|  |  |  |
| --- | --- | --- |
| Screenshot_6 | Univerzitet u Novom Sadu  Tehnički fakultet »MihajloPupin«  Zrenjanin | LOGOpupin |

***-Diplomski rad-***

Tema: Refaktorisanje PHP koda u cilju unapredjenja održavanja

Refactoring PHP code in aim to improve maintainability

Mentor: Doc.dr Ljubica Kazi Student: Uglješa Bajić

Smer: Informacione tehnologije

Broj indeksa: IT 62/15

Zrenjanin, 2023.

# **1. UVOD**

Veb-aplikacija je aplikacija kojoj se pristupa od strane korisnika preko mreže kao sto je internet ili intranet. Termin takođe označava i primenu kompjuterskog softvera koji je kodiran u veb-pretraživaču koji podržava programske jezike (kao npr. Javascript, PHP, u kombinaciji sa jezikom koji služi za obeležavanje kao što je HTML, itd.) i oslanja se na zajedničkom veb-pretraživaču da donese izvršnu aplikaciju.

Veb-aplikacije su popularne zbog sveprisutnosti veb-pretraživača. Sposobnost da ažurira i održava veb-aplikaciju bez distribucije i instaliranja softvera na hiljade potencijalnih klijentskih racunara je ključni razlog za njihovu popularnost. Uobičajene veb-aplikacije uključuju slanje, internet trgovinu, kao i mnoge druge funkcije.

Cilj diplomskog rada je refaktorisanje, tj. izmena php koda, a da se pritom u samoj veb aplikaciji ništa suštinski ne menja. Sam rad se svodi na prikaz dva primera koda, jednog koji je „loše“ napisan i drugog koji je funkcionalniji, pregledniji, laksi za održavanje, itd.

Diplomski rad se sastoji iz teorijskog istraživanja problematike razvoja veb aplikacija kao i višeslojne arhitekture. Praktičan primer koji ilustruje primenu navedenih koncepata realizovan je na primeru aplikacije “Evidencija filmova“.

# **2. TEORIJSKO - TEHNOLOŠKE OSNOVE**

## **2.1. Refaktorisanje programskog koda**

Refaktorisanje je postupak poboljšanja tako da se radi interno rekonstruisanje koda bez izmene ili dodavanja funkcionalnosti. To mogu biti manje izmene kao što je promena naziva lokalne promenljive, pa sve do krupnijih izmena kao što je kreiranje novih klasa ili metoda.

Programeti ovaj korak preduzimaju uz dogovore sa menadzerima ili kompanijama, nekad i samostalno u toku ili po završetku implementacije tražene funkcionalnosti.

Nije redak slučaj da se nakon završetka kompletnog projekta pristupi refaktorisanju, najčešće zajedno sa procesom dodavanja nove funkcionalnosti jer postojeća organizacija koda ne omogućava lako dodavanje nove.

Refaktorisanje koda je preporučljjivo u odredjenim slučajevima, a neki od njih su:

1. Velika klasa (neko bi rekao ’’kraljevska klasa’’) – kada klaa radi previše stvari, sto mežemo zaključiti na osnovu naziva metode ili ako klasa ima puno lokalnih instancnih promenljivih. Efikasan način rešavanja ovog problema jeste da uradimo reviziju koda i uočimo dupliranja implementacija i i kreiranjem manjih metoda pregrupišemo implementacije i kreiranje izvedene klase.
2. Dugačke metode – lakše je održavati program ukoliko su metode manje, ali se u preksi često dešava da pravimo dugačke metode, koje rade više stvari nego što bi trebalo. Ove metode ne samo da su teže za održavanje, već su teže i za tumačenje. U takvim situacijama ćemo najpre sve parametre metode zameniti sa objektom i njega prosleđivati. Deo koda za novu metodu identifikovaćemo putem komentara. Oni najčešće predstavljaju celine u metodi i mogu se prebaciti u novu metodu.
3. Velika lista parametara – tradicionalno je učeno da sve sto treba metodi prosledimo putem parametara. Pojavom objekata, te metode sad mogu imati veliku listu parametara. Ovakve liste parametara su teže za razumevanje, a posebno je otežavajuća okolnost kada ih treba pozvati ili dodati/oduzeti. U praksi ove metode možemo zameniti objektima, koji će zadržati podatke iz parametara što će na duže staze svakako olakšati održavanje. **[1]**

