

Proqram Təminatının Versiyalarının Vizuallaşdırılması Problemləri Haqqında

Məkrufə Hacırahimova¹, Mərziyə İsmayılova²
İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan
¹makrufa@science.az, ²imarziya@gmail.com

Xülasə— Məqalə proqram təminatının vizuallaşdırılması məsələsinə həsr olunmuşdur. İşdə proqram təminatı, proqram təminatının vizuallaşdırılması və təkamülü problemləri araşdırılmış və bəzi vizuallaşdırma alətlərinə baxılmışdır.

Açar sözlər— proqram təminatı, vizuallaşdırma, proqram təminatının vizuallaşdırılması, proqram təminatının vizuallaşdırılmasının təkamülü

I. GİRİŞ

Bu günkü proqram təminatı (PT) sistemləri olduqca böyük və mürəkkəbdir, onların hazırlanması və istismarı bir qayda olaraq, çoxlu sayda insanların birgə əməkdaşlığını tələb edir. Bu işə PT-nin proqramlaşdırılması, anlaşılması və dəyişdirilməsi məsələlərində, xüsusilə də müxtəlif sahənin mütəxəssislərinin yazdığı kod ilə işləyən zaman çətinlik yaradır. Çətinliyin aradan qaldırılmasının əsas yollarından biri PT-nin vizuallaşdırılmasıdır (*Software visualization, SV*). Effektiv vizuallaşdırma yuxarıda qeyd olunan məsələlərə sərf olunan səyləri əhəmiyyətli dərəcədə azalda bilər. SV-nin modelləşdirmə, yoxlama, sazlama və xidmət kimi fəaliyyətini dəstəkləyən çoxlu texniki üsullar və alətlər mövcuddur. SV PT-nin işlənilib hazırlanması, proqramların paralel icrası, obyekt yönümlü kodların statistik və dinamik vizuallaşdırılması, nasazlığın diaqnostikası, sazlama kimi müxtəlif sahələrə tətbiq edilir [1]. Bütün bunlara baxmayaraq SV-nin bir sıra problemləri onların yenidən tədqiq olunmasına ehtiyac yaradır. SV böyük layihələr üçün nəzərdə tutulduqda bir neçə istifadəçinin birgə işini dəstəkləyən vasitələr yaradılmalıdır. SV texniki yönümlü proqramçılardan başlayaraq biznes dünyasını idarə edən şəxslərə qədər geniş istifadəçi dairəsinin sorğularına və tələblərinə cavab verməlidir. Bütün bu məhdudiyyətlər ciddi problemlər yaradır və SV üçün yeni tədqiqatların əsasını qoyur.

Son zamanlar nüfuzlu beynəlxalq təşkilatlar, elmi qurumlar tərəfindən SV problemlərinə həsr olunmuş konfranslar, simpoziumlar, seminarlar, forumlar keçirilir. Qeyd etmək lazımdır ki, vizuallaşdırma üzrə ilk konfrans 1990-cı ildə IEEE (Institute of Electrical and Electronic Engineers) tərəfindən keçirilmişdir [2].

II. PROQRAM TƏMİNATININ PROBLEMLƏRİ

Bu gün proqram təminatı cəmiyyətin bütün aspektlərinə nüfuz etmişdir. Bu nüfuz etmə son 20 ildə daha sürətlə gerçəkləşmiş və artmaqda davam edir. C++ proqramlaşdırma dilinin yaradıcısının qeyd etdiyi kimi “Bizim sivilizasiya proqram təminatı ilə işləyir”. Ancaq PT sənayesi iki problemlə qarşılaşır. PT sənayesinin ilk problemi PT-nin mürəkkəbliyi ilə əlaqədardır. İyirmi il əvvəl orta ölçülü PT-nin kod sətirləri bir

