Življenjski cikel cevovodov neprekinjene namestitve informacijskih rešitev

Luka Četina, Luka Pavlič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Maribor, Slovenija
luka.cetina@um.si, luka.pavlic@um.si

CI/CD cevovodi postajajo nepogrešljivo orodje pri razvoju informacijskih sistemov, vendar je njihov življenjski cikel razmeroma slabo raziskan. V članku na podlagi pregleda literature opredelimo CI/CD cevovode, njihovo pogostost, najpogostejše gradnike ter orodja za vzpostavitev CI/CD cevovodov. Predstavimo tudi empirično raziskavo, ki vključuje podrobno analizo 1000 repozitorijev na platformi GitHub. Da bi raziskali življenjski cikel CI/CD cevovodov smo preučili strukturo cevovodov v repozitorijih in spremembe tekom projekta. Ugotovili smo, da so CI/CD cevovodi prisotni v 42% repozitorijev, povprečen čas do njihove vključitve pa je sedem mesecev. Skoraj vsi analizirani cevovodi vključujejo gradnjo, 62% jih vključuje tudi izdajo, 46% testiranje, 26% analizo kode in zgolj 18% namestitev. Analizirali smo tudi spremembe programske kode in cevovodov tekom projekta, ter ugotovili, da se ob povečanju sprememb v kodi poveča tudi število sprememb cevovodov in da spremembe cevovodov v povprečju predstavljajo 4% vseh sprememb v repozitoriju. Ugotovili smo tudi, da se cevovodi najbolj spreminjajo na začetku in koncu razvoja projektov, kar odraža njihovo vzpostavitev in kasnejše prilagoditve ter optimizacije.

Ključne besede:
CI/CD cevovodi
razvoj programske opreme
življenjski cikel CI/CD cevovodov
neprekinjena namestitev
neprekinjena dostava

Univerzitetna založba Univerze v Mariboru

ISBN 978-961-286-894-9
DOI https://doi.org/10.18690/um.feri.4.2024.4

1 Uvod

Razvijalci neprestano iščejo načine, kako se z manj truda in brez zmanjšanja kakovosti odzivati na spremembe pri razvoju informacijskih rešitev. Prakse neprekinjene integracije (*ang.* continuous integration), neprekinjene dostave (*ang.* continuous delivery) in neprekinjene namestitve (*ang.* continuous deployment), ki jih z okrajšavo zapišemo kot CI/CD, to pomagajo doseči z uvedbo rednih integracij, izdaj ali namestitev programske opreme [1]. Ker so lahko omenjene aktivnosti zelo zamudne oz. zahtevajo veliko človeškega posredovanja, je avtomatizacija pri tem ključnega pomena. Avtomatiziran proces, ki aktivnosti v sklopu neprekinjenih praks poveže, in jih samodejno izvaja, poimenujemo cevovod neprekinjene integracije (*ang.* continuous integration pipeline), dostave (*ang.* continuous delivery pipeline) ali namestitve (*ang.* continuous deployment pipeline). Ta poskrbi za avtomatizacijo pogostih operacij kot so gradnja, testiranje in nameščanje [2].

Kljub temu da uporaba tovrstnih cevovodov dokazano skrajša čas do izdaje produkta ter izboljša produktivnost in učinkovitost [3], vpeljevanje le-teh v proces razvoja vseeno ni brez pasti. Neustrezno implementiran cevovod ne bo nudil omenjenih prednosti in morda celo otežil razvoj, saj bodo neustrezno avtomatizirane aktivnost pogosto zahtevale ročno posredovanje [4]. Avtorji [5] so kot glavne ovire pri vzpostavitvi cevovoda identificirali pomanjkanje lahko razumljive dokumentacije, težavno iskanje ter odpravljanje napak, težavno ponovno uporabo delov cevovoda in raznoliko sintakso pri različnih orodjih. Zato je pomembno, da se še pred vzpostavitvijo cevovodov zavedamo potreb ter morebitnih izzivov pri vzpostavitvi. Pomembna pa ni le vzpostavitev, temveč tudi vzdrževanje in dopolnjevanje cevovodov tekom razvoja, saj ti niso zgolj statični elementi, ampak dinamični procesi, ki se prilagajajo spremembam v razvojnem okolju, tehnologijah ter zahtevah projekta. Kljub temu, da je cevovod podporen element, je vseeno samostojen izdelek, ki ga je potrebno po vzpostavitvi vzdrževati in njegov nadaljnji razvoj tudi načrtovati. Življenjski cikel programske opreme je dobro poznan in definiran, medtem ko je razvoj cevovodov tekom projekta razmeroma slabo raziskan. Zampetti idr. [4] poudarjajo, da se cevovodi nenehno spreminjajo in razvijajo skozi čas. Spremembe v cevovodih pogosto odražajo prilagoditve na nove zahteve projektov, optimizacijo procesov in izboljšanje učinkovitosti. Najpogostejše spremembe v cevovodih povezane z optimizacijo faz, dodajanjem novih funkcionalnosti ter izboljšanjem obstoječih procesov. Spremljanje in analiza teh sprememb sta ključnega pomena za razumevanje življenjskega cikla cevovodov ter za zagotavljanje njihove učinkovitosti in zanesljivosti. Pomembno je, da razvijalci aktivno spremljajo in vodijo evolucijo cevovodov, saj to omogoča pravočasno prepoznavanje težav in uvajanje potrebnih izboljšav, kar posledično vodi do bolj stabilnega in učinkovitega razvojnega procesa.