## **2.2. Održavanje softvera**

Održavanje softvera obuhvata sve modifikacije na softveru nakon njegove isporuke u javnost, sve do trenutka njegovog povlačenja iz upotrebe. Pored namere da se detaljnije predstavi sam termin održavanja softvera, cilj rada jeste i da čitaoci spoznaju značaj ovog dela razvoja softvera, da dobiju uvid u to koji alati im mogu pomoći da olakšaju sam proces, ali i da se upoznaju sa nekim merama koje odreduju kvalitet softvera i omogućavaju da se čitav posao olakša. **[2]**

Održavanje softvera se može definisati kao disciplina koja se bavi promenama nekog softverskog sistema nakon njegovog puštanja u rad. Intuitivno isprva možemo pomisliti da se tu radi prevashodno o promenama korektivne prirode (ispravljanje bagova). Medutim, okolnosti u kojima se koristi konkretan softver neprestano evoluiraju, razvijaju se postojeće softverske i hardverske tehnologije i pojavljuju nove. Vremenom se menja poslovna dinamika, pa i trendovi u samom vizuelnom dizajnu grafičkih interfejsa. Sve ove promene nalažu da postojeći softverski sistem ostaje u koraku sa svetom, zadovoljavajući zahteve za boljim operativnim performansama i novim funkcionalnim mogućnostima koje postavljaju korisnici. **[3]**

Medunarodni standard ISO/IEC 14764 svaki zahtev za modifikacijom softvera nakon njegove isporuke klasifikuje ili u korektivnu kategoriju, ili u kategoriju unapredenja (eng. enhancement). Zahtev se na dalje identifikuje kao korektivan, preventivan, adaptivan ili perfektivan. **[4]**

**Korekcije** su ispravke trenutno prisutnih grešaka, dok se preventivne modifikacije tiću promena na mestima u dizajnu koja bi kasnije mogla postati izvor operativnih greški pri korišćenju. **[4]**

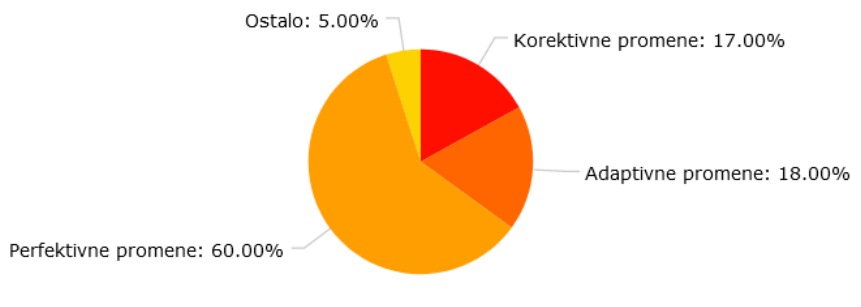
**Adaptivne** promene se bave održavanjem softvera usled promena okruženja u kojima se koristi, bilo gledano u poslovnom ili u softverskohardverskom kontekstu. **[4]**

**Perfektivne** modifikacije u jednom smislu mogu podsećati na preventivne, s tim da svrha promena nije izbegavanje grešaka (kao npr. pad sistema), već se žele izbeći neuspesi poslovne prirode. Dodatno, u ovu grupu spadaju zahtevi za poboljšanje radnih performansi softvera, unapredenje dokumentacije i dodavanje novih funkcionalnosti. Poboljšavanje održivosti softvera je takode tematika ovih promena, gde se održivost (eng. maintainability) opisuje kao fleksibilnost sa kojom se postojeći kod može održavati. **[4]**

**Preventivno** održavanje softvera ima za cilj sprečavanje eventualnih problema.



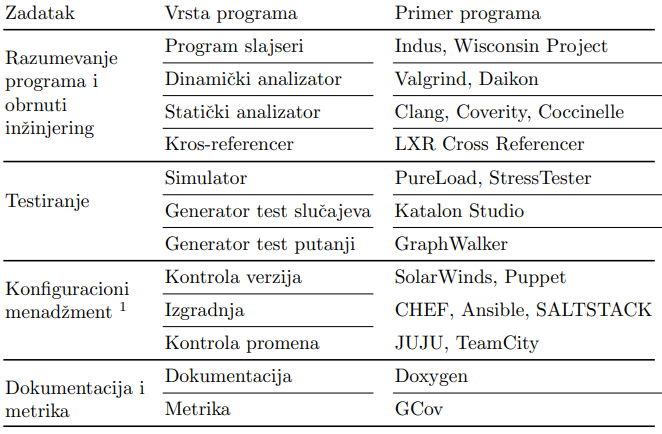
*Slika 1. Tipovi održavanja softvera* ***[5]***