neçə min, yaxud on minlərlə sətirdən ibarət idi. Hazırda orta ölçülü PT-nin kod sətirləri on milyonlara bərabərdir. Hətta nisbətən sadə proqram əlavəsi, məsələn, Microsoft Windows Paint proqramı on minlərlə kod sətirdən ibarətdir. Həmin proqram əlavəsi uzun illər ərzində onlarla proqramçı tərəfindən işlənilib hazırlanmışdır. Bu rəqəmlər bank, telekommunikasiya, yaxud sənaye əlavələri üçün olduqca böyükdür. PT kodu müxtəlif üsullarla: fayl iyerarxiyası şəklində; komponentlərin, funksiyaların və yaxud paketlərin şəbəkəsi kimi; layihələndirmə şablonları və ya aspektləri dəsti kimi strukturlaşdırıla bilər [3-5]. PT-nin başa düşülməsi üçün bir iyerarxiya kifayət deyildir. İyerarxiyalar arasındakı münasibətlər isə çox mürəkkəbdir. Əgər mənbə koduna dinamik verilənlər daxil edilərsə, PT daha da çətin anlaşılır.

PT sənayesinin ikinci problemi PT-nin daima inkişaf etməsi və dəyişikliklərə məruz qalması ilə əlaqədardır. PT-nin inkişafına bir sıra amillər - tələblərin, texnologiyaların, platformaların dəyişməsi, həmçinin səhvlərin aradan qaldırılması üçün dəyişikliklər və funksionallığın yaxşılaşdırılması təsir edir. PT inkişaf etdikcə onun mürəkkəbliyi artır. Bu hadisə PT-nin inkişaf qanunu və ya PT-nin entropiyasının artması ilə xarakterizə olunur [6, 7]. Bu problemin həll yollarından biri proqram sistemlərinin yenidən – sıfırdan başlayaraq yazılmasıdır, ancaq bunun üçün böyük maliyyə xərcləri tələb olunur. Ona görə də əksər proqram layihələri mövcud infrastrukturuna saxlamağa cəhd edir və yalnız yeni tələbləri ödəmək üçün dəyişdirilir. Nəticədə kodun böyük miqdarı dəstəklənməli və hər il yenilənməlidir (məsələn, köhnə sistemlər). [8]-də 2005-ci ildə təşkil olunmuş sənaye araşdırılmasında xidmətdə olan kodun sətirlərinin sayının ümumi miqdarı qiymətləndirilmiş və onun təxminən 800 milyarda yaxın olduğu qeyd edilmişdir. Kodun sətirlərinin otuz milyardının yeni olmasının və ya hər il on beş milyon mühəndis-proqramçı tərəfindən dəyişdirilməsinin lazım olduğu bildirilmişdir. Bu da böyük sayda resurs tələb edir. Sənaye araşdırmalarının qiymətləndirilməsinə uyğun olaraq proqram təminatına çəkilən ümumi xərcin 80-90 %-ni texniki xidmətə çəkilən xərc, layihənin əməkdaşlarının ümumi sayının 60-80%-ni isə xidmət göstərən heyət təşkil edir [9, 10]. [11, 12]-də PT-nin anlaşılması üzrə qiymətləndirmə aparılmışdır. Araşdırmalar göstərir ki, PT-nin işlənilib hazırlanması sahəsindəki cəhdlərin yarısından çoxu anlaşılma fəaliyyətinə yönəlmişdir.

PT sənayesinin yuxarıda göstərilən məsələlərinin həllində çoxlu üsullar mövcuddur və onlar aşağıda göstərilən iki yanaşmadan birinə əsaslanmalıdır [13]:

• **qabaqlayıcı** (*preventive*) yanaşma. Bu yanaşma işlənmə prosesində qəbul edilmiş sistemin strukturunun, həll keyfiyyətinin yaxşılaşdırılmasını yüksəltməyə cəhd edir;

• **təsdiqləyici** (*assertive*) yanaşma. Bu yanaşma düzəlişlərə, uyğunlaşmaya və nöqsansız xidmət fəaliyyətinə yönəldilmişdir. Yanaşma proqram və proseslərin başa düşülməsi və nasazlıqların lokallaşdırılması alətləri ilə dəstəklənir.

