репозиторијума. Пример одговора претходног захтева дат је у Листингу 2.6 (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [42]). Али, да би се дохватили детаљи предлога измена, као што су фајлови (Листинг 2.7) и коментари рецензената (Листинг 2.8), потребно је послати додатне захтеве.

```
1 GET https://api.github.com/repos/OWNER/REPO/pulls/PULL_NUMBER/files
```

Листинг 2.7: Пример захтева за дохватање измењених фајлова на сервису GitHub.

```
1 GET https://api.github.com/repos/OWNER/REPO/pulls/comments
```

Листинг 2.8: Пример захтева за дохватање коментара рецензената на сервису GitHub.

2.4 Секундарни извори телеметрије

Осим самих сервиса за колаборативни развој софтвера, који представљају оригиналне изворе података, током времена услед повећане потребе за приступом различитој телеметрији појавили су се секундарни извори телеметрије. Наиме, уместо слања великог броја захтева самом сервису за колаборативни развој од стране више корисника (у којем случају треба водити рачуна да се не прекорачи максималан број захтева у јединици времена који сервис намеће по кориснику, иначе долази до привремене блокаде корисника), секундарни сервиси једном дохвате све релевантне податке, трансформишу их у облик погодан за брзу претрагу, и затим их нуде свим заинтересованим корисницима. У овој секцији биће обрађени примери два секундарна сервиса за телеметрију са система GitHub, и то GH Archive и GH Torrent.

2.4.1 GH Archive

GH Archive је пројекат који има за циљ да забележи сву јавну телеметрију сервиса GitHub, и да је учини доступном на једноставан начин за даљу анализу [44]. Овај пројекат сакупља оригиналне догађаје (*event*), врши њихово агрегирање у JSON формату на нивоу сваког сата, ZIP-ује сакупљене догађаје и чини их јавно доступним на одговарајућим URL адресама (Листинг 2.9).

```
1 https://data.gharchive.org/2015-01-01-15.json.gz
```

Листинг 2.9: Пример URL адресе за дохватање GitHub телеметрије са GH Archive сервиса за период од 15 до 16 часова 01.01.2015. године.

Осим опције дохватања ZIP-ова са подацима, GH Archive нуди могућност коришћења сервиса BigQuery компаније Google [45] за анализу података. Сви подаци које GH Archive сакупи доступни су као јавни скуп података на BigQuery сервису, и веома једноставно им се приступа. На пример, да би се избројало колико догађаја различитог типа се десило 01. 01. 2024., довољно је да се у BigQuery веб едитор ископира Листинг 2.10 и покрене извршавање. BigQuery ће сам алоцирати довољан број машина за обраду овог упита, дохватити податке, филтрирати их и агрегирати, и на крају представити резултате у табеларном формату. Иако је овај начин приступа и обраде података јако брз и једноставан, и иако BigQuery нуди 1 ТВ бесплатне обраде података сваког месеца [46], то је за веће анализе (које укључују агрегирање телеметрије из више година) недовољно, па је потребно купити додатне кредите.

```
1 SELECT COUNT(*) AS cnt, type
2 FROM 'githubarchive.day.20240101'
3 GROUP BY type
```

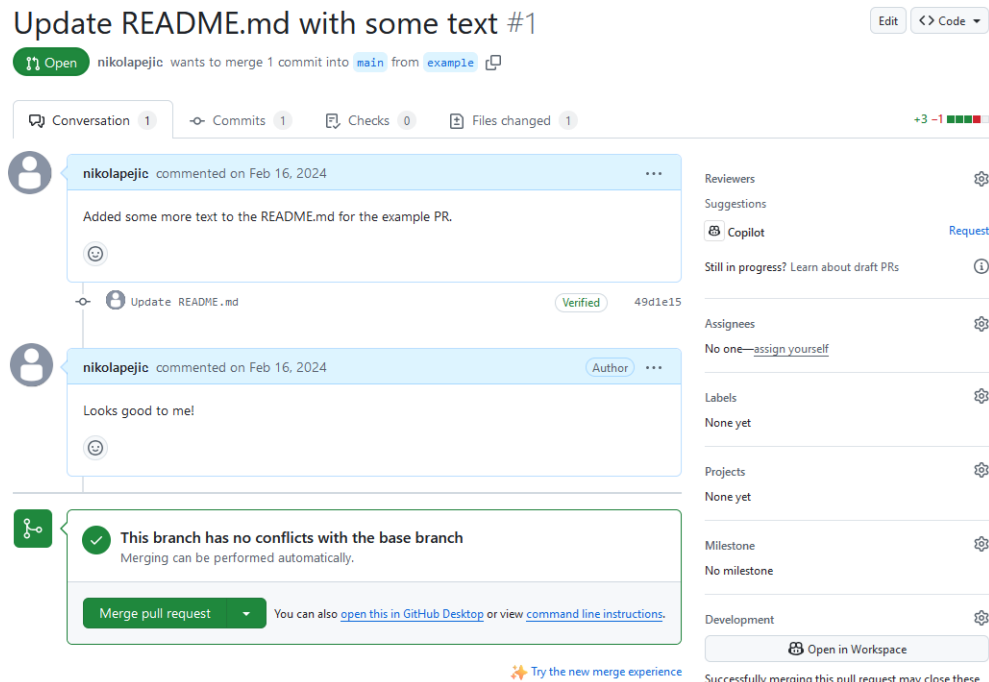
Листинг 2.10: Пример BigQuery упита за обраду GH Archive података за 01.01.2024. године.

2.4.2 GH Torrennt

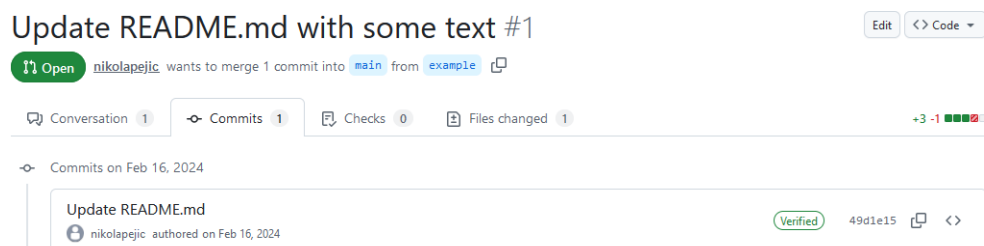
GH Torrennt [47] је још један пројекат који пружа индиректан приступ GitHub телеметрији. Користи се више налога како би се паралелизовао приступ GitHub REST API-ју и заобишло ограничење броја захтева по кориснику. Као део овог пројекта, креирана је MongoDB база података [48] која би требало сваког месеца да се освежи новим подацима.

Дељење података врши се преко BitTorrent протокола. Више радова је користило GH Torrennt као извор података [49, 50], али делује да је престало одржавање овог пројекта. Наиме, домен главне веб странице¹ је истекао још почетком 2023. године [51] и изгледа да није било било каквих активности од тада (макар у периоду рада на овом истраживању).

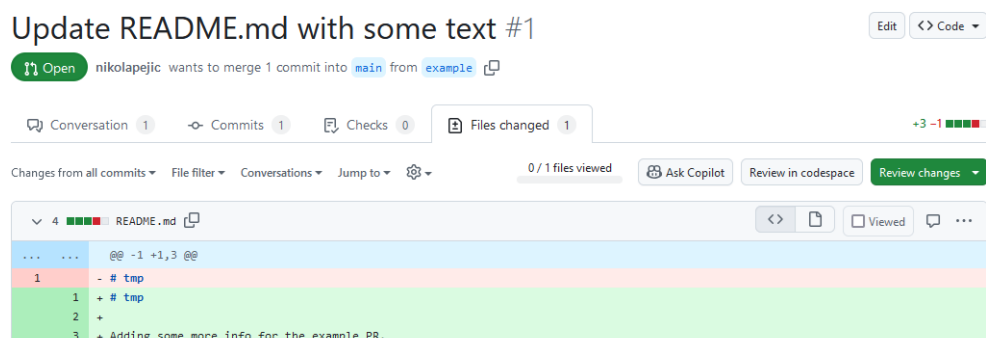
¹<https://ghtorrent.org>



Слика 2.8: Пример предлога измена на сервису GitHub - преглед.



Слика 2.9: Пример предлога измена на сервису GitHub - измене.



Слика 2.10: Пример предлога измена на сервису GitHub - фајлови.

3 Утицај COVID-19 пандемије на GitHub

Засновано на адаптацији текста публикације [52]:

N. Pejić, Z. Radivojević and M. Cvetanović, "Analyzing the Impact of COVID-19 on GitHub Event Trends. Sustainability", Sustainability 15, св. 19, 2023, doi: <https://doi.org/10.3390/su151914622>

Са почетком COVID-19 пандемије дошло је до великих промена у читавом свету. Услед рестрикција у кретању, велики број неесенцијалних послова морао је да се обавља или у потпуности на даљину („од куће”), или у хибридном моду са минималним присуством уживо. То је био случај и са open source развојем софтвера на платформи GitHub. Поставља се питање како су пандемија и промена у начину рада утицали на продуктивност open-source заједнице GitHub-а.

Како би се боље разумео временски оквир напредовања COVID-19 вируса и различитих мера које су предузете како би се спречило његово ширење, у наставку је дат преглед најбитнијих датума пандемије:

2019-12-31	СЗО јавља о случајевима „виралне пнеумоније” у Народној Републици Кини [53];
2020-01-09	вирус је идентификован као нови сој коронавируса [53];
2020-01-11	први пријављени смртни случај новог коронавируса [53];
2020-01-13	први случај коронавируса детектован ван Кине [53];

2020-01-23	Народна Република Кина уводи делимичне изолације/рестрикције кретања [54];
2020-01-30	уводи се ванредно стање јавног здравља од међународног значаја услед ширења новог коронавируса [53];
2020-02-11	нови коронавирус је означен као COVID-19 [53];
2020-02-25	неке области у Италији уводе потпуне рестрикције кретања [54];
2020-03-07	сто хиљада потврђених случајева COVID-19 на глобалном нивоу [53];
2020-03-09	цела Италија уводи потпуне рестрикције кретања [54];
2020-03-11	COVID-19 се категоризује као пандемија [53];
2020-03-16	креће прва фаза клиничког тестирања COVID-19 вакцина [55];
2020-04-04	преко милион случајева COVID-19 потврђено широм света [53];
2020-04-16	СЗО даје препоруке за рестрикције кретања [53];
2023-05-05	СЗО објављује крај COVID-19 пандемије [56];

У овој секцији прво су описана постојећа истраживања која су се бавила испитивањем утицаја пандемије на open source развој на платформи GitHub (3.1), а затим је описана потреба за додатним истраживањем на које се ово поглавље фокусирало, као и истраживачко питање на које се тражи одговор (3.2). Затим је описано само истраживање које је спроведено зарад испитивања дугорочног утицаја пандемије на развој софтвера на сервису GitHub (3.3), уз детаље имплементације (3.4), након чега су представљени резултати евалуације (3.5). На крају је дат преглед постигнутих резултата, уз дискусију о претњама њиховој валидности и коментарима о даљим правцима истраживања (3.6).

3.1 Постојећа истраживања

Од почетка пандемије било је неколико радова који су проучавали промене у трендовима телеметрије и активности на сервису GitHub, и њихов преглед је дат у Табели 3.2. Као што се може видети, истраживања су се фокусирала углавном на период током почетка пандемије (првих неколико месеци 2020. године), или закључно са крајем 2020. године. Током претраге литературе за истраживање у овом поглављу није пронађен ни један рад који се бавио дугорочнијим утицајем пандемије на open source развој на платформи GitHub.

Табела 3.2: Преглед претходних истраживања о утицају COVID-19 пандемије на open source развој на сервису GitHub.

	Публикација	Година пуб.	Период догађаја
Истраживање 1	Octoverse spotlight: An analysis of developer productivity, work cadence, and collaboration in the early days of COVID-19 [57]	2020	2019–2020
Истраживање 2	Octoverse—Finding balance between work and play [58]	2021	2018–2020
Истраживање 3	Public software development activity during the pandemic [59]	2021	2017–2020

3.1.1 Истраживање 1

GitHub је анализирао своју телеметрију током раних дана COVID-19 пандемије [57]. Закључили су да је активност развоја софтверског кода (број *push*-ованих измена, број предлога измена, број рецензија) остала непромењена или је чак и порасла у односу на претходну годину. Постојала је већа разлика у броју *issue*-а око периода када је пандемија кренула (март 2020. године), и како се активност око *issue*-а везује за планирање и организацију посла, претпоставка је да је било мање састанака на којима је планиран развој, па су коришћени *issue*-и као замена за састанке. Осим тога, радни дан инжењера постао је дужи за до једног сата, што може указати на преоптерећење инжењера који су покушавали да одрже темпо у промењеним условима.

Метрике које су анализирале:

- географска дистрибуција корисника
- број отворених предлога измена по активном кориснику по дану/радном дану
- број послатих измена (*push*) по активном кориснику по дану/радном дану
- број креираних *issue*-а по активном кориснику по дану/радном дану
- временска разлика између првих и последњих послатих измена (*push*) у току дана по дану

3.1.2 Истраживање 2

GitHub је наставио да анализира своју телеметрију и након завршетка 2020. године [58], и дошли су до следећих запажања:

1. Смањивањем величине предлога измена, оне постају једноставније за рецензију. Тиме се укупно време почев од креирања предлога измена, кроз рецензију, и коначно прихватање измена може смањити за до 7.5h.
2. Аутоматизацијом процеса коришћењем GitHub акција (*Actions*) постигнуто је смањење времена од креирања до прихватања измена и повећан укупни број предлога измена.
3. Анализа је показала да, иако инжењери праве паузу од свог посла током празника и одмора, развој у open source заједници тада показује скокове, што значи да се на овај вид рада гледа другачије него на стандардни посао.
4. Повећана је активност инжењера, не само у погледу времена које проводе у развоју кода, већ и у количини посла који се уради. То може показивати да инжењерима годи флексибилност у организацији свог радног дана која је настала услед рада на даљину, али може представљати проблем уколико је повећана продуктивност дошла на уштрб слободног времена и здравих пауза у раду.

У датом истраживању коришћене су следеће метрике:

- географска дистрибуција корисника
- број послатих измена (*push*) по активном кориснику по дану/радном дану
- број креираних *issue*-а по активном кориснику по дану/радном дану
- временска разлика између првих и последњих послатих измена (*push*) у току дана по дану
- број активних корисника по дану
- време од креирања до прихватања предлога измена
- време од креирања предлога измена до прве активности рецензената
- време од последње активности рецензената до прихватања (примене) измена

3.1.3 Истраживање 3

Такође у 2021. години, извршено је истраживање [59] које се фокусирао на:

1. поређење почетка пандемије (март-април) у односу на посматрани период (2017-2020) - показало се да је брзина промене броја догађаја током почетка пандемије 3 пута већа у односу на стандардну девијацију брзина промене броја догађаја током посматраног периода.
2. поређење 2020. године у односу на претходне године (2017-2019) - утврђено је да иако постоје разлике у средњим вредностима *PushEvent* догађаја пре и током 2020. године, 2020. година се није статистички значајно разликовала у односу на претходне године.

Ово истраживање користило је следеће метрике:

- Дневни/месечни број различитих GitHub догађаја (*PushEvent*, *CreateEvent*, *PullRequestEvent*, *ReleaseEvent*, *ForkEvent*, *IssueCommentEvent*, *DeleteEvent*, *PullRequestReviewCommentEvent*, *WatchEvent*, *PublicEvent*, *MemberEvent*, *GollumEvent*, *IssueEvent*, *CommitCommentEvent*).

3.2 Поставка проблема

Претходна истраживања вршила су анализу утицаја пандемије на GitHub само током 2020. године, иако је пандемија трајала до средине 2023. године [53]. Како није било радова који су испитивали дугорочнији утицај пандемије на open source заједнице, сматра се да управо ту постоји простор за научни допринос. Зато се поставља следеће истраживачко питање:

ИП3.1. Да ли постоје дугорочне промене у трендовима догађаја open source заједнице?

3.3 Предлог истраживања

Како би се испитао дугорочни утицај пандемије на GitHub, одлучено је да се посматра период од почетка 2020. до краја 2022. године (као период током трајања пандемије) у односу на период од почетка 2017. до краја 2019. године (као период пре пандемије). Одабрано је да се анализирају исти подаци као у Истраживању 3 (3.1.3), односно апсолутни бројеви

различитих GitHub догађаја, али током продуженог временског периода. За анализу промена трендова предлажу се следеће методе:

1. **Коефицијент праве линеарне регресије** - како делује да трендови промене броја догађаја кроз време прате линеарну зависност (Слика 3.1), над телеметријом се може извршити линеарна регресија, чиме се добијају праве које представљају линеарну апроксимацију трендова пре и током пандемије. Поређењем коефицијената ових двеју права може се добити релативна промена трендова догађаја.
2. **ADF тест** - слично као у Истраживању 3 (3.1.3), анализа стационарности може се урадити коришћењем ADF (*Augmented Dickey-Fuller*) теста.

3.4 Детаљи имплементације

За прикупљање телеметрије коришћен је BigQuery и GH Archive скуп података. Подаци за сваку годину прикупљени су и агрегирани на дневном нивоу коришћењем одговарајућег упита, а пример за 2022. годину дат је у Листингу 3.1, где *suffix* садржи месец и дан агрегираних података.

```

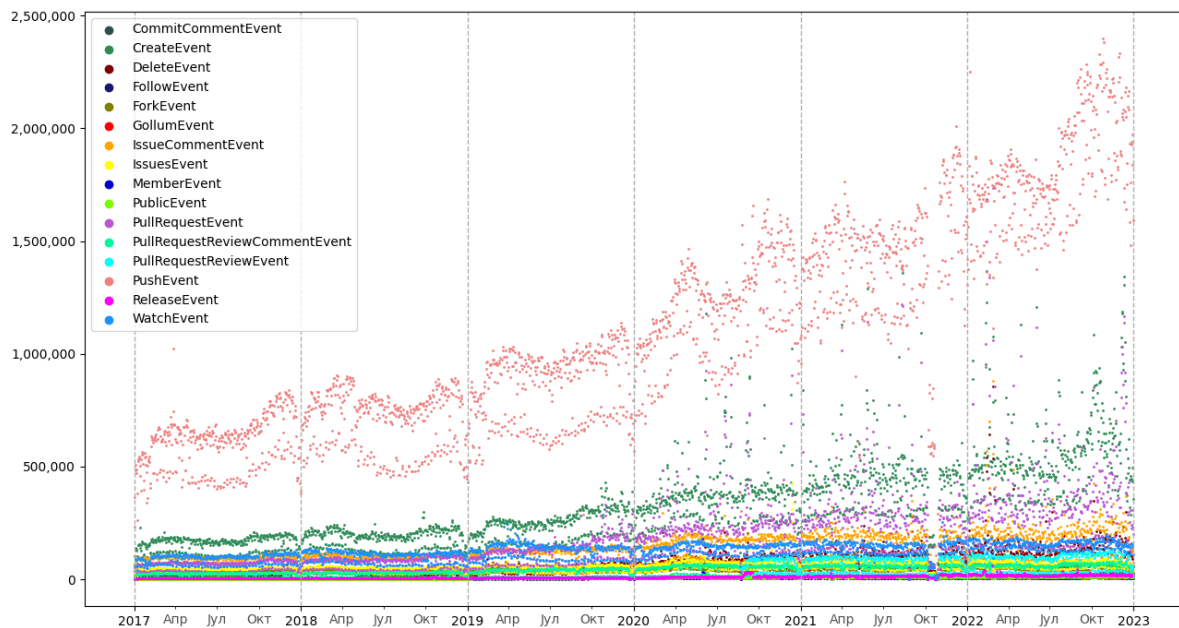
1 SELECT COUNT(*) AS cnt, type, 2022 AS year, _TABLE_SUFFIX AS suffix
2 FROM `githubarchive.day.2022*`
3 GROUP BY type, _TABLE_SUFFIX

```

Листинг 3.1: Пример упита који је коришћен за дохватање једне године (2022) скупа података помоћу BigQuery система.

Прикупљена је телеметрија о 16 различитих типова догађаја (Слика 3.1), али како неки типови (**FollowEvent** и **PullRequestReviewEvent**) нису емитовани током целог посматраног периода, они нису разматрани у даљој анализи. Преосталих 14 догађаја су:

- **CommitCommentEvent** - емитује се када постоји активност везана за рецензију појединачне измене (нпр. када се остави коментар везан за појединачну верзију измене).
- **CreateEvent** - емитује се када се креирају *git* грана или таг.
- **DeleteEvent** - емитује се када се бришу *git* грана или таг.
- **ForkEvent** - емитује се када корисник направи копију (*fork*) репозиторијума.



Слика 3.1: Преглед броја GitHub догађаја различитог типа од почетка 2017. до краја 2022. године.

- **GollumEvent** - емитије се када се креирају или модификују *wiki* странице.
- **IssueCommentEvent** - емитије се када дође до активности везане за коментаре на проблемима (нпр. када се остави или измени коментар).
- **IssueEvent** - емитије се када постоји активност везана за пријаве проблема (нпр. када се креира или затвори проблем, односно *issue*).
- **MemberEvent** - емитије се када се деси нека активност која је везана за кориснике који одржавају репозиторијум (нпр. када корисник прихвати да буде део групе која одржава репозиторијум).
- **PublicEvent** - емитије се када се репозиторијум јавно објави.
- **PullRequestEvent** - емитије се када постоји активност везана за предлоге измена (нпр. када се предлог измена креира или затвори).
- **PullRequestReviewCommentEvent** - емитије се када постоји активност везана за рецензију предлога измена (нпр. када рецензент остави коментар).
- **PushEvent** - емитије се када се локалне измене пошаљу са рачунара корисника на GitHub сервис.

Табела 3.3: Трендови раста броја догађаја кроз време за два посматрана периода (2017-2019 и 2020-2022), уз релативну промену и ознаку о смеру промене ($\nearrow$ у случају раста и $\searrow$ у случају пада тренда).

Тип догађаја	2017-2019	2020-2022	промена
CommitCommentEvent	0.56	18.16	32.18x $\nearrow$
CreateEvent	107.25	198.24	1.85x $\nearrow$
DeleteEvent	26.83	49.21	1.83x $\nearrow$
ForkEvent	15.42	0.43	0.03x $\searrow$
GollumEvent	-0.54	-1.57	2.89x $\searrow$
IssueCommentEvent	28.56	29.12	1.02x $\nearrow$
IssuesEvent	11.00	2.94	0.27x $\searrow$
MemberEvent	1.70	2.47	1.45x $\nearrow$
PublicEvent	4.24	4.42	1.04x $\nearrow$
PullRequestEvent	84.03	107.94	1.28x $\nearrow$
PullRequestReviewCommentEvent	14.39	1.52	0.11x $\searrow$
PushEvent	322.37	786.43	2.44x $\nearrow$
ReleaseEvent	2.93	10.25	3.50x $\nearrow$
WatchEvent	40.00	20.89	0.52x $\searrow$

- **ReleaseEvent** - емитије се када се објави ново издање софтвера.
- **WatchEvent** - емитије се када корисник крене да прати репозиторијум.

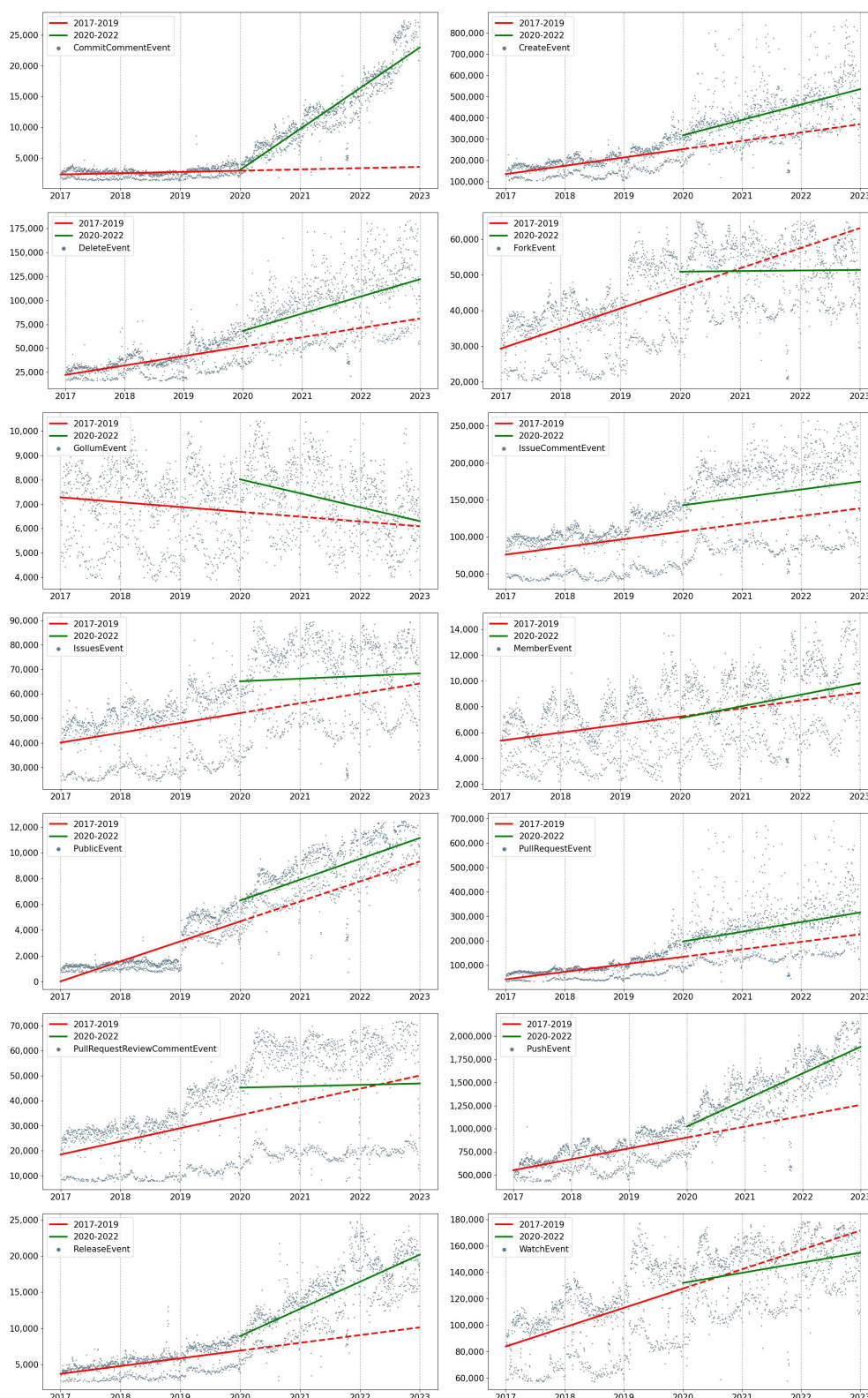
Како би се елиминисао утицај екстремних вредности, одлучено је да се пре било какве анализе филтрира по 2% најекстремнијих вредности сваког типа догађаја. За анализу стационарности одабрано је да се користи *statsmodels.tsa.stattools.adfuller* имплементација ADF теста из *statsmodel* Python 3 пакета [60].

3.5 Евалуација

ИПЗ.1. Да ли постоје дугорочне промене у трендовима догађаја open source заједнице?

Трендови линеарне регресије

Након прикупљања података и након филтрирања по 2% најекстремнијих вредности за сваки тип догађаја, за сваки догађај су посматрана 2 периода: 2017-2019 (период пре почетка пандемије), и 2020-2022 (период од почетка пандемије [53], до последње навршене године



Слика 3.2: Преглед броја GitHub догађаја сваког типа од почетка 2017. до краја 2022. године. Црвене и зелене линије представљају трендове промене броја догађаја кроз време, и добијене су линеарном регресијом над догађајима из периода 2017-2019 и 2020-2022, респективно. Испрекидана црвена линија представља тренд из 2017-2019 настављен током периода 2020-2022.

у тренутку рађења ове анализе). Коришћењем линеарне регресије одређени су трендови броја догађаја током ових периода, и резултати поређења трендова дати су у Табели 3.3, док су визуелизације дате на Слици 3.2. Највеће разлике у трендовима могу се видети за **CommitCommentEvent** догађај, чији се тренд (коефицијент праве) повећао ≈ 32 пута, и **ForkEvent** и **PullRequestReviewCommentEvent**, чији су трендови опали 0.03 и 0.11 пута, респективно.

Пораст тренда **CommitCommentEvent** могао би бити директан утицај пандемије - како комуникација уживо углавном више није била могућа, дискусије о коду су природно премештене у асинхрони вид комуникације остављањем коментара на самом коду. Помало је неочекиван пад **PullRequestReviewCommentEvent**, јер се везује за рецензију предлога измена, али могуће је да је велики пораст коментара на појединачним верзијама измена (**CommitCommentEvent**) одвукао поједине типове комуникација од других догађаја. Такође, може се приметити пад тренда **GollumEvent**, који се емитује при уређивању документације (*wiki* страница), па би се могло закључити да овај вид асинхроне комуникације није био толико важан.

Са друге стране, пад **ForkEvent** догађаја може се протумачити слабљењем интересовања open source заједнице за прикључивање пројектима, јер је копирање (*fork*-овање) репозиторијума стандардни начин укључивања у колаборативни развој неког репозиторијума - како су за директан рад на репозиторијуму потребна додатна права приступа, обично сваки корисник направи копију репозиторијума коју може слободно да мења, уз опцију да касније предложи да се имплементирани измене примене и над оригиналним репозиторијумом. Ово донекле има смисла јер је током пандемије било важнијих дешавања, па је започињање колаборације на новом пројекту вероватно имало мањи приоритет него раније. Слично би се могло рећи и за благи пад **WatchEvent** догађаја, који се везује за укључивање нотификација о дешавањима на датом репозиторијуму.

Догађаји који су везани за продуктивност, као што су **PushEvent** и **ReleaseEvent**, показали су нешто повећан тренд у односу на претходни период (≈ 2.4 и ≈ 3.5 пута, респективно), што би могло да се протумачи као наставак развоја open source софтвера сличним или нешто бржим темпом него у односу на претходни период. Остали догађаји нису имали веће промене трендова у два посматрана периода.

