ZESZYTY NAUKOWE UNIWERSYTETU SZCZECIŃSKIEGO

NR 508 STUDIA INFORMATICA NR 22 2008

ŁUKASZ RADLIŃSKI

PRZEGLĄD PUBLICZNIE DOSTĘPNYCH BAZ DANYCH PRZEDSIĘWZIĘĆ INFORMATYCZNYCH

Wprowadzenie

Celem wielu badań naukowych podejmowanych w zakresie inżynierii oprogramowania jest znalezienie zależności między czynnikami wpływającymi na proces tworzenia oprogramowania oraz na wytworzone oprogramowanie. Do ich znalezienia zwykle potrzebne są wiarygodne dane empiryczne. Często są one trudno dostępne, gdyż stanowią tajemnicę firm programistycznych. Taka sytuacja utrudnia prowadzenie badań poznawczych. W Internecie są jednak dostępne bazy danych o rzeczywistych przedsięwzięciach[1] informatycznych, na przykład ISBSG, PROMISE, NASA czy oparte na Bugzilli. Bazy te są zróżnicowane pod względem ilości i jakości danych. W efekcie nie jest możliwe wybranie jednej z nich jako uniwersalnej do każdego zastosowania. Niektóre, jak na przykład udostępniane przez ISBSG, zawierają dużą liczbę zmiennych, co zwiększa zakres ich stosowalności. Inne, jak na przykład zestaw baz PROMISE, to dane tematyczne do szacowania nakładów, defektów czy innych czynników.

Głównym celem niniejszego artykułu jest dokonanie analizy danych empirycznych o przedsięwzięciach informatycznych. Zostaną scharakteryzowane dane z poszczególnych baz, a także wskazane cele, do jakich można je wykorzystywać. Omówione zostaną także zauważone zalety i wady poszczególnych

[1] Pomimo że terminy ‘przedsięwzięcie’ i ‘projekt’ mogą mieć niekiedy inne znaczenie w zależności od kontekstu ich użycia, w tej pracy będą traktowane jako synonimy. Termin ‘projekt’ nie jest bowiem tu używany w wąskim znaczeniu, jako ‘opracowanie, dokumentacja tworzonego systemu’, lecz jako ‘całość działań zmierzających do zbudowania systemu informatycznego’.

baz. Nie jest celem analizy wskazanie bazy danych najlepszej w każdej sytuacji, ponieważ taka nie istnieje.

Wyniki analizy będą wykorzystane przy budowie modeli do szacowania ryzyka w przedsięwzięciach informatycznych. Niektóre z baz charakteryzowanych w artykule (ISBSG, Eclipse, Mozilla, PROMISE: qqdefects) zostały już wykorzystane w prowadzonych badaniach[2]. Również inni autorzy stosowali je w swoich badaniach. Bazy te są nadal rozwijane – są zasilane nowymi danymi (z wyjątkiem repozytorium NASA).

1. Bazy danych przedsięwzięć informatycznych

ISBSG

Baza danych ISBSG R10[3] stanowi repozytorium 4106 przedsięwzięć informatycznych wdrażanych w latach 1989–2004. Przedsięwzięcia te opisano, podając ich 102 parametry, zarówno jakościowe, jak i ilościowe. Istotną cechą tej bazy jest fakt istnienia niepełnych danych – ani jedno przedsięwzięcie nie zostało opisane za pomocą wszystkich dostępnych parametrów. Największe braki występują w danych słownych charakteryzujących przedsięwzięcie i system.

Dane zawarte w repozytorium zostały dobrowolnie przekazane przez firmy informatyczne z całego świata, nie udostępniono jednak danych dotyczących tych firm. Zaobserwować można duże podobieństwo niektórych danych, na przykład zbliżony termin wdrożenia, typ organizacji (klienta), rodzaj aplikacji czy brak danych do tej samej grupy parametrów. To może sugerować, że projekty nie były realizowane przez jedną firmę informatyczną. Repozytorium

---

[2] Ł. Radliński, N. Fenton, D. Marquez, P. Hearty, *Empirical Analysis of Software Defect Types, Information Systems Architecture and Technology. Information Technology and Web Engineering: Models, Concepts & Challenges*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2007; Ł. Radliński, N. Fenton, M. Neil, D. Marquez, *Modelling Prior Productivity and Defect Rates in a Causal Model for Software Project Risk Assessment*, „Polish Journal of Environmental Studies" 2007, vol. 16, no. 4A; N. Fenton, M. Neil, W. Marsh, P. Hearty, Ł. Radliński, P. Krause, *Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction, Proceedings of the Third international Workshop on Predictor Models in Software Engineering* (May 20–26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 2. http://dx.doi.org/10.1109/PROMISE.2007.11.