III. PROQRAM TƏMİNATININ VİZUALLAŞDIRILMASI VƏ TƏKAMÜLÜ

A. Proqram təminatının vizuallaşdırılması (*Software visualization*)

Vizuallaşdırma latın dilində “visualis” sözündən olub görmə, müşahidə mənasını verir və rəqəmsal informasiyanın və ya fiziki hadisələrin münasib şəkildə müşahidə və analizinin nəticələrinin görüntüsünü təqdim etmək üsuludur. Vizuallaşdırma haqqında çoxlu təriflər mövcuddur. [14]-də vizuallaşdırmanın ən çox yayılmış təriflərindən biri verilmişdir: “Vizuallaşdırma – informasiyanı vizual formaya çevirən və istifadəçilərə informasiyanı müşahidə etməyə imkan verən prosesdir. Alınmış vizual görüntü alim və ya mühəndislərə verilənlərdə gizlənən, lakin verilənlərin analizi və tədqiqi üçün lazım olan vizual xüsusiyyətləri başa düşməyə imkan verir”.

Müasir araşdırmalar və təcrübələrdən irəli gələn vizuallaşdırmanın əsas üstünlükləri aşağıdakı kimi ümumiləşdirilmişdir [15]:

- vizuallaşdırma böyük həcmli verilənlərin başa düşülməsi imkanını təmin edir;
- vizuallaşdırma gözlənilmədən meydana çıxan xüsusiyyətləri anlamağa imkan verir;
- vizuallaşdırma kiçik və irimiqyaslı obyektlərin anlaşılmasını asanlaşdırır;
- vizuallaşdırma hipotezlərin yaradılmasına imkan verir.

Vizuallaşdırma bir fənn kimi 1980-ci ildə meydana çıxmışdır [16]. Verilənlərin vizuallaşdırılması – verilənləri analiz etmək məqsədilə verilənlər kolleksiyasının vizuallaşdırılmasının prinsip və metodlarını öyrənir. Verilənlərin vizuallaşdırılması bu gün iki əsas tədqiqat sahəsinə malikdir: elmi vizuallaşdırma və informasiya vizuallaşdırması. Bu iki sahə arasında elə bir ciddi fərq olmasa da, praktikada onlar aşağıdakı kimi fərqləndirilir. Elmi vizuallaşdırmada verilənlər adətən fasiləsiz fiziki obyektlərin seçilməsini təşkil edir (məsələn, ölçmə və ya modelləşmədən alınan temperaturun göstərilməsi, tibbi skanlaşdırma qurğu vasitəsilə alınan toxumaların sıxlığı). Belə verilənlər fərz edilən məkan kodlaşdırmasına malikdirlər və rəqəm tiplidirlər. İnformasiya vizuallaşdırmasında bunun əksinə olaraq verilənlər abstrakt xarakter daşıyır (məsələn, PT artefaktları, mətn sənədləri, qrafiklər və ya adi verilənlər bazasının cədvəlləri). Belə verilənlər çox vaxt seçim prosesinin nəticəsi hesab edilmir, fərz edilən məkan kodlaşdırmasına malik olmur və rəqəm tipli deyildir. Verilənləri vizuallaşdırmaq üçün vizual təsviri aydın şəkildə işləyib hazırlamaq lazımdır.

Abstrakt verilənləri görünən etmək üçün istifadə edilən konkret təsvirin seçilməsi məsələnin qoyuluşundan, verilənlərdən asılıdır və vizuallaşdırmanın effektivliyinə təsir edə bilər.