Табела 3.4: Анализа стационарности броја догађаја кроз време коришћењем ADF теста. У табели је приказана тест статистика и резултат ADF теста за сваки тип догађаја.

Тип догађаја	ADF тест	
	тест статистика	резултат
CommitCommentEvent	−2.75	није стационаран*
CreateEvent	−7.34	стационаран
DeleteEvent	−6.61	стационаран
ForkEvent	−7.47	стационаран
GollumEvent	−10.81	стационаран
IssueCommentEvent	−6.20	стационаран
IssuesEvent	−7.30	стационаран
MemberEvent	−9.22	стационаран
PublicEvent	−3.72	није стационаран**
PullRequestEvent	−6.99	стационаран
PullRequestReviewCommentEvent	−9.14	стационаран
PushEvent	−3.70	није стационаран**
ReleaseEvent	−4.41	стационаран
WatchEvent	−6.48	стационаран

* p вредност > 0.1

** p вредност > 0.01

Анализа стационарности

Како би се утврдило да ли је било статистички значајних промена у трендовима догађаја, извршена је анализа стационарности коришћењем ADF (*Augmented Dickey–Fuller*) теста. Након филтрирања по 2% екстремних вредности из целокупног скупа података (2017–2022) за сваки догађај, вредности су хронолошки сортиране и над њима је примењена *statsmodels.tsa.stattools.adfuller* имплементација ADF теста из *statsmodel* Python 3 пакета [60]. Како је на Слици 3.2 видљив тренд линеарног раста/пада броја догађаја кроз време, коришћена је опција за **ct** регресију (*constant and trend*). Нулта хипотеза ADF теста је да низ има јединични корен, док је алтернативна хипотеза да је низ стационаран. Тест статистика се пореди са различитим критичним вредностима (у овом раду 1% и 10%). Уколико је вредност тест статистике већа од одговарајуће критичне вредности, нулта хипотеза се одбацује, и низ се сматра стационарним. Уколико је вредност тест статистике мања од одговарајуће критичне вредности, онда се низ сматра нестационарним са p вредношћу изнад одговарајућег нивоа значајности.

Резултати ADF теста дати су у Табели 3.4, одакле се може видети да су једино **CommitCommentEvent**, **PublicEvent**, и **PushEvent** показали извешан ниво

нестационарности. **CommitCommentEvent** има највећи ниво нестационарности, што је уочљиво и на Слици 3.1. Види се да је до промене дошло са почетком 2020. године, и то је уједно и граница између посматраних периода, па би се могло закључити да је пандемија утицала на промену. **PublicEvent** има нешто мањи ниво нестационарности, и са Слике 3.1 се може видети да је до промене дошло са почетком 2019 године, што је годину дана пре почетка пандемије, тако да она није утицала на нестационарност овог догађаја. Са Слике 3.1 се не види јасан прекид у тренду **PushEvent**, као што је био случај са претходна два догађаја. Са друге стране, то је донекле и очекивано, јер нестационарност тренда овог догађаја није имала велики статистички значај.

3.6 Дискусија

Ово истраживање представља наставак и допуну Истраживање 3 које је иницијално спроведено над подацима до краја 2020. године, док су сада обухваћене и 2021. и 2022. година током којих је трајала пандемија. Као што је показано, већина догађаја је имала благи пораст али није имала статистички значајне промене током пандемије. Једини догађај који је показао нестационарност са већим статистичким значајем је **PushEvent**, који је на основу претходних истраживања деловао стационарно. Даље, иако ADF тест није показао да је промена имала велики статистички значај, **CommitCommentEvent** је имао релативно велико повећање тренда од $\approx 34x$. Са друге стране, догађаји који су имали већи пад у тренду су: **GollumnEvent**, који је везан за рад на документацији, и **IssuesEvent** (чији тренд је има пад од ≈ 0.01), и **ForkEvent** (≈ 0.09), али су сви ови догађаји класификовани као стационарни.

Када се посматра семантика догађаја, делује да су догађаји који се везују за развој кода и комуникацију о коду (*commit* и *issue* коментари) одржали или увећали своје трендове током пандемије, што би се могло протумачити као одржавање нивоа продуктивности. Са друге стране, догађаји који су имали пад су они везани за активност open source заједнице (нпр. *fork*-овање репозиторијума, креирање и затварање *issue*-а), као и рад на документацији, али сви ти догађаји су се показали као стационарни на ADF тесту.

Претње валидности

Истраживање у овом поглављу је посматрало само укупан број догађаја сваког типа, и није вршено никакво филтрирање бот налога и других система аутоматизације, што би могло да

утиче на процену активности стварних корисника. Такође, није вршена анализа садржаја догађаја, тако да су рецимо предлози измена који мењају само пар коментара третирани исто као и предлози измена који додају нове функционалности.

Перспективе за будућа истраживања

Као што је поменуто, није вршена анализа садржаја догађаја. За дубљу анализу утицаја пандемије на продуктивност, пожељно је да се у обзир узму и метрике као што су број додатих, избрисаних или измењених линија, време од креирања до прихватања предлога измена, дужина коментара, итд.

Такође, ово истраживање посматрало је само догађаје на целој платформи GitHub као меру продуктивности и наставка open source развоја, без дубље анализе продуктивности на нивоу појединачних инжењера. Истраживање 1 и Истраживање 2 вршили су анализе одређених метрика као што су временске разлике од прве до последње активности инжењера у току дана, број креираних предлога измена/промењених линија кода по броју активних корисника, број активних корисника по дану, итд. Као што је учињено за Истраживање 3, вреди поновити и ова два истраживања тако да се анализирају метрике не само током 2020. године већ до краја пандемије, ради дубље анализе утицаја пандемије на open source развој на платформи GitHub.

4 Претраживање експерата и предлога измена

Када на једном пројекту (репозиторијуму) ради мали тим од неколико инжењера, тада се може очекивати да већина инжењера буде упозната са целокупним кодом и тренутним променама у репозиторијуму, и да свако може да врши рецензије било којих измена. Али, у случајевима када у истом репозиторијуму ради неколико стотина или хиљада инжењера, тада је неефикасно и практично немогуће да сви инжењери буду упознати са свим детаљима. Уместо тога, са порастом броја инжењера долази до све веће специјализације, па се свака група инжењера фокусира само на део кода и није упозната са детаљима остатка репозиторијума [4]. Зато може бити компликовано наћи одговарајућег експерта за насумично изабрану област. Исто се може рећи и у случају претраге предлога измена - уколико је њихов број довољно велики, и уколико не постоји могућност претраге по тексту, претрага се може свести на пролажење кроз све претходне измене, што може бити мукотрпан и временски захтеван посао.

Након статистичке анализе телеметрије из претходног поглавља, ова секција описује први систем за побољшање искуства софтверских инжењера који је представљен у овој дисертацији. У овој секцији прво су описана постојећа решења (4.1), а затим је дефинисан проблем који се решава као и истраживачко питање на које се тражи одговор (4.2). Након тога је описан једноставан систем који омогућава претрагу предлога измена на основу задатог текста и који генерише експертизу (кључне речи) сваког инжењера на основу описа његових предлога измена, и тиме омогућава једноставну претрагу релевантних експерата (4.3). Затим су представљени детаљи имплементације система (4.4), након чега је анализирана телеметрија о активности и коришћењу једног таквог система који је имплементиран и учињен доступним током периода од неколико месеци групи од око 300 инжењера у индустрији (4.5). На крају

је дат преглед постигнутих резултата, уз дискусију о претњама њиховој валидности и коментарима о даљим правцима истраживања (4.6).

4.1 Постојећа решења

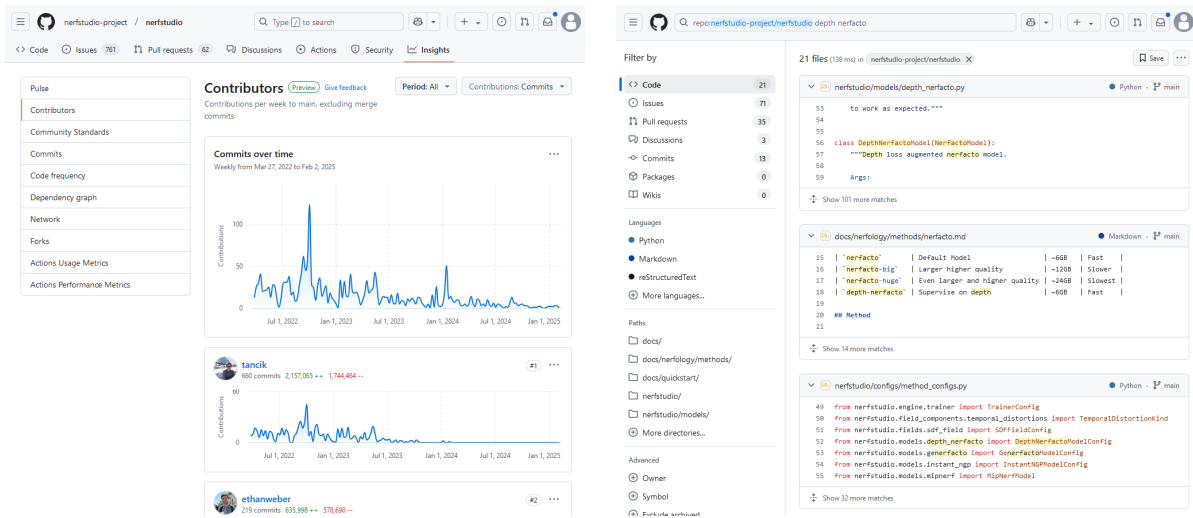
Претрагом литературе пронађено је више метода за препоруку рецензената који су користили TF-IDF за извлачење кључних речи из описа измена зарад одређивања експертизе инжењера [61, 62]. Зато је у овом поглављу дат кратак опис TF-IDF методе (4.1.1), док је детаљнији преглед детаља и различитих варијација у имплементацији дат у Поглављу 6. Осим тога, у наставку је дат и кратак преглед неколико начина за проналажење експерата у постојећим алатима за колаборативни развој софтвера (4.1.2).

4.1.1 TF-IDF

Један од начина на који неколико система за препоруку рецензената [61, 62] (Поглавље 6) одређује експертизу софтверских инжењера јесте коришћењем TF-IDF (6.1), односно *term frequency - inverse document frequency*, над речима из наслова и описа измена и ауторима измена као документима. На тај начин могу да се добију кључне речи са тежинским факторима који су већи што је неки аутор чешће користио дату кључну реч, а мањи уколико је већи број аутора користио исту кључну реч.

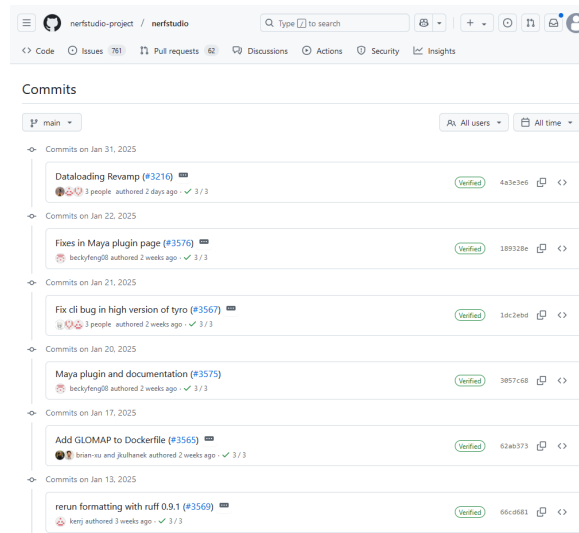
4.1.2 Постојећи алати у окружењима за колаборативни развој софтвера

Да би се нашао експерт за неку област или за неки део софтверског кода, тренутно не постоји превише алата који би могли бити од помоћи. На пример, сервис GitHub нуди могућност прегледа свих инжењера који су мењали код у датом репозиторијуму (Слика 4.1 (а)), али такав преглед не нуди никакве информације о њиховој експертизи нити претходним претходним изменама осим укупног броја измена и броја додатих/избрисаних линија кода. Уместо тога, често је прво потребно извршити неку претрагу кода или претходних измена на подстрингове за које знамо да се налазе у коду/коментарима/опису измена, како би се нашла почетна тачка потраге за експертима (Слика 4.1 (б)). Када се дође до фајла који је релевантан, обично се посматра његова историја како би се нашле измене које припадају области за коју се тражи експерт (Слика 4.1 (в)). Друга опција је да се контактирају инжењери који су најскорије (или најчешће) мењали дати фајл како би се проверило да



(а) преглед инжењера који су мењали код у репозиторијуму

(б) претрага репозиторијума на задате кључне речи



(в) преглед историје измена једног фајла

Слика 4.1: Различити видови претраге инжењера (експерата) на платформи GitHub. Пример показује стварно стање јавног репозиторијума (`nerfstudio-project/nerfstudio`).

ли су они експерти, и (у случају да нису) како би се добила смерница кога даље треба контактирати.

4.2 Поставка проблема

Иако GitHub има могућност претраге претходних предлога измена на задати текст, ова функционалност не постоји у свим сервисима за колаборативни развој кода (нпр. сервис Azure DevOps није поседовао ову функционалност у периоду вршења овог истраживања). Такође,

процес описан у 4.1.2 може бити релативно једноставан у случају да се ради о тиму који има десетак инжењера, али у већим тимовима са неколико стотина или хиљада инжењера, процес тражења експерта може бити релативно компликован поступак са више итерација. Иако је неки инжењер некада мењао дати фајл, не мора да значи да је он и даље експерт за дату област - можда је напустио тим, или је променио област, или можда никада и није био експерт да ту област већ је само мењао фајл као спорадичну измену у оквиру друге области. Са друге стране, TF-IDF као релативно једноставан начин екстрактовања експертизе инжењера из историја њихових измена није примењен у најпопуларнијим окружењима за колаборативан развој, па самим тим примена TF-IDF делује као одлична прилика за унапређење постојећих окружења.

Како би се испитала потреба за таквим решењем и његова адекватност, дефинишу се следеће истраживачко питање:

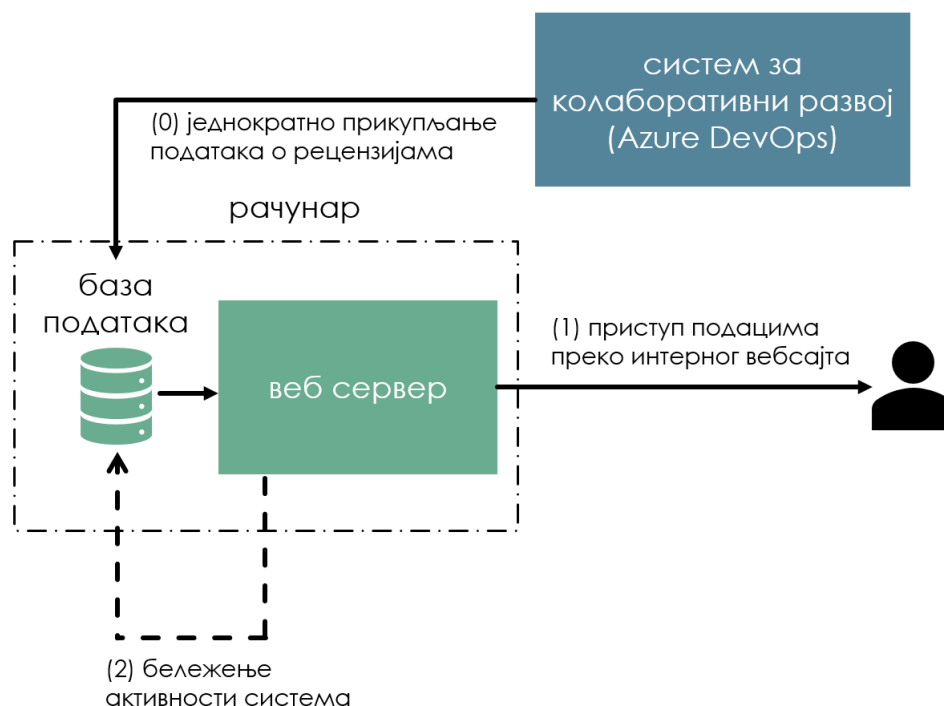
ИП4.1. Да ли постоји потреба за додатним системом за претрагу експерата и предлога измена у системима за колаборативни развој софтвера?

Како би се одговорило на ово истраживачко питање користиће се телеметрија прототипа система за претраживање експерата.

4.3 Предлог решења

Решење које је предложено у овом поглављу омогућава претрагу предлога измена по задатом тексту, као и инжењера по њиховој експертизи коришћењем кључних речи и њихових тежинских фактора који су генерисани применом TF-IDF над насловима и описима измена. Како би систем био доступан свим инжењерима без потребе за инсталацијом додатног софтвера, одлучено је да се направи веб апликација којој инжењери могу да приступе кроз било који веб претраживач (*web browser*). Како би се добила иницијална процена о искоришћењу система, одлучено је да се систем имплементира у две фазе:

Фаза 1. представља прототип решења - укључује веб апликацију кроз коју је могуће претраживати експерте. Подаци су једнократно дохваћени, обрађени и похрањени у базу података - не врши аутоматски (нити мануелно) освежавање података. На овај начин, могуће је тестирати идеју решења уз минимална улагања. Овакво решење се очекује да је задовољавајуће јер је претпоставка да се експертиза



Слика 4.2: Шематски приказ „PR Insights” веб сајта. Плави елементи су системи који су већ постојали (Azure DevOps). Зелени елементи су наменски направљени за потребе једноставне претраге експерата.

инжењера не мења превише брзо/често.

Фаза 2. представља целокупно решење - у односу на фазу један систем периодично врши аутоматско дохватање, обраду и ажурирање кључних речи и тежинских фактора у оквиру базе података, чиме се постиже освежавање података.

4.4 Детаљи имплементације

Како би се олакшала потрага за релевантним експертима и предлозима измена, осмишљено је решење чији је шематски приказ дат на Слици 4.2. Решење је дизајнирано да буде коришћено од стране групе инжењера из индустрије, и сви подаци о предлозима измена су доступни на сервису Azure DevOps. За потребе овог истраживања направљен је само прототип (**Фаза 1.**), а не потпуно функционални систем (**Фаза 2.**), па су подаци са Azure DevOps сервиса једнократно прикупљени (није се вршило касније освежавање података).

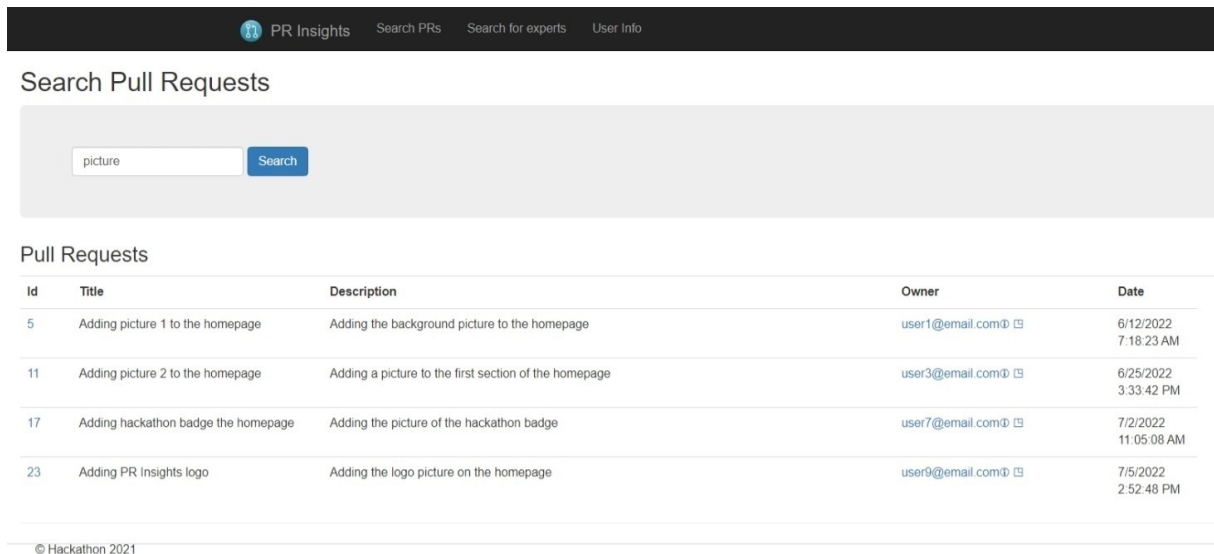
На једном рачунару подигнута је база података (SQL Server) у коју су похрањени подаци. Прво, сачувани су релевантни подаци о предлозима измена, а то су за потребе овог

истраживања били *ID*, *наслов*, *опис*, *датум креирања*, и *аутор*. Затим, како би се омогућило лако претраживање експерата, за сваког инжењера (аутора измена) извучене су кључне речи из наслова и описа његових предлога измена. Након једноставног филтрирања којим су избачене речи које не садрже информације (нпр. *the*, *a*, *an*, итд., и речи које се превише често употребљавају у развоју кода, као што су *test*, *bug*, *fix*, итд.), коришћењем TF-IDF [63] алгорита за сваку кључну реч дефинисан је тежински фактор који говори колико је она релевантна за ауторе који су је користили да опишу своје измене. Користећи TF-IDF, кључне речи које се јако често користе добијају мале тежинске факторе (јер већина аутора има експертизу за њих), док кључне речи које користе само поједини аутори имају већу тежину. Такође, што чешће аутор користи дату кључну реч она има већи тежински фактор. Свака комбинација аутора, кључне речи, и не-нула тежинског фактора је сачувана у бази, где се тежински фактори сматрају експертизом инжењера у датим областима.

Како би се олакшало претраживање тимова који имају експертизу у некој области, уведен је појам менаџерске експертизе. Након екстраковања кључних речи и тежинских фактора инжењера, дохваћена је хијерархија организације (*org chart*), а затим је за сваког менаџера посматрана експертиза његових подређених. Уколико довољан проценат подређених поседује одређену кључну реч, онда се она додељује менаџеру као менаџерска експертиза, и то са тежинским фактором који је једак просеку тежинских фактора дате кључне речи код његових подређених. Тиме се добија увид у области (односно кључне речи) за које је дати менаџер задужен.

Како би се омогућио једноставан интерфејс за приступ информацијама, одлучено је да се на рачунару поред базе података подигне и веб сервер са „PR Insights” веб сајтом, који је нудио следеће функционалности:

1. Прва функционалност сајта је релативно проста - како изабрани систем за колаборативни развој софтвера (Azure DevOps) не нуди опцију претраге наслова и описа измена по тексту, одлучено је да се та функционалност омогући корисницима „PR Insights” сајта. Претрагом на одређени текст прослеђује се упит бази којим се траже све измене које садрже задати текст било у наслову било у опису измена (*substring search*). Сви резултати претраге се врате кориснику у табеларном формату, као што је приказано на Слици 4.3. За пуне информације о предложеној измени, сваки *Id* представља линк до одговарајућег предлога измена (рецензије) у Azure DevOps систему. Такође, сваки аутор садржи три линка: први је *mailto* линк који омогућава



PR Insights Search PRs Search for experts User Info

Search Pull Requests

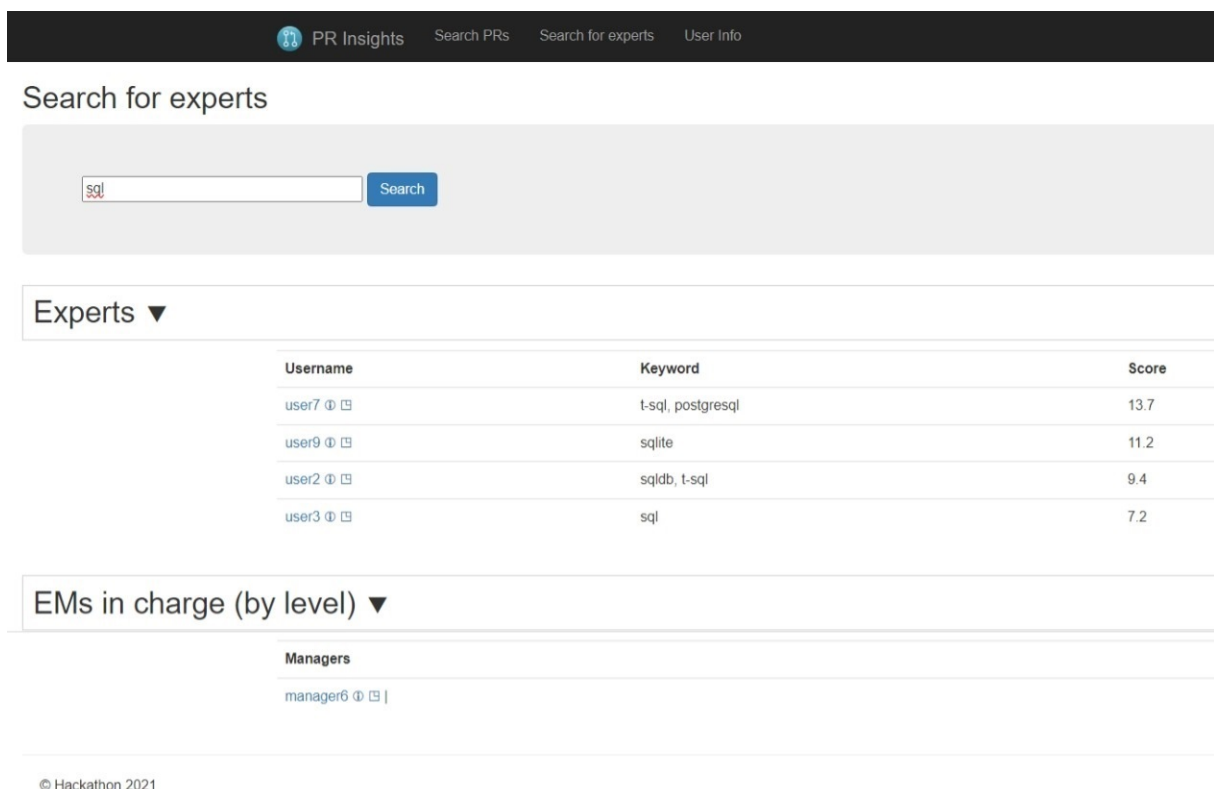
picture Search

Pull Requests

Id	Title	Description	Owner	Date
5	Adding picture 1 to the homepage	Adding the background picture to the homepage	user1@email.com	6/12/2022 7:18:23 AM
11	Adding picture 2 to the homepage	Adding a picture to the first section of the homepage	user3@email.com	6/25/2022 3:33:42 PM
17	Adding hackathon badge the homepage	Adding the picture of the hackathon badge	user7@email.com	7/2/2022 11:05:08 AM
23	Adding PR Insights logo	Adding the logo picture on the homepage	user9@email.com	7/5/2022 2:52:48 PM

© Hackathon 2021

Слика 4.3: Изглед „PR Insights” веб странице - претрага предлога измена. Приказани подаци представљају пример резултата претраге и генерисани су зарад демонстрирања начина функционисања сајта.



PR Insights Search PRs Search for experts User Info

Search for experts

sql Search

Experts ▼

Username	Keyword	Score
user7	t-sql, postgresql	13.7
user9	sqlite	11.2
user2	sqldb, t-sql	9.4
user3	sql	7.2

EMs in charge (by level) ▼

Managers
manager6

© Hackathon 2021

Слика 4.4: Изглед „PR Insights” веб странице - претрага експерта. Приказани подаци представљају пример резултата претраге и генерисани су зарад демонстрирања начина функционисања сајта.

 PR Insights Search PRs Search for experts User Info

User Info

Search

Name

Mail user38@email.com

Top expertise backend databases php

Top EM expertise backend

Top Reviewers ▼

Reviewer	Reviewer Activity Score
user15@email.com 	7
user17@email.com 	5
user11@email.com 	2

Expertise ▼

Keyword	Relevancy
backend	4.7
databases	3.6
php	2.1

EM Expertise ▼

Keyword	Relevancy
backend	2.5

© Hackathon 2021

Слика 4.5: Изглед „PR Insights” веб странице - профил корисника. Приказани подаци представљају пример резултата претраге и генерисани су зарад демонстрирања начина функционисања сајта.

писање емајла аутору; други је линк који води до постојећег интерног сајта за преглед профила запослених; трећи је линк до профила инжењера на „PR Insights” сајту (трећа функционалност).

- Друга функционалност је претрага експерата на основу кључних речи из наслова и описа њихових предлога измена (Слика 4.4). Као што је претходно описано, кључне речи су екстраховане и похрањене у базу података. Када се почне са куцањем кључних речи, омогућена је *autocomplete* функционалност на основу података похрањених у бази. Претрагом на задате кључне речи биће приказани сви инжењери који су их претходно користили у опису својих измена, и поред алиаса сваког инжењера биће

приказане и све његове пронађене кључне речи и њихов сумирани тежински фактор. Такође ће бити излистани и сви менаџери који садрже менаџерске кључне речи, и то уређени по хијерархији рапортирања. Сваки листинг инжењера и менаџера такође садржи сва три претходно описана линка.

3. На крају ту је директна претрага корисника преко њихових алијаса (односно имејл адреса) и приказивање корисничког профила (Слика 4.5). Приказ корисничког профила садржи пуно име корисника, имејл адресу, кратак преглед првих 10 најбитнијих кључних речи инжењера, као и првих 10 најбитнијих кључних речи менаџера (уколико је у питању менаџер и оне постоје). Затим, ту је приказ рецензената који су извршили највише рецензија измена датог корисника, сортираних према опадајућој активности. Након тога дат је преглед свих кључних речи са њиховим тежинским факторима, као и преглед свих менаџерских кључних речи.

4.5 Евалуација

Да би се одредило колико је „PR Insights” веб сајт значао инжењерима, имплементација сајта укључивала је и телеметрију о посетама. Сакупљени су подаци о предлозима измена из једног репозиторијума, и сајт је учињен доступним групи од око 300 инжењера из индустрије који су радили на изабраном репозиторијуму.

Резултати - ИП4.1.