[3] ISBSG, Repository Data Release 10, International Software Benchmarking Standards Group 2007, www.isbsg.org.

ISBSG jest płatne, przy wykorzystaniu do celów naukowych opłata jest znacząco niższa[4].

Promise

PROMISE[5] stanowi repozytorium 36 niezależnych baz danych. Bazy te są zróżnicowane tematycznie. Pochodzą z różnych źródeł – najczęściej od osób indywidualnych, które udostępniają w ten sposób dane empiryczne do dalszych badań. Większość baz danych zapisana jest w plikach *arff*[6]. W plikach tych, oprócz samych danych i opisu ich znaczenia, zawarte są dodatkowe informacje dotyczące pochodzenia danych, wykorzystania ich w minionych badaniach, często łącznie z przytoczeniem najważniejszych wniosków z tych badań i modeli powstałych z tych danych.

NASA MDP

NASA IV&V Facility Metrics Data Program[7] (MDP) to projekt rozwijany przez Galaxy Global Corporation, Inc. dla NASA. Jego głównym celem jest zbieranie, walidacja, organizowanie, przechowywanie i dostarczanie danych o różnych metrykach oprogramowania rozwijanego przez NASA. Repozytorium składa się z 13 baz danych, zawierających łącznie dane o około 14 tys. produktów[8]. W bazach zapisana jest historia powstawania tych systemów, głównie w odniesieniu do ich jakości (raporty o problemach, defektach, zmianach wymagań itp.). Pojedyncza baza zawiera dane o jednym projekcie lub wydzielonej jego części: strukturze i modułach. Ze względu na dość wąski obszar zainteresowań NASA, również systemy rozwijane przez tę organizację mają specyficzne zastosowania – w dziedzinie obronności, lotów kosmicznych, obsługi satelitów itp.

---

[4] Więcej informacji na temat możliwości wykorzystania repozytorium do celów naukowych na stronie http://www.isbsg.org/Isbsg.Nsf/weben/Academic%20&%20students.

[5] G. Boetticher, T. Menzies and T. Ostrand, *PROMISE Repository of Empirical Software Engineering Data*, West Virginia University, Department of Computer Science, 2007, http://promisedata.org/repository.

[6] Pliki '*arff*' to pliki tekstowe ASCII, które opisują listę instancji opisanych wspólną listą atrybutów. Format ten został opracowany na Uniwersytecie Waikato. Jest on głównie używany w aplikacjach *data-mining* i maszynowego uczenia. Więcej: http://weka.sourceforge.net/wekadoc/index.php/en:ARFF_%283.5.1%29.

[7] http://mdp.ivv.nasa.gov/index.html.

[8] Termin 'produkty' odnosi się w tym przypadku raczej do poszczególnych części systemu (podprojektów, komponentów), a nie całych produktów programowych.

Projekty wykorzystujące Bugzilla

Bugzilla[9] jest aplikacją służącą do zarządzania błędami w tworzonym oprogramowaniu. Wiele znanych organizacji tworzących oprogramowanie typu *open-source* z niej korzysta, między innymi: Apache Software Foundation[10], Eclipse[11], GNOME[12], KDE[13], Mozilla[14], Linux Kernel[15], NASA[16], OpenOffice[17] czy twórcy różnych dystrybucji Linuxa[18]. Lista znanych instalacji Bugzilli obejmuje ponad 750 pozycji[19], przy czym każda stanowi osobną bazę, która może zawierać dane o wielu przedsięwzięciach. Twórcy wymienionego oprogramowania udostępniają nieodpłatnie bazy danych dotyczące rejestrowanych defektów w oprogramowaniu. W przykładach zastosowań Bugzilli przedstawionych w tej pracy pojedyncza baza zawiera, zależnie od podejścia twórcy oprogramowania, dane o jednym produkcie składającym się z podsystemów (modułów, bibliotek itp.) lub o wielu produktach powiązanych często w ramach jednego dużego przedsięwzięcia.

---

[9] The Mozilla Organisation, Bugzilla, 2007, http://www.bugzilla.org.