Qeyd etmək lazımdır ki, bəzi mənbələrdə elmi vizuallaşdırma, informasiya vizuallaşdırması ilə yanaşı SV üçüncü tədqiqat sahəsi kimi təqdim edilir [16]. Bəzi mənbələrdə isə SV informasiyanın vizuallaşdırılmasının bir qolu kimi təqdim olunur. SV-yə “Başa düşülmə və qarşılıqlı anlamın, həmçinin baxılan mövcud PT sisteminin mürəkkəbliyinin azaldılmasını təmin etmək üçün təsvirlərin müxtəlif formalarından istifadə edən bir fənn” kimi tərif verilmişdir [1]. SV kompüter proqramları və alqoritmlərinin vizuallaşdırılmasına aiddir. Burada əsas məqsəd PT artefaktlarının yaxşı anlaşılmasını təmin etməkdən ibarətdir. SV qabaqlayıcı və həm də təsdiqləyici imkanları dəstəkləməklə PT sənayesinin mürəkkəblik və təkamül problemlərində çox ümidverici bir həldir. Təəccüblü görünsə də, PT işləyib hazırlayan mühəndislər indiyə qədər PT sistemlərinin yaradılması, tətbiqi və xidmət göstərməsi üçün bir alət kimi vizuallaşdırmadan məhdud şəkildə istifadə etmişlər [1].

[17]-də SV və onun proqram mühəndisliyində tətbiqinə dair geniş icmal verilmişdir. İcmaldə PT istehsalçıları cəmiyyətində SV vacibliyini tədqiq edən iki araşdırma göstərilmişdir. Birinci araşdırma üzrə sorğu PT hazırlayan 111 tədqiqatçı arasında aparılmışdır. Texniki xidmət, PT-nin yenidən işlənilməsi (*re-engineering*) və tərs işləmə (*reverse engineering*) üzrə işlərin yerinə yetirilməsi üçün vizuallaşdırmadan istifadənin vacibliyi araşdırılmışdır. Sorğuda iştirak edən tədqiqatçıların 40%-i onun tamamilə vacibliyini, 42%-i əhəmiyyətliliyini, 7%-i münasibliyini, 1%-i isə əhəmiyyətsizliyini qeyd etmişdir.

İkinci araşdırmada PT vizuallaşdırmasının istifadəsinin səbəbləri öyrənilmişdir. Sorğuda iştirak edən 107 nəfərdən 71-i sənaye, 36-sı elmi dairənin tədqiqatçıları təmsil edirdi. Bu sorğunun nəticəsi göstərir ki, PT işlənilməsində vizuallaşdırmadan istifadənin ən vacib üstünlükləri aşağıdakılardan ibarətdir:

- PT-yə xərclərin azaldılması;
- daha yaxşı anlaşılması;
- məhsuldarlığın artması;
- mürəkkəbliyin idarə edilməsi;
- səhvlərin tapılmasına kömək;
- keyfiyyətin yüksəldilməsi.

B. Proqram təminatının təkamülünün vizuallaşdırılması (*Software evolution visualization*)

PT daima dəyişir və inkişaf edir. PT sistemlərində dəyişikliklərin əl ilə analiz edilməsi böyük zəhmət tələb edir və bu zaman tez-tez səhvlərə yol verilir. PT sistemlərinin təkamülünün vizuallaşdırılmasına PT sistemlərinin versiyalarının vizuallaşdırılması yolu ilə nail olmaq olar. Versiyaların vizuallaşdırılmasına adətən, sistemə daxil olan fayılın konkret versiyasında kod sətirlərinin sayı, sistemdə

artım faizi və dəyişmə faizi, qüsurların sıxlığı və mürəkkəblik meyarlarının dəyişdirilməsi kimi göstəricilərin vizuallaşdırılması daxildir.