*Slika 2. Podela troškova održavanja prema kategorijama* ***[5]***

## **2.2.1. Alati koji olakšavaju održavanje softvera**

Kada je održavanje softvera u pitanju, ne postoji magični alat koji bi rešio probleme, što je i oćekivano iz obima problema koje održavanje softvera rešava, ali postoje alati koji automatski odraduju neki zadatak koji je vezan za željenu promenu. Neke generalne kategorije zadataka koji mogu da se automatizuju i time povećaju produktivnost i olakšaju obiman posao održavanja su: razumevanje programa, obrnuti inženjering, debagovanje, testiranje, održavanje dokumentacije i konfiguracioni menadžment. **[6]**

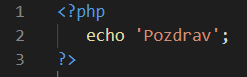


*Slika 3. Pregled alata za date zadatke u održavanju* ***[6]***

## **2.3. PHP programski jezik**

PHP (Hypertext Preprocessor) je jedan od najpopularnijih jezika koji se koriste za razvoj veb aplikacija. On omogućava programeru da brzo razvije dobro formirane programe bez greške, koristeći tehnike proceduralnog i objektno-orijentisanog programiranja. Obezbeđuje mogućnost upotrebe mnogih postojećih biblioteka koda koje su uključene u osnovnu instalaciju ili mogu da se instaliraju unutar PHP okruženja. To korisniku pruža mogućnost da izvrši određeni zadatak na više načina.

PHP kod je ugradjen sa početnim i završnim tagom ’’<?php’’ i ’’?>’’ i pomoću njih ubacuje kod u HTML.



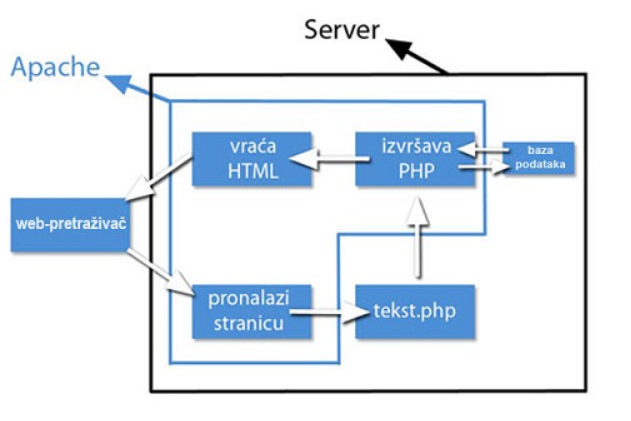
*Slika 4. Isečak PHP koda*

PHP obezbeđuje veću fleksibilnost od mnogih drugih jezika. Lakoća dodavanja biblioteka koda u okruženje je jedna od glavnih prednosti i ono što PHP čini popularnim. **[7]**

PHP je skriptni jezik dizajniran da popuni jaz između SSI i Perl, koji su u velikoj meri namenjeni veb okruženju. PHP je prilično dobio popilarnost u poslednje vreme, i jedan je od prethodnika otvorenog koda. Njegova popularnost potiče od sintakse nalik C, njegove brzine i njene sposobnosti jednostavnosti. Trenutna verzija PHP-a je PHP 7 koji je mnogo brži i napredniji od prehnodnih verzija.

Prednosti PHP programskog jezika:

* PHP se može pokrenuti na svim platformama
* Kompatibilan je sa skoro svim serverima
* Može da radi sa svim bazama podataka
* Besplatan je (open source)
* Lak je za učenje i korišćenje