[10] Apache Software Foundation, ASF Bugzilla, 2007, http://issues.apache. org/bugzilla/report.cgi.

[11] Eclipse, Eclipse Bugs, 2007, https://bugs.eclipse.org/bugs.

[12] GNOME Foundation, GNOME Bugzilla, 2007, http://bugzilla.gnome.org.

[13] KDE e.V., KDE Bug Tracking System, 2007, http://bugs.kde.org.

[14] Mozilla.org, Bugzilla@Mozilla, 2007, https://bugzilla.mozilla.org.

[15] The Linux Kernel Organisation, Inc., Kernel Bug Tracker, 2007, http://bugzilla.kernel.org/report.cgi.

[16] NASA, Bug Reporting System, Goddard Space Flight Center, NASA, 2007, http://itos.gsfc.nasa.gov/~bugzilla/report.cgi.

[17] OpenOffice.org, OpenOffice.org Issues, 2007, http://qa.openoffice.org/issues/query.cgi.

[18] Fedora Bugzilla, 2007, https://bugzilla.redhat.com; Mandriva Bugzilla, 2007, http://qa.mandriva.com; Bugzilla Gentoo, 2007, http://bugs.gentoo.org; Novell's Bugzilla, 2007, https://bugzilla.novell.com/index.cgi.

[19] The Mozilla Organisation, Bugzilla Installation List, 2007, http://www.bugzilla.org/installation-list.

2. Porównanie baz danych przedsięwzięć informatycznych

W tabeli 1 przedstawione są dane porównawcze charakteryzujące analizowane bazy danych. W trakcie analizy zawartości baz stwierdzono, że:

1. W przedsięwzięciach wykorzystujących Bugzillę twórcy oprogramowania w różny sposób traktowali pojęcia: produkt, komponent, subkomponent itp. Spowodowało to, że liczby projektów podane w tabeli 1 nie są porównywalne między bazami. Na przykład baza KDE wydaje się zawierać największą liczbę produktów. Jest to jednak efekt odmiennego nazewnictwa stosowanego w tych przedsięwzięciach. W tej tabeli przez pojęcie „projekt" rozumiano zarówno kompletne produkty programowe, jak i komponenty – zawsze najbardziej syntetyczne elementy w każdej bazie.

2. Większość bazy projektów z repozytorium NASA zostało również umieszczonych w repozytorium PROMISE – w innym formacie (pojedynczy plik dla jednej bazy) i nieco okrojonej wersji (brak szczegółów dotyczących historii wymagań czy defektów).

3. Niektóre bazy z repozytorium PROMISE nie zawierają liczby defektów, a jedynie oznaczenia, czy dany moduł je zawierał czy nie.

Tabela 1

Porównanie baz danych przedsięwzięć informatycznych

Nazwa grupy baz danych/ Nazwa bazy danych	Opis bazy danych	Liczba projektów	Liczba parametrów liczbowych/słownych/ suma	Główne parametry opisujące projekt	Format danych
1	2	3	4	5	6
ISBSG: ISBSG Repository Data Release 10	dane o projektach z wielu dziedzin zastosowań	4106	38/60/102	rozmiar, nakłady, defekty, charakterystyka projektu, dokumenty i techniki	plik *xls*
PROMISE: coc81	baza wykorzystana do modelu COCOMO	63	17/0/17[20]	rozmiar, nakłady, jakość ludzi i procesu	plik *arff*

[20] Pomimo że wszystkie dane są ujęte w skali liczbowej, 15 z nich opisuje natężenie danej cechy w skali porządkowej. Plik z danymi zawiera informacje na temat pastowanych transformacji danych.