PT təkamülünün vizuallaşdırılması – SV-nin yeni qoludur. PT təkamülünün vizuallaşdırılması - sistemlərin necə meydana gəlməsini göstərməklə çox böyük PT layihələrinin xidmət mərhələlərinin sadələşdirilməsinə yönəldilmişdir. PT-nin təkamülünün vizuallaşdırılmasında diqqət mərkəzində olan əsas tezis: “PT sistemlərinin təkamülü haqqında təsəvvür almaq imkanını istifadəçilərə necə vermək?”

PT təkamülünün vizuallaşdırılmasının izləyiciləri böyük proqram layihələrinin texniki xidmət mərhələsində iştirak edən idarəetmə və PT mühəndisləri komandasından ibarətdir. Bu mütəxəssislər adətən işləmə prosesinin son mərhələlərində PT ilə üzləşirlər və mənbə kodundan başqa heç bir dəstək olmadan onu anlamağa məcburdurlar. Proqram mühəndisliyində PT-nin xidmət prosesinə cəlb edilmiş fərqli rollar daşıyan fərqli şəxslər iştirak edir. Eyni rol fərqli şəxs tərəfindən oynanıla bilər və eyni şəxs proqram layihəsinin bütün xidmət müddəti ərzində bir və ya bir neçə rol oynaya bilər. PT təkamülünün vizuallaşdırılmasına yönəldilmiş ümumi rollar aşağıdakılardır:

- layihə rəhbərləri məhsulun mənbə kodunun icmalını ala bilər və qərarların qəbul edilməsində aşkar edilmiş tendensiyalardan istifadə edə bilərlər;
- menecerlər heyəti məhsulun təkamül vəziyyətini izləyə və onun yeni buraxılış üçün hazır olmasını həll edə bilərlər;
- layihə hazırlayanlar yenidən təşkil olunmağa ehtiyacı olan və yaxud arxitektura pozulması yaranan alt sistemləri müəyyən edə bilərlər;
- test edənlər sistemin miqyasıyında tələb olunan əvvəlki vəziyyətə qayıdışı müəyyənləşdirə bilərlər və s.

Mühəndis-proqramçılar SV təkamülünün effektiv bir alətə çevrilməsi üçün bir sıra məsələləri yerinə yetirməlidirlər. Bu problemlərdən bəziləri vizuallaşdırma üçün ümumi problemlərdir. Bəzi problemlər bütövlükdə PT-nin hazırlanması sənayesi konteksti üçün və xüsusi halda PT təkamülü üçün spesifikdir. Bütövlükdə, bu problemlər istənilən vizuallaşdırmanın son məqsədi ilə əlaqədardır, yəni istifadəçiyə konkret problemin effektiv həlli üçün köməkdir. SV təkamülünün problemləri aşağıdakılardır:

- **miqyaslanma (scalability)**: müasir PT sistemləri çox böyükdür. Belə sistemlərin bütün təkamülünün vizuallaşdırılması sadə məsələ deyildir. Birincisi, bu interaktiv analiz və aşkarlanmanın sadələşdirilməsi üçün böyük miqdarda informasiyanın effektiv analizini tələb edir. İkincisi, analizin nəticəsi effektiv şəkildə təsvir edilməlidir. Əgər əldə olunan verilənlər həddən artıq böyükdürsə təqdimatı böyük ekranlarda və ya çox ekranlı konfigurasiyalarda təsvir etmək olar. Tipik PT-nin hazırlanması baxımından istifadəçinin bir ekranlı monitorla işləməsi tələb olunur. Bu da çox böyük informasiya fəzasının səmərəli və effektiv göstərilməsində məhdudiyət yaradır.
- **intuisiya (intuitiveness)**: PT ilə bağlı artefaktlar və obyektlər, fayllar, kod sətirləri, funksiyalar, modullar,

proqramçılar, səhvlər və xəbərlər kimi obyektlər mürəkkəb şəbəkə əlaqəsi ilə birləşmiş abstrakt obyektlərdir. Asan başa düşülən və bu yüksək ölçülü verilənlər fəzasının qavranılmasının effektiv ötürülməsinə uyğun olan vizual təsvirin yaradılması PT təkamülünün vizuallaşdırılmasının ən böyük problemlərindən biridir.