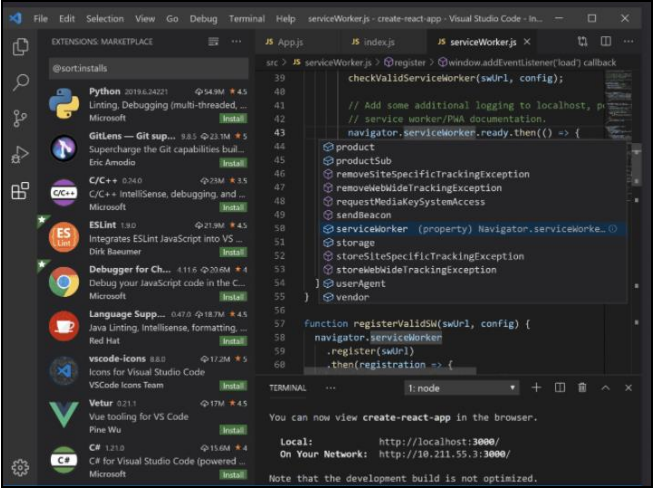


*Slika 5. Prikaz rada PHP aplikacije* ***[7]***

# **3. OPIS KORISCENIH ALATA U REALIZACIJI PRIMERA**

## **3.1. Visual Studio Code**

Visual Studio Code je besplatan program za uređivanje izvornog koda koji se pokreće na kompjuteru i dostupan je za Windows, mcOS i Linux sisteme. Ima ugrađenu podršku za JavaScript, TypeScript i Node.js kao i velikih izbor dodatnih besplatnih ekstenzija za druge jezike kao što su C++, C#, Java, Pyton, PHP. Lak je za upotrebu i programeru daje široki spektar mogućnosti. **[8]**



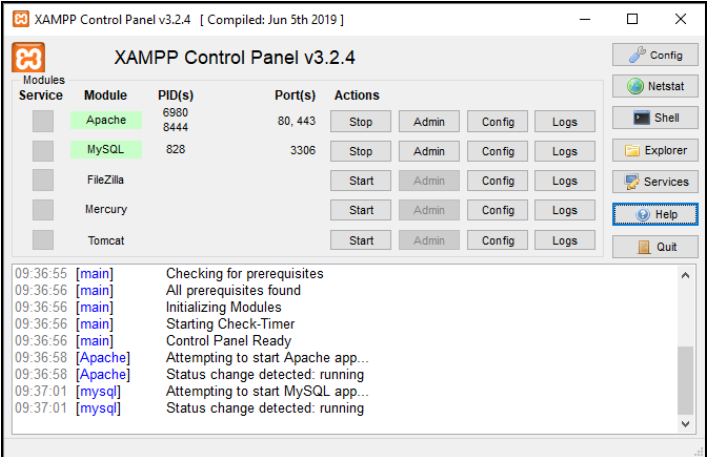
*Slika 6. Korisnički interfejs Visual Studio Code* ***[8]***

## **3.2. XAMPP**

XAMPP je besplatni i open-source cross-platformski veb paket rešenja veb servera koji je razvio Apache Friends, koji se sastoji od Apache HTTP servera, MariaDB baze podataka i tumači za skripte napisane na programskim jezicima PHP i Perl. Budući da većina stvarnih implementacija veb servera koristi iste komponente kao XAMPP, omogućava prelazak sa lokalnog testnog servera na live server. Jednostavnost primene XAMPP-a znači da WAMP ili LAMP stek može da se instalira brzo i jednostavno na operativni sistem od strane programera. Prednost su brojne uobičajene dodatne aplikacije kao što su Wordpress i Joomla! takođe se može instalirati sa sličnom lakoćom pomoću Bitnami-ja.

Da biste pristupili tom folderu i počeli sa podešavanjem vašeg „servera”, treba da u vašem Internet pretraživaču kao adresu upišete <http://localhost> ili „http://127.0.0.1” i bićete prebačeni na stranicu za izbor između devet jezika (među kojima nema našeg) kontrolnog panela vašeg servera.

U kontrolnom panelu ima pregršt stvarčica koje bi svaki Web dizajner poželeo. Odavde možete da proverite koji je od servisa (MySQL, PHP, HTTPS, CGI, SSI, SMTP i FTP) uključen. **[9]**



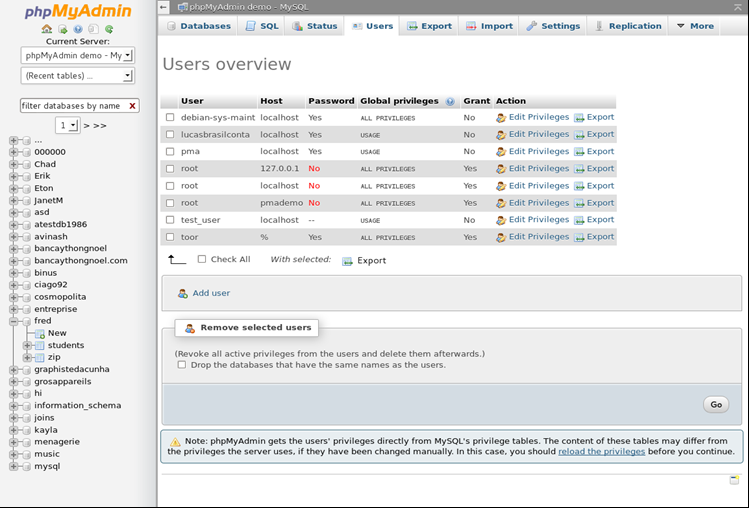
*Slika 7. Interfejs XAMPP Control Panel-a*