1	2	3	4	5	6
PROMISE: cocomona-sa_v1	dane z różnych ośrodków NASA z lat 80. i 90. XX wieku	60	2/15/17	rozmiar, nakłady, jakość ludzi i procesu	plik *arff*
PROMISE: datatreive	dane projektu DATATRIVE realizowanego w Digital Engineering we Włoszech	130	8/0/9	rozmiar, jakość ludzi, defekty	plik *arff*
PROMISE: desharnais		81	11/1/12	rozmiar, nakłady, poziom doświadczenia ludzi, charakterystyka projektu	plik *arff*
PROMISE: humans	dane o osobach i dokonanych przez nie szacunkach dotyczących komponentów oprogramowania	122 osoby	20/2/22	dane demograficzne, wykształcenie i doświadczenie zawodowe, błędy szacowowane oprogramowania przez poszczególne osoby	plik *arff* zapisany jako *csv*
PROMISE: humans2	dane o osobach i dokonanych przez nie szacunkach dotyczących komponentów oprogramowania	75 osób	14/0/14[21]	wykształcenie i doświadczenie zawodowe, błędy szacowowane oprogramowania przez poszczególne osoby	plik *arff*
PROMISE: nasa93	syntetyczne dane o projektach realizowanych w NASA	93	4/20/24	charakterystyka projektu, jakość ludzi i procesu, rozmiar, nakłady	plik *arff*
PROMISE: usp05-ft	dane o projektach studenckich	76	10/5/15	charakterystyka projektu, rozmiar, nakłady	plik *arff*
PROMISE: usp05	dane o projektach studenckich	203	17/6/23	charakterystyka projektu, rozmiar, nakłady	plik *arff*
PROMISE: cm1	dane o jednym z podzespołów statku kosmicznego (realizowanym przez NASA)	498 modułów	21/1/22	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: jm1	system predykcyjny oparty na metodach symululacyjnych (realizowany przez NASA)	10885 modułów	21/1/22	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: kc1	system zarządzający przyjmowaniem i przetwarzaniem danych (realizowany przez NASA)	2109 modułów	21/1/22	rozmiar i inne miary kodu, defekty	plik *arff*

[21] Jeden z parametrów jest słowny (wykształcenie), został jednak zakodowany w bazie jako liczba.

1	2	3	4	5	6
PROMISE: kc1-class-level-numericdefect	system zarządzający przyjmowaniem i przetwarzaniem danych (realizowany przez NASA), dane na poziomie klas	145 klas	95/0/95	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: kc1-class-level-defectiveornot	system zarządzający przyjmowaniem i przetwarzaniem danych (realizowany przez NASA), dane na poziomie klas	145 klas	94/1/95	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: kc1-class-level-top5percentDF	system zarządzający przyjmowaniem i przetwarzaniem danych (realizowany przez NASA), dane na poziomie klas	145 klas	94/1/95	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: kc2	inna część systemu kc1 realizowanego przez NASA przez inny zespół	522 modułów	21/1/22	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: kc3	system zbierający i przetwarzający dane satelitarne (realizowany przez NASA)	458 modułów	39/1/40	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: ivvbayes		4	177/0/177[22]	rozmiar, nakłady, defekty jakość ludzi i procesu	pliki *xls* i *pdf*
PROMISE: mb2		1: 211 2: 433	1: 26/2/28 2: 22/2/24	*brak objaśnień*	pliki *csv*, *txt* i *awk*
PROMISE: mc1	system związany z zapłonem wahadłowca (realizowany przez NASA)	9466 modułów	38/1/39	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: mozilla4	dane o czasach obsługi problemów (defektów lub innych) w Mozilli	15545 zmian	6/0/6	rozmiar, czasy modyfikacji	plik *arff*
PROMISE: mw1	system związany z obsługą startu wahadłowca (realizowany przez NASA)	403 moduły	37/1/38	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: pc1	system obsługi lotu sztucznego satelity (realizowany przez NASA)	1109 modułów	21/1/22	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: pc2	dynamiczny symulator kontroli ustawienia w czasie lotu (realizowany przez NASA)	5589 modułów	36/1/37	rozmiar i inne miary kodu, defekty	plik *arff*

[22] 161 parametrów w skali porządkowej zostało zakodowanych w bazie jako liczbowe.