- **istifadəlilik (usability)**: PT-nin anlaşılmasının – proqram kompleksinin müxtəlif (bir-biri əlaqəli) aspektlərində bir çox sorğular tələb edən dinamik və tez-tez təkrar olunan prosesdir. Adətən, istifadəçilər hipotezlər formalaşdırır, sonra onu təsdiqləməyə çalışırlar. Bu prosesdə onlar hipotezlərin dəyişdirilməsinə səbəb olan və yoxlamanın yeni dövrəsini tələb edən yeni faktlar aşkar edə bilərlər. İstifadəçi aktivliyinin tələblərinin və xüsusiyyətlərinin nəzərə alınmasına uyğun olaraq PT təkamülünün vizuallaşdırılması əlavələrinin işlənməsi müvəffəqiyyət qazanmaq üçün çox önəmlidir.

- **inteqrasiya (integration)**: PT-nin vizuallaşdırılma əlavələrinin qəbul olunması, uzunmüddətli perspektivdə uğur qazanması üçün PT-nin işlənməsi prosesində yaradılmış kod analizatorları, kompilyatorlar, sazlama və PT-nin konfigurasiyasının idarəedilməsi sistemləri kimi alətləri ilə asan inteqrasiya olunmalıdır [18].

IV. VIZUALLAŞDIRMA ALƏTLƏRİ

PT təkamülünün vizuallaşdırılması böyük həcmli informasiyanı istifadəçilər üçün əlverişli etməli və istifadəçilərə sərbəst şəkildə təsvirləri və tendensiyaları tapmağa imkan verməlidir. Bu istiqamətdə çox az sayda alətlər təklif olunmuş, onların bəziləri aşağıda təqdim olunur.

PT təkamülünə aid olan ilk vizuallaşdırma alətlərindən biri SeeSoft alətidir [19]. Verilmiş sorğuya müfəviq olan dəyişikliklər üçün kod fraqmentlərinin təsvirində düz xətt və rəngdən istifadə olunur. [20]-də analoji yanaşmanın istifadəsilə daha müasir alət təqdim edilmişdir. Bu alət müəyyən vaxtda PT layihəsinin artefaktları və aktivləri haqqında informasiyanı vahid təsvirdə birləşdirir. SeeSoft və Xia [21] alətlərinin hər ikisi PT-nin strukturunu göstərmək üçün rənglənmiş ağacvari maketdən istifadə edir. Belə alətlər PT sistemlərinin strukturunu və dəyişikliklərin əlaqəsini üzə çıxarmağa, həyata keçirilməsi ilə sıx əlaqədar olan faylların tapılmasına kömək edir (şəx. 1).

UNIX əməliyyat sistemi üçün hazırlanmış Diff utiliti və onun Windows versiyası üçün olan WinDiff utiliti qlobal təkamülə gedən yolda ilk addım kimi faylların iki versiyası arasındakı fərqləri (yerləşdirmə, silmə və dəyişdirmə) göstərir [22, 23]. Hipikat – bu analoji alət layihənin gedişində qeydə alınmış səhvlər haqqında və ya elektron poçt məlumatları kimi kontekst informasiya ilə versiyalarda fərqlər haqqında informasiyanı zənginləşdirir [24]. Bu informasiya müxtəlif versiyalar arasındakı dəyişiklikləri başa düşmək üçün çox faydalıdır. Lakin iki versiya üçün effektiv hesab edilən bu alət hər biri yüzlərlə versiyaya malik fayldan təşkil olunmuş real layihədə təkamülü nəzərdən keçirməyə imkan vermir. Bundan başqa, onlar PT-nin konfigurasiyasının idarə edilməsində (SCM – *software configuration management*) informasiya potensialından tam istifadə etməyə imkan vermir (məsələn, iki versiya arasında vaxta və müəllifə aid olan informasiya dəyişiklikləri).