## **3.3. MySQL**

MySQL je sistem za upravljanje relacijskim bazama otvorenog koda (RDBMS). MySQL je besplatni i open-source softver pod uslovima GNU General Public License, a dostupan je i pod raznim vlasničkim licencama. MySQL je bio u vlasništvu i sponzorisanju švedske kompanije MySQL AB, koju je kupila kompanija Sun Microsistems (danas Oracle korporacija). 2010. godine, kada je Oracle kupio Sun, Videnius je forsirao MySQL projekat otvorenog koda i stvorio MariaDB.

MySQL je komponenta skupa softverskog softvera za veb aplikacije (i drugih), što je akronim za Linux, Apache, MySQL, Perl / PHP / Python. MySQL koriste mnoge veb aplikacije koje pokreću baze podataka, uključujući Drupal, Joomla, phpBB i WordPress. MySQL se takođe koristi na mnogim popularnim veb lokacijama, uključujući Facebook, Flickr, MediaWiki, Twitter i YouTube.

MySQL je poznat po tome da može da upravlja bazama podataka sa 60.000 tabela i nekoliko milijardi redova. MySQL može da radi sa tabelama od čak nekoliko hiljada terabajta na pojedinim operativnim sistemima, a inače podržava dobrih 4 GB. MySQL, među ostalima, koriste NASA i uprava za popis SAD. **[10]**



*Slika 8. Interfejs phpMyAdmin veb aplikacije* ***[11]***

**[1] Definicija i cilj refaktorisanja:** <https://www.ucim-programiranje.com/2012/11/refaktoring-kad-kako-da-li/>

**[2] Održavanje softvera:** <http://poincare.matf.bg.ac.rs/~milena/msnr/2021/01_OdrzavanjeSoftvera_MarkovicRadisicZivanovic.pdf>

**[3] Koncept održavanja softvera:** Penny Grubb and Armstrong A Takang. Software maintenance: concepts and practice. World Scientific, 2003.

**[4] Tipovi održavanja softvera:** <https://www.iso.org/obp/ui/#iso:std:iso-iec:14764:ed-2:v1:en>

**[5] Slika - tipovi održavanja softvera:** <http://www.tfzr.uns.ac.rs/Content/files/1/Cas%208%20Odrzavanje%20softvera.pdf>

**[6] Alati za održavanje softvera:** <http://poincare.matf.bg.ac.rs/~milena/msnr/2021/01_OdrzavanjeSoftvera_MarkovicRadisicZivanovic.pdf>

**[7] PHP programski jezik:** <https://www.its.edu.rs/2013/10/03/php-programski-jezik/>, <http://www.tizag.com/phpT/ifelse.php>

**[8]** **Informacije o programu Visual Studio Code i slika primera istog:** https://code.visualstudio.com/

**[9] Opšti opis XAMPP aplikacije:** https://www.sk.rs/2005/08/skpd11.html

**[10] Opšti opis MySQL:** <https://cet.rs/wp-content/uploads/2018/09/PHP-i-MySQL-za-sajt.pdf>

**[11] Primer intefejsa phpMyAdmin veb aplikacije:** <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-18-04>

~~~~~~~~~~~~

1. UVOD

2. TEORIJSKO-TEHNOLOSKE OSNOVE

2.1. Refaktorisanje programskog koda (definicija, karakteristike, tipovi)

2.2. Odrzavanje softvera

2.3. PHP programiranje

3. POSTOJECA RESENJA REFAKTORISANJA U RAZVOJNIM OKRUZENJIMA

3 - 5 razvojnih alata koji imaju u sebi opcije za refaktorisanje

4. OPIS KORISCENIH ALATA U REALIZACIJI PRIMERA

Visual studio code

XAMPP

MYSQL

5. REALIZOVAN PRIMER - stavke kao kod Zubera, softver uzeti iz kolokvijuma koji je pogresan i uraditi isto ali kako treba

6. REZULTATI REFAKTORISANJA KODA

6.1. Pogresno koncipiran programski kod

6.2. Refaktorisan programski kod radi boljeg odrzavanja

7. ZAKLJUCAK

8. LITERATURA