Cilj članka je podrobneje raziskati ter definirati cevovode neprekinjene integracije, izdaje in namestitve, ugotoviti kako pogosto se uporabljajo in katere gradnike vsebujejo. Zanima nas tudi, katera orodja se uporabljajo za vzpostavitev cevovodov ter kako se ti čez čas spreminjajo.

Zaradi omejitev GitHub API in ker smo morali ročno analizirati aktivnost repozitorijev smo v analizo vključili le 1000 repozitorijev. Pri repozitorijih smo se omejili na tiste, napisane v programskem jeziku java, saj smo lahko tako na podlagi ukazov, uporabljenih v cevovodu, bolj zanesljivo identificirali vključene gradnike. Kot predstavnika cevovodov smo določili orodje GitHub Actions, zato smo pridobili le repozitorije, ki so bili ustvarjeni po izidu omenjenega orodja – po maju 2019. Aktivnost repozitorijev smo pregledovali ročno, kar pomeni, da obstaja možnost napak.

V naslednjem poglavju opredelimo cevovode, njihove najpogostejše gradnike in pogostost uporabe. V tretjem poglavju povzemamo sorodna dela, sledi pa mu predstavitev zajema in analize podatkov iz platforme GitHub. Peto poglavje vsebuje omejitve raziskave, v zaključku pa povzamemo ugotovitve in podamo načrte za prihodnje raziskave.

2 CI/CD cevovodi

Izraza »dostava« in »namestitev« se v kontekstu neprekinjenih praks pogosto uporabljata izmenjujoče, uporaba se med avtorji pogosto celo razlikuje, zato se je uveljavil izraz CI/CD cevovod (*ang.* CI/CD pipeline), ki zaobjema cevovode neprekinjene integracije, dostave in namestitve. Tudi ta izraz je pogosto napačno uporabljen saj se ga uporablja za označevanje vsake avtomatizacije oz. cevovoda, ki se uporablja med procesom razvoja programske opreme. Uporaba cevovoda ne pomeni, da pri razvoju sledimo neprekinjeni integraciji, dostavi oz. namestitvi, vendar le, da smo del aktivnosti avtomatizirali [6].

CI/CD cevovode lahko zato definiramo kot proces, ki avtomatizira in vodi informacijsko rešitev skozi posamezne aktivnosti razvoja. Izvajanje aktivnosti se lahko sproži glede na vnaprej definiran urnik, kot odziv na dogodke znotraj repozitorija (npr. nalaganje nove kode v repozitorij in ustvarjanje nove zahteve za združevanje kode - *ang.* pull request) ali ročno [2], [6], [7], [8].

2.1. Orodja za vzpostavitev CI/CD cevovodov

Orodja za vzpostavitev CI/CD cevovodov, nameščena pri stranki (ang. self hosted CI/CD tools), so orodja, ki jih morajo uporabniki oz. razvijalci namestiti in upravljati sami. Ker delujejo lokalno, jih je lažje prilagoditi potrebam razvoja, vendar to pomeni več dela z vzdrževanjem in upravljanjem. Leta 2001 je izšlo prvo širše uporabljano tovrstno orodje z imenom CruiseControl, čez 4 leta pa se pojavi tudi orodje Hudson, ki se zaradi pravnega spora preimenuje v Jenkins [9].

Orodja za vzpostavitev CI/CD cevovodov v oblaku (*ang.* cloud hosted CI/CD tools), so orodja, ki jih oblačni ponudniki ponujajo kot storitev in ne zahtevajo vzpostavitve ali vzdrževanja, zato jim rečemo tudi CI/CD kot storitev (*ang.* CI/CD as a Service). Eno izmed prvih tovrstnih orodij je bil Travis, ki je zaradi preproste integracije s platformo GitHub postal izredno priljubljen, kmalu pa mu je sledil CircleCI, pri katerem je bil poudarek na zanesljivosti, hitrosti ter podpori Docker zabojnikov. Podobna orodja so izdali tudi večji oblačni ponudniki kot so AWS, Google Cloud in Azure [9].

Orodja za vzpostavitev CI/CD cevovodov, integrirana z repozitorji za gostenje kode (*ang.* git hosting integrated CI/CD tools), so podobna tistim v oblaku, le da jih ponujajo platforme za gostenje kode in so že vključena v repozitorije. To pomeni, da orodij ni potrebno povezovati z repozitorji, kar olajša vzpostavitev cevovodov. Prvi so takšno orodje ponudili pri podjetju GitLab leta 2015, čez 3 leta pa je podobno orodje izdalo tudi podjetje GitHub, ki je zaradi tržnice vnaprej pripravljenih akcij (*ang.* actions) v zelo kratkem času postalo izredno priljubljeno [9]. Orodja integrirana z repozitoriji so danes najbolj priljubljena, med njimi pa prevladuje orodje GitHub Actions [2]. Rostami idr. [1] so v sklopu raziskave ugotovili, da sta najbolj priljubljeni orodji GitHub Actions in Travis, saj vsaj enega od njiju uporablja kar 90% GitHub repozitorijev, ki vključujejo CI/CD cevovode.