Şəkil 1. Son yoxlama tarixinin rənglərlə kodlaşdırılmış sxemi: yaşıl köhnə, mavi yeni.

Daha müasir alətlər bunu real layihələrin təkamülündə ümumiləşdirməyə çalışırlar. Məsələn, Historian aləti Qant diaqramından istifadə etməklə üst-üstə düşən versiyalar sistemi (Concurrent Versions System, CVS) repozitoriyalarının bəsit vizuallaşmasını təqdim edir [25, 26]. Bu vizuallaşma çox az sayda fayllarla daha yaxşı işləyir və bütün layihə üçün təkamülünü izləməyə imkan vermir. Digər bir yanaşma PT strukturunu və təkamül mexanizmlərini qrafiklər ardıcılığı kimi vizuallaşdırır [27].

VRCS və 3DsoftVis alətləri 3D qurmasında vaxtdan ölçü parametri kimi istifadə etməklə miqyaslama problemini yaxşılaşdırmağa çalışır [28, 29]. Bu yanaşma daha çox sayda versiyaları vizuallaşdırmağa imkan versə də, 3D vizual mühitində ona xas olan “sədd” (occlusion) problemi vizuallaşmanın bəzi imkanlarını azaldır.

[30]-da siniflər səviyyəsində obyekt yönümlü PT təkamülünü vizuallaşdırmaq üçün təkamül matrisindən istifadə olunmuşdur. [31]-də bütün layihənin təkamülünü fayllar səviyyəsində vizuallaşdırmaq və təkamül anlarını vizual olaraq qeyd etmək üçün spektroqrafdan istifadə edilmişdir.

NƏTİCƏ

Hazırda PT-nin işlənməsi prosesində hələ də PT-nin vizuallaşdırılmasından tam istifadə olunmur. Aparılan tədqiqatlardan məlumdur ki, vizuallaşdırmanın proqramçılar tərəfindən qəbul edilməsinin əsas maneələrindən biri PT-nin hazırlanması və dəstəklənməsi üçün qəbul edilmiş alətlər, metodologiyalar və proseslərdə vizuallaşdırmanın inteqrasiyasının olmamasıdır. PT-nin vizuallaşdırılmasının ən mühüm problemlərindən biri də müasir PT sistemlərinin böyük ölçülünə görə onların məhdudlaşdırılmış miqyaslanmaya malik olmasıdır. Eyni zamanda PT-nin işlənilməsinin konkret kontekstləri, məsələn, konkret proqramlaşdırma dilindən istifadə və ya hazırlanma metodologiyaları, ixtisaslaşdırılmış interaktiv vizual texnologiyaların və alətlərin hazırlanmasını tələb edə bilər. SV təkamülü proqramçıların və istifadəçilərin sorğularına və tələblərinə cavab verməlidir. Bütün bunlar SV təkamülü üçün yeni araşdırmalara geniş yol açır.