1	2	3	4	5	6
PROMISE: pc3	system obsługi lotu sztucznego satelity (realizowany przez NASA)	1163 moduły	37/1/38	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: pc4	system obsługi lotu sztucznego satelity (realizowany przez NASA)	1458 modułów	37/1/38	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: pc5	*upgrade* kokpitu (realizowany przez NASA)	17186 modułów	38/1/39	rozmiar i inne miary kodu, defekty	plik *arff*
PROMISE: qqdefects	projekty z branży elektroniki (AGD); dodatkowo model (sieć Bayesa) do szacowania liczby defektów	31	3/28/31	jakość procesu, rozmiar, nakłady, defekty	plik *arff*, *cmp*
PROMISE: cm1trace	wymagania dla systemu CM1 (realizowanego przez NASA)	220 wymagań niskiego poziomu, 235 wymagań wysokiego poziomu	0/1/1	lista wymagań wraz zależnościami między nimi	pliki tekst.
PROMISE: modis		49 wymagań niskiego poziomu, 19 wymagań wysokiego poziomu	0/1/1	lista wymagań wraz zależnościami między nimi	pliki tekst.
PROMISE: nfr		625 wymagań	1/2/3	wymagania i powiązania z klasami	plik *arff*
PROMISE: freebsd	lista problemów do naprawienia lub usunięcia	33 335 problemów	3/3/7	problem, data rejestracji, odnośniki do klas	plik *arff*
PROMISE: nickle	dane o projektach napisanych w języku Nickle	2972 zmian	3/6/10	typ i rozmiar modyfikacji, data zatwierdzenia	plik *arff*
PROMISE: reuse	dane o projektach wykorzystujących wcześniej napisane części kodu	24	28/1/29	charakterystyka projektu, jakość ludzi i procesu, sukces/porażka projektu	plik *arff*
PROMISE: xfree	dane o zmianach w kodzie projektu XFree86	175658 zmian	3/6/10	typ i rozmiar modyfikacji, data zatwierdzenia	plik *arff*
PROMISE: xorg	dane o zmianach w kodzie projektu x.org	136435 zmian	3/6/10	typ i rozmiar modyfikacji, data zatwierdzenia	plik *arff*

1	2	3	4	5	6
NASA MDP: cm1	dane o jednym z podzespołów statku kosmicznego (realizowanym przez NASA)	505 modułów, 128 defektów, 160 wymagań	60/3/64	rozmiar i inne miary kodu, historia: problemów i ich rozwiązań oraz wymagań i ich realizacji	pliki *csv*
NASA MDP: jm1	system predykcyjny oparty na metodach symulacyjnych (realizowany przez NASA)	10878 modułów, 1659 defektów, 74 wymag.	40/8/49	rozmiar i inne miary kodu, historia: problemów i ich rozwiązań oraz wymagań i ich realizacji	pliki *csv*
NASA MDP: kc1	system zarządzający przyjmowaniem i przetwarzaniem danych (realizowany przez NASA)	145 modułów, 1101 defektów	45/10/56	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: kc3	system zbierający i przetwarzający dane satelitarne (realizowany przez NASA)	458 modułów, 92 defekty	52/11/64	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: kc4		125 modułów, 103 defekty	54/8/63	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: mc1	system związany z zapłonem wahadłowca (realizowany przez NASA)	9466 modułów, 202 defekty	47/2/50	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: mc2	system naprowadzania wizualnego (realizowany przez NASA)	161 modułów, 228 defektów	47/5/52	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: mw1	system związany z obsługą startu wahadłowca (realizowany przez NASA)	403 moduły, 101 defektów	48/4/53	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: pc1	system obsługi lotu sztucznego satelity (realizowany przez NASA)	1107 modułów, 575 defektów, 320 wymagań	58/7/66	rozmiar i inne miary kodu, historia: problemów i ich rozwiązań oraz wymagań i ich realizacji	pliki *csv*
NASA MDP: pc2	dynamiczny symulator kontroli ustawienia w czasie lotu (realizowany przez NASA)	5589 modułów, 72 defekty	48/6/55	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: pc3	system obsługi lotu sztucznego satelity (realizowany przez NASA)	1563 moduły, 1273 defekty	48/4/53	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*

1	2	3	4	5	6
NASA MDP: pc4	system obsługi lotu sztucznego satelity (realizowany przez NASA)	1158 modułów, 986 defektów	48/4/53	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
NASA MDP: pc5	*upgrade* kokpitu (realizowany przez NASA)	17186 modułów, 1079 defektów	45/3/48	rozmiar i inne miary kodu, historia problemów i ich rozwiązań	pliki *csv*
Bugzilla: Apache Software Foundation	baza danych o produktach Apache Software Foundation	63	2/12/15	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: Eclipse	baza danych o Eclipse i dodatkowych produktach, modułach i bibliotekach	85	2/14/17	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: GNOME	baza danych o GNOME i dodatkowych modułach i bibliotekach	367	1/13/15	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: KDE	baza danych o KDE i dodatkowych modułach i bibliotekach	361	2/14/17	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: Mozilla	baza danych o Mozilla i produktach powiązanych	35	2/14/17	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: Linux Kernel	baza danych o Linux Kernel i dodatkowych modułach i bibliotekach	14	1/11/13	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: NASA	baza danych projektów – składników ITOS – system kontroli i monitoringu satelitarnego	4	2/13/16	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: OpenOffice	baza danych o OpenOffice i dodatkowych modułach i bibliotekach	151	2/14/17	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy, pliki *sxc*, *sxw* i in.
Bugzilla: Fedora	baza danych o Fedora i produktach powiązanych	49	2/12/15	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: Mandriva	baza danych o Mandriva i produktach powiązanych	15	2/14/17	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy

1	2	3	4	5	6
Bugzilla: Gentoo	baza danych o Gentoo i produktach powiązanych	21	2/11/14	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy
Bugzilla: Novell	baza danych o różnych produktach rozwijanych przez Novell	37	2/14/17	słowny opis projektu, liczba defektów wg różnych klasyfikacji	raporty: tabele, wykresy i listy

3. Zalety i wady poszczególnych repozytoriów danych

Analiza zawartości danych w poszczególnych repozytoriach ujawniła zalety i wady poszczególnych repozytoriów. Zilustrowano je w tabeli 2.

Tabela 2

Zalety i wady poszczególnych repozytoriów danych

Repozytorium	Zalety	Wady
ISBSG	• duża różnorodność projektów • zunifikowana lista parametrów opisujących projekt	• duża liczba brakujących danych • brak parametrów opisujących jakość procesu i ludzi
PROMISE	• duża różnorodność tematyczna baz – obszary zastosowań projektów i parametry opisujące projekt • duża ilość dodatkowych informacji o danych i ich wykorzystaniu: publikacje, modele • mała liczba brakujących danych	• brak ujednoliconej listy parametrów • częsta obecność parametrów jedynie z jednej dziedziny (rozmiar, defekty, nakłady)
NASA MDP	• szczegółowe dane na temat defektów – aż do poziomu raportu o problemie • duża liczba parametrów opisujących poszczególne moduły i defekty	• kłopotliwe korzystanie z danych z powodu zapisu w wielu plikach • duża liczba brakujących danych o defektach w niektórych bazach • brak danych o nakładach • brak parametrów opisujących jakość procesu i ludzi • projekty jedynie z obszaru działalności NASA
wykorzystujące Bugzilla	• szczegółowe dane na temat defektów – aż do poziomu raportu o problemie • różne formy prezentacji danych: raporty, wykresy, listy	• nastawienie jedynie na defekty – brak danych o nakładach oraz parametrów opisujących jakość procesu i ludzi • brak łatwego dostępu do zestawienia danych w postaci źródłowej

4. Możliwości zastosowania poszczególnych repozytoriów danych

Ze względu na różną zawartość poszczególnych baz danych nie jest możliwe wykorzystanie każdej z nich w dowolnym celu. Baza ISBSG i kilka baz z PROMISE zawierają dane o różnych projektach, są to jednak dane syntetyczne, charakteryzujące pojedynczy projekt bez uwzględnienia jego struktury. Pozostałe bazy zawierają dane o pojedynczych systemach, są one bardziej szczegółowe, odnoszą się do poszczególnych części systemu, głównie modułów. Zatem do analiz produktów jako całości bardziej będą nadawały się te pierwsze, a do analiz poszczególnych części systemów – drugie.

Drugim wyróżnikiem możliwości zastosowania poszczególnych baz jest obecność parametrów opisujących wybrany fragment przedsięwzięcia. W tabeli 1 umieszczono dane o najważniejszych parametrach występujących w poszczególnych bazach. Wyróżnić można kilka możliwych zastosowań tych baz:

- szacowanie wielkości oprogramowania – tam, gdzie umieszczono szczegółowe dane o poszczególnych modułach (głównie bazy NASA),
- szacowanie nakładów – tam, gdzie umieszczono dane o nakładach (ISBSG i niektóre z PROMISE),
- szacowanie defektów – tam, gdzie umieszczono dane o liczbie defektów, fakcie ich występowania w poszczególnych modułach, historii defektów i innych problemów (prawie wszystkie bazy z wyjątkiem niektórych z PROMISE),
- analiza zależności między rozmiarem, nakładami a defektami – tam, gdzie umieszczono razem te dane (ISBSG i kilka baz z PROMISE),
- analiza wpływu jakości procesu i ludzi – tam, gdzie umieszczono dane opisujące jakość procesu i ludzi (kilka z baz PROMISE).