2.2. Najpogostejši gradniki

CI/CD cevovodi lahko avtomatizirajo vsako aktivnost, ki jo je mogoče zapisati v obliki ukaza ali zunanje skripte, kar pomeni, da so lahko posamezni gradniki cevovodov precej raznoliki. Obstajajo pa aktivnosti, ki so ključne med razvojem in se najbolj pogosto vključujejo v CI/CD cevovode [1], [7], [8], [10]:

- gradnja (*ang.* Build),
- testiranje (*ang.* Testing),
- analiza kode (*ang.* Code Analysis),
- izdaja (*ang.* Release),
- namestitev (*ang.* Deploy).

2.3. Pogostost uporabe

Po poročilu organizacije CD Foundation vsak drugi razvijalec uporablja neprekinjene prakse razvoja programske opreme, medtem ko so drugi avtorji [2] ugotovili, da zgolj 33% projektov na GitHub uporablja orodja za vzpostavitev CI/CD cevovodov. Čas do vključitve cevovoda v projekt se krajša, saj so leta 2012 cevovod v povprečju vključili po enem letu, medtem ko se je do leta 2023 to zmanjšalo na 3 mesece. Avtorji [5] so med študenti računalniških smeri izvedli anketo, ki je pokazala, da se je 42% udeležencev s CI/CD praksami že srečalo v sklopu služb na IT področju.

3 Sorodna dela

Raziskave na področju CI/CD cevovodov so se v zadnjih letih precej razširile, kljub temu pa se le redke ukvarjajo s spremembami cevovoda skozi čas. Najpogosteje naslavljajo orodja za vzpostavitev, pogostost uporabe ter izzive pri vzpostavitvi CI/CD cevovodov.

Rostami Mazrae idr. [1] so na podlagi poglobljenih intervjujev z izkušenimi razvijalci izvedli kvalitativno študijo o uporabi in migracijah CI/CD orodij. Identificirali so razloge za uporabo specifičnih tehnologij, razloge za sočasno uporabo več CI/CD orodij v istem projektu ter pogostost in vzroke za menjavo orodij. Njihove ugotovitve kažejo jasen trend migracije s Travis CI na GitHub Actions, pri čemer so razvijalci kot glavne razloge za spremembo navajali boljšo integracijo s platformo GitHub in lažje upravljanje cevovodov. Avtorji navajajo, da pogostost uporabe več CI/CD tehnologij hkrati kaže na potrebo po boljši podpori za sočasno uporabo teh tehnologij.

Da Gião idr. [2] so izvedli obsežno analizo CI/CD orodij, ki se uporabljajo v GitHub repozitorijih. Analizirali so 612.557 repozitorijev in ugotovili, da 33% teh projektov vključuje vsaj eno orodje za vzpostavitev CI/CD cevovodov. Podobno kot pri [1], se je tudi tukaj izkazalo, da je orodje GitHub Actions daleč najbolj priljubljeno, saj ga uporablja 58% CI/CD projektov. Raziskava je tudi pokazala, da veliko projektov uporablja več kot eno CI/CD orodje hkrati, pri čemer so GitHub Actions, Travis CI in Jenkins najbolj pogosto so-uporabljena orodja. Avtorji navajajo, da je uporaba več CI/CD orodij v istem projektu povezana z željo po izkoriščanju specifičnih prednosti posameznih orodij ter z omejitvami in pomanjkljivostmi posameznih tehnologij.

Zampetti idr. [4] so preučevali razvoj in spremembe CI/CD cevovodov v odprtokodnih projektih. Njihova študija je temeljila na analizi zgodovine sprememb v CI/CD cevovodih izbranih projektov, pri čemer so identificirali vzorce sprememb in prilagoditev, ki so jih izvajali razvijalci. Ugotovili so, da se CI/CD cevovodi skozi čas nenehno spreminjajo, pri čemer so najpogostejše spremembe povezane z optimizacijo procesov, izboljšanjem učinkovitosti in odpravljanjem napak. Kot najpogosteje spremenjeni elementi cevovodov so se pokazale faze in opravila (ang. jobs), sledijo pa jim dodajaje komentarjev in izvajanje refaktoriranja konfiguracije cevovoda.

4 Pridobivanje in analiza repozitorijev iz platforme GitHub

Da bi ugotovili kako pogosti so CI/CD cevovodi, kaj vsebujejo in kako se čez čas spreminjajo, smo tudi sami pridobili in analizirali cevovode odprtokodnih projektov. Za pridobivanje in analizo repozitorijev smo uporabili različna orodja in tehnologije. Za pisanje skript in avtomatizacijo postopka pridobivanja podatkov smo uporabili python, medtem ko smo za dostop do podatkov o repozitorijih in commitih uporabili GitHub API. Za kloniranje repozitorijev in pridobivanje podatkov o commitih smo uporabili Git, za obdelavo in analizo podatkov ter shranjevanje rezultatov v CSV datoteke pa python knjižnico Pandas.