ИП4.1. Да ли постоји потреба за додатним системом за претрагу експерата и предлога измена у системима за колаборативни развој софтвера?

На Слици 4.6 приказани су бројеви укупних (а) и јединствених (б) посета страницама сајта, агрегираним по сваком дану од тренутка имплементације система (24.12.2021.) до завршетка експеримента (24.10.2022.). Као што се може видети, постојали су тренуци када је сајт био јако користан инжењерима, и тада је број посета (захтева) скакао и до скоро 50 у току једног дана. То би упућивало на ситуацију у којој је корисник желео да пронађе јако специфичне информације (нпр. да је претраживао базу предлога измена на онај који садржи одређени подстринг али да није могао да се сети тачног подстринга који је предлог измена садржао). Током већине посматраних дана није било активности, и када је било активности ретко када је број посета био већи од 10 током једног дана. Са друге стране, гледајући јединствене

посете (број различитих корисника који су посетили сајт у току једног дана), може се видети да током дана када је било више посета оне су углавном долазиле од једног до два корисника, са само два дана који су имали више од тога.

Када се посматра дистрибуција броја корисника према укупном броју посета (Слика 4.7), може се видети да је већина корисника приступила сајту до 10 пута (16/24, односно $\approx 67\%$ корисника), да је мањи број приступио сајту нешто више од тога (6/24 корисника, односно $\approx 25\%$ корисника посетило је сајт између 10 и 25 пута), и само 2 корисника је посетило сајт више од тога, где је максималан број посета по кориснику био ≈ 90 . На основу тога би се могло претпоставити да је већина корисника приступала сајту како би га испробали или када не би могли да пронађу информације на други начин, али да је пар корисника вероватно укључило „PR Insights” сајт у свој скуп алата за претраживање информација.

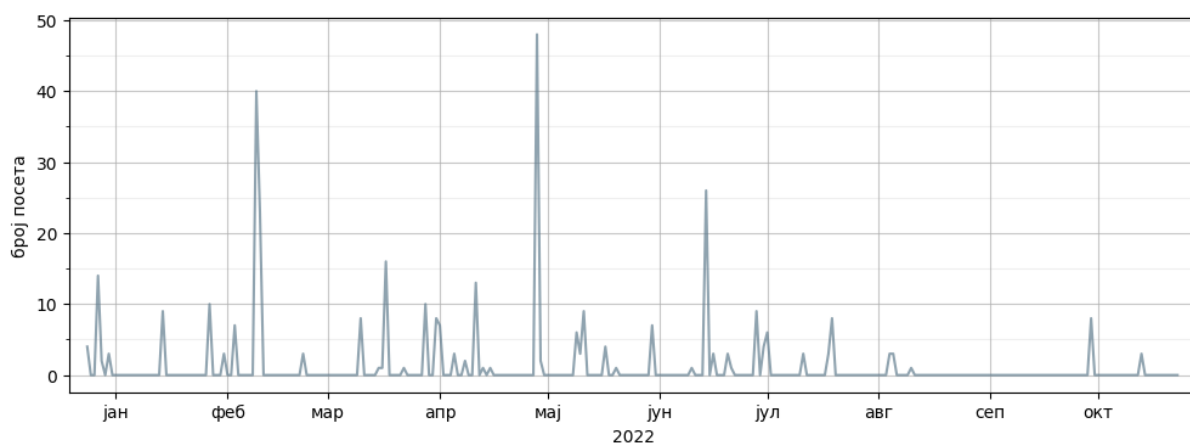
На основу ових резултата може се закључити да је само неколико корисника активније приступало веб сајту, што показује извесну потребу за овим или сличним решењем. На жалост, како искоришћеност није била превелика, није се даље улагало у имплементацију пуне верзије система (**Фаза 2.**), већ је истраживање завршено прототипом из прве фазе.

4.6 Дискусија

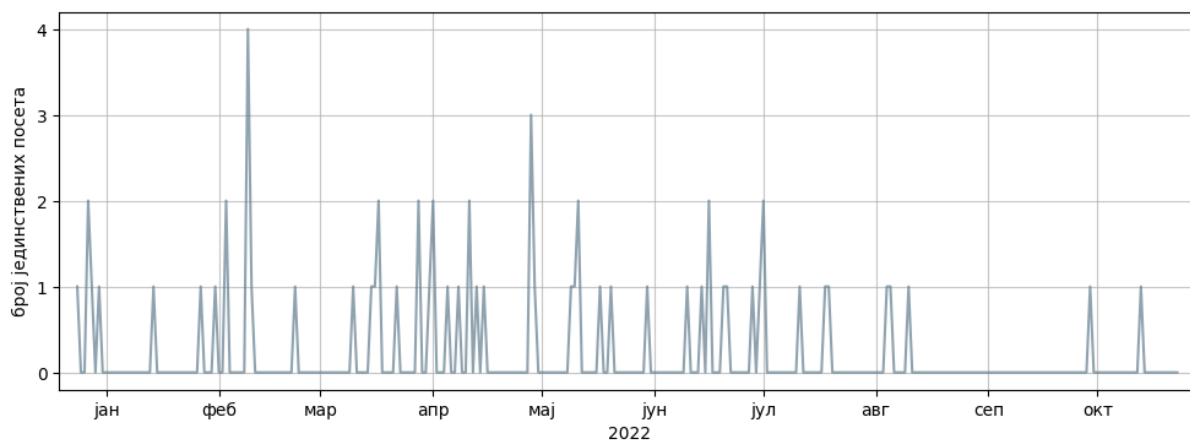
Како је телеметрија показала, иако је предложено решење учињено доступним групи од око 300 инжењера, само 24 њих је макар једном посетило веб сајт, што је релативно мали број. Од оних који су посетили сајт, само неколико њих га је искористило у већој мери. За тих неколико би могло да се каже да је постојала потреба за предложеним решењем. Али, чак и у случају да остали корисници немају потребу за предложеним решењем, очекивало би се да је већи проценат базе корисника посетио сајт макар једном да испроба функционалности. Зато се сматра да већи број инжењера није био упознат са постојањем решења, и да је требало више да се ради на његовој промоцији.

Претње валидности

Један од разлога зашто је посећеност веб сајта могла бити релативно ниска је услед чињенице да је су подаци сакупљени само једном и да нису касније ажурирани. Иако је план истраживања био да се не имплементира пуно решење (**Фаза 2.**) уколико прототип (**Фаза 1.**) не достигне задовољавајућу искоришћеност/посећеност, постоји могућност да би



(а) укупне посете



(б) јединствене посете

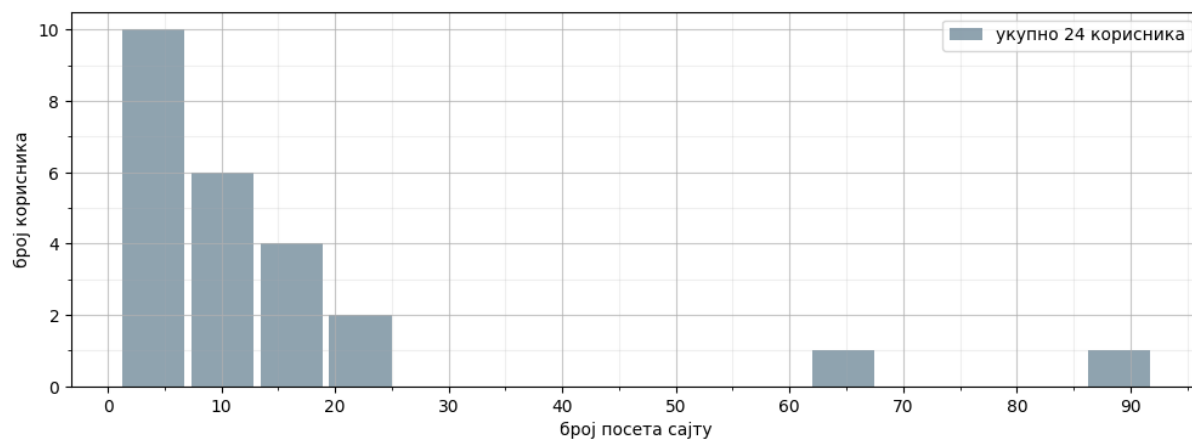
Слика 4.6: Укупне и јединствене дневне посете „PR Insights” веб страници кроз време.

најсвежији подаци могли да привуку већи број корисника и постигну боље резултате.

Није вршено озбиљније рекламирање веб сајта осим неколико мејлова који су послати целој групи инжењера. Постоји могућност да већина групе није ни била свесна постојања сајта, и да га је зато само 20-ак корисника посетило током трајања експеримента. Било би пожељно да је испитано колико инжењера је знало за овај сајт, као и да је више напора уложено у његово рекламирање скупу инжењера.

Перспективе за будућа истраживања

Поред мерења посећености сајта, пожељно би било да је урађена анализа релевантности кључних речи и њихових тежина. То је могло да се постигне анкетирањем одређеног броја корисника који би за себе (и неколико колега са чијим су радом упознати) проценили колико



Слика 4.7: Дистрибуција броја корисника „PR Insights” веб странице према укупном броју посета.

су њихове кључне речи релевантне. Од корисника се могло тражити да сами дају тежинске факторе својим кључним речима, и да се тако добијени тежински фактори упореде са TF-IDF тежинским факторима како би се проценила ваљаност изабране методе.

Осим додатног рекламирања веб сајта које је поменуто у 4.6, зарад лакшег приступа решењу могла је да се направи Azure DevOps екстензија [64]. Тиме корисници не би морали да посећују додатни веб сајт како би приступили предложеном решењу, већ би све могли да ураде у оквиру самог Azure DevOps сервиса. Овај приступ би вероватно додатно повећао потражњу и искоришћење решења.

5 Подсећање рецензената

Када аутор имплементира измену и затражи њену рецензију од стране остатка тима, некад се дешава да прођу дани (чак и недеље) до прве активности рецензената [65–68]. Понекад се дешава да рецензенти, иако добију иницијалну нотификацију о новој измени и рецензији која се очекује од њих, просто забораве на дату рецензију, поготову уколико у краћем временском периоду добију више захтева за рецензију [69]. Како је софтверским инжењерима веома битно да се рецензија њихових измена изврши у што краћем року [70], у случају да у току очекиваног периода нема активности на рецензији, аутор тада мора да контактира рецензенте и подсети их на своју измену [66], јер време до прве активности рецензената је корелисано са дужином трајања рецензије [65].

У овом поглављу представљен је проблем неактивних рецензената и потреба за слањем подсетника за активне рецензије. Представљена су два решења за подсећање корисника (5.1), где је једно директно везано за подсећање корисника на рецензије софтверског кода (5.1.1), док је друго дизајнирано да подсећа кориснике на активности из имејлова (5.1.2). Затим је дефинисан проблем који истраживање у овом поглављу покушава да реши уз дефиниције 5 истраживачких питања (5.2), након чега је представљено идејно решење за подсећање рецензената које представља примену *briefing* мејлова сервиса Viva Insights на проблем рецензије софтверског кода. Како би се испитала валидност идејног решења, приступљено је његовој евалуацији анкетирањем групе софтверских инжењера (5.4), чиме су добијени одговори на прва 3 истраживачка питања. Након успешне валидације идејног решења, приступљено је његовој имплементацији (5.5). Имплементирано решење је учињено доступним групи софтверских инжењера једног репозиторијума, и након више месеци коришћења система извршена је додатна евалуација (5.6), чиме су добијени одговори на

преостала 2 истраживачка питања. На крају је дат преглед постигнутих резултата, уз дискусију о претњама њиховој валидности и коментарима о даљим правцима истраживања (5.7).

5.1 Постојећа решења

Претрагом литературе пронађено је једно постојеће решење (*Nudge*) које се бави подсећањем рецензената и аутора на активне рецензије софтверског кода, и његов кратак опис је дат у наставку. Поред тога дат је и опис решења које се примењује на сличан проблем (*briefing* мејловима сервиса Viva Insights) где се, између осталог, периодично добија подсетник са списком мејлова на које је корисник заборавио да одговори.

5.1.1 Nudge

Један од начина на који се може помоћи инжењерима јесте имплементација система који аутоматски подсећа рецензенте на рецензије које чекају на њихову активност. Један такав систем (*Nudge*) креиран је и тестиран прво на око 150, а затим на 8000 репозиторијума компаније Microsoft [71]. Систем је осмишљен тако да се на свакој рецензији за коју се процени да је потребно подсећање остави коментар који помиње (*@mention*) [72] аутора или рецензента од кога се очекује активност. Корисник на чију активност се чека одређује се на основу неколико фактора који зависе од верзије система. Систем је направљен у две варијанте:

- **стандардна верзија (*Nudge-LT*)** - садржи само подсистем за предвиђање дужине трајања рецензије, али без подсистема за идентификацију корисника и детекцију активности. Сходно томе, корисници су подсећани на рецензије само уколико је тренутни животни век рецензије већи од предвиђеног.
- **„паметна” верзија (*Nudge-FULL*)** - укључује подсистеме за предвиђање дужине трајања рецензије, идентификацију корисника, и детекцију активности.

Систем ради у неколико фаза:

1. **предвиђање дужине трајања рецензије** - представља линеарну регресију која користи 29 метрика како би предвидео очекивано време трајања рецензије. Наставља

се на следећу фазу само уколико је тренутно трајање рецензије премашило очекивано. Ова фаза је укључена у обе варијанте система.

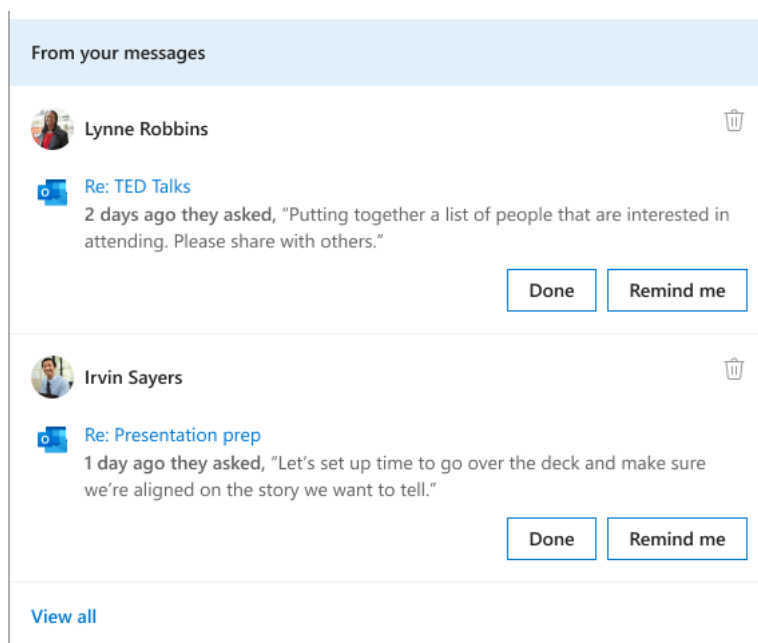
2. **детекција активности** - систем који проверава да ли је било активности у последњих 24h. Прелази се на следећу фазу само уколико није било активности. Ова фаза постоји само у *Nudge-FULL* варијанти система.
3. **идентификација корисника** - на основу информација из рецензије закључује се на којег корисника (аутора или рецензента) се чека на следећу активност. Ова фаза постоји само у *Nudge-FULL* варијанти система.
4. **слање нотификација** - свим одабраним корисницима се шаљу нотификације тако што се на рецензији остави коментар у којем се помињу (*@mention*) идентификовани корисници, што додатно доводи до слања мејла поменутиим корисницима. Ова фаза је укључена у обе варијанте система.

Метрике које се користе у првој фази зарад одређивања очекиваног трајања рецензије су:

1. дан у недељи када је креиран предлог измена
2. просечно време трајања рецензије за предлоге измена датог аутора
3. укупан број обавезних рецензената на датој рецензији
4. да ли је модификован пројектни фајл (*.csproj*)
5. просечно време рецензије оних предлога измена које мењају исте пројекте као дата рецензија
6. број различитих типова фајлова који се мењају у датој измени
7. број речи у опису предлога измена
8. да ли су измењени неки конфигурациони фајлови
9. број активних рецензија у датом репозиторијуму
10. број линија кода које су промењене по класи
11. број измена у целом предлогу измена
12. број функција које имају измене у датом предлогу измена

13. да ли предлог измена уводи нову функционалност
14. број измењених линија у целом предлогу измена
15. број различитих путања које су промењене у датом предлогу измена
16. број условних исказа које су промењене
17. број петљи које су промењене
18. број додатих/модификованих/избрисаних класа
19. да ли предлог измена рефакторише постојећи код
20. промена у броју референци на екстерне пројекте/библиотеке
21. број модификованих фајлова
22. да ли је предлог измена заправо ажурирање кода у једног грани изменама из друге гране
23. да ли се избацују неке застареле функционалности
24. број речи у наслову предлога измена
25. да ли је предлог измена креиран током стандардног радног времена или ван њега
26. да ли предлог измена садржи исправке проблема
27. колико дуго је аутор измена део тренутног тима
28. време од прве активности аутора у датом репозиторијуму
29. колико дуго је аутор запослен у компанији

Nudge систем је био активан 18 месеци на прво над 147 а затим над 8 000 репозиторијума, и током тог периода систем је слао подсетнике за рецензију 210 000 предлога измена које су креиране од стране 40 000 инжењера. Сматра се да је слањем подсетника Nudge-LT довео до смањења времена рецензије за 42.9%, а да је Nudge-FULL верзија убрзала рецензије за 60.6%. Али, и поред ових оптимизација, више од половине ($\approx 53\%$) нотификација основне верзије система биле су негативно прихваћене, а чак и код „пааметнијег” система незанемарљив проценат нотификација је имао негативан сентимент ($\approx 22\%$).



Слика 5.1: Приказ дела „briefing” мејла сервиса Viva Insights који се односи на мејлове који чекају на одговор [76] (© 2025 Microsoft).

5.1.2 Briefing emails (Viva Insights)

Viva Insights [73, 74] је сервис компаније Microsoft који генерише корисне информације из мејлова и чет порука како би повећао продуктивност. Једана од опција коју нуди овај сервис су „briefing” мејлови [75] - то су мејлови који сваког радног дана дају преглед:

1. активности које се очекују од корисника, на основу претходних имејл преписки (нпр. уколико је од корисника тражена нека информација у мејлу на који он није одговорио).
2. докумената који би могли бити релевантни за састанке које корисник има у току дана.
3. термина у којима корисник може да закаже „фокус” време, током којег ће му бити искључене нотификације.

Од интереса за ово поглавље је прва ставка претходне листе. Пример мејла који подсећа корисника на активности које се очекују од њега дат је на Слици 5.1. У овом примеру може се видети да је систем Viva Insights закључио да је корисник добио два мејла у којима се тражи његова активност и да се онда вероватно није десио. У првом мејлу то је закључено на основу реченице „*Please share with others.*”, а у другом на основу захтева за састанак („*Let's set up time to...*”).

5.2 Поставка проблема

Један од коментара незадовољних корисника *Nudge* система био је да су рецензенти свесни рецензија и да им нотификација није корисна, што указује на то да део корисника није желео да прима овакав вид подсетника. Са друге стране, како једном рецензенту може бити додељено више рецензија истовремено, добијање подсетника за сваку од њих одвојеном нотификацијом можда није идеално решење. Зато се уводи претпоставка да би (осим могућности да се одјаве од нотификација) рецензентима више одговарало да периодично добију преглед свих активних рецензија у једној нотификацији, слично *briefing* мејловима сервиса Viva Insights.

Како би се испитала потреба за таквим решењем и његова адекватност, дефинишу се следећа истраживачка питања:

ИП5.1. Да ли инжењери желе да примају периодичне нотификације о рецензијама софтверског кода?

ИП5.2. Колико често би инжењери преферирали да примају нотификације о рецензијама софтверског кода?

ИП5.3. Да ли у периодичне нотификације треба укључити све активне рецензије софтверског кода, или одређени подскуп?

ИП5.4. Колико би рецензенти интераговали са периодичним нотификацијама о рецензијама софтверског кода?

ИП5.5. Да ли периодичне нотификације о рецензијама софтверског кода могу имати утицај на продуктивност рецензената?

Како би се одговорило на ова истраживачка питања коришћени су упитници и телеметрија система за подсећање рецензената.

5.3 Предлог решења

Предложено решење базира се на приступу *briefing* мејлова Viva Insights сервиса - идеја је да рецензентима периодично стиже мејл са прегледом свих релевантних рецензија. У односу на преференције већине корисника би се прецизније одредио период слања мејлова, као и

критеријум релевантности рецензија (све рецензије, или само рецензије са новим изменама, итд.). Пријављивање на сервис за подсећање би се вршило на добровољној бази, и уколико би неки корисник пожелео да престане да прима нотификације, једноставном акцијом („на клик”) би могао да се одјави са сервиса.

5.4 Евалуација идејног решења

Како би се одредило да ли постоји потреба за системом за периодично подсећање рецензената, прво се приступило анкетирању инжењера. Анкета је послата групи од нешто више од 300 инжењера једног репозиторијума из индустрије. Анкета је садржала 4 питања и може се видети у Табели 5.1, а резултати анкете дати су у Табели 5.2.

Резултати - ИП5.1.

ИП5.1. Да ли инжењери желе да примају периодичне нотификације о рецензијама софтверског кода?

Као што је и описано, циљ првог питања у анкети била је провера да ли би инжењерима било корисно да предложени систем постоји, тако што би се сваки испитаник изјаснио да ли би желео да га користи. Свих 17 испитаника који су се одазвали позиву да попуне анкету одговорило је да би желели да користе описани систем.

Резултати - ИП5.2.

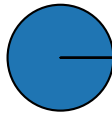
ИП5.2. Колико често би инжењери преферирали да примају нотификације о рецензијама софтверског кода?

Друго питање односило се на периодичност подсетника. 4/17 ($\approx 23.5\%$) испитаника одговорило је да жели подсетнике сваког дана, 8/17 ($\approx 47.0\%$) испитаника желело би да их добија сваки други дан, док су осталих 5/17 ($\approx 29.4\%$) сматрали да би подсетници требали да се шаљу са периодом од три дана.

Табела 5.1: Упитник о периодичним подсетницима на активне рецензије.

PR Reviewer briefings - PR Insights	
We're planning on adding a new feature to the PR Insights project, which is periodic briefings about your pending PR review requests, which you would get by email. The goal of this questionnaire is to gain insight into how often those notifications should be sent, what should the notifications contain, etc. The form is anonymous and the answers are non-binding, meaning that you will not be opted-in for the briefing emails based on your answers - this form is basically a market research questionnaire. General remarks The idea is to add a feature similar to Microsoft Viva's daily briefing emails, which gives you an overview of your daily meetings. Similarly, we would like to add a briefing email about your pending PR reviews, in order to increase the speed at which PRs are getting reviewed, and in order for the PR owner to have a more engaging experience with the reviewers - we all know of the times when some review was published and left for days without even a comment, and we want to change that. This questionnaire will be anonymous, and will be used to determine the starting parameters for the briefing emails. The emails will initially be sent only to a subset of engineers who opt in for it (the opt-in process would come later), so answer the questionnaire only based on how often YOU would like to see this kind of notifications, but do consider both sides - when you're the PR owner, you would like your reviewers to be notified frequently in order to quickly finish the PR, but as a reviewer, getting a bunch of notifications might be overwhelming. The email you would get (every N days) would look like: Hi Name, These are the active PRs which you're a reviewer on: - PR 1 - PR 2 We also plan on sending the emails for PRs with some filter - e.g. only active PRs with new iterations, so in case you reviewed a PR, you won't be notified again for it until needed, so when a new iteration is added, that PR will be mentioned in the next periodic briefing email.	
1	I would like to receive emails briefing me about my pending PR reviews. We just want to see if a feature like this makes sense. This is just to check if engineers would like to get these briefing emails. ☐ Y ☐ N
2	I would like to receive the notifications every N days. Currently, some idea we had would be to send these emails every 2-3 days (e.g. every Mon and Thu, or Mon-Wed-Fri). —
3	I want Draft PRs to be included. For draft PRs, they are usually WIP, and probably shouldn't be included. What do you think? ☐ Y ☐ N
4	I want a PR which I'm reviewing to be a part of the periodic briefings if: The idea here is to determine what filter to use. Currently, we're considering adding the PR to the email if an iteration is added after your last review (comment or vote) - meaning "there are some changes which you haven't seen, take a look". ☐ no ifs, I want all active PRs which I'm a reviewer on to be included in the emails ☐ a new comment is added (by anyone) ☐ a new comment is added as a reply to my comment ☐ a new iteration is added which I haven't reviewed ☐ only if I haven't already reviewed a PR (only if I haven't already accepted the changes)

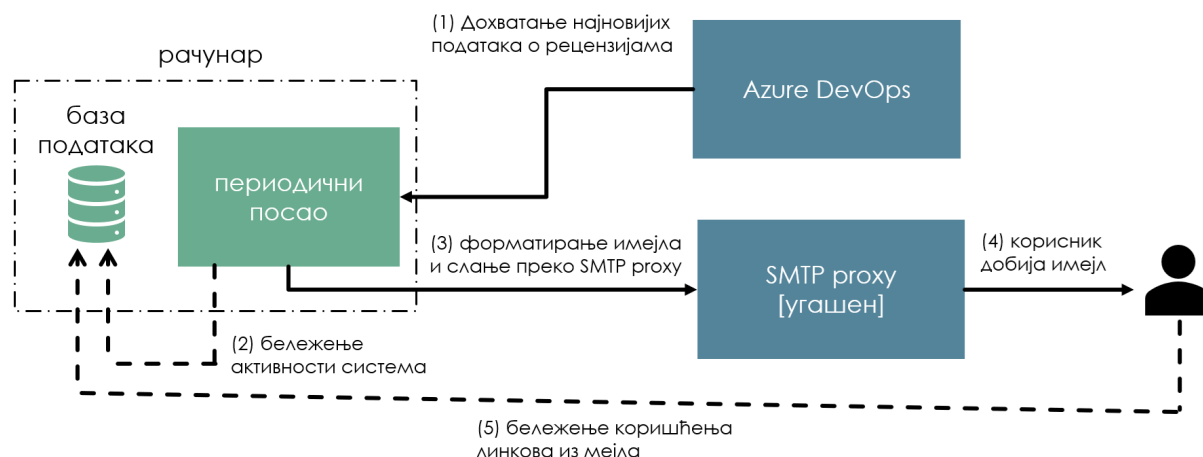
Табела 5.2: Резултати упитника о периодичним подсетницима на активне рецензије.

1	I would like to receive emails briefing me about my pending PR reviews. ● Y 17 ● N 0	
2	I would like to receive the notifications every N days. ● 1 4 ● 2 8 ● 3 5	
3	I want Draft PRs to be included. ● Y 3 ● N 14	
4	I want a PR which I'm reviewing to be a part of the periodic briefings if: ● no ifs, I want all active PRs which I'm a reviewer on to be included in the emails 11 ● a new comment is added (by anyone) 1 ● a new comment is added as a reply to my comment 1 ● a new iteration is added which I haven't reviewed 2 ● only if I haven't already reviewed a PR (only if I haven't already accepted the changes) 2	

Резултати - ИП5.3.

ИП5.3. Да ли у периодичне нотификације треба укључити све активне рецензије софтверског кода, или одређени подскуп?

Треће питање односило се на нацрте измена. Наиме, у неким случајевима може се креирати нацрт измене и он може да се пошаље на рецензију иако сама измена није довршена. На пример, када се ради о јако компликованим изменама, пожељно је да рецензенти што раније процене да ли је приступ решавању проблема ваљан, па самим тим није потребно да сви детаљи буду разрађени, па се у таквим случајевима могу креирати нацрти измена. Већина испитаника (14/17, односно $\approx 82,4\%$) изјаснила се да не би желела да буде подсећана на рецензије нацрте измена.



Слика 5.2: Шематски приказ решења за периодично подсећање рецензената на активне рецензије. Плави елементи су већ постојали пре имплементације решења. Зелени елементи су креирани за потребе овог решења.

Четврто питање било је везано за потенцијално филтрирање предлога рецензија. Како се предложене измене могу наћи у неколико различитих фаза рецензије, од тога да се пошаље нова итерација кода, или да неки други рецензент остави нови коментар (било потпуно независан, било као одговор на коментар датог рецензента), до тога да рецензент није прегледао измену, или је прегледао само неку претходну верзију измене/коментара, валидно је питање да ли неке од новина на предлогу измена треба игнорисати. Иако је постојала разлика у мишљењима међу испитаницима, већина (11/17, односно $\approx 64.7\%$) је одговорила да би желели да имају преглед свих рецензија, независно од фазе рецензије.

5.5 Детаљи имплементације

Како је утврђено да постоји интересовање за описаним идејним решењем, приступљено је његовом имплементирању. Шематски приказ финалне имплементације дат је на Слици 5.2. На једном рачунару подешен је периодични посао користећи *Windows Task Scheduler* [77]. Како је већина испитаника одговорила да би желели да примају периодичне подсетника сваког другог дана (и како су остали одговори били у опсегу од ± 1 дан), одлучено је да се подсетници шаљу сваког понедељка, среде, и петка ујутру, па је периодични посао тако и подешен. Када се посао окине, он прво прође кроз списак корисника који су се пријавили да примају подсетнике, и за сваког користећи ADO REST интерфејс дохвати списак активних предлога измена на којима су наведени као рецензенти.