ƏDƏBİYYAT

- [1] D. Gračanin, K. Matkovic, M. Eltoweissy, “Innovations Syst Softw Eng”, 2005, Springer-Verlag: pp. 221–230.
- [2] B. Zhu, H. Chen, Information visualization, Annual Review of Information Science and Technology, 2005, Vol. 39, no. 1, pp. 139-177, <http://ai.arizona.edu/>
- [3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, “Design Patterns - Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995.
- [4] T. Elrad, R. Filman, A. Bader, “Aspect-oriented programming”, CACM: Communications of the ACM press, 2001.
- [5] J. Hhannemann, G. Kiczales, “Design pattern implementation in java and aspect”, Proceedings of the Annual ACM SIGPLAN Conferences on Object-Oriented Programming 2002, pp. 161–173.
- [6] M. Lehman, “Laws of software evolution revisited”, Proceedings of the 5th European Workshop on Software Process Technology (London, UK, Springer-Verlag, 1996, pp. 108–124.
- [7] M. Godfrey, L. Zou, “Using origin analysis to detect merging and splitting of source code entities”, IEEE Transactions on Software Engineering, 2005, pp. 166–181.
- [8] G. Booch, www.booch.com.
- [9] L. Erlikh, “Leveraging legacy system dollars for e-business”, IEEE IT Pro 2000, pp. 17–23.
- [10] D.M. Coleman, D. Ash, B. Lowther, P. OMAN, “Using metrics to evaluate software system maintainability”, IEEE Computer, 1994, pp. 44–49.
- [11] T. Standish, “An essay on software reuse”, IEEE Transactions on Software Engineering, 1984, pp. 494–497.
- [12] T. Corbi, “Program understanding challenge for the 1990’s. IBM Syst”, 1989, pp. 294–306.
- [13] P. Bennet, “Software Maintenance Management: A Study of the Maintenance of Computer Application Software in Data Processing Organizations”, Addison-Wesley, 1980.
- [14] N. Gershon, “Information visualization: The next frontier”, Proceedings of International Conference and Exhibition on Computer Graphics and Interactive Techniques, 1994, pp. 485–486.
- [15] C. Ware, “Information Visualization. Perception for Design”, Morgan Kaufmann, 2000.
- [16] E. Olshannikova et.al. “Visualizing Big Data with augmented and virtual reality: challenges and research agenda”, Journal of Big Data, 2015, vol. 2, pp. 2–22..
- [17] S. Diehl, “Software Visualization - Visualizing the Structure, Behaviour, and Evolution of Software”, Springer, 2007.
- [18] Stefan-Lucian, “Software Evolution Visualization”, Technische Universiteit Eindhoven”, 2007.
- [19] S. G. Eick, J. Steffen and E. Sumner, “Seesoft – a tool for visualizing line oriented software statistics”, IEEE Transactions on Software Engineering, 1992, pp. 957–968.
- [20] J. Froehlich, P. Dourish, “Unifying artifacts and activities in a visual tool for distributed software development teams”, Proceedings of the 26th International Conference on Software Engineering, 2004, IEEE CS, pp. 387–396.
- [21] X. WU, “Visualization of version control information”. Master’s thesis, University of Victoria, 2003.
- [22] DIFF, www.gnu.org.
- [23] WINDIFF, www.microsoft.com.
- [24] D. Cubranic, G. Murphy, J. Singer, K. Booth, “Hipikat: A project memory for software development”, IEEE Transactions on Software Engineering, 2005, pp. 446–465.
- [25] www.historian.tigris.org.
- [26] H. Gantt, “Work, wages and profits”, The Engineering Magazine 1910, 312.
- [27] C. Collberg, S. Kobourov, J. Nagra, J. Pitts, K. Wampler, “A system for graph-based visualization of the evolution of software”, Proceedings of

- the 2003 ACM Symposium on Software Visualization, 2003, ACM press, pp. 77–86.
- [28] H. Koike, H.-C. Chu, “Vrcs: integrating version control and module management using interactive three-dimensional graphics”, Proceedings of the 1997 IEEE Symposium on Visual Languages 1997, IEEE CS, pp. 168–173.
- [29] C. Riva, “Visualizing software release histories with 3dsoftvis”, Proceedings of the 22nd International Conference on Software Engineering, 2000, ACM Press, p. 789.
- [30] M. Lanza, “The evolution matrix: recovering software evolution using software visualization techniques”, Proceedings of the 4th International Work- shop on Principles of Software Evolution, 2001, ACM Press, pp. 37–42.
- [31] J. Wu, C. Spitzer, A. Hassan, R. Holt, “Evolution spectrographs Visualizing punctuated change in software evolution”, Proceedings of the 7th International Workshop on Principles of Software Evolution, 2004, IEEE CS, pp. 57–66.