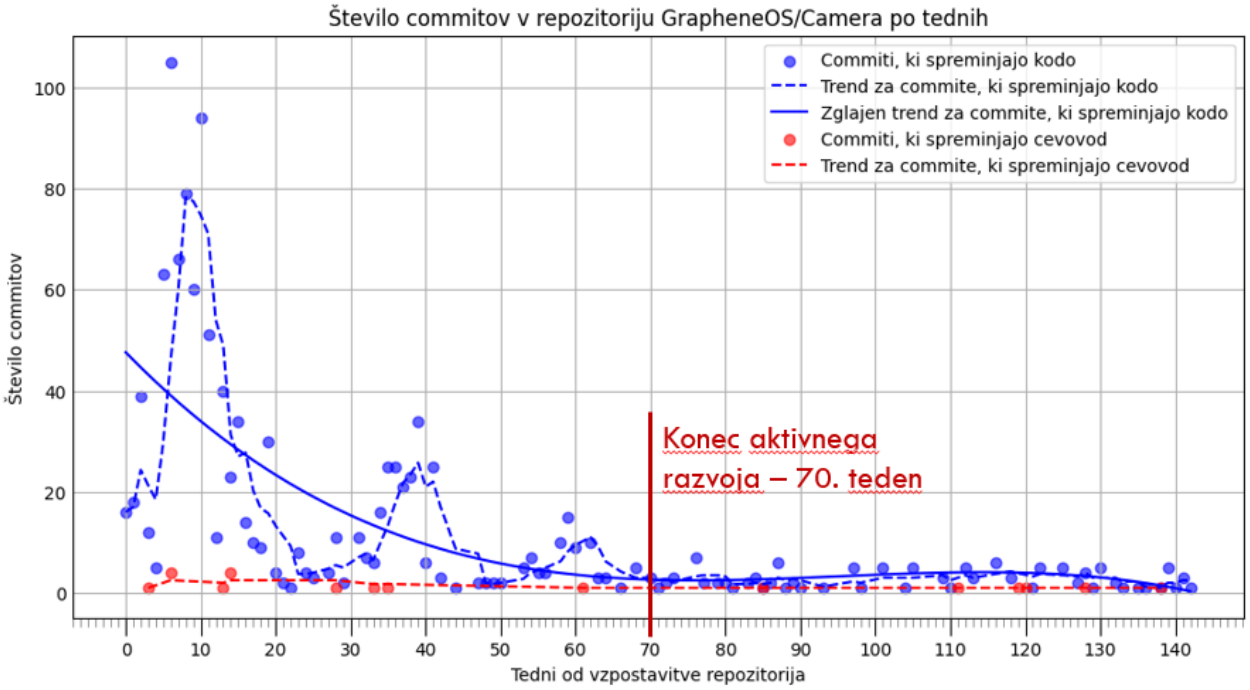
4.1. Pridobivanje in zajem podatkov

Iz platforme GitHub pridobili 1000 najbolj pribljubljenih (*ang.* starred) repozitorijev, napisanih pretežno v programskem jeziku java. Ustrezne repozitorije smo poiskali s pomočjo GitHub API, preko katerega smo pridobili tudi osnovne podatke o repozitoriju. Ali repozitorij vsebuje CI/CD cevovod smo preverili tako, da smo preko GitHub API prenesli vsebino mape .github/workflows in preverili ali vsebuje .yml datoteke.

Za vsak repozitorij smo med drugim pridobili: ime lastnika, ime repozitorija, prisotnost CI/CD cevovoda, privzeto vejo (*ang.* branch), datum vzpostavitve ter zadnje posodobitve repozitorija in velikost. Za vsak repozitorij, ki vsebuje CI/CD cevovod, smo pridobili dodatne informacije o številu in vsebini datotek, ki določajo cevovod ter imena in število opravil (ang. jobs). Na podlagi uporabljenih ukazov in imen opravil smo preverili, ali cevovod vključuje katerega izmed petih identificiranih najpogostejših gradnikov.

Za vsak repozitorij prenesli vse commite in preverili, če ti spreminjajo datoteke v mapi s cevovodi. Ta postopek je vključeval kloniranje repozitorija, pridobivanje podatkov o commitih, ter pregled spremenjenih datotek. Commiti so bili razdeljeni na tiste, ki spreminjajo cevovod in tiste, ki ga ne.

Da bi lahko cevovode projektov med seboj primerjali, smo poiskali repozitorije, kjer se je aktiven razvoj že zaključil. To smo storili tako, da smo vizualizirali število commitov v repozitoriju za vsak teden trajanja razvoja in ročno določili teden, ko je aktivnost upadla in je projekt prešel v fazo vzdrževanja. Primer vizualizacije aktivnosti repozitorija je prikazana na Slika .



Slika 1: Vizualizacija aktivnosti v repozitoriju.