Како је већина испитаника изразила жељу да добија преглед свих активних рецензија, то је и

урађено. Али, зарад бољег прегледа као и бољег адресирања преференци осталих испитаника, рецензије су подељене у више категорија, тако да они рецензенти који би желели да имају преглед само једне категорије могу то лако да учине. Одабране су 3 категорије:

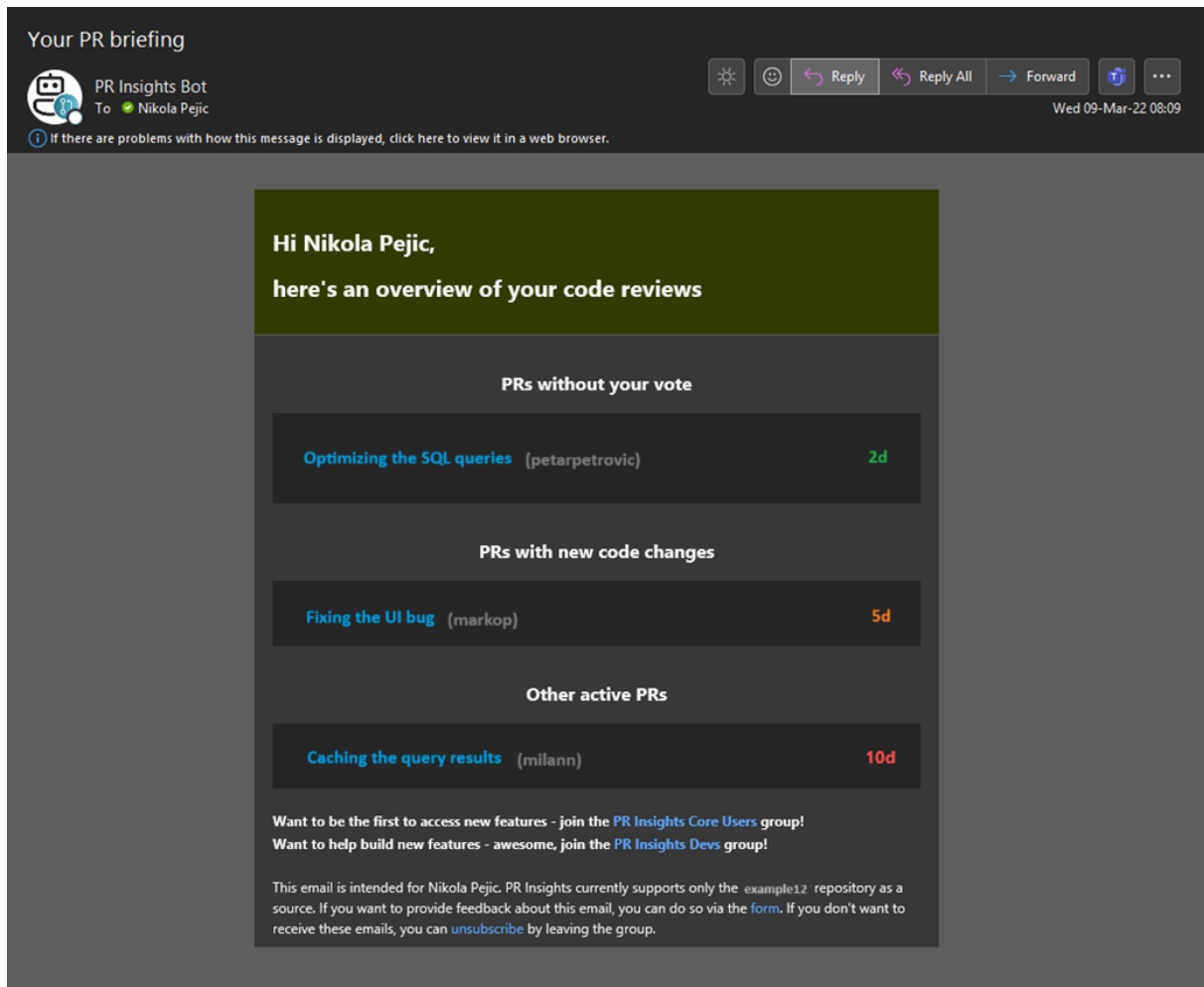
- Нове рецензије које рецензент није обрадио (на којима рецензент није гласао да се прихвати или одбије измена).
- Рецензије које је рецензент обрадио (на којима је гласао), али које су у међувремену измењене/допуњене новом верзијом кода.
- Рецензије које је рецензент обрадио и које немају нове измене.

Након категоризације рецензија, у базу (SQL Server инстанца) која је подигнута на истом рачунару уписују се подаци везани за активност система. Конкретно, бележи се број рецензија у свакој категорији за сваког корисника система. Затим се унапред припремљени HTML шаблон мејла попуњава подацима о рецензијама. На крају се припремљени мејлови шаљу одговарајућим корисницима помоћу *Simple Mail Transfer Protocol* (SMTP) проху сервера. Како би се пратиле интеракције корисника са послатим мејловима, свака рецензија који је излистана у мејлу садржи линк који прво води до REST сервера чији је задатак да само упише у базу да је дошло до клика на линк, и да затим проследи (*http redirect*) захтев корисника на ствари линк до ADO странице за одговарајућу рецензију.

Пример мејла који стиже корисницима дат је на Слици 5.3. У примеру свака категорија рецензија има по једну предложену измену, које су описане својим насловом, налогом аутора (у заградама), и бројем дана који су прошли од креирања рецензије. Мејл је послат од стране *PR Insights Bot* налога за аутоматизацију. На дну мејла дато је упутство за давање повратних информација о сервису, као и упутство за одјављивање са сервиса и престанак слања подсетника датом кориснику.

5.6 Евалуација имплементираних решења

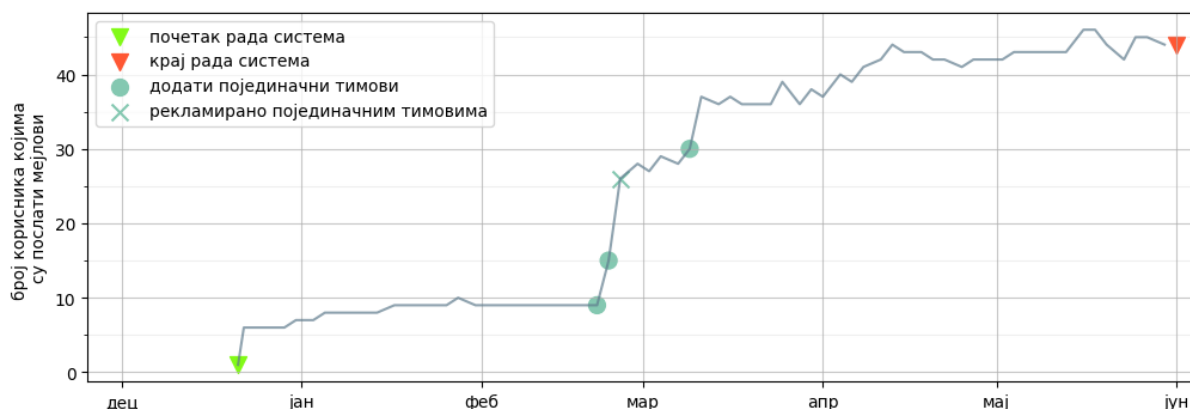
Након имплементације решења, систем је покренут 21. децембра 2021. године и група од више од 300 инжењера једног репозиторијума из индустрије обавештена ја о постојању система, начину пријављивања и одјављивања, и могућности да се затражи једнократни пример мејла. Систем је угашен нешто више од 5 месеци касније (1. јуна 2022. године) услед гашења SMTP проху сервера и немогућности да се довољно једноставно пређе на неки други



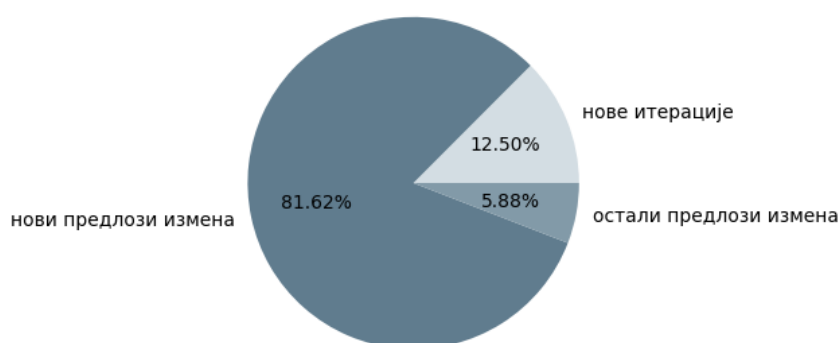
Слика 5.3: Пример мејла који систем за периодично подсећање рецензената на активне рецензије шаље корисницима. Могу се видети све 3 категорије рецензија. Свака рецензија представљена је својим насловом (у плавој боји), налогом аутора (у сивој боји у заградама), као и бројем дана који су прошли од креирања рецензије (у зеленој, наранџастој, или црвеној боји, у зависности од критичности).

начин слања мејлова.

Током периода када је сервис био активан, укупно 4 инжењера је затражило једнократни пример мејла, и укупно 46 инжењера је пријављено на сервис током његовог постојања. На Слици 5.4 може се видети број корисника којима су слати мејлови кроз време, уз назначене битне датуме. Како би се повећао број корисника, уместо различитих кампања које би циљале појединачне кориснике, током фебруара и марта директно су контактирани поједини менаџери са захтевом да се њихови подређени аутоматски пријаве на систем. Уколико би менаџери то прихватили, њихови запослени били би обавештени о томе да ће бити аутоматски пријављени на сервис, уз основне информације о сервису и упутство за одјаву.



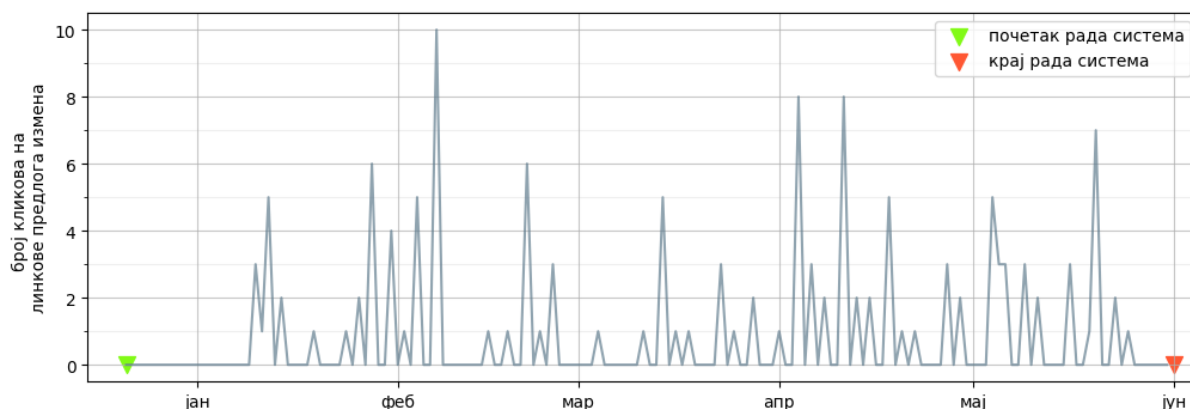
Слика 5.4: Промена броја корисника система којима су послати мејлови кроз време. На графику су означене битне тачке, као што су почетак и крај рада система, датуми када су на систем додати цели тимови (менаџери са својим директним подређенима), и датуми када је систем директно препоручен појединим тимовима.



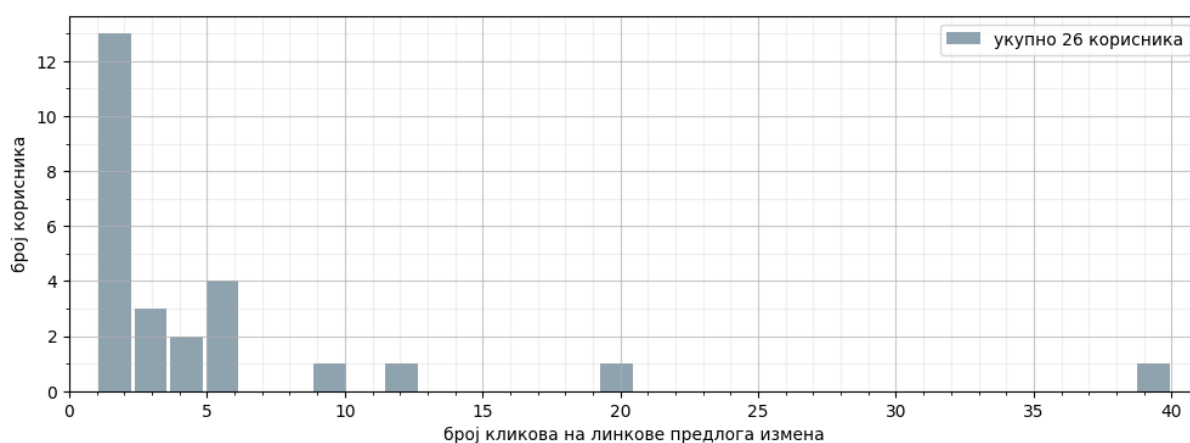
Слика 5.5: Преглед дистрибуције предлога измена према категоријама у послатим мејловима.

Већина контактираних менаџера (3/4) подржала је аутоматску пријаву њихових подређених, што је резултовало већим скоковима у броју корисника, док је преостали менаџер преферирао само додатно рекламирање сервиса њиховим подређеним. На основу телеметрије о раду система закључено је да је послато ≈ 1800 мејлова током периода када је систем био активан.

На основу промене броја корисника кроз време (Слика 5.4), могло би се закључити да је систем био добро прихваћен и да додатни мејлови нису довели до превелике сметње која би довела до одјављивања са сервиса. Већина подсетника на предлоге измена се односила на потпуно нове измене које нису претходно биле обрађене од стране рецензента (Слика 5.5), што је био и главни разлог за креирање система.



Слика 5.6: Преглед временске дистрибуције кликова на линкове рецензија у послатим мејловима. На графику су означене битне тачке, као што су почетак и крај рада система.



Слика 5.7: Дистрибуција корисника према броју кликова на линкове рецензија.

Резултати - ИП5.4.

ИП5.4. Колико би рецензенти интераговали са периодичним нотификацијама о рецензијама софтверског кода?

Када се посматра телеметрија о броју кликова на линкове предлога измена у мејловима (Слика 5.6), може се видети да је током целог трајања рада система постојала релативно слична дистрибуција кликова. Током већине дана када је било активности, број кликова углавном није прелазио 6 кликова. Како је пред крај рада система било пријављено 46 корисника, дистрибуција кликова делује помало разочаравајуће, јер указује да је већина корисника игнорисала мејлове, мада постоји мала могућност да корисници само нису користили линкове у мејловима као пречице за приступ рецензијама. Слична ствар се може закључити посматрајући дистрибуцију броја корисника према броју кликова на линкове

Табела 5.3: Анкета о сентименту корисника према систему за подсећање рецензената, као и субјективном осећају утицаја система на њихову продуктивност.

PR Insights - Briefing Emails	
Checking if there is sufficient interest in the feature.	
Why? The email endpoint the briefing emails use is being deprecated, so this is to assess if a fix is needed, or if the PR Briefing Email service should be deprecated as well.	
The form is anonymous.	
1	Do you want to continue receiving the briefing emails? ☐ Yes ☐ No ☐ Whatever, don't really care
2	The PR Briefing emails have a _____ impact on my productivity. ☐ Absolutely positive ☐ Somewhat positive ☐ Neutral (no impact) ☐ Somewhat negative ☐ Absolutely negative

(Слика 5.7). Тек нешто више од половине корисника (26/46, односно $\approx 56\%$) је макар једном кликнуло на линкове у мејловима. Од оних који су кликнули макар једном, половина (13) је кликнуло само једном, док се скоро сви могу обухватити у категорији до пет кликова на мејлове (22). Тек четири корисника је нешто активније користило мејлове, где би се само за једног или два могло рећи да су их активно користили.

Резултати - ИП5.5.

ИП5.5. *Да ли периодичне нотификације о рецензијама софтверског кода могу имати утицај на продуктивност рецензената?*

Поред доношења закључака на основу анализе телеметрије, битно је испитати и сентимент корисника према систему. Пред престанак рада система услед гашења SMTP проху сервера, разматрана је могућност реимплементације коришћењем неког другог начина слања мејлова. Како би се одредило да ли додатна улагања у систем имају смисла, послата је анкета која је корисницима система објаснила ситуацију, и у којој су могли да дају своје мишљење поводом тога да ли би желели да систем настави са радом, као и њихов субјективни осећај везан за утицај мејлова које су добијали на њихову продуктивност (Табела 5.3).

Резултати анкете могу се видети у Табели 5.4. Од 46 корисника система, 19 ($\approx 41\%$) је одговорило на анкету. Већина испитаника (11/19, односно $\approx 58\%$) одговорило је да жели да

Табела 5.4: Резултати анкете о сентименту корисника према систему за подсећање рецензената.

1	Do you want to continue receiving the briefing emails?			
	● Yes		11	
	● No		2	
	● Whatever, don't really care		6	
2	The PR Briefing emails have a _____ impact on my productivity.			
	● Absolutely positive		2	
	● Somewhat positive		9	
	● Neutral (no impact)		8	
	● Somewhat negative		0	
	● Absolutely negative		0	

настави да прима мејлове са подсетницима. Уколико се посматрају позитивни и неутрални одговори (испитаници којима је свеједно да ли ће наставити да примају мејлове), долази се до броја од 17/19 (односно $\approx 89\%$) испитаника којима би наставак рада система могао да има смисла. Само 2/19 (односно $\approx 10\%$) испитаника не би желело да настави да прима мејлове са подсетницима. По питању субјективног утиска о утицају мејлова на њихову продуктивност, већина испитаника (11/19, односно $\approx 58\%$) одговорила је да су мејлови имали позитиван утицај на њихову продуктивност, док је остатак испитаника одговорило да мејлови нису имали утицај на њихову продуктивност. Ни један испитаник није осећао да су мејлови имали негативни утицај на продуктивност.

5.7 Дискусија

Истраживање у овом поглављу фокусирао се на примену постојећег формата за подсећање корисника (*briefing* мејлова сервиса Viva Insights) на област рецензије софтверског кода и подсећање рецензената. Дефинисано је 5 истраживачких питања, где су се прва 3 бавила валидацијом идејног решења. Анкетирањем групе софтверских инжењера валидирано је да постоји жеља за коришћењем предложеног решења, и прецизирана је његова конфигурација. Затим је идејно решење имплементирано и учињено доступним корисницима током периода од пет месеци, након чега је анализирана телеметрија о искоришћењу система, а поред тога је извршена још једна анкета корисника. Закључено је да је неколико корисника активније користило систем, док је већина спорадично интераговала са њим. Велика већина корисника



Слика 5.8: Приказ релативне промене 90-ог перцентиала броја рецензија по послатом мејлу. На графику су означене битне тачке, као што су почетак и крај рада система, и тренутак промене аутоматске полисе за доделу рецензената предлозима измена. Са $1x$ означена је просечна вредност до промене полисе, док $2x$, $3x$, итд., представљају вредности које су 2, 3, итд., пута веће од иницијалне просечне вредности.

је желела да настави да користи сервис или је била неутрална ($\approx 89\%$), док су сви корисници сматрали да је сервис имао позитиван ($\approx 58\%$) или неутралан ($\approx 42\%$) утицај на њихову продуктивност, и није било корисника којима је продуктивност била нарушена.

Претње валидности

Потенцијални разлог малог искоришћења мејлова би могао бити услед чињенице да је крајем фебруара дошло је до промене полисе на основу које су се аутоматски додељивали рецензенти предлозима измена. Нуспојава промене полисе била је да су поједини рецензенти јако често додавани на рецензије, што је могло да доведе до њиховог преоптерећења. Ово се може видети када се посматра релативна промена 90-ог перцентиала броја предлога измена по послатом мејлу кроз време (Слика 5.8). Због тога постоји могућност да је одређени број рецензената само одустао од активног праћења својих рецензија.

Треба напоменути да би резултати анкета у овом истраживању могли да пате од пристрасности у узорковању. Наиме, учешће у анкетама је било на потпуно добровољној бази, па је у случају прве анкете (која је спроведена над групом софтверских инжењера који раде на једном репозиторијуму) могло доћи до тога да су одговарали само инжењери који су били заинтересовани за описани систем и који су желели да издвоје пар минута како би одговорили на анкету, па би њени резултати могли да нагињу позитивном сентименту и не морају да осликавају мишљење просечног инжењера. Такође, у другој анкети испитивани

су само корисници који су били пројављени на систем у тренутку спровођења анкете. Иако су у одређеним тренуцима корисници аутоматски пријављивани на систем у већим групама, они су имали могућност да се у било којем тренутку одјаве са система. Зато би се могло очекивати да је крајња група корисника обухватала кориснике који су желели (или макар нису били против) тога да наставе да користе систем. На основу телеметрије о броју корисника којима су послати мејлови/подсетници (Слика 5.4), не делује да је било значајнијег броја корисника који су се одјавили са система, али би могла постојати одређена пристрасност у узорковању.

Перспективе за будућа истраживања

Истраживање у овом поглављу је имало релативно конзервативан приступ, где се није приступило аутоматском пријављивању великог броја корисника на систем. Пријављивање је вршено углавном на добровољној бази уз додавање неколико мањих групица. Како је верификовано да предложено решење нема негативног утицаја на продуктивност и може је побољшати, требало би приступити представљању система већем броју корисника и нешто агресивнијем приступу аутоматског пријављивања нових корисника. Такође, систем треба учинити доступним и инжењерима који раде на другим репозиторијумима, потенцијално и open-source заједници, како би се детаљније испитао утицај у различитим условима, и како би се резултати учинили робуснијим на специфичности појединих репозиторијумима/тимовима (нпр. репозиторијум над којим је систем тестиран имао је промене у полиси за расподелу рецензената током посматраног периода). Затим, потребно је анализирати дужи временски период како би се отклонили шумови услед празника и годишњих одмора.

Било би пожељно да се осим субјективног утицаја на продуктивност, мери и објективан утицај на продуктивност. За тако нешто било би корисно пратити метрике као што су време од креирања предлога измена до прве активности рецензената, као и време до завршетка рецензије и прихватања измена, и упоредити дате метрике пре и после примене система. Друга могућност је А-Б тестирање где би само део инжењера једног репозиторијума био пријављен на систем, чиме би могао да се пореди утицај система на инжењере под истим условима.

6 Системи за препоруку рецензената

Засновано на адаптацији текста публикације:

N. Pejić, Z. Radivojević and M. Cvetanović, "Helping Pull Request Reviewer Recommendation Systems to Focus," IEEE Access, св. 11, стр. 71013-71025, 2023, doi: <https://doi.org/10.1109/ACCESS.2023.3292056>

Одабир рецензената предлога измена софтверског кода може утицати на квалитет [3, 5, 6] као и дужину трајања рецензије [8, 78]. Проблем одабира рецензената постаје још битнији на великим репозиторијумима са огромним бројем инжењера (и самим тим потенцијалних рецензената) не само због потешкоћа одабира релевантних рецензената из великог скупа кандидата, већ и услед чињенице да у великим репозиторијумима инжењери имају тенденцију да се групишу у мање тимове који су фокусирани само на део кода, тиме стварајући локалну али не и глобалну експертизу. Како глобални експерт за одређену област/технологију (нпр. C# језик) не би био рецензент сваке измене кода која има везе са датом облашћу/технологијом (нпр. сваки пут када се мења C# код), у неким репозиторијумима више смисла може имати да се такве рецензије дистрибуирају локалним експертима који заједно могу обрадити већи број једноставнијих измена, а који могу по потреби затражити помоћ глобалног експерта уколико се ради о некој компликованој или проблематичној измени кода. У таквим ситуацијама одабир најрелевантнијег експерта, иако релативно сигуран избор у случају малих репозиторијума, може бити контрапродуктиван када су у питању велики репозиторијуми са мноштвом инжењера. Зато се може очекивати да системи за препоруку рецензената, који би требало да олакшају и убрзају одабир рецензената, могу имати проблеме уколико су дизајнирани или тестирани само над мањим

репозиторијумима.

У овом поглављу дат је преглед неколико постојећих приступа за препоруку рецензената (6.1). Затим је представљен проблем препоруке рецензената у репозиторијумима са великим бројем инжењера - постављено је четири истраживачких питања на које се ово поглавље фокусирало и дефинисане су метрике које су коришћене за мерење перформанси (6.2). Након тога дат је предлог решења које се базира на филтрирању рецензената чиме се врши фокусирање постојећих приступа на мањи број кандидата (6.3), и укратко су описани детаљи имплементације (6.2). Затим је представљен експеримент који је спроведен зарад евалуације предложеног решења над подацима о рецензијама из неколико репозиторијума из индустрије, уз анализу и дискусију о резултатима сваког истраживачког питања (6.5). На самом крају овог поглавља дат је кратак осврт на истраживање које је урађено, уз додатну дискусију резултата и претњама валидности (6.6).

6.1 Постојећа решења

Претрагом литературе одабрано је 7 постојећих приступа који су имплементирани и коришћени за потребе истраживања у овом поглављу. У наставку је дат кратак опис сваког од изабраних приступа, а њихов преглед може се видети у Табели 6.1.

6.1.1 Приступ 1

Приступ 1 [61] представља комбинацију два метода:

1. **Први метод** извлачи вектор речи¹ из наслова, описа и списка рецензената који су оставили коментар, и свакој речи додељује тежински фактор користећи *TF-IDF* [63] који је дефинисан као

$$tfidf_I(t, p_r, P_R) = \log\left(\frac{n_t}{N_{p_r}} + 1\right) \cdot \log\left(\frac{N_{P_R}}{|p_r \in P_R : t \in p_r|}\right), \quad (6.1)$$

где је t реч, p_r је рецензија, P_R је скуп свих рецензија, n_t је број појављивања речи t у рецензији p_r , N_{p_r} је укупан број речи у p_r , и N_{P_R} је број рецензија у скупу рецензија P_R .

Из сваке нове рецензије прво се генерише вектор речи са $tfidf_I$ тежинским

¹Не користе се све речи, већ се прво избацују речи које садрже неалфабетске карактере, везници и друге речи које носе мало информација (*stop words*), а затим се речи стемују (*stemming*).

Табела 6.1: Преглед постојећих метода за препоруку рецензената.

Ознака приступа	Назив рада	Година објављивања
Приступ 1	Reviewer recommender of pull-requests in GitHub [61]	2014
Приступ 2	Who should review my code? A file location-based code-reviewer recommendation approach for Modern Code Review [78]	2015
Приступ 3	Automatically recommending code reviewers based on their expertise: An empirical comparison [79]	2016
Приступ 4	Who should comment on this pull request? Analyzing attributes for more accurate commenter recommendation in pull-based development [80]	2017
Приступ 5	Profile based recommendation of code reviewers [81]	2018
Приступ 6	WhoDo: Automating reviewer suggestions at scale [4]	2019
Приступ 7	Learning to Rank Reviewers for Pull Requests [62]	2019

коэффицијентима, и онда користећи косинусну сличност се одреде k најсличнијих рецензија из историје рецензија P_R . За две рецензије p_{r_1} и p_{r_2} , косинусна сличност одређује се као

$$\text{cosineSimilarity}(p_{r_1}, p_{r_2}) = \frac{\sum_{t \in p_{r_1} \cap p_{r_2}} \text{tfidf}_1(t, p_{r_1}, P_R) \cdot \text{tfidf}_1(t, p_{r_2}, P_R)}{\sqrt{\sum_{t \in p_{r_1}} \text{tfidf}_1(t, p_{r_1}, P_R)} \cdot \sqrt{\sum_{t \in p_{r_2}} \text{tfidf}_1(t, p_{r_2}, P_R)}}. \quad (6.2)$$

Затим се посматрају рецензенти који су остављали коментаре на тих k најсличнијих рецензија, њихове метрике се рачунају тако што се за сваког рецензента d изброји број коментара n_i које је оставио на рецензији p_{r_i} и помножи се косинусном сличношћу рецензије, односно

$$\text{Score}_1(p_r, d) = \sum_{i=1}^k n_i \cdot \text{cosineSimilarity}(p_r, p_{r_i}), \quad (6.3)$$

где је p_r нова рецензија за коју се врши препоручивање рецензента, а p_{r_i} су рецензије из историје репозиторијума P_R .

2. **Други метод** посматра граф коментарисања (*comment network*), који представља тежински усмерени граф у којем су чворови рецензенти, а гране између чвора представљају историју коментарисања и њихов тежински фактор се рачуна као

$$w_{ij} = \sum_{r=1}^k w_{(ij,r)} = P_c \cdot \sum_{r=1}^k \sum_{n=1}^m \lambda^{n-1} \cdot t_{(ij,r,n)}, \quad (6.4)$$

где је k укупан број рецензија које је креирао рецензент, $w_{(ij,k)}$ је парцијална тежина сваке рецензије, P_c је емпиријски коефицијент који одређује утицај сваког коментара на рецензију (аутори Приступа 1 користили су вредност 1.0), k је број број коментара које је рецензент i оставио на рецензији r . Како би се контролисао утицај рецензената који су остављали више коментара на једној рецензији, уведен је фактор пропадања λ (аутори Приступа 1 одабрали су вредност 0.8). Како би се придао већи значај новијим коментарима и рецензијама, додат је временски фактор $t_{(ij,r,n)}$ који се дефинише као

$$t_{(ij,r,n)} = \frac{\text{timestamp}_{(ij,r,n)} - \text{baseline}}{\text{deadline} - \text{baseline}} \in (0, 1], \quad (6.5)$$

где је $\text{timestamp}_{(ij,r,n)}$ тренутак када је рецензент j оставио коментар n у рецензији r коју је креирао рецензент i , baseline и deadline представљају временске границе скупа података над којим је метод трениран (baseline је тренутак непосредно пре креирања прве рецензије, док је deadline тренутак непосредно након креирања последње рецензије).

По другом методу, метрика која одређује подобност потенцијалног рецензента d да изврши рецензију p_r је заправо тежински фактор између аутора рецензије a и рецензента d , односно

$$\text{Score}_2(p_r, d) = w_{ad}. \quad (6.6)$$

Како су аутори Приступа 1 сматрали да су експертиза (прва метода) и историја интеракција рецензената (друга метода) подједнако битни фактори при одређивању подобности рецензената, они су нормализовали добијене метрике (Score_1 и Score_2), сабрали их, и користили тај збир да одреде најподобније рецензенте.