Spośród analizowanych baz danych tylko dwie zawierają jednocześnie dane o rozmiarze projektu, nakładach, defektach i jakości ludzi i procesu:

- PROMISE: qqdefects – 31 projektów,
- PROMISE: ivvbayes – 4 projekty.

Zatem tylko w tych dwóch wypadkach można przeprowadzić zintegrowaną analizę wzajemnych zależności.

Podsumowanie

W artykule przedstawiono wyniki analizy publicznie dostępnych baz danych przedsięwzięć informatycznych. Stwierdzono, że nie istnieje baza uniwersalna – najlepiej nadająca się do dowolnych badań z zakresu inżynierii oprogramowania. Zależnie od celu badań należy wybrać odpowiednią bazę pod względem ilości danych oraz występowania potrzebnych parametrów opisujących przedsięwzięcie informatyczne. Tabele umieszczone w pracy mogą stanowić wskazówki przy wyborze odpowiedniej bazy do prac badawczych.

Literatura

Apache Software Foundation, ASF Bugzilla, 2007, http://issues.apache.org/bugzilla/report.cgi.

Boetticher G., Menzies T., Ostrand T., *PROMISE Repository of Empirical Software Engineering Data*, West Virginia University, Department of Computer Science, 2007, http://promisedata.org/repository.

Bugzilla Gentoo, 2007, http://bugs.gentoo.org.

Eclipse, Eclipse Bugs, 2007, https://bugs.eclipse.org/bugs.

Fedora Bugzilla, 2007, https://bugzilla.redhat.com.

Fenton N., Neil M., Marsh W., Hearty P., Radliński Ł., Krause P., *Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction*, Proc. 3rd Int. Workshop on Predictor Models in Software Engineering (May 20–26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 2. http://dx.doi.org/10.1109/PROMISE.2007.11.

GNOME Foundation, GNOME Bugzilla, 2007, http://bugzilla.gnome.org.

ISBSG, Repository Data Release 10, International Software Benchmarking Standards Group 2007, www.isbsg.org.

KDE e.V., KDE Bug Tracking System, 2007, http://bugs.kde.org.

Mandriva Bugzilla, 2007, http://qa.mandriva.com.

Mozilla.org, Bugzilla@Mozilla, 2007, https://bugzilla.mozilla.org.

NASA, Bug Reporting System, Goddard Space Flight Center, NASA, 2007, http://itos.gsfc.nasa.gov/~bugzilla/report.cgi.

Novell's Bugzilla, 2007, https://bugzilla.novell.com/index.cgi.

OpenOffice.org, OpenOffice.org Issues, 2007, http://qa.openoffice.org/issues/query.cgi.

Radliński Ł., Fenton N., Marquez D., Hearty P., *Empirical Analysis of Software Defect Types*, *Information Systems Architecture and Technology. Information Technology and Web Engineering: Models, Concepts & Challenges*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2007.

Radliński Ł., Fenton N., Neil M., Marquez D., *Modelling Prior Productivity and Defect Rates in a Causal Model for Software Project Risk Assessment*, „Polish Journal of Environmental Studies" 2007, vol. 16, no. 4A.

The Linux Kernel Organisation, Inc., Kernel Bug Tracker, 2007, http://bugzilla.kernel.org/report.cgi.

The Mozilla Organisation, Bugzilla Installation List, 2007, http://www.bugzilla.org/installation-list.

The Mozilla Organisation, Bugzilla, 2007, http://www.bugzilla.org.

A REVIEW OF PUBLICLY AVAILABLE DATABASES OF SOFTWARE PROJECTS

Summary

A set of reliable empirical data is often required for a scientific research. For many years in the software engineering domain such datasets were usually not easily publicly available. However, recently some repositories have been established and opened for public access. This paper focuses on the analysis of databases of software projects. I have analyzed such databases grouped in four major repositories: ISBSG, PROMISE, NASA and the databases based on Bugzilla. The contents are diverse among the databases: different parameters describing projects, different level of data granularity and different number of observations. Because of that there is no single database which could be used in each type of research analysis. Rather the aim of the research and the need for specific type of data determines which database could be used. A couple of possible types of analysis can be supported by these datasets: estimation of software size, effort and defects. A few of them allow the trade-off analysis between these factors. Some also contain data about the process and people quality. The comparison of the databases may be a useful tip when the choice of the database needs to be made.

Translated by Łukasz Radliński