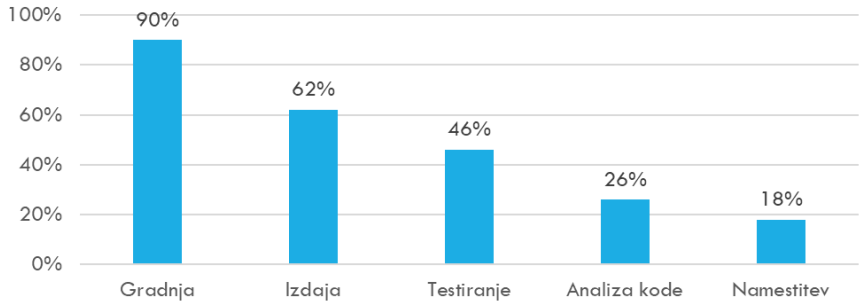
4.2. Pogostost uporabe

Ugotovili smo, da ima izmed 1000 analiziranih repozitorijev, cevovode vzpostavljenih 42% (*N*=415). Povprečen čas do vključitve cevovoda v repozitorij (t.j. čas do prve spremembe datotek v mapi .github/workflows) znaša 7 mesecev, mediana pa 4 mesece. Povprečna starost repozitorijev, ki vsebujejo cevovode, je ob zajemu znašala 38 mesecev, pri čemer je bil najmlajši repozitorij ustvarjen 2 meseca in pol, najstarejši pa 59 mesecev pred zajemom. V primerjavi z da Gião idr. [2], ki so ugotovili, da cevovode uporablja 33% repozitorijev na platformi GitHub, je

naša raziskava pokazala višjo stopnjo uporabe cevovodov. Razlog za takšno razliko lahko tiči v analiziranih projektih, saj so da Gião idr. v raziskavo vključili repozitorije, ustvarjene po letu 2012, medtem ko so bili v našo raziskavo vključeni le tisti, ki so nastali po letu 2019. V času od leta 2012 do 2019 so CI/CD cevovodi postali precej bolj priljubljeni, zato je tudi smiselno, da bodo prisotni v večjem številu repozitorijev. Precej daljši čas do vključitve CI/CD cevovodov v repozitorij bi lahko pripisali peščici dolgo trajajočih projektov, ki nikoli (ali zelo pozno) niso vključili CI/CD cevovoda in so zaradi manjšega števila analiziranih repozitorijev precej vplivali na povprečen čas. V skladu s tem je tudi mediana, ki se precej bolj približa povprečnemu času treh mesecev, ki so ga navedli Giao idr.

4.3. Najpogostejši gradniki

Za vsakega izmed najbolj pogostih gradnikov smo izračunali število projektov, ki ta gradnik vključuje. 90% projektov v cevovodih vključuje aktivnost gradnje, 62% vključuje izdajo in 46% vključuje testiranje. Statična analiza kode in namestitev sta precej bolj redki, saj ju vključuje le 26% oz. 18% projektov. Pogostost vključenosti posameznih gradnikov v cevovod je prikazana na Slika .



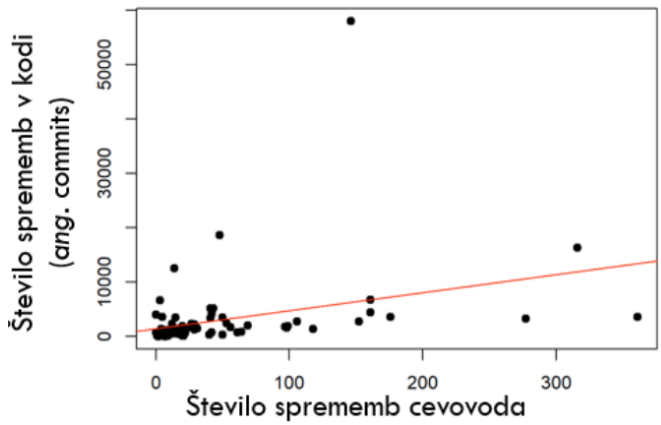
Slika 2: Odstotek projektov, ki v cevovodu vključujejo posamezne gradnike.

Analizirali smo tudi imena opravil znotraj cevovodov in ugotovili, da se skladajo z identificiranimi najpogostejšimi gradniki. Največ projektov je vsebovalo opravilo z imenom »build«, ki mu sledijo »test«, »release«, »publish«, »analyze« in »deploy«. Pojavilo se je tudi ime »job1«, kar kaže, da se pri gradnji cevovodov pogosto uporabijo predloge, ki jih razvijalci kasneje ne spremenijo. Ker se je izkazalo, da nekaj cevovodov ne vključuje nobenega izmed najpogostejših gradnikov, smo preverili katera opravila vsebujejo. V tovrstnih cevovodih so bila najbolj pogosta opravila, ki se ukvarjajo z upravljanjem poročil o napakah, prošenj za nove funkcionalnosti in nalog pri razvoju. Najbolj pogosta imena teh opravil so bila: »issue«, »stale«, »check«, »add« in »label«.

4.4. Spremembe cevovodov

Analiza je pokazala, da spremembe cevovodov predstavljajo približno 4% vseh commitov repozitorija in 8% spremenjenih vrstic v repozitoriju.