6.1.2 Приступ 2

Приступ 2 [78] рангира потенцијалне рецензенте користећи 4 функције за поређење сличности путања фајлова. За сваког потенцијалног рецензента d , посматрају се све рецензије које су му биле додељене P_d . За нову рецензију r , метрика која одређује релевантност рецензента d се дефинише као

$$score(r, d) = \sum_{r' \in P_d} \sum_{f \in F_r} \sum_{f' \in F_{r'}} filePathSimilarity(f, f'), \quad (6.7)$$

где је F_r скуп свих фајлова који су промењени у рецензији r . $filePathSimilarity(f, f')$ је функција за одређивање сличности путања и она има следећи облик

$$filePathSimilarity(f_1, f_2) = \frac{StringComparison(f_1, f_2)}{\max(Length(f_1), Length(f_2))}, \quad (6.8)$$

где су f_1 и f_2 путање фајлова који се пореде, $Length(f)$ је функција која броји сегменте путање фајла, и $StringComparison(f_1, f_2)$ је нека од функција за поређење путања.

Аутори рада су одабрали да користе следеће функције за поређење путања:

- *LCP (Longest Common Prefix)* - максимална дужина заједничких континуалних сегмената на почетку апсолутних путања.
- *LCS (Longest Common Suffix)* - максимална дужина заједничких континуалних сегмената на крају апсолутних путања.
- *LCSubstr (Longest Common Substring)* - максимална дужина заједничких континуалних сегмената апсолутних путања.
- *LCSubseq (Longest Common Subsequence)* - максимална дужина заједничких сегмената апсолутних путања, идући слева у десно, где је дозвољено прескакање сегмената.

Користећи (6.7) и сваку од наведених имплементација функција за поређење путања одвојено, врши се рангирање свих рецензената и као резултат се добију 4 уређене ранг листе, тако да се сваки рецензент d у i -тој ранг листи има ранг $ord_i(d)$. Затим се врши спајање ранг листи користећи Borda count [82], тако што се финална метрика подобности рецензента $Score(d)$

рачуна као

$$Score(d) = \sum_{i=1}^4 (N - ord_i(d)), \quad (6.9)$$

где је N број кандидата рецензената.

6.1.3 Приступ 3

Приступ 3 [79], полазећи од идеје да фајлови који имају сличне путање имају сличну функционалност, прво дефинише FPS метрику (*File Path Similarity*). Нека су дати скуп свих фајлова F , скуп свих рецензија R , и скуп свих софтверских инжењера D . Нека је $Files : R \rightarrow P(F)$ функција која пресликава сваку рецензију у скуп фајлова који су били део ње, и нека је $CommonPath : F \times F \rightarrow \mathbb{N}$ функција која за свака два фајла броји (с лева у десно) колико нивоа директоријума је заједничко у апсолутним путањама датих фајлова. Слично, нека је $Length : F \rightarrow \mathbb{N}$ функција која броји директоријуме у апсолутној путањи датог фајла. Затим се уводе функције за одређивање сличности два фајла $Similarity : F \times F \rightarrow \mathbb{R}$, као и функција која одређује сличност две рецензије $ReviewSimilarity : R \times R \rightarrow \mathbb{R}$, које су дефинисане као:

$$Similarity(f_1, f_2) = \frac{CommonPath(f_1, f_2)}{\max[Length(f_1), Length(f_2)]}, \quad (6.10)$$

$$ReviewSimilarity(r_1, r_2) = \frac{\sum_{\substack{f_1 \in Files(r_1) \\ f_2 \in Files(r_2)}} Similarity(f_1, f_2)}{|Files(r_1)| \cdot |Files(r_2)|}. \quad (6.11)$$

Уколико су $r_1, r_2, \dots, r_n$ све рецензије из R у хронолошком редоследу, тада се функција $_dR_x$ која даје скуп свих рецензија пре рецензије r_x у којима је рецензент d учествовао дефинише као

$$_dR_x(r_x, d) = \{r_i \in R | 0 < i < x \wedge d \in Reviewers(r_i)\}. \quad (6.12)$$

Затим се дефинише $WRC_\delta : F \times D \times R \rightarrow \mathbb{R}$ функција (*Weighted Review Count*) која представља претпроцесирану метрику експертизе рецензента за дати фајл. WRC_δ се

дефинише као

$$WRC_{\delta}(f, d, r_x) = \sum_{\substack{r_i \in_d R_x \\ f \in Files(r_i)}} \delta^{x-i-1} |Files(r_i)|^{-1}. \quad (6.13)$$

Коначно, дефинише се $FPSScore_{\delta}$ метрика која за задату рецензију r_x и потенцијалног рецензента d као

$$FPSScore_{\delta}(r_x, d) = \sum_{\substack{f_1 \in Files(r_x) \\ f_2 \in F}} \frac{Similarity(f_1, f_2) \cdot WRC_{\delta}(f_2, d, r_x)}{|Files(r_x)|}. \quad (6.14)$$

Оваквом дефиницијом $FPSScore_{\delta}$ метрике могуће је унапред израчунати WRC метрику пре него што је познато који фајлови су промењени у рецензији r_x , чиме је омогућено релативно брзо давање препоруке за сваку нову рецензију без потребе проласка кроз целу историју.

6.1.4 Приступ 4

Аутори Приступа 4 [80] су испробали неколико приступа за препоруку рецензената, али онај који је дао најбоље резултате био је приступ који се базира на активности рецензената. Нека је r_{new} нова рецензија, нека је d_i потенцијални рецензент, нека је R_{d_i} скуп рецензија у којима је рецензент d_i оставио макар један коментар, и нека је $time_{r_j}$ тренутак креирања рецензије r_j . Тада се метрика која одређује релевантност рецензента d_i дефинише као

$$activeness(r_{new}, d_i) = \sum_{(r_j \in R_{d_i}) \wedge ((time_{r_{new}} - time_{r_j}) \leq \gamma)} (time_{r_{new}} - time_{r_j})^{-\lambda}, \quad (6.15)$$

где је γ параметар који дефинише колико историје (претходних рецензија) утиче на релевантност рецензента, а λ је параметар временског пропадања. Аутори приступа су за ова два параметра експериментално утврдили да су оптималне вредности $\gamma = +\infty$ (или $\gamma = 60$ дана) и $\lambda = 1$.

6.1.5 Приступ 5

Приступ 5 [81] конструише профил сваког рецензента. Нека је $f(C_t) = \{f_1^t, f_2^t, \dots, f_n^t\}$ функција која дату рецензију C_t мапира у скуп фајлова који су промењени у њој. Сваки фајл f_i^t

престављен је својом апсолутном путањом, која се састоји од назива фолдера раздвојених специјалним карактером (`'\'` или `'/'`) и коначно назива самог фајла. Затим, нека је $m(f_i^t)$ функција која мапира апсолутну путању фајла f_i^t у мултисет сегмената путање фајла (назива фолдера и фајла). Слично, нека је $m(C_t)$ функција која даје мултисет сегмената путања за све фајлове у рецензији C_t . Нека је $P_i(t)$ профил рецензента R_i у тренутку t дефинисан као мултисет добијен унијом свих мултисетова у којима је рецензент учествовао.

За сваку нову рецензију, прво се генерише мултисет сегмената путања измењених фајлова. Затим се пореди са профилем сваког рецензента користећи Тверски индекс који је дефинисан као

$$T(X, Y) = \frac{|X \cap Y|}{|X \cup Y| - \alpha|X \setminus Y| - \beta|Y \setminus X|}. \quad (6.16)$$

У мерењима које су аутори Приступа 5 вршили утврдили су да су вредности параметра $\alpha = 0$ и $\beta = 1$ оптималне, па су те вредности коришћене током истраживања у овом поглављу. Аутори су такође тестирали више варијација овог приступа, али је претходно описан облик давао најбоље резултате.

6.1.6 Приступ 6

Приступ 6 [4] дефинише неколико метрика које мере колико честе је рецензент r мењао или вршио рецензирање над сваким фајлом f или директоријумом d који је део рецензије rev , и агрегира их уз конфигурабилне тежинске факторе C_1, C_2, C_3, C_4 . Сматра се да је рецензент r прегледао дату рецензију rev када је одобрио или када је оставио макар један коментар на њој. За дату рецензију rev и потенцијалног рецензента r , агрегирана метрика има облик:

$$\begin{aligned} Score(r, rev) = & C_1 \cdot \sum_{f \in F} \frac{n_{change}(r, f)}{t_{change}(r, f)} + C_2 \cdot \sum_{d \in D} \frac{n_{change}(r, d)}{t_{change}(r, d)} + \\ & C_3 \cdot \sum_{f \in F} \frac{n_{review}(r, f)}{t_{review}(r, f)} + C_4 \cdot \sum_{d \in D} \frac{n_{review}(r, d)}{t_{review}(r, d)}, \end{aligned} \quad (6.17)$$

где су F и D скупови фајлова и вршних директоријума који су промењени у рецензији rev , а n_{change} и n_{review} функције враћају целобројне вредности које означавају колико пута је рецензент r мењао односно вршио рецензију над датим фајлом или директоријумом. Слично,

Табела 6.2: Преглед категорија метрика које Приступ 7 користи.

Категорија	Метрике
Сличност путања фајлова и директоријума	ϕ_1, ϕ_2
Сличност наслова рецензије	ϕ_3
Социјалне везе	$\phi_4, \phi_5, \phi_6, \phi_7$
Активност	$\phi_8, \phi_9, \phi_{10}, \phi_{11}, \phi_{12}$
Дан у недељи	ϕ_{13}
Број измењених фајлова	ϕ_{14}

t_{change} и t_{review} су функције које враћају број дана од када је рецензент r последњи пут интераговао (мењао или вршио рецензију, респективно) дате рецензије rev .

Одабиром коефицијената C_1, C_2, C_3, C_4 могуће је придати већу тежину писању измена кода у односу на рецензирање, и сл. Аутори Приступа 6 су у својој евалуацији све коефицијенте поставили на 1, па су те вредности коришћене током истраживања у овом поглављу.

6.1.7 Приступ 7

Приступ 7 [62] користи 14 метрика како би описао однос рецензије и потенцијалних рецензената, а које су према значењу груписане у 6 група (6.2)

Користећи функцију (6.10), дефинишу се прве две метрике ϕ_1 и ϕ_2 као

$$\phi_1(r, d) = \begin{cases} \sum_{f_i \in F_r} \sum_{f_j \in F_d} \frac{Similarity(f_i, f_j)}{|F_r| \cdot |F_d|} & \text{ако је } |F_d| > 0 \\ 0 & \text{ако је } |F_d| = 0 \end{cases} \quad (6.18)$$

$$\phi_2(r, d) = \begin{cases} \sum_{f_i \in F_r} \sum_{f_j \in F_s} \frac{Similarity(f_i, f_j)}{|F_r| \cdot |F_s|} & \text{ако је } |F_s| > 0 \\ 0 & \text{ако је } |F_s| = 0 \end{cases}, \quad (6.19)$$

где је F_r скуп фајлова који су промењени у r , F_d је скуп фајлова које је рецензент d променио у претходном периоду, и F_s је скуп фајлова над којима је рецензент d извршио рецензију у претходном периоду.

Наредна метрика, слично као (6.1) из Приступа 1, користи *TF-IDF* функцију у нешто другачијем облику

$$tfidf_7(t, p_r, P_R) = \left(0.5 + \frac{0.5 \cdot n_t}{\max_{t \in p_r} (n_t)} \right) \cdot \log \frac{N_{P_R}}{|p_r \in P_R : t \in p_r|}. \quad (6.20)$$

И слично као и у Приступу 1, користи се косинусна сличност (6.2) како би се поредила два вектора речи са *TF-IDF* тежинским факторима

$$\phi_3(r, d) = \sum_{r_{old} \in pr(d)} cosineSimilarity(r, r_{old}). \quad (6.21)$$

Нека $pulls_reviewed(q, d)$ представља скуп рецензија које је инжењер q креирао и које је рецензент d прегледао, нека је $pulls_reviewed_{30}(q, d)$ подскуп $pulls_reviewed(q, d)$ који обухвата само рецензије које су обрађене од стране рецензента d у последњих 30 дана од тренутка када је креирана рецензија r . Слично, нека је $pulls_commented(q, d)$ скуп рецензија које је креирао инжењер q и на којима је рецензент d оставио макар један коментар, и нека је $pulls_commented_{30}(q, d)$ подскуп који укључује само рецензије на којима је рецензент d оставио макар један коментар у последњих 30 дана од тренутка креирања рецензије r . Тада се наредне 4 метрике дефинишу као

$$\phi_4(r, d) = |pulls_reviewed(q, d)|, \quad (6.22)$$

$$\phi_5(r, d) = |pulls_reviewed_{30}(q, d)|, \quad (6.23)$$

$$\phi_6(r, d) = |pulls_commented(q, d)|, \quad (6.24)$$

$$\phi_7(r, d) = |pulls_commented_{30}(q, d)|. \quad (6.25)$$

Нека $pulls_reviewed(d)$ представља скуп рецензија које је рецензент d обрадио, и нека је $pulls_reviewed_{30}(d)$ подскуп тог скупа који укључује само рецензије које су обрађене

максимално 30 дана пре креирања рецензије r . На сличан начин су дефинисане и функције $pulls_commented(d)$ и $pulls_commented_{30}(d)$ које представљају скупове рецензија на којима је рецензент d оставио макар један коментар у одговарајућим периодима. Тада се наредне 4 метрике дефинишу као:

$$\phi_8(r, d) = |pulls_reviewed(d)|, \quad (6.26)$$

$$\phi_9(r, d) = |pulls_reviewed_{30}(d)|, \quad (6.27)$$

$$\phi_{10}(r, d) = |pulls_commented(d)|, \quad (6.28)$$

$$\phi_{11}(r, d) = |pulls_commented_{30}(d)|. \quad (6.29)$$

Нека је $last(d)$ тренутак последње рецензије коју је рецензент d извршио, и нека је $last(d).day$ датум који одговара том тренутку. Слично, нека је $r.day$ датум када је рецензија r креирана. Тада се наредна метрика дефинише као

$$\phi_{12}(r, d) = (r.day - last(d).day + 1)^{-1}. \quad (6.30)$$

За дати дан у недељи day када је рецензија r креирана, метрика ϕ_{13} се дефинише као

$$\phi_{13}(r, d) = |pulls_{day}(r, d)|, \quad (6.31)$$

где је $pulls_{day}$ скуп рецензија које је рецензент d током своје историје обрадио током задатог дана у недељи. Идеја са овом метриком је да неки рецензенти преферирају одређене дане у недељи за вршење рецензије, па би најбољи рецензенти представљали оне који су претходно били најактивнији истог дана у недељи када је креирана рецензија.

Нека је $files(r)$ број фајлова који су модификовани у рецензији r . Нека је $avg(files(d))$ просечан број фајлова који су мењани у рецензијама које је рецензент d обрадио. Тада се

последња метрика дефинише као

$$\phi_{14}(r, d) = (|files(r) - avg(files(d))| + 1)^{-1}. \quad (6.32)$$

Како ове метрике могу да имају вредности у веома различитим опсезима то може да утиче на тренирање модела машинског учења, па је пре него што се приступи њиховом коришћењу потребно је да се изврши нормализација. За дату метрику ϕ , нека су $\phi.min$ и $\phi.max$ минимална и максимална вредност израчунате на скупу података за тренинг. Тада се нормализација дате метрике врши на следећи начин

$$\phi' = \begin{cases} 0 & \text{ако је } \phi < \phi.min \\ \frac{\phi - \phi.min}{\phi.max - \phi.min} & \text{ако је } \phi.min < \phi < \phi.max \\ 1 & \text{ако је } \phi > \phi.max \end{cases} \quad (6.33)$$

Са овако дефинисаним метрикама, финална агрегирана вредност релевантности рецензента d за рецензију r се одређује као

$$s(r, d) = \sum_{i=1}^{14} w_i \cdot \phi'_i(r, d), \quad (6.34)$$

где су w_i тежински фактори који одређују релевантност метрике. Да би се они одредили, коришћена је SVM^{rank} имплементација [83] *learning-to-rank* [84] метода машинског учења.

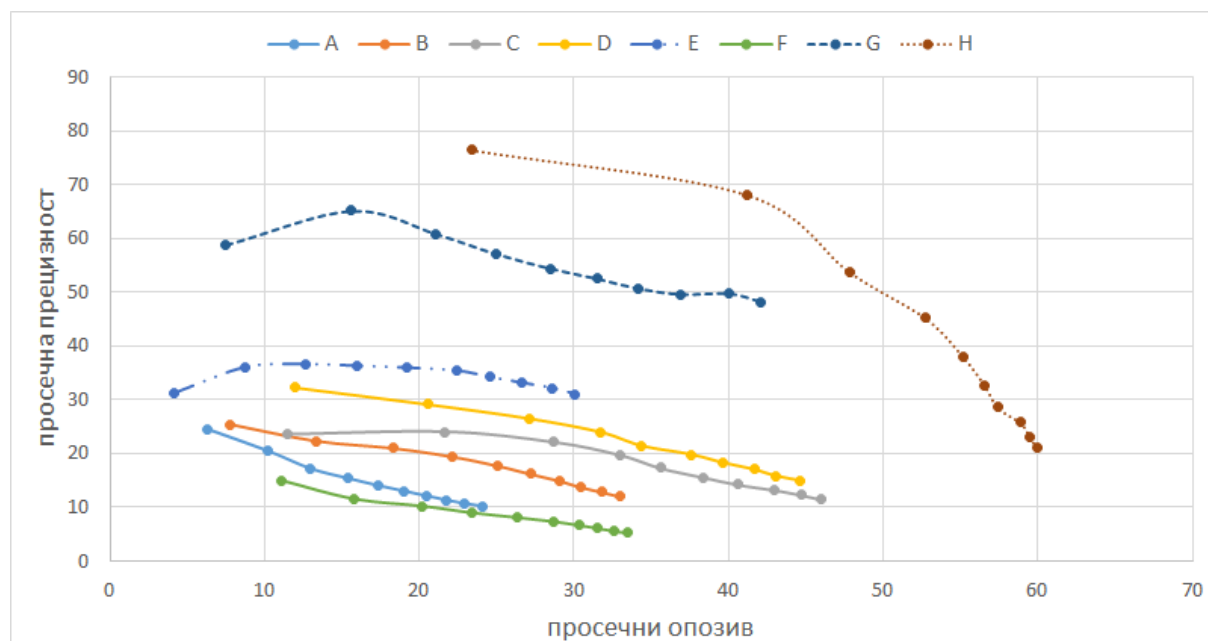
6.2 Поставка проблема

Аутори постојећих приступа описаних у поглављу 6.1 су углавном тестирали своја решења над релативном мањим *open-source* репозиторијумима са платформе GitHub. Од изабраних приступа, само три (Приступ 3, Приступ 4, и Приступ 7) су тестирани над репозиторијумима који имају више од 200 инжењера² (Табела 6.3), и само два (Приступ 3 и Приступ 4) су тестирани над јако великим репозиторијумима који су имали више од 1000 инжењера. Како инжењери у већим репозиторијумима имају тенденцију да се групишу и раде у мањим, уско специјализованим тимовима, могло би се очекивати да неки приступи имају потешкоћа са препоручивањем рецензената у великим репозиторијумима.

²Приступ 2 и Приступ 5 тестирани су над репозиторијумима који су имали максимално 202 инжењера, што није значајно изнад 200.

Табела 6.3: Преглед постојећих решења са статистикама о скупу података над којима су тестирани (© 2023 IEEE) [25].

Приступ	укупан број репозиторијума	дистрибуција репозиторијума према броју инжењера
Приступ 1	10	
Приступ 2	4	
Приступ 3	4	
Приступ 4	8	
Приступ 5	4	
Приступ 6	5	
Приступ 7	12	



Слика 6.1: График просечних прецизност-опозив перформанси свих постојећих метода за различите скупове података (A-H) за 10 најрелевантнијих препорука рецензената (© 2023 IEEE) [25].

Мерењем перформанси постојећих метода над скуповима података A-H (детаљније о скуповима података у 6.5), примећено је да у просеку перформансе постојећих метода имају тенденцију да опадају са порастом величине репозиторијума, што се може видети на Слици 6.1. Ово додатно поткрепљује претпоставку овог истраживања да су тимови у већим репозиторијумима уже специјализовани у односу на тимове у мањим репозиторијумима, и да је у случају већих репозиторијума потребно додатно фокусирати приступе за препоруку рецензената. То је у складу са запажањима да су у већим репозиторијумима инжењери експерти само за мали део кода репозиторијума, са неколицином искуснијих инжењера који су упознати са већим деловима [4].

Уводи се претпоставка да би постојећи методи могли имати користи од смањивања скупа кандидата за рецензију са скупа свих софтверских инжењера на скуп инжењера који имају неку врсту конекције са датом изменом (нпр. интераговали су са аутором измене на претходним рецензијама, вршили су рецензије фајлова који су модификовани у датој измени, итд.), чиме се постојеће методе фокусирају само на релевантне кандидате. Како се може дефинисати неколико критеријума релевантности, дефинише се појам **базичног филтера**, који врши филтрирање на основу једног критеријума, као и појам **композитног филтера**, који врши филтрирање на основу уније више базичних филтера.

Како би се испитао утицај различитих филтера на постојеће приступе, као и на различите скупове података, дефинишу се следећа истраживачка питања:

ИП6.1. Да ли филтрирање скупа потенцијалних рецензената коришћењем базичног филтера доводи до побољшаних перформанси постојећих приступа за препоруку рецензената?

ИП6.2. Да ли филтрирање скупа потенцијалних рецензената коришћењем композитног филтера доводи до побољшаних перформанси постојећих приступа за препоруку рецензената?

ИП6.3. Који базични филтери су најбитнији за сваки од постојећих приступа?

ИП6.4. Који базични филтери су најбитнији за сваки од тестираних скупова података (репозиторијума)?

Како би се измериле перформансе сваког метода, као и потенцијална побољшања, дефинишу се следеће метрике:

- **Прецизност@k** (*precision@k*) представља број исправно препоручених рецензената од k препоручених рецензената, подељен са k .

$$\text{прецизност@k} = \frac{\text{број исправно препоручених}}{\text{укупан број препоручених рецензената}}, \quad (6.35)$$

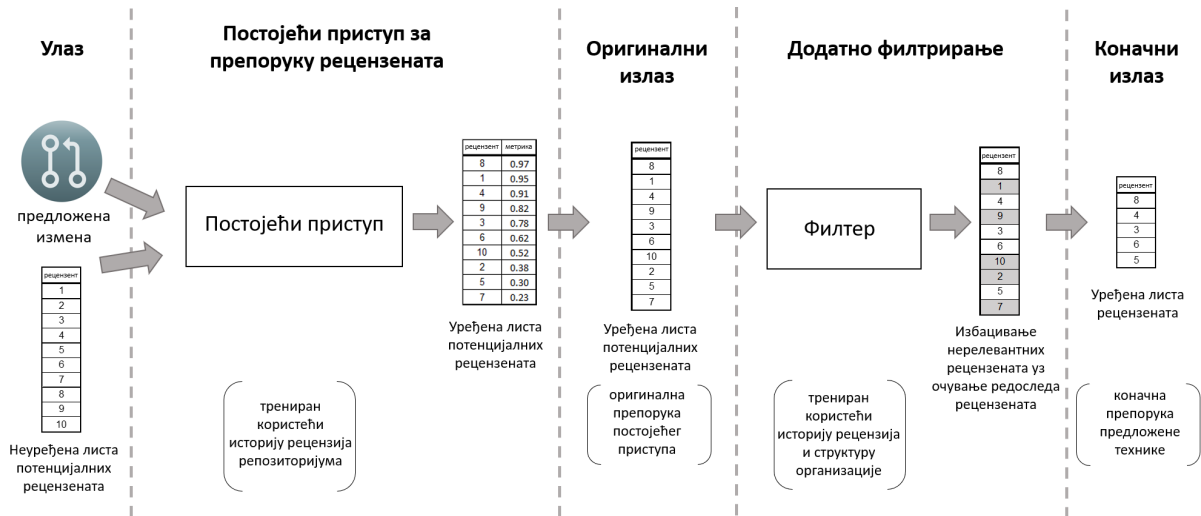
где је k = укупан број препоручених рецензената.

- **Опозив@k** (*recall@k*) представља број исправно препоручених рецензената од k препоручених рецензената, подељен са укупним бројем рецензената присутних на датој рецензији.

$$\text{опозив@k} = \frac{\text{број исправно препоручених}}{\text{укупан број рецензената на рецензији}}. \quad (6.36)$$

За сваки од филтера, дефинисане су следеће нулте хипотезе како би се испитао њихов утицај на сваку од метрика:

- H_{0-1} : Не постоји статистички значајна разлика између прецизности@k постојећег приступа са и вез примењеног филтера.



Слика 6.2: Шематски приказ предложене технике за побољшање перформанси постојећих приступа (© 2023 IEEE) [25]. Постојећи приступ процењује релевантност сваког потенцијалног рецензента и додељује му метрику која одговара његовој процењеној релевантности, и затим врши сортирање потенцијалних рецензената према опадајућим вредностима метрике. Након тога се примењује филтер који избацује све рецензенте који нису релевантни према критеријуму филтера, и тиме се производи коначна препорука система.

- H_{0-2} : Не постоји статистички значајна разлика између опозива@ k постојећег приступа са и вез примењеног филтера.

6.3 Предлог решења

Како би се побољшале перформансе постојећих метода, дизајнирана је техника која редукује скуп потенцијалних рецензената тако да се разматрају само рецензенти који имају неку локалност било са аутором предложених измена, било са фајловима који су измењени, јер су се оба фактора показала као битна при препоруци рецензената [78, 85, 86]. Предложена техника је релативно једноставна и илустрована је на Слици 6.2. Прво се користи постојећи приступ без било каквих модификација, и као резултат се добија листа рецензената уређена према њиховој процењеној релевантности. Затим се користи филтер који, према одређеном критеријуму, избацује све рецензенте које сматра нерелевантним, одржавајући уређеност листе. Коначну препоруку представљају преосталих N најрелевантнијих рецензената који су преостали у филтрираној лист, где је N број рецензената који су тражени од система.

Разматране су 2 категорије филтера: **базични филтери**, који користе само један критеријум по којем филтрирају, и **композитни филтери**, који представљају унију више критеријума

и филтрирају рецензенте који не испуњавају ни један од њих. Предлажу се следећи **базични филтери**:

- **Рецензенти фајлова (РФ)** - овај филтер посматра историју рецензија фајлова који су промењени у предложеној измени, и пропушта само оне рецензенте који су претходно учествовали у макар једној рецензији која је садржала бар један од измењених фајлова.
- **Аутори фајлова (АФ)** - овај филтер посматра историју измена фајлова који су промењени у предложеној измени, и пропушта само оне рецензенте који су претходно мењали макар један од измењених фајлова.
- **Претходни рецензенти (ПР)** - овај филтер посматра историју интеракција аутора предложене измене и свих осталих рецензената, и пропушта само оне рецензенте који су претходно вршили рецензију макар једне предложене измене коју је аутор креирао.
- **Претходни гласачи (ПГ)** - овај филтер посматра историју интеракција аутора предложене измене и свих осталих рецензената, и пропушта само оне рецензенте који су претходно гласали макар једном да предложена измена коју је аутор креирао буде прихваћена.
- **Колеге (КО)** - овај филтер пропушта само рецензенте који имају истог менаџера као и аутор предложене измене. Као извор информација користи организациону структуру (*org chart*) тимова који раде на репозиторијуму.
- **Менаџер (МЕ)** - овај филтер пропушта само менаџера аутора предложене измене. Као извор информација користи организациону структуру (*org chart*) тимова који раде на репозиторијуму.

6.4 Детаљи имплементације

Сваки постојећи приступ описан у 6.1 имплементиран је у програмском језику Python на основу описа из респективних радова. При имплементацији Приступа 7 коришћена је постојећа имплементација *learning-to-rank* [84] метода машинског учења, SVM^{rank} [83]. Сваки од базичних филтера имплементиран је у језику Python, и њиховим комбиновањем добијени су композитни филтери.

Табела 6.4: Преглед различитих скупова података (репозиторијума) који су коришћени у евалуацији.

Скуп података	укупан број:		
	рецензија	рецензената	фајлова
A	15 816	1532	327 640
B	7663	585	40 298
C	1930	264	57 813
D	5011	385	22 775
E	2465	679	7289
F	2000	682	16 012
G	2551	174	8275
H	1318	157	1984

6.5 Евалуација

Скупови података који су коришћени у овој секцији сакупљени су из индустрије и представљају 8 активних репозиторијума са подацима из 2021. године, и детаљи ових скупова података дати су у Табели 6.4. Као што се може видети, изабрани репозиторијуми су различитих величина, од релативно мањих са 100-200 активних инжењера/рецензената, док је највећи репозиторијум имао око 1500 активних рецензената током посматраног периода. Такође, репозиторијуми имају и добру дистрибуцију броја фајлова који су промењени у рецензијама, где је најмањи имао око 2000, док је највећи имао више од 300 000 промењених фајлова.

Како би се извршила валидација, сваки од скупова података је хронолошки уређен и подељен на два дела тако да оба садрже једнак број рецензија. Постојећи приступи као и предложени филтери су тренирани искључиво користећи тренинг („старију”) половину података, док је валидација вршена користећи валидациону („новију”) половину. Након тренирања, за сваку рецензију из валидационог скупа података генерисана је уређена листа рецензената користећи сваки од имплементираних постојећих приступа. Затим, применом различитих филтера, из датих уређених листи су избачени нерелевантни рецензенти (уз очување поретка).

Коначно, прецизност@ k и опозив@ k су мерени тако што је посматрано најрелевантнијих k препорука рецензената из филтриране листе рецензената и вршена је провера да ли су они заиста присутни на датој рецензији. Како би се одредило побољшање које филтери доносе, перформансе приступа са филтером се пореде са перформансама приступа без филтера.

Табела 6.5: Преглед агрегираних резултата експеримената за **ИП6.1.** - прецизност (© 2023 IEEE) [25].

Техника	Приступи						
	1	2	3	4	5	6	7
Менаџер	-7.81	-21.26	-18.44	-5.44	7.49	-46.54	-72.60
Колеге	-3.58	-12.02	-9.24	3.02	12.79	-34.56	-17.27
Претходни гласачи	5.83	-3.47	-1.08	11.12	16.24	-24.01	-8.33
Претходни рецензенти	4.42	1.17	2.89	7.31	10.29	-12.19	-1.10
Аутори фајлова	0.60	-9.56	-7.22	4.93	11.07	-31.19	-14.98
Рецензенти фајлова	7.45	1.34	1.96	11.12	11.03	-12.34	-2.74

Табела 6.6: Преглед агрегираних резултата експеримената за **ИП6.1.** - опозив (© 2023 IEEE) [25].

Техника	Приступи						
	1	2	3	4	5	6	7
Менаџер	-10.55	-21.48	-19.86	-8.29	5.68	-55.44	-33.17
Колеге	-6.91	-12.80	-11.18	-0.31	12.09	-44.43	-23.21
Претходни гласачи	3.24	-0.43	0.91	11.35	19.66	-30.00	-10.43
Претходни рецензенти	4.77	5.86	6.81	11.12	15.75	-14.90	-0.13
Аутори фајлова	-1.90	-8.46	-7.23	3.75	12.44	-38.82	-18.89
Рецензенти фајлова	4.52	3.11	3.34	9.84	14.11	-18.70	-4.87

Како би се извршила провера статистичког значаја побољшања, користи се једнострани Студентов t -тест и одговарајућа нулта хипотеза се одбацује уколико је p -вредност $p < 0.05$ и t -вредност $t < t_{crit}$.

Резултати - ИП6.1.

ИП6.1. *Да ли филтрирање скупа потенцијалних рецензената коришћењем базичног филтера доводи до побољшаних перформанси постојећих приступа за препоруку рецензената?*

Преглед резултата ИП6.1 експеримента приказан је у Табели 6.5 и Табели 6.6, док су пуни резултати приказани у Табели 6.7. Као што се може видети, **Менаџер** није донео користи постојећим приступима, што се и очекивало. Како овај филтер пропушта максимално једног рецензента, очекивано је да филтрирањем само на основу њега долази до деградације

Табела 6.7: Детаљнији резултати **ИП6.1.** експеримента који приказују постигнута побољшања за сваки базични филтер, скуп података и постојећи приступ. Сива поља представљају случајеве када је одговарајућа нулта хипотеза (H_{0-1} за прецизност, H_{0-2} за опозив) прихваћена (што значи да није било статистички значајних побољшања).

Техника	Подаци	Приступ 1		Приступ 2		Приступ 3		Приступ 4		Приступ 5		Приступ 6		Приступ 7	
Менаџер	A	H_{0-1}	-5.04	H_{0-1}	-12.12	H_{0-1}	-8.67	H_{0-1}	3.67	H_{0-1}	8.13	H_{0-1}	-30.07	H_{0-1}	-16.94
		H_{0-2}	-6.50	H_{0-2}	-10.33	H_{0-2}	-8.68	H_{0-2}	1.94	H_{0-2}	6.21	H_{0-2}	-33.25	H_{0-2}	-19.95
	B	H_{0-1}	-5.39	H_{0-1}	-6.71	H_{0-1}	-3.56	H_{0-1}	-0.30	H_{0-1}	6.37	H_{0-1}	-38.77	H_{0-1}	-14.86
		H_{0-2}	-8.29	H_{0-2}	-11.81	H_{0-2}	-7.56	H_{0-2}	-1.54	H_{0-2}	6.19	H_{0-2}	-55.46	H_{0-2}	-23.98
	C	H_{0-1}	-1.22	H_{0-1}	-6.20	H_{0-1}	-3.26	H_{0-1}	2.64	H_{0-1}	7.61	H_{0-1}	-36.51	H_{0-1}	-16.00
		H_{0-2}	-7.75	H_{0-2}	-17.25	H_{0-2}	-13.92	H_{0-2}	-2.26	H_{0-2}	10.62	H_{0-2}	-69.14	H_{0-2}	-38.49
	D	H_{0-1}	-5.09	H_{0-1}	-18.63	H_{0-1}	-15.49	H_{0-1}	-2.24	H_{0-1}	3.96	H_{0-1}	-45.26	H_{0-1}	-28.78
		H_{0-2}	-12.26	H_{0-2}	-29.81	H_{0-2}	-26.82	H_{0-2}	-7.43	H_{0-2}	4.06	H_{0-2}	-67.44	H_{0-2}	-46.51
Колеге	E	H_{0-1}	-7.12	H_{0-1}	-25.93	H_{0-1}	-11.98	H_{0-1}	-5.40	H_{0-1}	10.32	H_{0-1}	-74.36	H_{0-1}	-30.12
		H_{0-2}	-5.85	H_{0-2}	-13.95	H_{0-2}	-7.02	H_{0-2}	-4.46	H_{0-2}	3.68	H_{0-2}	-49.59	H_{0-2}	-16.74
	F	H_{0-1}	-1.04	H_{0-1}	0.59	H_{0-1}	0.06	H_{0-1}	0.75	H_{0-1}	1.55	H_{0-1}	-22.07	H_{0-1}	-5.53
		H_{0-2}	-3.91	H_{0-2}	-6.38	H_{0-2}	-8.92	H_{0-2}	-2.64	H_{0-2}	0.60	H_{0-2}	-60.20	H_{0-2}	-29.21
	G	H_{0-1}	-19.98	H_{0-1}	-56.77	H_{0-1}	-60.69	H_{0-1}	-12.89	H_{0-1}	13.73	H_{0-1}	-74.31	H_{0-1}	-65.28
		H_{0-2}	-15.82	H_{0-2}	-25.52	H_{0-2}	-29.80	H_{0-2}	-15.23	H_{0-2}	4.41	H_{0-2}	-42.13	H_{0-2}	-34.77
	H	H_{0-1}	-17.61	H_{0-1}	-44.28	H_{0-1}	-43.93	H_{0-1}	-29.73	H_{0-1}	8.25	H_{0-1}	-50.99	H_{0-1}	-43.33
		H_{0-2}	-23.97	H_{0-2}	-56.82	H_{0-2}	-56.18	H_{0-2}	-34.73	H_{0-2}	9.66	H_{0-2}	-66.31	H_{0-2}	-55.71
Претходни гласачи	A	H_{0-1}	-2.28	H_{0-1}	-6.94	H_{0-1}	-3.70	H_{0-1}	8.26	H_{0-1}	10.41	H_{0-1}	-20.53	H_{0-1}	-9.35
		H_{0-2}	-3.76	H_{0-2}	-3.20	H_{0-2}	-1.65	H_{0-2}	8.87	H_{0-2}	11.66	H_{0-2}	-22.60	H_{0-2}	-10.55
	B	H_{0-1}	0.32	H_{0-1}	4.66	H_{0-1}	7.71	H_{0-1}	10.57	H_{0-1}	11.33	H_{0-1}	-23.36	H_{0-1}	-2.13
		H_{0-2}	-2.14	H_{0-2}	3.79	H_{0-2}	8.03	H_{0-2}	13.08	H_{0-2}	17.50	H_{0-2}	-37.28	H_{0-2}	-7.42
	C	H_{0-1}	-0.29	H_{0-1}	-3.67	H_{0-1}	-0.77	H_{0-1}	4.69	H_{0-1}	7.04	H_{0-1}	-32.24	H_{0-1}	-12.85
		H_{0-2}	-5.99	H_{0-2}	-13.84	H_{0-2}	-10.49	H_{0-2}	0.98	H_{0-2}	12.02	H_{0-2}	-64.27	H_{0-2}	-34.18
	D	H_{0-1}	4.56	H_{0-1}	-6.16	H_{0-1}	-3.06	H_{0-1}	9.85	H_{0-1}	9.75	H_{0-1}	-29.53	H_{0-1}	-15.91
		H_{0-2}	2.32	H_{0-2}	-10.80	H_{0-2}	-7.79	H_{0-2}	11.71	H_{0-2}	17.88	H_{0-2}	-45.46	H_{0-2}	-26.61
Претходни рецензенти	E	H_{0-1}	0.23	H_{0-1}	-12.26	H_{0-1}	1.69	H_{0-1}	8.07	H_{0-1}	18.30	H_{0-1}	-58.28	H_{0-1}	-15.60
		H_{0-2}	-3.07	H_{0-2}	-7.42	H_{0-2}	-0.49	H_{0-2}	1.98	H_{0-2}	9.46	H_{0-2}	-41.94	H_{0-2}	-9.55
	F	H_{0-1}	-1.07	H_{0-1}	3.16	H_{0-1}	2.71	H_{0-1}	3.27	H_{0-1}	3.20	H_{0-1}	-16.16	H_{0-1}	-1.74
		H_{0-2}	-3.86	H_{0-2}	0.97	H_{0-2}	-1.48	H_{0-2}	3.80	H_{0-2}	6.89	H_{0-2}	-46.87	H_{0-2}	-18.04
	G	H_{0-1}	-8.66	H_{0-1}	-32.78	H_{0-1}	-36.67	H_{0-1}	11.37	H_{0-1}	34.06	H_{0-1}	-47.14	H_{0-1}	-38.66
		H_{0-2}	-9.30	H_{0-2}	-14.74	H_{0-2}	-19.00	H_{0-2}	-4.81	H_{0-2}	13.49	H_{0-2}	-30.21	H_{0-2}	-23.00
	H	H_{0-1}	-21.47	H_{0-1}	-42.21	H_{0-1}	-41.87	H_{0-1}	-31.90	H_{0-1}	8.20	H_{0-1}	-49.25	H_{0-1}	-41.92
		H_{0-2}	-29.50	H_{0-2}	-57.19	H_{0-2}	-56.56	H_{0-2}	-38.13	H_{0-2}	7.80	H_{0-2}	-66.80	H_{0-2}	-56.32
Аутори фајлова	A	H_{0-1}	5.15	H_{0-1}	1.29	H_{0-1}	3.92	H_{0-1}	14.33	H_{0-1}	16.63	H_{0-1}	-10.63	H_{0-1}	0.76
		H_{0-2}	3.98	H_{0-2}	6.86	H_{0-2}	8.03	H_{0-2}	17.24	H_{0-2}	20.21	H_{0-2}	-10.83	H_{0-2}	1.20
	B	H_{0-1}	6.74	H_{0-1}	9.53	H_{0-1}	12.12	H_{0-1}	13.81	H_{0-1}	17.76	H_{0-1}	-13.86	H_{0-1}	4.48
		H_{0-2}	7.33	H_{0-2}	13.31	H_{0-2}	17.24	H_{0-2}	21.32	H_{0-2}	26.18	H_{0-2}	-22.55	H_{0-2}	3.79
	C	H_{0-1}	6.28	H_{0-1}	4.40	H_{0-1}	6.79	H_{0-1}	13.03	H_{0-1}	12.06	H_{0-1}	-17.59	H_{0-1}	-3.44
		H_{0-2}	7.27	H_{0-2}	9.50	H_{0-2}	12.36	H_{0-2}	23.52	H_{0-2}	28.63	H_{0-2}	-34.28	H_{0-2}	-9.15
	D	H_{0-1}	11.09	H_{0-1}	1.08	H_{0-1}	3.39	H_{0-1}	13.09	H_{0-1}	13.82	H_{0-1}	-14.21	H_{0-1}	-4.54
		H_{0-2}	14.38	H_{0-2}	1.96	H_{0-2}	4.12	H_{0-2}	19.56	H_{0-2}	23.71	H_{0-2}	-23.32	H_{0-2}	-9.06
Рецензенти фајлова	E	H_{0-1}	12.17	H_{0-1}	-2.74	H_{0-1}	10.42	H_{0-1}	16.99	H_{0-1}	22.00	H_{0-1}	-47.67	H_{0-1}	-5.95
		H_{0-2}	2.28	H_{0-2}	-1.82	H_{0-2}	4.77	H_{0-2}	7.35	H_{0-2}	11.82	H_{0-2}	-36.18	H_{0-2}	-4.44
	F	H_{0-1}	-0.69	H_{0-1}	2.18	H_{0-1}	1.69	H_{0-1}	3.02	H_{0-1}	1.52	H_{0-1}	-20.64	H_{0-1}	-4.28
		H_{0-2}	-2.58	H_{0-2}	3.82	H_{0-2}	1.34	H_{0-2}	6.96	H_{0-2}	6.05	H_{0-2}	-50.14	H_{0-2}	-19.35
	G	H_{0-1}	11.39	H_{0-1}	-17.51	H_{0-1}	-21.26	H_{0-1}	26.94	H_{0-1}	29.75	H_{0-1}	-33.61	H_{0-1}	-26.38
		H_{0-2}	4.00	H_{0-2}	-1.26	H_{0-2}	-5.44	H_{0-2}	9.54	H_{0-2}	18.19	H_{0-2}	-16.36	H_{0-2}	-10.02
	H	H_{0-1}	-5.48	H_{0-1}	-26.09	H_{0-1}	-25.70	H_{0-1}	-12.26	H_{0-1}	16.36	H_{0-1}	-33.87	H_{0-1}	-27.25
		H_{0-2}	-10.72	H_{0-2}	-35.80	H_{0-2}	-35.11	H_{0-2}	-14.69	H_{0-2}	22.50	H_{0-2}	-46.35	H_{0-2}	-36.43

перформанси када се посматрају првих 10 препорука. Даље, **Колеге** су успеле да у просеку донесу побољшања за 2 постојећа приступа када се посматра прецизност, и само за 1 када се посматра опозив. **Претходни гласачи** су побољшали прецизност 3 и опозив 4 постојећа приступа, и довели до највећих побољшања прецизности од свих базичних филтера у случају Приступа 5 и највећих побољшања опозива у случају Приступа 4 и Приступа 5.

Затим, **Претходни рецензенти** су довели до побољшања прецизности и опозива 5 постојећих приступа, и до највећих побољшања прецизности у случају Приступа 3 (и најмањег смањења у случајевима Приступа 6 и Приступа 7), односно до највећих побољшања опозива у случају Приступа 1, Приступа 2, и Приступа 3 (и најмањег смањења у случајевима Приступа 6 и Приступа 7). **Аутори фајлова** довели су до побољшања прецизности на 3 постојећа приступа, и до побољшања опозива на 2. Коначно, **Рецензенти фајлова** довели су до побољшања перформанси и опозива на 5 постојећих приступа, и до највећег побољшања прецизности у случају Приступа 1, Приступа 2, и Приступа 4.

Приступ 6 и Приступ 7 у просеку нису остварили побољшања ни прецизности ни опозива. Са друге стране, **Претходни рецензенти** су довели до максималних побољшања у највећем броју случајева, а слична побољшања показали су и **Рецензенти фајлова**, што говори о томе да су они најбитнији базични филтери.

Резултати - ИП6.2.

***ИП6.2.** Да ли филтрирање скупа потенцијалних рецензената коришћењем композитног филтера доводи до побољшаних перформанси постојећих приступа за препоруку рецензената?*

Експериментална мерења за **ИП6.2.** вршена су тако што су генерисане све могуће комбинације базичних филтера, и онда су примењене над препорукама свих постојећих приступа. Приказ резултата свих комбинација дат је у Табели 6.8, док је преглед најбољих резултата дат у Табели 6.9 и Табели 6.10.

Као што се може видети, комбиновањем базичних филтера у композитне филтере могуће је постићи додатна побољшања. У ИП6.1 су само 5 од 7 приступа искусили побољшања услед примене базичних филтера (сви осим Приступа 6 и Приступа 7), док је у случају композитних филтера то био случај са 6 приступа (једино Приступ 6 није имао користи). Такође, повећао се и ниво побољшања, где је применом композитних филтера могуће постићи

Табела 6.9: Преглед најбољих резултата композитних филтера у експерименту за **ИП6.2.** - прецизност (© 2023 IEEE) [25].

Приступ	Техника						Резултати		
	РФ	АФ	ПР	ПГ	КО	МЕ	Побољшања на:	прец.	опоз.
Приступ 1	1	0	0	1	1	1	A,B,C,D,E,G	9.49	7.32
Приступ 2	1	1	0	1	1	1	A,B,C,D,E,F,G	5.34	8.87
Приступ 3	0	1	1	b	1	1	A,B,C,D,E,F	6.34	9.98
Приступ 4	0	1	0	1	1	1	A,B,C,D,E,F,G,H	15.39	16.57
Приступ 5	0	0	0	1	0	1	A,B,C,D,E,F,G	17.60	21.17
Приступ 6	1	1	1	b	1	1		-2.57	-3.83
Приступ 7	0	1	1	b	1	1	A,B,C,D,E,F	3.05	4.48

у случају неких постојећих приступа чак 17.60% бољу прецизност и 21.23% бољи опозив, док је применом базичних филтера максимум био 16.24% и 19.66%, респективно.

Када се погледа који базични филтери су присутни у најбољим композитним филтерима, можемо видети да су **Менаџер** и **Претходни гласачи** увек присутни, што говори о њиховом значају. У случају филтера **Менаџер** је велика разлика у односу на ИП6.1, где је овај филтер имао јако слабе перформансе услед своје ограничености пропуштањем максимално једног рецензента. Сада видимо да је тај један рецензент јако релевантан, и да је корисно разматрати га.

Интересантно је приметити да постоје више композитних филтера који имају најбоље перформансе. Конкретно, најбољи композитни филтери за Приступ 2, Приступ 3, Приступ 6 и Приступ 7 могу али не морају да садрже филтер **Претходни гласачи**, што је у табели означено симболом *b*. Ово је очекивано, јер у свим тим ситуацијама је активан филтер **Претходни рецензенти**, који увек представља надскуп филтера **Претходни гласачи**. Зато када год су **Претходни рецензенти** активни, додавање **Претходних гласача** неће донети додатне користи. Али, већи број рецензената не даје увек боље резултате, што се може видети у случају најбољих композитних филтера Приступа 1, Приступа 4 и Приступа 5. Код њих су активни само **Претходни гласачи** без **Претходних рецензената**, што је још један показатељ да разматрање превише рецензената може довести до превеликог шума.

Резултати - ИП6.3.

***ИП6.3.** Који базични филтери су најбитнији за сваки од постојећих приступа?*

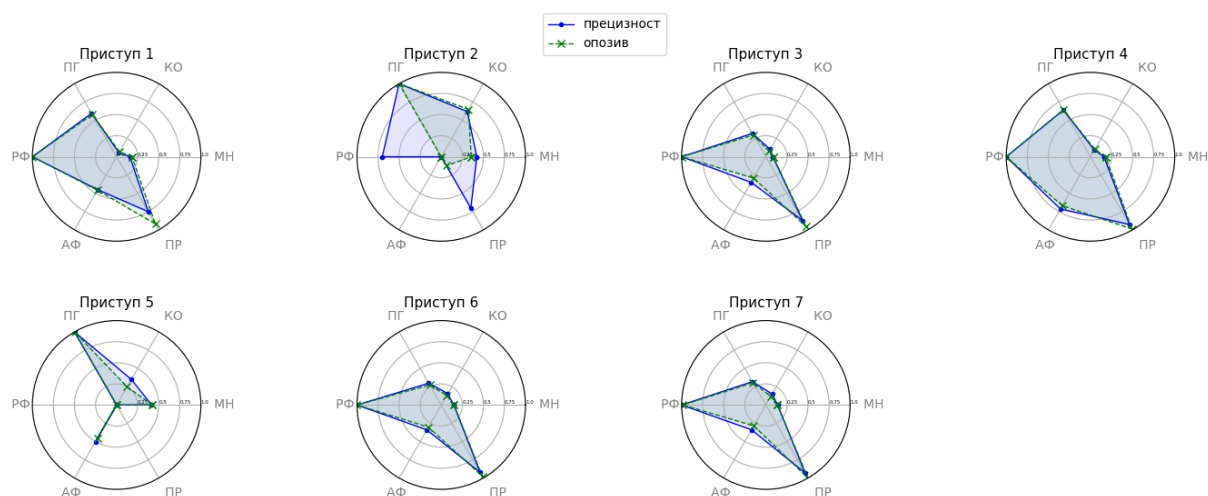
Табела 6.10: Преглед најбољих резултата композитних филтера у експерименту за **ИП6.2.** - опозив (© 2023 IEEE) [25].

Приступ	Техника						Резултати		
	РФ	АФ	ПР	ПГ	КО	МЕ	Побољшања на:	прец.	опоз.
Приступ 1	1	0	0	1	1	1	A,B,C,D,E,G	9.49	7.32
Приступ 2	0	1	1	b	1	1	A,B,C,D,E,F,G	4.84	9.21
Приступ 3	0	1	1	b	1	1	A,B,C,D,E,F	6.34	9.98
Приступ 4	0	1	0	1	1	1	A,B,C,D,E,F,G,H	15.39	16.57
Приступ 5	0	0	0	1	1	1	A,B,C,D,E,F,G	17.10	21.23
Приступ 6	1	1	1	b	1	1		-2.57	-3.83
Приступ 7	0	1	1	b	1	1	A,B,C,D,E,F	3.05	4.48

Како би се одредио значај неког базичног филтера при креирању композитних филтера, посматрани су парови композитних филтера који садрже тај базични филтер и композитних филтера који се добију када се из датих комбинација избаци посматрани базични филтер. Просечна разлика између ових парова означена је као допринос датог базичног филтера. Након што су израчунати доприноси свих базичних филтера за сваки постојећи приступ, вредности су нормализоване и затим су исцртани графици приказани на Слици 6.3.

Као што се може видети, **Рецензенти фајлова** и **Претходни рецензенти** су једни од битнијих филтера за све приступе осим за Приступ 5, и представљају најбитније филтере за Приступ 1, Приступ 3, Приступ 4, Приступ 6 и Приступ 7. Додавањем **Менаџера** у композитни филтер долази до мањих побољшања, али она ни за један од постојећих приступа нису била значајнија, што се и могло очекивати, јер додавањем овог филтра се пропушта максимално још један потенцијални рецензент, што неће превише утицати на резултат.

Интересантан је случај са **Претходним гласачима**, где они имају утицај на све постојеће приступе, али је тај утицај најизраженији код Приступа 2 и Приступа 5. Ово се може објаснити чињеницом да ови приступи не разматрају акције рецензената, па се може очекивати да филтер који припушта само рецензенте који су гласали и прихватили претходне измене датог аутора (односно, који избацује неактивне рецензенте) доноси велику корист у овим случајевима. У овом погледу чудно је понашање Приступа 3, који иако не узима у обзир активност рецензената није имао значајнију корист од филтера **Претходни гласачи**.



Слика 6.3: Нормализовани утицај сваког базичног филтера при комбиновању са другим филтерима, за сваки постојећи приступ (© 2023 IEEE) [25].

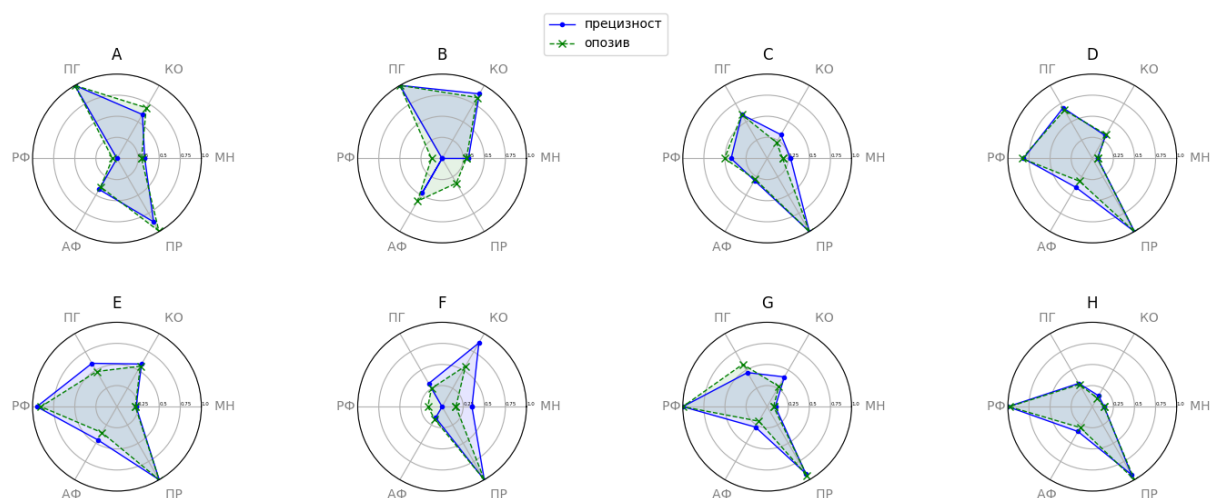
Резултати - ИП6.4.

ИП6.4. Који базични филтери су најбитнији за сваки од тестираних скупова података (репозиторијума)?

Као што је претходно описано у случају **ИП6.3.**, сличан поступак одређивања утицаја базичних филтера на композитне филтере поновљен је и у случају **ИП6.4.**, али са агрегирањем резултата по скуповима података. Тако добијени резултати графички су приказани на Слици 6.4.

Претходни рецензенти су најбитнији филтер на свим скуповима података осим на *B*. Оно шта се разликује у односу на **ИП6.3.** је да **Рецензенти фајлова** нису толико доминанти, и представљају битан филтер само на скуповима података *D*, *E*, *G* и *H*. То може да указује да ови скупови података имају јасно дефинисане компоненте и власнике фајлова, и када год се мења одређени фајл, постоји јасно дефинисана особа или део тима чији је посао да изврше рецензију и одобре те измене.

Даље, делује да су **Претходни гласачи** највише побољшања донели већим репозиторијумима (скуповима података), односно *A* и *B*, и донекле *C*, *D* и *E*. **Менаџер** поново није показао превелики утицај на композитне филтере, али су се **Колеге** показале као релативно битан фактор у више скупова података (*A*, *B* и *F*), што може да буде показатељ тога да тимови који раде на тим репозиторијумима врше расподелу послова



Слика 6.4: Нормализовани утицај сваког базичног филтера при комбиновању са другим филтерима, за сваки скуп података (© 2023 IEEE) [25].

Табела 6.11: Максимално побољшање прецизности применом било којег филтера за парове постојећих приступа и скупова података (© 2023 IEEE) [25].

	A	B	C	D	E	F	G	H
Приступ 1	7.73	8.25	9.06	12.71	20.79	0.48	23.35	2.33
Приступ 2	5.53	13.31	8.92	3.87	16.04	9.12	6.45	-4.59
Приступ 3	7.98	15.92	10.89	5.87	26.37	8.77	3.72	-4.10
Приступ 4	17.74	16.57	16.10	15.10	28.59	7.35	43.28	-1.71
Приступ 5	18.27	18.03	13.62	13.82	24.13	8.19	42.48	18.70
Приступ 6	-0.31	-0.99	-3.56	-1.34	-3.90	-2.11	-2.55	-5.44
Приступ 7	6.54	9.06	3.08	1.05	13.00	6.83	0.01	-3.74

према хијерархији, тако да инжењери који имају истог менаџера раде на сличним пословима и у току су са радом својих директних колега.

6.6 Дискусија

Коришћењем базичних филтера постигнута су побољшања 5 од 7 постојећих приступа, док је коришћење композитних филтера донело додатна побољшања на свим тестираним приступима осим на Приступу 6. Али, то су све били резултати који су агрегирани по приступима и/или скуповима података. Уколико се посматрају максималне перформансе применом било којег базичног или композитног за сваки од парова постојећих приступа и

Табела 6.12: Максимално побољшање опозива применом било којег филтера за парове постојећих приступа и скупова података (© 2023 IEEE) [25].

	A	B	C	D	E	F	G	H
Приступ 1	7.19	10.03	14.15	17.48	8.58	0.75	7.52	0.09
Приступ 2	11.45	19.26	19.11	7.25	12.50	27.19	8.20	-5.29
Приступ 3	12.54	23.25	21.48	8.66	16.95	25.19	5.02	-4.43
Приступ 4	21.33	26.33	30.11	23.47	15.63	21.71	15.70	-1.18
Приступ 5	21.71	26.63	30.94	24.13	15.46	27.21	19.74	25.30
Приступ 6	0.51	-2.43	-7.57	-2.62	-3.56	-5.16	-1.23	-6.92
Приступ 7	8.51	10.63	5.10	1.50	10.33	13.56	1.00	-4.28

репозиторијума (Табела 6.11 и Табела 6.12), могуће је уочити додатна побољшања. Уколико се занемаре скуп података *H* и Приступ 6, све остале комбинације приступа и скупа података су имале побољшања услед примене филтера. Чак и за скуп података *H*, неки приступи као што су Приступ 1 и Приступ 5 су имали побољшања. Приступ 6 остаје једини који није имао озбиљнијих побољшања, али је чак и у том случају успео да има нешто бољи опозив на скупу података *A*, што даје наду да би и овај приступ могао да има додатна побољшања на другим скуповима података уколико би се увео неки нови филтер, или уз неку додатну модификацију предложене технике филтрирања. Из ових табела се може видети да је максимално побољшање прецизности 43.28%, и да је максимално побољшање опозива 30.94%.

Претње валидности

Стање хијерархије је динамична ствар која се може мењати кроз време, док је за потребе овог истраживања коришћен узорак хијерархије из јуна 2022. године. Иако су сви скупови података садржали искључиво податке из календарске 2021. године, историја хијерархије није била доступна, и услед тога је могло доћи до незнатних разлика у односу на стварну хијерархију у тренутку креирања предлога измена и одређивања релевантности рецензената, што би могло да утиче на резултате у случају филтера **Менаџер и Колеге**.

Такође, иако је уложен труд да скупови података садрже само инжењере, и да се очисте од било каквих „бот” налога (који припадају различитим системима за аутоматизацију), као и било каквих дистрибуционих листи, не постоји гаранција да су коришћени скупови података били потпуно очишћени, што би могло у некој мери да утиче на резултате. На

пример, уколико је у неком скупу података остао „бот” налог који се аутоматски додаје на сваку рецензију, онда би такав налог био сигуран избор на свакој рецензији и постојећи приступи би вероватно конвергирали ка њему као првој препоруци. Али, сам налог не би био превише релевантан када би аутор измене предлагао рецензенте. Са друге стране, вероватно би га поједини филтери (нпр. **Колеге**, **Аутори фајлова**, итд.) избацили из разматрања јер тај налог не би имао уобичајене везе са осталим инжењерима као што би се очекивало од стварне особе, и самим тим би се сматрао нерелевантним.