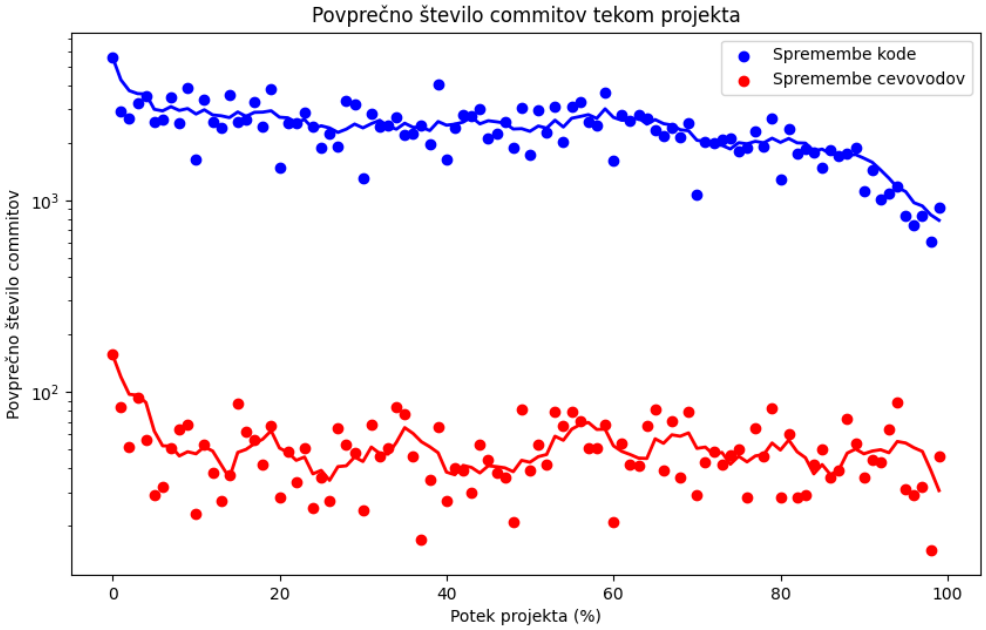
Zanimalo nas je, ali večje število sprememb v kodi v repozitoriju pomeni tudi več sprememb cevovoda, zato smo izvedli Pearsonov test korelacije med številom commitov, ki spreminjajo kodo in številom commitov, ki spreminjajo cevovod. Analiza je pokazala neznatno pozitivno a vseeno statistično značilno korelacijo med omenjenima spremenljivkama, $r(413)=0{,}16$, $p<0{,}01$. To pomeni, da s številom sprememb v kodi raste tudi število sprememb cevovodov. Korelacija je prikazana na Slika .



Slika 3: Korelacija skupnega števila sprememb v kodi in skupnega števila sprememb cevovodov.

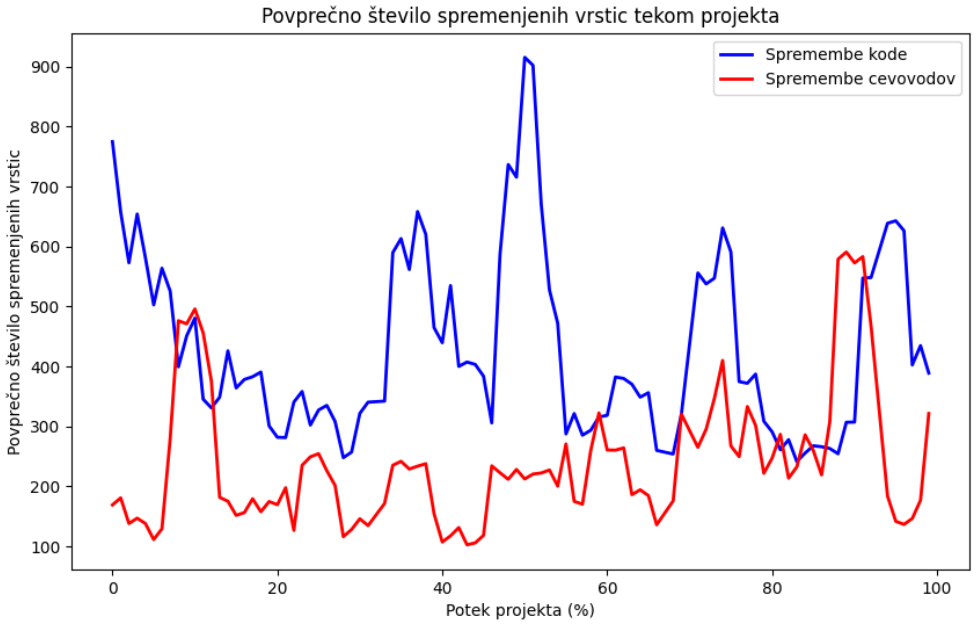
Zanimalo nas je tudi ali enako velja za posamezne mesece. Torej, če se število sprememb v kodi v poljubnem mesecu poveča, se bo povečalo tudi število sprememb cevovoda? Ponovno smo izvedli Pearsonov test korelacije, ki je ponovno pokazal statistično značilno a precej nižjo pozitivno korelacijo med spremenljivkama: $r(15519)=0,08$, $p<0,001$. To nakazuje, da spremembe cevovodov nekoliko sledijo spremembam v kodi, vendar pa se cevovodi spreminjajo z zamikom in ne hkrati s kodo.