Коначно, сви постојећи приступи који су разматрани у овој секцији су у потпуности имплементирани од стране аутора. Иако је при имплементацији праћен опис постојећих приступа из њихових респективних радова, уколико су постојали неки ситнији детаљи имплементације који нису поменути у радовима (а који би могли да утичу на перформансе приступа), или у случају грешака у тумачењу или реимплементацији, имплементације приступа које су коришћене у овој секцији се могу разликовати у односу на оне које су користили аутори њихових респективних радова.

Перспективе за будућа истраживања

Истраживање у овом поглављу вршено је *offline*, односно над историјским подацима. Како би се испитао утицај предложеног решења у реалним условима, пожељно је примени предложено решење над „живим” системима за препоруку рецензената. Такође, потребно је тестирати перформансе система у случају дужих временских периода. На пример, уколико је један рецензент био релевантан пре 10 година, не мора да значи да је и даље релевантан. Потребно је испитати колики временски период прошлости предложено решење треба да посматра, затим када и како треба ажурирати метрике релевантности, итд.

Било би пожељно тестирати постојеће методе и предложено решење над подацима из других репозиторијума из индустрије, као и над open-source репозиторијумима. Треба утврдити да ли предложене технике помажу или штете у случају малих репозиторијума. Такође, од када је ово истраживање кренуло и од када су узорковани постојећи приступи, појавили су се нови приступи за препоруку рецензената. Зарад потпуније слике о користима предложених техника ваљало би поновити мерења са новијим приступима.

Коначно, данас су све популарније платформе које нуде окружење за једноставно тестирање и поређење различитих метода - *Papers with code* [87] и *WebArena* [88] само су неке од платформи нуде могућност тестирања метода из различитих области над стандардизованим

скуповима података и рангирање њихових перформанси. Било би корисно направити (или проширити постојећу) платформу која нуди сличну услугу за област препоруке рецензената софтверског кода, како би се поређења и унапређења која су вршена у овом раду могла једноставније обавити.

7 Закључак

Ова теза бавила се облашћу развоја софтверског кода и конкретније рецензијама софтверског кода. На почетку је дат кратак преглед историје процеса развоја и рецензије (инспекције) софтверског кода, почев од првих формалних дефиниција 70-их година прошлог века, преко појаве првих алата и транзиције на дистрибуирани и асинхрони вид рецензије кода, све до модерног процеса рецензије. Представљен је и преглед неколико модерних платформи за колаборативни развој и рецензију кода, како из уобичајеног корисничког угла, тако и из угла истраживача којем је потребно да на једноставан начин директно или индиректно дохвати податке о великом броју рецензија.

Затим је у трећем поглављу представљено истраживање које је испитивало утицај COVID-19 пандемије на open source развој на платформи GitHub. Дат је кратак преглед најбитнијих датума везаних за пандемију, као и претходних истраживања која су се бавила овом темом, а која су углавном била фокусирана на првих годину дана пандемије. Зато се истраживање из трећег поглавља фокусирао на испитивање дугорочнијег утицаја пандемије на GitHub. Истраживање је користило GH Archive и BigQuery као једноставан и брз начин приступа и обради структурираних података. Анализом броја различитих догађаја и њихових трендова пре почетка и током пандемије, закључено је да је већина догађаја наставила са претходним трендовима, али да су догађаји који се везују за активност open source заједнице и рада на документацији били у паду, док су догађаји који се везују за развој кода и комуникацију о коду имали нешто већи пораст.

Четврто поглавље бавило се екстракцијом информација из претходних предлога измена. Претрагом постојећих система за колаборативни развој утврђено је да није једноставно утврдити експертизу произвољног инжењера у репозиторијуму, или да макар захтева време

за претрагу и анализу претходних измена. Зато је уведена претпоставка да би корисници имали користи од система који би омогућио једноставнију и бржу претрагу инжењера, и истраживање се фокусирало на утврђивање валидности донете претпоставке. Осмишљен је систем који би користећи TF-IDF извадио кључне речи и њихове тежинске факторе из описа претходних измена и уз помоћ њих омогућио претраживање инжењера. Како би се верификовала претпоставка, имплементирана је једноставна верзија осмишљеног система заснована на веб технологијама и статичном скупу података, и имплементација је учињена доступним инжењерима једног тима из индустрије. Анализом телеметрије о коришћењу имплементираних система утврђено је да у просеку решење није било превише често коришћено, али да је постојало неколико корисника који су били активнији и за које се сматра да су имали користи од имплементираних решења.

У петом поглављу описан је проблем неактивних рецензената и једно постојеће решење овог проблема - систем Nudge. Иако је овај систем био релативно успешан, било је корисника који су сматрали да нотификација које су добијали нису корисне. Користећи сервис Viva Insights (који између осталог подсећа кориснике на мејлове у којима се очекује њихов одговор) као инспирацију, претпостављено је да би корисницима више одговарао обједињени периодични преглед њихових рецензија, уместо о свакој рецензији посебно. Поглавље се фокусирало прво на валидацију идеје анкетирањем потенцијалних корисника, на основу чега је закључено да постоје корисници који би желели овакав вид нотификација, као и неколико релевантних параметара за конфигурацију таквог система. Систем је затим имплементиран и био активан неколико месеци, и током његовог рада послато је око 1800 мејлова подсетника на адресе 46 корисника који су се пријавили да користе систем. Пред гашење система покренута је анкета која је испитала сентимент пријављених корисника о систему, и већина је желела да настави да прима подсетнике ($\approx 89\%$), и сви испитаници су сматрали да је систем имао или позитиван ($\approx 58\%$) или неутралан ($\approx 42\%$) утицај на њихову продуктивност.

Шесто поглавље бавило се системима за препоруку рецензената и представљено је неколико постојећих приступа за проблем препоруке рецензената софтверског кода. Сваки приступ је укратко описан, а затим је представљена техника којом се могу побољшати њихове перформансе. Користећи неколико експеримената који су спроведени над историјским подацима 8 репозиторијума из индустрије, поређене су перформансе постојећих приступа са и без примене технике, и закључено је да је могуће постићи преко 40% бољу прецизност и преко 40% бољи опозив на појединим репозиторијума. Најбоље перформансе постигнуте су применом композитних филтера, и то углавном оних који садрже **Претходне гласаче**

и **Претходне рецензенте**, и (у зависности од приступа и репозиторијума) **Рецензенти фајлова** и **Колеге**. Већина комбинација скупова података и приступа је имала побољшане перформансе применом предложене технике, и само најмањи репозиторијум и један од седам тестираних приступа нису имали значајнијих користи.

Ова теза је представила неколико истраживања из различитих области оптимизације процеса рецензије софтверског кода и екстракције корисних информација из претходних предлога измена. Описана истраживања могу се продубити применом предложених решења над већим скуповима података (како из индустрије тако и из open source заједнице), како би се добио бољи преглед резултата предложених решења у ширем контексту. Такође, описани приступи и области нису једини, већ су само они на које се аутор тезе фокусирао током свог истраживања.

Током писања ове тезе десио се значајан скок у способностима вештачке интелигенције, конкретно LLM (*Large Language Model*) и *foundational* модела [89, 90]. Услед тога, отварају се нове могућности за аутоматизацију које раније или нису биле могуће или су биле непрецизне - нпр. аутоматско генерисање описа предлога измена на основу садржаја измена [91], аутоматске рецензије кода [92] (и аутоматско генерисање кода), итд. Зато делује да процес развоја и рецензије софтверског кода посебно сада садржи прилике за даље напретке не само у областима које су поменуте у овом раду, већ и за истраживања у новим областима и очекује се да ће у наредним годинама бити доста значајних и интересантних резултата.

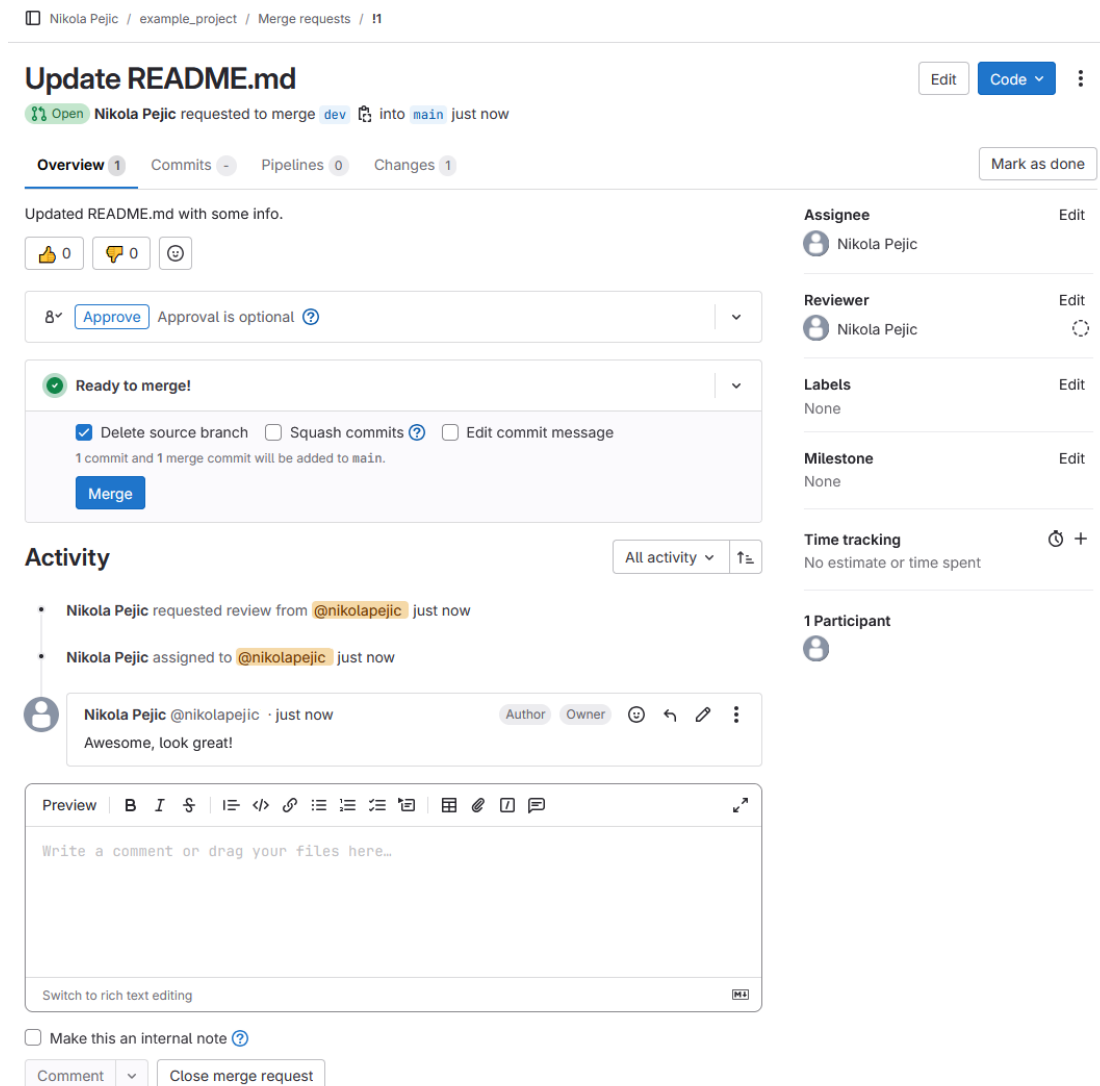
А Системи за колаборативни развој и рецензију софтвера (додатни примери)

А.1 GitLab

GitLab је још један сервис за колаборативни развој софтвера [93], али за разлику од сервиса Azure DevOps и GitHub, целокупан код платформе GitLab је open source и доступан под MIT лиценцом [94]. Осим *git* система за верзионисање кода, GitLab нуди сервисе и за организацију и праћење послова (tasks [95], issues [96], milestones [97], iterations [98], time tracking [99]), системе за аутоматско компајлирање, тестирање и објављивање софтвера (pipelines [100], package registry [101], releases [102]). Слично сервису GitHub, и GitLab нуди могућност хостовања статичких веб страница на основу кода из репозиторијума (GitLab Pages [103]).

На Слици А.1 приказан је пример предлога измена, где се може видети да је наслов „*Update README.md*”, да је аутор ове тезе уједно и аутор измене („*Nikola Pejic*”), да је опис измене „*Update README.md with some info*”, и може се видети да је предлог измене још увек активан („*Open*”). Рецензија је додељена аутору тезе („*Assignee*” је „*Nikola Pejic*”), који је такође и рецензент ове измене („*Reviewer*”). Рецензија садржи један коментар који гласи „*Awesome, look great!*” од стране аутора тезе.

Пример прегледа појединачних измена (*commits*) које дати предлог измена садржи приказан је на Слици А.2. Може се видети да је направљена само једна измена са називом „*Update README.md*”, и она се десила 12. октобра 2024. године. На Слици А.3 приказан је садржај измена по сваком фајлу. Може се видети да је измењен само један фајл („*README.md*”), и то тако што је додата једна празна линија (линија 2) и једна линија која садржи текст „*Adding some info to the readme.*” (линија 3).



Слика А.1: Пример предлога измена на сервису GitLab - преглед.

GitLab нуди REST интерфејс [104] зарад лакшег приступа подацима о предлозима измена и рецензијама. Слањем захтева из Листинга А.1 могу се излистати сви предлози измена датог пројекта (назив који сервис GitLab користи уместо репозиторијума).

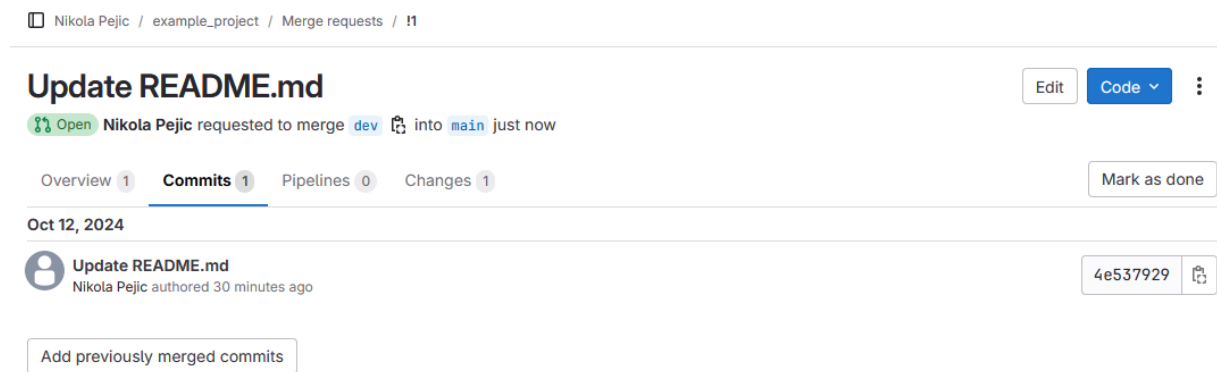
```
1 GET https://gitlab.example.com/api/v4/projects/<PROJECT>/merge_requests
```

Листинг А.1: Пример захтева за дохватање предлога измена на сервису GitLab.

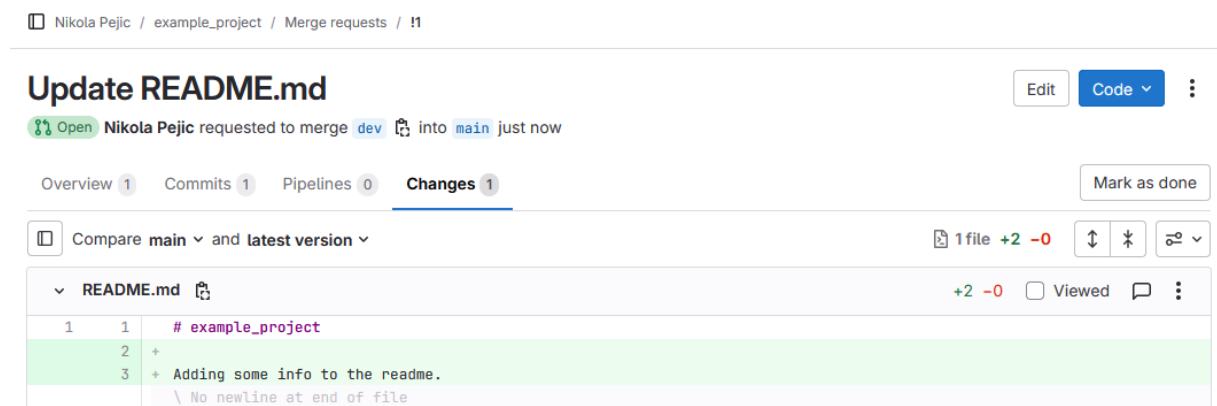
Пример одговора претходног захтева дат је у Листингу А.2 (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [105]).

[illegible]

Листинг А.2: Пример одговора на захтев за дохватање података о предлозима измена сервиса GitLab (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [105]).



Слика А.2: Пример предлога измена на сервису GitLab - измене.

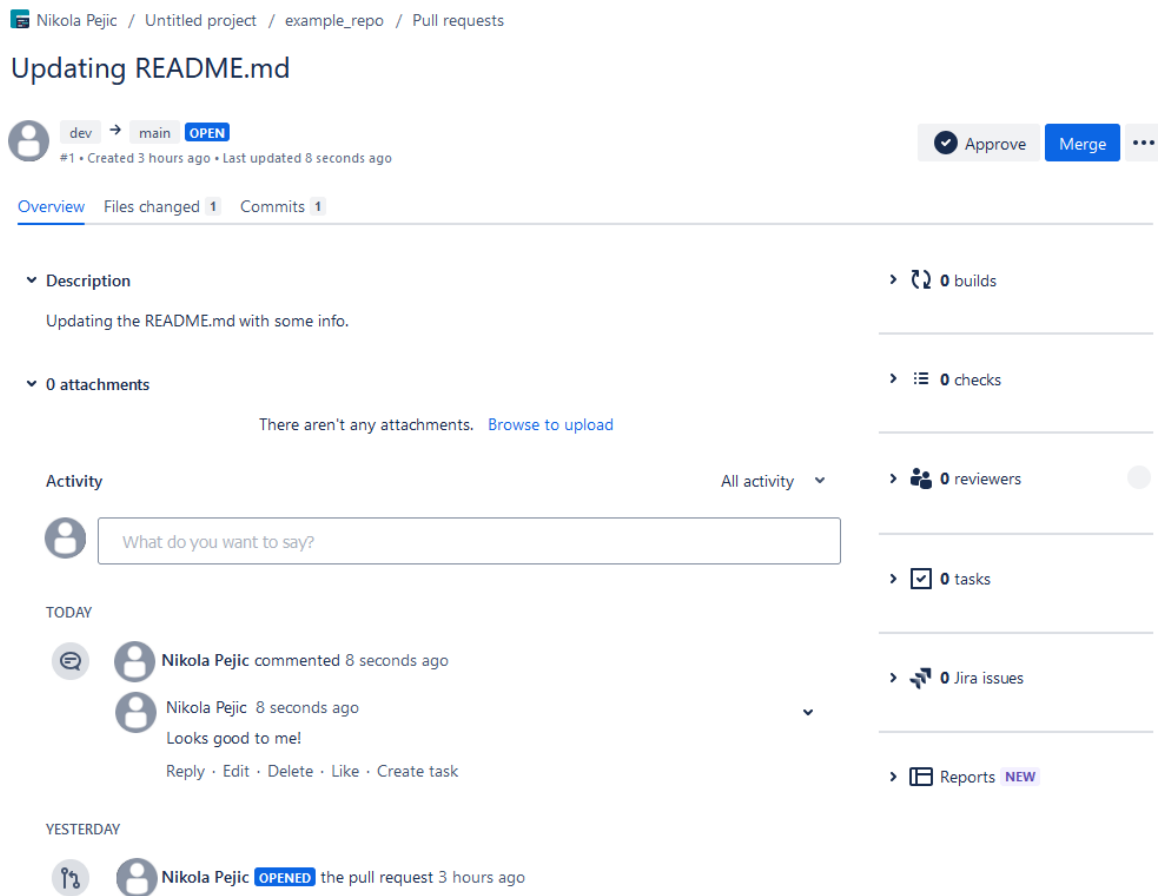


Слика А.3: Пример предлога измена на сервису GitLab - фајлови.

A.2 BitBucket

Bitbucket је сервис за колаборативни развој софтвера компаније Atlassian [106]. Као и други модерни сервиси, Bitbucket осим *git* система за верзионисање кода нуди и функционалности за организацију и праћење послова (issues [107], или кроз сервис Jira [108]), као и системе за аутоматско компајлирање, тестирање и објављивање софтвера (pipelines [109], deployments [110]). Слично сервисима GitHub и GitLab, и Bitbucket нуди могућност хостовања статичких веб страница на основу кода из репозиторијума [111].

Пример предлога измена на сервису Bitbucket може се видети на Слици А.4, где се може видети да је наслов „*Update README.md*”, да је аутор измене уједно и аутор ове тезе, да је предлог измене креиран 3 сата од тренутка када је направљена ова слика. Затим, може се видети да измена има само један коментар који гласи „*Looks good to me!*”, и да је коментар оставио аутор тезе пре 8 секунди.



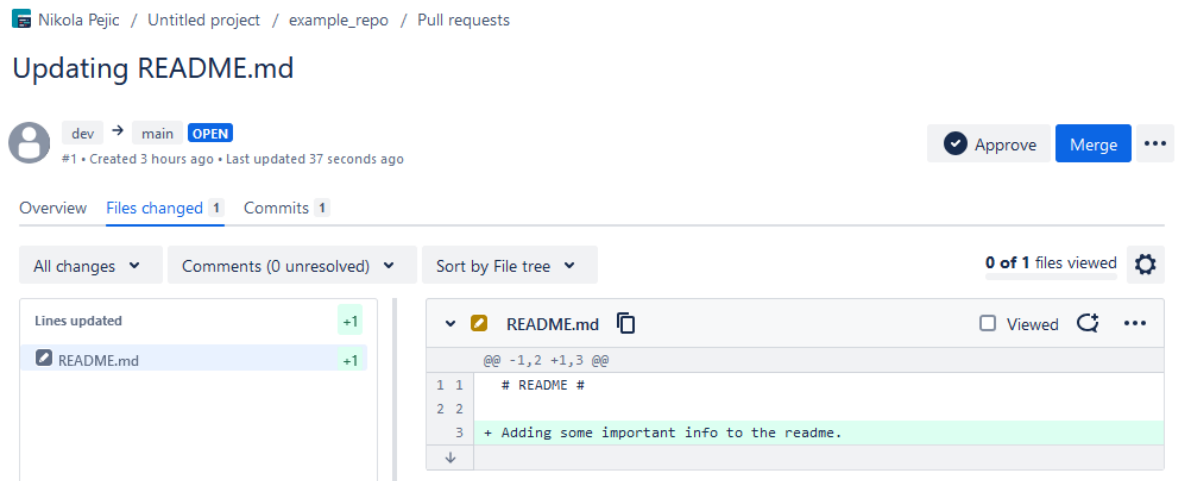
Слика А.4: Пример предлога измена на сервису Bitbucket - преглед.

На Слици А.5 дат је преглед фајлова који су измењени у датом предлогу измена, и може се видети да је измењен само један фајл („*README.md*”), ткао што је додата једна линија текста који гласи „*Adding some important info to the readme.*” (линија 3). Затим, на Слици А.6 дат је преглед измена (*commits*), и може се видети да је направљена само једна измена са поруком „*README.md edited online with Bitbucket*”.

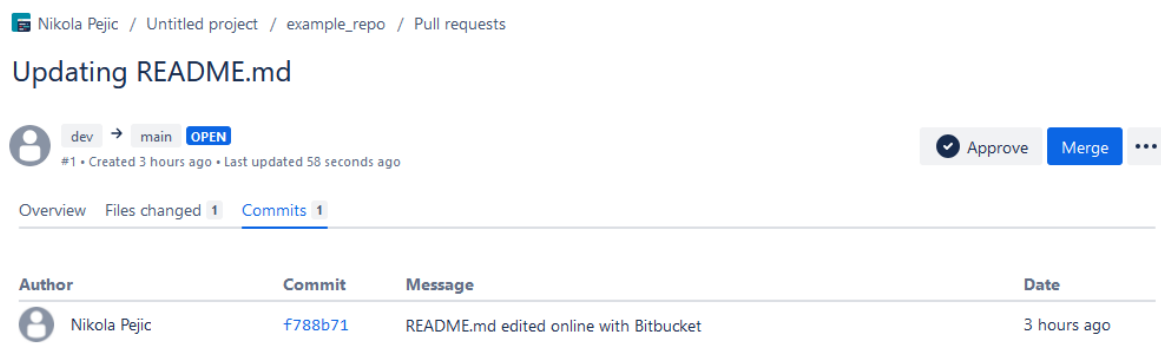
Bitbucket нуди REST интерфејс [112] зарад лакшег приступа подацима о предлозима измена и рецензијама. Слањем захтева из Листинга А.3 могу се излистати сви предлози измена датог репозиторијума.

- 1 GET `https://api.bitbucket.org/2.0/repositories/{workspace}/{repo_slug}/pullrequests`

Листинг А.3: Пример захтева за дохватање предлога измена на сервису Bitbucket.



Слика А.5: Пример предлога измена на сервису Bitbucket - фајлови.



Слика А.6: Пример предлога измена на сервису Bitbucket - измене.

Пример одговора претходног захтева дат је у Листингу А.4 (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [112]).


```

1 {
2   "size": 142,
3   "page": 102,
4   "pagelen": 159,
5   "next": "<string>",
6   "previous": "<string>",
7   "values": [
8     {
9       "type": "<string>",
10      "id": 108,
11      "title": "<string>",
12      "rendered": {
13        "title": {
14          "raw": "<string>",
15          "markup": "markdown",
16          "html": "<string>"
17        },
18        "description": {
19          "raw": "<string>",
20          "markup": "markdown",
21          "html": "<string>"
22        },
23        "reason": {
24          "raw": "<string>",
25          "markup": "markdown",
26          "html": "<string>"
27        }
28      },
29      "summary": {
30        "raw": "<string>",
31        "markup": "markdown",
32        "html": "<string>"
33      },
34      "state": "OPEN",
35      "author": {
36        "type": "<string>"
37      },
38      "source": {},
39      "destination": {},
40      "merge_commit": {
41        "hash": "<string>"
42      },
43      "comment_count": 51,
44      "task_count": 53,
45      "close_source_branch": true,
46      "closed_by": {
47        "type": "<string>"
48      },
49      "reason": "<string>",
50      "created_on": "<string>",
51      "updated_on": "<string>",
52      "reviewers": [
53        {
54          "type": "<string>"
55        }
56      ],
57      "participants": [
58        {
59          "type": "<string>"
60        }
61      ]
62    }
63  ]
64 }

```

Листинг А.4: Пример одговора на захтев за дохватање података о предлозима измена сервиса Bitbucket (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [112]).

Литература

- [1] G. Gousios, M. A. Storey и A. Bacchelli, „Work practices and challenges in pull-based development: The contributor’s perspective”, y *Proceedings - International Conference on Software Engineering*, св. 14-22-May-2016, 2016. DOI: 10.1145/2884781.2884826.
- [2] G. Gousios, A. Zaidman, M. A. Storey и A. Van Deursen, „Work practices and challenges in pull-based development: The integrator’s perspective”, y *Proceedings - International Conference on Software Engineering*, св. 1, 2015. DOI: 10.1109/ICSE.2015.55.
- [3] C. Sadowski, E. Söderberg, L. Church, M. Sipko и A. Bacchelli, „Modern code review: A case study at google”, y *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, мај 2018., стр. 181–190, ISBN: 9781450356596. DOI: 10.1145/3183519.3183525.
- [4] S. Asthana и др., „WhoDo: Automating reviewer suggestions at scale”, y *ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019. DOI: 10.1145/3338906.3340449.
- [5] O. Kononenko, O. Baysal, L. Guerrouj, Y. Cao и M. W. Godfrey, „Investigating code review quality: Do people and participation matter?”, y *2015 IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015 - Proceedings*, 2015. DOI: 10.1109/ICSM.2015.7332457.
- [6] A. Bosu, J. C. Carver, C. Bird, J. Orbeck и C. Chockley, „Process Aspects and Social Dynamics of Contemporary Code Review: Insights from Open Source Development and Industrial Practice at Microsoft”, *IEEE Transactions on Software Engineering*, св. 43, бр. 1, 2017., ISSN: 00985589. DOI: 10.1109/TSE.2016.2576451.
- [7] P. Thongtanunam, C. Tantithamthavorn, R. G. Kula, N. Yoshida, H. Iida и K.-i. Matsumoto, „Who Should Review My Code ?”, *Proceedings of the the 22nd IEEE*

- International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2015.
- [8] Y. Yu, H. Wang, G. Yin и T. Wang, „Reviewer recommendation for pull-requests in GitHub: What can we learn from code review and bug assignment?”, *Information and Software Technology*, св. 74, 2016., ISSN: 09505849. DOI: 10.1016/j.infsof.2016.01.004.
 - [9] M. E. Fagan, „Design and code inspections to reduce errors in program development”, *IBM Systems Journal*, св. 15, бр. 3, стр. 182–211, 1976. DOI: 10.1147/sj.153.0182.
 - [10] B. Runge, „The Inspection Method applied to small projects”, в *Proceedings of the 6th International Conference on Software Engineering*, сер. ICSE '82, Tokyo, Japan: IEEE Computer Society Press, 1982., стр. 416–417.
 - [11] D. Bisant и J. Lyle, „A Two-Person Inspection Method to Improve Programming Productivity”, *IEEE Transactions on Software Engineering*, св. 15, бр. 10, стр. 1294–1304, 1989. DOI: 10.1109/TSE.1989.559782.
 - [12] S. Kusumoto, A. Chimura, T. Kikuno, K.-i. Matsumoto и Y. Mohri, „A promising approach to two-person software review in educational environment”, *Journal of Systems and Software*, св. 40, бр. 2, стр. 115–123, 1998.
 - [13] D. Parnas и D. Weiss, „Active design reviews: Principles and practices”, *Journal of Systems and Software*, св. 7, бр. 4, стр. 259–265, 1987., ISSN: 0164-1212. DOI: [https://doi.org/10.1016/0164-1212\(87\)90025-2](https://doi.org/10.1016/0164-1212(87)90025-2). адреса: <https://www.sciencedirect.com/science/article/pii/0164121287900252>.
 - [14] J. C. Knight и E. A. Myers, „An improved inspection technique”, *Commun. ACM*, св. 36, бр. 11, стр. 51–61, ноя. 1993., ISSN: 0001-0782. DOI: 10.1145/163359.163366. адреса: <https://doi.org/10.1145/163359.163366>.
 - [15] T. Thelin, H. Petersson, P. Runeson и C. Wohlin, „Applying sampling to improve software inspections”, *Journal of Systems and Software*, св. 73, бр. 2, стр. 257–269, 2004., Applications of statistics in software engineering, ISSN: 0164-1212. DOI: [https://doi.org/10.1016/S0164-1212\(03\)00249-8](https://doi.org/10.1016/S0164-1212(03)00249-8). адреса: <https://www.sciencedirect.com/science/article/pii/S0164121203002498>.
 - [16] L. G. Votta, „Does every inspection need a meeting?”, *SIGSOFT Softw. Eng. Notes*, св. 18, бр. 5, стр. 107–114, дек. 1993., ISSN: 0163-5948. DOI: 10.1145/167049.167070. адреса: <https://doi.org/10.1145/167049.167070>.

- [17] C. Sauer, D. Jeffery, L. Land и P. Yetton, „The effectiveness of software development technical reviews: a behaviorally motivated program of research”, *IEEE Transactions on Software Engineering*, св. 26, бр. 1, стр. 1–14, 2000. DOI: 10.1109/32.825763.
- [18] D. Kelly и T. Shepard, „Task-directed software inspection”, *Journal of Systems and Software*, св. 73, бр. 2, стр. 361–368, 2004., Applications of statistics in software engineering, ISSN: 0164-1212. DOI: <https://doi.org/10.1016/j.jss.2003.10.027>. адреса: <https://www.sciencedirect.com/science/article/pii/S016412120300270X>.
- [19] D. Mishra и A. Mishra, „Efficient software review process for small and medium enterprises”, *IET Software*, св. 1, бр. 4, стр. 132–142, 2007. DOI: 10.1049/IET-SEN:20070011.
- [20] L. Brothers, V. Sembugamoorthy и M. Muller, „ICICLE: groupware for code inspection”, сер. CSCW '90, Los Angeles, California, USA: Association for Computing Machinery, 1990., стр. 169–181, ISBN: 0897914023. DOI: 10.1145/99332.99353. адреса: <https://doi.org/10.1145/99332.99353>.
- [21] J. C. Knight и E. A. Myers, „An improved inspection technique”, *Commun. ACM*, св. 36, бр. 11, стр. 51–61, нов. 1993., ISSN: 0001-0782. DOI: 10.1145/163359.163366. адреса: <https://doi.org/10.1145/163359.163366>.
- [22] J. Gintell, J. Arnold, M. Houde, J. Kruszelnicki, R. McKenney и G. Memmi, „Scrutiny: A collaborative inspection and review system”, у *Software Engineering — ESEC '93*, I. Sommerville и M. Paul, уредници, Berlin, Heidelberg: Springer Berlin Heidelberg, 1993., стр. 344–360, ISBN: 978-3-540-47972-7.
- [23] V. Mashayekhi, J. Drake, W.-T. Tsai и J. Riedl, „Distributed, collaborative software inspection”, *IEEE Software*, св. 10, бр. 5, стр. 66–75, 1993. DOI: 10.1109/52.232404.
- [24] H. Hedberg, „Introducing the next generation of software inspection tools”, у *International Conference on Product Focused Software Process Improvement*, Springer, 2004., стр. 234–247.
- [25] N. Pejić, Z. Radivojević и M. Cvetanović, „Helping Pull Request Reviewer Recommendation Systems to Focus”, *IEEE Access*, св. 11, стр. 71 013–71 025, 2023. DOI: 10.1109/ACCESS.2023.3292056.
- [26] P. C. Rigby и C. Bird, „Convergent contemporary software peer review practices”, у *2013 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2013 - Proceedings*, 2013. DOI: 10.1145/2491411.2491444.

- [27] Microsoft. „What is Azure DevOps?” (2022.), адреса: <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>. (преузето: 04.02.2024.)
- [28] „git: –fast-version-control”, адреса: <https://git-scm.com/>. (преузето: 04.02.2024.)
- [29] Microsoft. „What is Team Foundation Version Control?”, адреса: <https://learn.microsoft.com/en-us/azure/devops/repos/tfvc/what-is-tfvc?view=azure-devops>. (преузето: 04.02.2024.)
- [30] Microsoft. „What is Azure Boards?”, адреса: <https://learn.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops>. (преузето: 04.02.2024.)
- [31] Microsoft. „What is Azure Pipelines?”, адреса: <https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>. (преузето: 04.02.2024.)
- [32] Microsoft. „Overview of services”, адреса: <https://learn.microsoft.com/en-us/azure/devops/user-guide/services?toc=%2Fazure%2Fdevops%2Fget-started%2Ftoc.json&view=azure-devops>. (преузето: 04.02.2024.)
- [33] R. Fielding, „Architectural Styles and the Design of Network-based Software Architectures”, PhD dissertation, University of California, 2000. адреса: https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.
- [34] Microsoft. „Azure DevOps Services REST API Reference”, адреса: <https://learn.microsoft.com/en-us/rest/api/azure/devops/?view=azure-devops-rest-7.2>. (преузето: 07.02.2024.)
- [35] Microsoft. „Pull Requests - Get Pull Requests”, адреса: <https://learn.microsoft.com/en-us/rest/api/azure/devops/git/pull-requests/get-pull-requests?view=azure-devops-rest-7.2&tabs=HTTP>. (преузето: 07.02.2024.)
- [36] Microsoft. „Pull Request Threads - List”, адреса: <https://learn.microsoft.com/en-us/rest/api/azure/devops/git/pull-request-threads/list?view=azure-devops-rest-7.2&tabs=HTTP>. (преузето: 16.02.2024.)
- [37] GitHub. „About GitHub and Git”, адреса: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>. (преузето: 16.02.2024.)
- [38] GitHub. „About issues”, адреса: <https://docs.github.com/en/issues/tracking-your-work-with-issues/about-issues>. (преузето: 16.02.2024.)

- [39] GitHub. „About Projects”, адреса: <https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>. (преузето: 16.02.2024.)
- [40] GitHub. „Understanding GitHub Actions”, адреса: <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>. (преузето: 16.02.2024.)
- [41] GitHub. „About GitHub Pages”, адреса: <https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages>. (преузето: 16.02.2024.)
- [42] GitHub. „Pulls”, адреса: <https://docs.github.com/en/rest/pulls/pulls?apiVersion=2022-11-28>. (преузето: 16.02.2024.)
- [43] GitHub. „About the REST API”, адреса: <https://docs.github.com/en/rest/about-the-rest-api/about-the-rest-api?apiVersion=2022-11-28>. (преузето: 16.02.2024.)
- [44] I. Grigorik. „GH Archive”, адреса: <https://www.gharchive.org/>. (преузето: 16.02.2024.)
- [45] Google. „BigQuery overview”, адреса: <https://cloud.google.com/bigquery/docs/introduction>. (преузето: 17.02.2024.)
- [46] Google. „BigQuery pricing”, адреса: <https://cloud.google.com/bigquery/pricing#queries>. (преузето: 17.02.2024.)
- [47] G. Gousios и D. Spinellis, „GHTorrent: Github’s data from a firehose”, у *IEEE International Working Conference on Mining Software Repositories*, 2012. DOI: 10.1109/MSR.2012.6224294.
- [48] K. Chodorow, *MongoDB: The Definitive Guide*. 2013., св. 203.
- [49] G. Gousios, M. Pinzger и A. V. Deursen, „An exploratory study of the pull-based software development model”, у *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, мај 2014., стр. 345–355. DOI: 10.1145/2568225.2568260.
- [50] D. Pletea, B. Vasilescu и A. Serebrenik, „Security and emotion: Sentiment analysis of security discussions on GitHub”, у *11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings*, 2014. DOI: 10.1145/2597073.2597117.
- [51] „domain expired”, адреса: <https://github.com/gousiosg/github-mirror/issues/108>. (преузето: 16.06.2024.)
- [52] N. Pejić, Z. Radivojević и M. Cvetavnović, „Analyzing the Impact of COVID-19 on GitHub Event Trends”, *Sustainability*, св. 15, бр. 19, 2023., ISSN: 2071-1050. DOI: 10.3390/su151914622. адреса: <https://www.mdpi.com/2071-1050/15/19/14622>.
- [53] W. H. O. (WHO). „Timeline: WHO’s COVID-19 response”, адреса: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/interactive-timeline>. (преузето: 17.03.2024.)

- [54] CNN Editorial Research, *Covid-19 Pandemic Timeline Fast Facts*, Available online: <https://edition.cnn.com/2021/08/09/health/covid-19-pandemic-timeline-fast-facts/index.html> (accessed on 1 Jul 2023).
- [55] T. Carvalho, F. Krammer и A. Iwasaki, *The first 12 months of COVID-19: a timeline of immunological insights*, 2021. DOI: 10.1038/s41577-021-00522-1.
- [56] *WHO chief declares end to COVID-19 as a global health emergency*, Available online: <https://news.un.org/en/story/2023/05/1136367> (accessed on 24 Jun 2023). адреса: <https://news.un.org/en/story/2023/05/1136367>.
- [57] *Octoverse spotlight: An analysis of developer productivity, work cadence, and collaboration in the early days of COVID-19*, Available online: <https://github.blog/2020-05-06-octoverse-spotlight-an-analysis-of-developer-productivity-work-cadence-and-collaboration-in-the-early-days-of-covid-19/> (accessed on 26 Jul 2023). адреса: <https://github.blog/2020-05-06-octoverse-spotlight-an-analysis-of-developer-productivity-work-cadence-and-collaboration-in-the-early-days-of-covid-19/>.
- [58] Microsoft, „Octoverse - Finding balance between work and play”, *Github.Com*, 2021.
- [59] V. Klotzman, F. Farmahinifarahani и C. Lopes, „Public software development activity during the pandemic”, y *International Symposium on Empirical Software Engineering and Measurement*, 2021. DOI: 10.1145/3475716.3475778.
- [60] statsmodels. „statsmodels.tsa.stattools.adfuller”, адреса: <https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html>. (презенто: 17.03.2024.)
- [61] Y. Yu, H. Wang, G. Yin и C. X. Ling, „Reviewer recommender of pull-requests in GitHub”, y *Proceedings - 30th International Conference on Software Maintenance and Evolution, ICSME 2014*, 2014. DOI: 10.1109/ICSME.2014.107.
- [62] X. Ye, „Learning to Rank Reviewers for Pull Requests”, *IEEE Access*, т. 7, 2019., ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2925560.
- [63] A. Aizawa, „An information-theoretic perspective of tf-idf measures”, *Information Processing and Management*, т. 39, кн. 1, 2003., ISSN: 03064573. DOI: 10.1016/S0306-4573(02)00021-3.
- [64] Microsoft. „Extensions overview”, адреса: <https://learn.microsoft.com/en-us/azure/devops/extend/overview?view=azure-devops>. (презенто: 31.08.2024.)

- [65] Y. Yu, H. Wang, V. Filkov, P. Devanbu и B. Vasilescu, „Wait for it: Determinants of pull request evaluation latency on github”, у *2015 IEEE/ACM 12th working conference on mining software repositories*, IEEE, 2015., стр. 367–371.
- [66] L. MacLeod, M. Greiler, M.-A. Storey, C. Bird и J. Czerwonka, „Code reviewing in the trenches: Challenges and best practices”, *IEEE Software*, св. 35, бр. 4, стр. 34–42, 2017.
- [67] G. Kudrjavets, A. Kumar, N. Nagappan и A. Rastogi, „Mining code review data to understand waiting times between acceptance and merging: An empirical analysis”, у *Proceedings of the 19th International Conference on Mining Software Repositories*, 2022., стр. 579–590.
- [68] X. Zhang, Y. Yu, T. Wang, A. Rastogi и H. Wang, „Pull request latency explained: An empirical overview”, *Empirical Software Engineering*, св. 27, бр. 6, стр. 126, 2022.
- [69] L. Yang и др., „Evaluating Learning-to-Rank Models for Prioritizing Code Review Requests using Process Simulation”, у *Proceedings - 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2023*, 2023. DOI: 10.1109/SANER56733.2023.00050.
- [70] G. Kudrjavets и A. Rastogi, „Does code review speed matter for practitioners?”, *Empirical Software Engineering*, св. 29, бр. 1, 2024., ISSN: 15737616. DOI: 10.1007/s10664-023-10401-z.
- [71] C. S. Maddila, S. S. Upadrasta, C. Bansal, N. Nagappan, G. Gousios и A. van Deursen, „Nudge: Accelerating Overdue Pull Requests Towards Completion. CoRR abs/2011.12468 (2021)”, *arXiv preprint arXiv:2011.12468*, 2021.
- [72] Microsoft. „Use @mentions in work items and pull requests”, адреса: <https://learn.microsoft.com/en-us/azure/devops/organizations/notifications/at-mentions?view=azure-devops>. (преузето: 11.05.2024.)
- [73] Microsoft. „Microsoft Viva Insights”, адреса: <https://www.microsoft.com/en-us/microsoft-viva/insights>. (преузето: 18.08.2024.)
- [74] B. John, Z. Alsamarra'i и N. Panteli, „Enhancing employee experience in the era of hybrid work: the case of Microsoft viva”, *IEEE Software*, св. 40, бр. 2, стр. 70–79, 2022.
- [75] Microsoft. „Briefing email overview”, адреса: <https://learn.microsoft.com/en-us/viva/insights/personal/briefing/be-overview>. (преузето: 11.05.2024.)
- [76] Microsoft. „Follow up on tasks”, адреса: <https://learn.microsoft.com/en-us/viva/insights/personal/briefing/be-tasks>. (преузето: 18.08.2024.)

- [77] Microsoft. „About the Task Scheduler”, адреса: <https://learn.microsoft.com/en-us/windows/win32/taskschd/about-the-task-scheduler>. (преузето: 22.02.2024.)
- [78] P. Thongtanunam, C. Tantithamthavorn, R. G. Kula, N. Yoshida, H. Iida и K. I. Matsumoto, „Who should review my code? A file location-based code-reviewer recommendation approach for Modern Code Review”, y *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2015 - Proceedings*, 2015. DOI: 10.1109/SANER.2015.7081824.
- [79] C. Hannebauer, M. Patalas, S. Stünkely и V. Gruhn, „Automatically recommending code reviewers based on their expertise: An empirical comparison”, y *ASE 2016 - Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 2016. DOI: 10.1145/2970276.2970306.
- [80] J. Jiang, Y. Yang, J. He, X. Blanc и L. Zhang, „Who should comment on this pull request? Analyzing attributes for more accurate commenter recommendation in pull-based development”, *Information and Software Technology*, св. 84, 2017., ISSN: 09505849. DOI: 10.1016/j.infsof.2016.10.006.
- [81] M. Fejzer, P. Przymus и K. Stencel, „Profile based recommendation of code reviewers”, *Journal of Intelligent Information Systems*, св. 50, стр. 597–619, 2018.
- [82] R. Ranawana и V. Palade, „Multi-Classifer Systems: Review and a roadmap for developers”, *International Journal of Hybrid Intelligent Systems*, св. 3, бр. 1, 2016., ISSN: 14485869. DOI: 10.3233/his-2006-3104.
- [83] T. Joachims, „Training linear SVMs in linear time”, y *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, св. 2006, 2006. DOI: 10.1145/1150402.1150429.
- [84] T. Joachims, „Optimizing search engines using clickthrough data”, y *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002. DOI: 10.1145/775047.775067.
- [85] P. Thongtanunam и A. E. Hassan, *Review Dynamics and Their Impact on Software Quality*, 2021. DOI: 10.1109/TSE.2020.2964660.
- [86] P. Thongtanunam, S. McIntosh, A. E. Hassan и H. Iida, „Revisiting code ownership and its relationship with software quality in the scope of modern code review”, y *Proceedings - International Conference on Software Engineering*, св. 14-22-May-2016, 2016. DOI: 10.1145/2884781.2884852.

- [87] „Papers with code”, адреса: <https://paperswithcode.com/>. (преузето: 24.08.2024.)
- [88] S. Zhou и др., „WebArena: A Realistic Web Environment for Building Autonomous Agents”, *arXiv preprint arXiv:2307.13854*, 2023. адреса: <https://webarena.dev>.
- [89] M. Wermelinger, „Using github copilot to solve simple programming problems”, *y Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 2023., стр. 172–178.
- [90] B. Roziere и др., „Code llama: Open foundation models for code”, *arXiv preprint arXiv:2308.12950*, 2023.
- [91] M. N. Sakib, M. A. Islam и M. M. Arifin, „Automatic Pull Request Description Generation Using LLMs: A T5 Model Approach”, *arXiv preprint arXiv:2408.00921*, 2024.
- [92] Z. Rasheed и др., „AI-powered Code Review with LLMs: Early Results”, *arXiv preprint arXiv:2404.18496*, 2024.
- [93] B. GitLab, „Gitlab”, *The One DevOps Platform*, 2022. адреса: https://mycourses.aalto.fi/pluginfile.php/1436285/mod_resource/content/2/GitLab_and_GitLab_CI.pdf.
- [94] „MIT License”, адреса: <https://choosealicense.com/licenses/mit/>. (преузето: 12.10.2024.)
- [95] GitLab. „Tasks”, адреса: <https://docs.gitlab.com/ee/user/tasks.html>. (преузето: 12.10.2024.)
- [96] GitLab. „Issues”, адреса: <https://docs.gitlab.com/ee/user/project/issues/>. (преузето: 12.10.2024.)
- [97] GitLab. „Milestones”, адреса: <https://docs.gitlab.com/ee/user/project/milestones/>. (преузето: 12.10.2024.)
- [98] GitLab. „Iterations”, адреса: <https://docs.gitlab.com/ee/user/group/iterations/>. (преузето: 12.10.2024.)
- [99] GitLab. „Time tracking”, адреса: https://docs.gitlab.com/ee/user/project/time_tracking.html. (преузето: 12.10.2024.)
- [100] GitLab. „CI/CD pipelines”, адреса: <https://docs.gitlab.com/ee/ci/pipelines/>. (преузето: 12.10.2024.)
- [101] GitLab. „Package registry”, адреса: https://docs.gitlab.com/ee/user/packages/package_registry/. (преузето: 12.10.2024.)
- [102] GitLab. „Releases”, адреса: <https://docs.gitlab.com/ee/user/project/releases/>. (преузето: 12.10.2024.)

- [103] GitLab. „GitLab Pages”, адреса: <https://docs.gitlab.com/ee/user/project/pages/>. (преузето: 12.10.2024.)
- [104] GitLab. „REST API”, адреса: <https://docs.gitlab.com/ee/api/rest/>. (преузето: 13.10.2024.)
- [105] GitLab. „Merge requests API”, адреса: https://docs.gitlab.com/ee/api/merge_requests.html. (преузето: 13.10.2024.)
- [106] Atlassian. „A brief overview of Bitbucket”, адреса: <https://bitbucket.org/product/guides/getting-started/overview#bitbucket-software-hosting-options>. (преузето: 13.10.2024.)
- [107] Atlassian. „Understand Bitbucket issues”, адреса: <https://support.atlassian.com/bitbucket-cloud/docs/understand-bitbucket-issues/>. (преузето: 13.10.2024.)
- [108] Atlassian. „All great projects start with Jira”, адреса: <https://www.atlassian.com/software/jira>. (преузето: 13.10.2024.)
- [109] Atlassian. „Get started with Bitbucket Pipelines”, адреса: <https://support.atlassian.com/bitbucket-cloud/docs/get-started-with-bitbucket-pipelines/>. (преузето: 13.10.2024.)
- [110] Atlassian. „Bitbucket deployment guidelines”, адреса: <https://support.atlassian.com/bitbucket-cloud/docs/bitbucket-deployment-guidelines/>. (преузето: 13.10.2024.)
- [111] Atlassian. „Publishing a Website on Bitbucket Cloud”, адреса: <https://support.atlassian.com/bitbucket-cloud/docs/publishing-a-website-on-bitbucket-cloud/>. (преузето: 13.10.2024.)
- [112] Atlassian. „Pullrequests”, адреса: <https://developer.atlassian.com/cloud/bitbucket/rest/api-group-pullrequests/#api-group-pullrequests>. (преузето: 14.10.2024.)

Списак слика

1.1	Ток истраживања дисертације	2
2.1	Дијаграм секвенци који упрошћено приказује процес рецензије кода 2000-их година	12
2.2	Дијаграм секвенци који упрошћено приказује процес развоја и рецензије кода	15
2.3	Упрошћени приказ структуре података који су доступни за сваки предлог измена	16
2.4	Пример предлога измена на сервису Azure DevOps - преглед	17
2.5	Пример предлога измена на сервису Azure DevOps - фајлови	17
2.6	Пример предлога измена на сервису Azure DevOps - итерације	17
2.7	Пример предлога измена на сервису Azure DevOps - измене	18
2.8	Пример предлога измена на сервису GitHub - преглед	25
2.9	Пример предлога измена на сервису GitHub - измене	25
2.10	Пример предлога измена на сервису GitHub - фајлови	25
3.1	Преглед броја GitHub догађаја различитог типа од почетка 2017. до краја 2022. године	33
3.2	Преглед броја различитих типова GitHub догађаја од почетка 2017. до краја 2022. године	35

4.1	Различити видови претраге инжењера (експерата) на платформи GitHub . . .	43
4.2	Шематски приказ „PR Insights” веб сајта	45
4.3	Изглед „PR Insights” веб странице - претрага предлога измена	47
4.4	Изглед „PR Insights” веб странице - претрага експерта	47
4.5	Изглед „PR Insights” веб странице - профил корисника	48
4.6	Укупне и јединствене дневне посете „PR Insights” веб странице кроз време . .	51
4.7	Дистрибуција броја корисника „PR Insights” веб странице према укупном броју посета	52
5.1	Приказ дела „briefing” мејла сервиса Viva Insights који се односи на мејлове који чекају на одговор	57
5.2	Шематски приказ решења за периодично подсећање рецензената на активне рецензије	62
5.3	Пример мејла који систем за периодично подсећање рецензената на активне рецензије шаље корисницима	64
5.4	Промена броја корисника система којима су послати мејлови кроз време . . .	65
5.5	Преглед дистрибуције предлога измена према категоријама у послатим мејловима	65
5.6	Преглед временске дистрибуције кликова на линкове рецензија у послатим мејловима	66
5.7	Дистрибуција корисника према броју кликова на линкове рецензија	66
5.8	Приказ релативне промене 90-тог перцентиала броја рецензија по послатом мејлу	69
6.1	График просечних прецизност-опозив перформанси свих постојећих метода за различите скупове података	84
6.2	Шематски приказ предложене технике за побољшање перформанси постојећих приступа.	86

6.3	Нормализовани утицај сваког базичног филтера при комбиновању са другим филтерима, за сваки постојећи приступ.	95
6.4	Нормализовани утицај сваког базичног филтера при комбиновању са другим филтерима, за сваки постојећи приступ.	96
A.1	Пример предлога измена на сервису GitLab - преглед	106
A.2	Пример предлога измена на сервису GitLab - измене	108
A.3	Пример предлога измена на сервису GitLab - фајлови	108
A.4	Пример предлога измена на сервису Bitbucket - преглед	109
A.5	Пример предлога измена на сервису Bitbucket - фајлови	110
A.6	Пример предлога измена на сервису Bitbucket - измене	110

Списак табела

2.1	Број грешака који су пронађени током различитих фаза развоја	9
3.2	Преглед претходних истраживања о утицају COVID-19 пандемије на open source развој на сервису GitHub.	29
3.3	Трендови промене броја догађаја кроз време за два посматрана периода . . .	34
3.4	Анализа стационарности броја догађаја кроз време	37
5.1	Упитник о периодичним подсетницима на активне рецензије.	60
5.2	Резултати упитника о периодичним подсетницима на активне рецензије. . . .	61
5.3	Анкета о сентименту корисника према систему за подсећање рецензената . .	67
5.4	Резултати анкете о сентименту корисника према систему за подсећање рецензената.	68
6.1	Преглед постојећих метода за препоруку рецензената.	73
6.2	Преглед категорија метрика које Приступ 7 користи.	79
6.3	Преглед постојећих решења са статистикама о скупу података над којима су тестирани.	83
6.4	Преглед различитих скупова података (репозиторијума) који су коришћени у евалуацији.	88
6.5	Преглед агрегираних резултата експеримената за ИП6.1. - прецизност	89

6.6	Преглед агрегираних резултата експеримената за ИП6.1. - опозив.	89
6.7	Детаљнији резултати ИП6.1. експеримента	90
6.8	Детаљнији резултати ИП6.2. експеримента	92
6.9	Преглед најбољих резултата композитних филтера у експерименту за ИП6.2. - прецизност.	93
6.10	Преглед најбољих резултата композитних филтера у експерименту за ИП6.2. - опозив.	94
6.11	Максимално побољшање прецизности применом било којег филтера за парове постојећих приступа и скупова података.	96
6.12	Максимално побољшање опозива применом било којег филтера за парове постојећих приступа и скупова података.	97

Списак листинга

2.1	HTTP GET захтев за дохватање података о предлозима измена и рецензијама са сервиса Azure DevOps	18
2.2	Пример одговора на захтев за дохватање података о предлозима измена и рецензијама са сервиса Azure DevOps (пример је преузет из званичне документације сервиса [35]).	19
2.3	Пример одговора на захтев за дохватање података о коментарима на рецензији са сервиса Azure DevOps (пример представља редуковану верзију из званичне документације сервиса [36]).	20
2.4	HTTP GET захтев за дохватање података о коментарима на појединачној рецензији са сервиса Azure DevOps	21
2.5	HTTP GET захтев за дохватање података о предлозима измена са сервиса GitHub	21
2.6	Пример одговора на захтев за дохватање података о предлозима измена сервиса GitHub (пример представља поједностављену верзију из званичне документације [42]).	22
2.7	Пример захтева за дохватање измењених фајлова на сервису GitHub.	23
2.8	Пример захтева за дохватање коментара рецензената на сервису GitHub.	23
2.9	Пример URL адресе за дохватање GitHub телеметрије са GH Archive сервиса за период од 15 до 16 часова 01.01.2015. године.	23
2.10	Пример BigQuery упита за обраду GH Archive података за 01.01.2024. године.	24

3.1	Пример упита који је коришћен за дохватање једне године (2022) скупа података помоћу BigQuery система.	32
A.1	Пример захтева за дохватање предлога измена на сервису GitLab.	106
A.2	Пример одговора на захтев за дохватање података о предлозима измена сервиса GitLab (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [105]).	107
A.3	Пример захтева за дохватање предлога измена на сервису Bitbucket.	109
A.4	Пример одговора на захтев за дохватање података о предлозима измена сервиса Bitbucket (пример представља поједностављену верзију примера одговора који је доступан у званичној документацији [112]).	111

Биографија аутора

Никола Пејић рођен је 28. октобра 1996. године у Лесковцу. Електротехнички факултет у Београду уписао је 2014. године. Дипломирао је 2018. године на модулу Рачунарска техника и информатика са просечном оценом 9.69. Дипломски рад под насловом „Креирање статистика екстерних табела лоцираних на кластеру“ израдио је под менторством ван. проф. др Милоша Цветановића и одбранио га са оценом 10.

Мастер академске студије уписао је 2018. године на Електротехничком факултету у Београду на модулу Рачунарска техника и информатика, и завршио их је 2019. године са просечном оценом 10.00. Мастер рад под насловом „Процена сличности процедура преведених различитим компајлерима применом неуралних мрежа“ израдио је под менторством ван. проф. др Милоша Цветановића и одбранио га је са оценом 10. Докторске академске студије уписао је 2020. године на Електротехничком факултету у Београду. Све испите предвиђене студијским програмом положио је са оценом 10. У току докторских студија објавио је 2 рада у међународним часописима.

Од 2019. године ради као софтверски инжењер у развојном центру компаније Microsoft у Београду. Током своје каријере био је део SQL тима, тима за мешовиту стварност (Mixed Reality) и примењене науке (Applied Sciences Group), а данас је део Azure DevOps тима.

Изјава о ауторству

Име и презиме аутора: Никола Пејић

Број индекса: 5029/2020

Изјављујем

да је докторска дисертација под насловом

Побољшање система за рецензију софтверских измена на основу информација генерисаних из историје рецензија

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио ауторска права и користио интелектуалну својину других лица

Потпис аутора

У Београду, 29.04.2025.

Никола Пејић

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: **Никола Пејић**

Број индекса: **5029/2020**

Студијски програм: **Рачунарска техника и информатика**

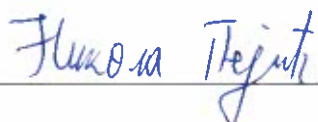
Наслов рада: **Побољшање система за рецензију софтверских измена на основу информација генерисаних из историје рецензија**

Ментор: **ван. проф. др Милош Цветановић**

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао ради похрањивања у Дигиталном репозиторијуму Универзитета у Београду. Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада. Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 29.04.2025.



Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Побољшање система за рецензију софтверских измена на основу информација генерисаних из историје рецензија

која је моје ауторско дело.

Дисертацију са свим прилозима предао сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative commons) за коју сам се одлучио.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
- ③ Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

Потпис аутора

У Београду, 29.04.2025.



1. **Ауторство.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.
2. **Ауторство – некомерцијално.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.
3. **Ауторство – некомерцијално – без прерада.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.
4. **Ауторство – некомерцијално – делити под истим условима.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.
5. **Ауторство – без прерада.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.
6. **Ауторство – делити под истим условима.** Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.