Projekte, kjer je aktivni razvoj že zaključen smo po trajanju razdelili na procente in za vsak procent trajanja projekta izračunali povprečno število commitov, ki spreminjajo kodo in commitov, ki spreminjajo cevovodov. Ugotovili smo, da se v povprečju največje število commitov zgodi na začetku razvoja. Pri spremembah v kodi je proti koncu projekta viden rahel upad, ki ga pri spremembah cevovodov ni. Povprečno število commitov tekom projekta je vizualizirano na Slika .



Slika 4: Povprečno število commitov tekom projekta.

Poleg števila sprememb smo analizirali tudi število spremenjenih vrstic za vsak procent trajanja projekta in ugotovili, da število spremenjenih vrstic v cevovodu ne sledi številu spremenjenih vrstic v kodi. Največ spremenjenih vrstic pri cevovodih je takoj po začetku in malo pred koncem razvoja. Povprečno število spremenjenih vrstic tekom projekta je vizualizirano na Slika .



Slika 5: Povprečno število spremenjenih vrstic tekom projekta.

Večje število spremenjenih vrstic v obdobju 5%-15% trajanja projekta je najverjetneje vzpostavitev cevovoda, nihanja med razvojem bi lahko predstavljale manjše popravke, ki zagotavljajo da je cevovod usklajen s spremembami v kodi, obdobje 85%-95% pa je najverjetneje končno usklajevanje cevovoda s projektom in optimizacija za stabilnost in učinkovitost. Po vzpostavitvi se število sprememb na cevovodu s trajanjem projekta počasi viša, kar nakazuje, da se cevovodi s kodo usklajujejo tudi vmes, vendar v precej manjši meri.

5 Diskusija

S pomočjo analize CI/CD cevovodov smo pridobili pomemben vpogled v njihov njihove značilnosti in spremembe skozi čas. Ugotovitve kažejo, da so CI/CD cevovodi prisotni v 42% analiziranih GitHub repozitorijev, povprečen čas do njihove vključitve pa je sedem mesecev. To potrjuje, da so CI/CD cevovodi postali pomemben del sodobnega razvoja programske opreme, vendar še vedno niso univerzalno sprejeti v vseh projektih. Struktura CI/CD cevovodov je prav tako razkrila pomembne trende. Gradnja je vključena v skoraj vseh cevovodih, kar poudarja njen temeljni pomen za avtomatizacijo razvoja. Fazi testiranja in izdaje sta vključeni v približno polovici cevovodov, vključevanje statične analize kode in namestitve pa je najmanj pogosto. Rezultati kažejo, da je ozaveščenost o pomenu gradnje in testiranja visoka, prav tako pa je ti fazi precej preprosto vzpostaviti. Za statično analizo kode in namestitev je potrebna integracija z zunanjimi orodji, ki so pogosto plačljiva, kar je lahko razlog za nizek procent projektov, ki ta gradnika vključujejo. Ugotovili smo tudi, da spremembe v CI/CD cevovodih predstavljajo približno 4% vseh commitov in 8% vseh spremenjenih vrstic v repozitoriju. To kaže na to, da čeprav so cevovodi pomemben del razvoja, predstavljajo le majhen delež celotnih sprememb v projektu. Korelacija med številom sprememb v kodi in spremembami v cevovodih je pokazala neznatno pozitivno, a statistično značilno povezavo, ki postane šibkejša če korelacijo računamo za določeno časovno obdobje. To nakazuje, da se spremembe v cevovodih običajno dogajajo z zamikom po spremembah v kodi. Ta ugotovitev poudarja potrebo po usklajevanju med razvojem kode in cevovoda, da se zagotovi učinkovita avtomatizacija in integracija. Analiza sprememb cevovodov tekom življenjskega cikla projektov je razkrila, da se cevovodi najbolj spreminjajo na začetku in koncu razvoja. Na začetku se spremembe najverjetneje nanašajo na vzpostavitev in zgodnje prilagoditve cevovoda, kar je kritično za postavitev temeljev za nadaljnji razvoj. Spremembe proti koncu razvoja pa lahko odražajo končno

usklajevanje in optimizacijo cevovoda za stabilnost in učinkovitost. Vmesne faze kažejo manjše, a postopno naraščajoče spremembe, kar nakazuje na stalno prilagajanje cevovoda.

Na podlagi teh ugotovitev lahko podamo nekaj napotkov:

- Vključitev CI/CD cevovodov že v zgodnjih fazah razvoja lahko pomaga izkoristiti prednosti zgodnje avtomatizacije in zaznavanja težav. Vključitev je priporočljiva v prvih nekaj mesecih razvoja oz. vsaj v sedmih mesecih.
- Prilagoditve cevovoda je priporočljivo izvajati sproti, da zagotovite usklajenost s spremembami v kodi in preprečite večje težave v kasnejših fazah projekta. Prav tako lahko s tem zmanjšamo število zaključnih uskladitev in prilagoditev.
- Poleg gradnikov gradnje in izdaje je potrebno več pozornosti nameniti tudi testiranju in statični analizi kode, saj lahko s tem pomagamo zagotavljati visok nivo kakovosti kode.

6 Zaključek

Uporaba CI/CD cevovodov je postala ključna za sodoben razvoj programske opreme, saj ti omogočajo hitrejšo in bolj zanesljivo integracijo, dostavo ter namestitev informacijskih rešitev. V sklopu našega dela smo se osredotočili na analizo življenjskega cikla CI/CD cevovodov, pogostost njihove uporabe, strukturo in spremembe skozi čas. Za ta namen smo iz platforme GitHub pridobili 1000 repozitorijev, napisanih pretežno v programskem jeziku java, in analizirali njihove CI/CD cevovode.

Rezultati so pokazali, da so CI/CD cevovodi prisotni v manj kot polovici analiziranih GitHub repozitorijev, pri čemer povprečen čas do vključitve znaša sedem mesecev. Večina cevovodov vključuje fazo gradnje, fazi testiranja in izdaje vključuje približno polovica repozitorijev, fazo namestitve pa manj kot petina. Spremembe cevovodov v povprečju le majhen delež vseh sprememb v repozitorijih, pri čemer obstaja statistično značilna korelacija med številom sprememb v kodi in cevovodih. Naša analiza je pokazala, da se število sprememb v cevovodih povečuje v zgodnjih in poznih fazah razvoja projekta, medtem ko je v vmesnem delu aktivnost manjša, a počasi raste. Večje število sprememb na začetku razvoja verjetno predstavlja vzpostavitev in zgodnje prilagoditve cevovoda, medtem ko spremembe proti koncu razvoja odražajo končno usklajevanje in optimizacijo cevovoda za stabilnost in učinkovitost.

V prihodnjih raziskavah se bomo osredotočili na:

- Raziskovanje sprememb cevovodov glede na značilnosti projekta, kot so velikost projekta, število sodelujočih razvijalcev in tematika projekta.
- Ugotavljanje vrste sprememb cevovodov, ne le kdaj se cevovod spremeni, ampak kateri deli cevovoda se spreminja.
- Identifikacijo najbolj pogostih sprememb v CI/CD cevovodih.

Te ugotovitve predstavljajo pomembno izhodišče za nadaljnje raziskave o uporabi ter značilnostih CI/CD cevovodov ter prispevajo k razumevanju njihove evolucije tekom razvoja. Razumevanje življenjskega cikla CI/CD cevovodov je ključno za boljše načrtovanje, vzpostavitev in vzdrževanje CI/CD cevovodov, kar posledično pomaga izboljšati celoten proces razvoja programske opreme.

7 Literatura

[1] P. Rostami Mazrae, T. Mens, M. Golzadeh, in A. Decan, „On the usage, co-usage and migration of CI/CD tools: A qualitative analysis“, *Empir. Softw. Eng.*, let. 28, št. 2, str. 52, mar. 2023, doi: 10.1007/s10664-022-10285-5.

[2] H. da Gião, A. Flores, R. Pereira, in J. Cunha, „Chronicles of CI/CD: A Deep Dive into its Usage Over Time“, 27. februar 2024, *arXiv*: arXiv:2402.17588. doi: 10.48550/arXiv.2402.17588.

[3] M. Shahin, M. Ali Babar, in L. Zhu, „Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices“, *IEEE Access*, let. 5, str. 3909–3943, 2017, doi: 10.1109/ACCESS.2017.2685629.

[4] F. Zampetti, S. Geremia, G. Bavota, in M. Di Penta, „CI/CD Pipelines Evolution and Restructuring: A Qualitative and Quantitative Study“, v *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, sep. 2021, str. 471–482. doi: 10.1109/ICSME52107.2021.00048.

[5] N. Hroncova in P. Dakic, „Research Study on the Use of CI/CD Among Slovak Students“, v *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*, sep. 2022, str. 458–461. doi: 10.1109/ACIT54803.2022.9913113.

[6] „What Are CI/CD And The CI/CD Pipeline? | IBM“. Pridobljeno: 29. julij 2024. [Na spletu]. Dostopno na: https://www.ibm.com/think/topics/ci-cd-pipeline

[7] „What is a CI/CD pipeline?“ Pridobljeno: 9. februar 2024. [Na spletu]. Dostopno na: https://www.redhat.com/en/topics/devops/what-cicd-pipeline

[8] „CI/CD pipelines | GitLab“. Pridobljeno: 1. julij 2024. [Na spletu]. Dostopno na: https://docs.gitlab.com/ee/ci/pipelines/

[9] D. HQ, „A Brief History of CI/CD Tooling“, Medium. Pridobljeno: 1. julij 2024. [Na spletu]. Dostopno na: https://medium.com/@DiggerHQ/a-brief-history-of-ci-cd-tooling-5a67c2638f3a

[10] „What is a CI/CD pipeline?“, CircleCI. Pridobljeno: 1. julij 2024. [Na spletu]. Dostopno na: https://circleci.com/blog/what-is-a-ci-cd-pipeline/