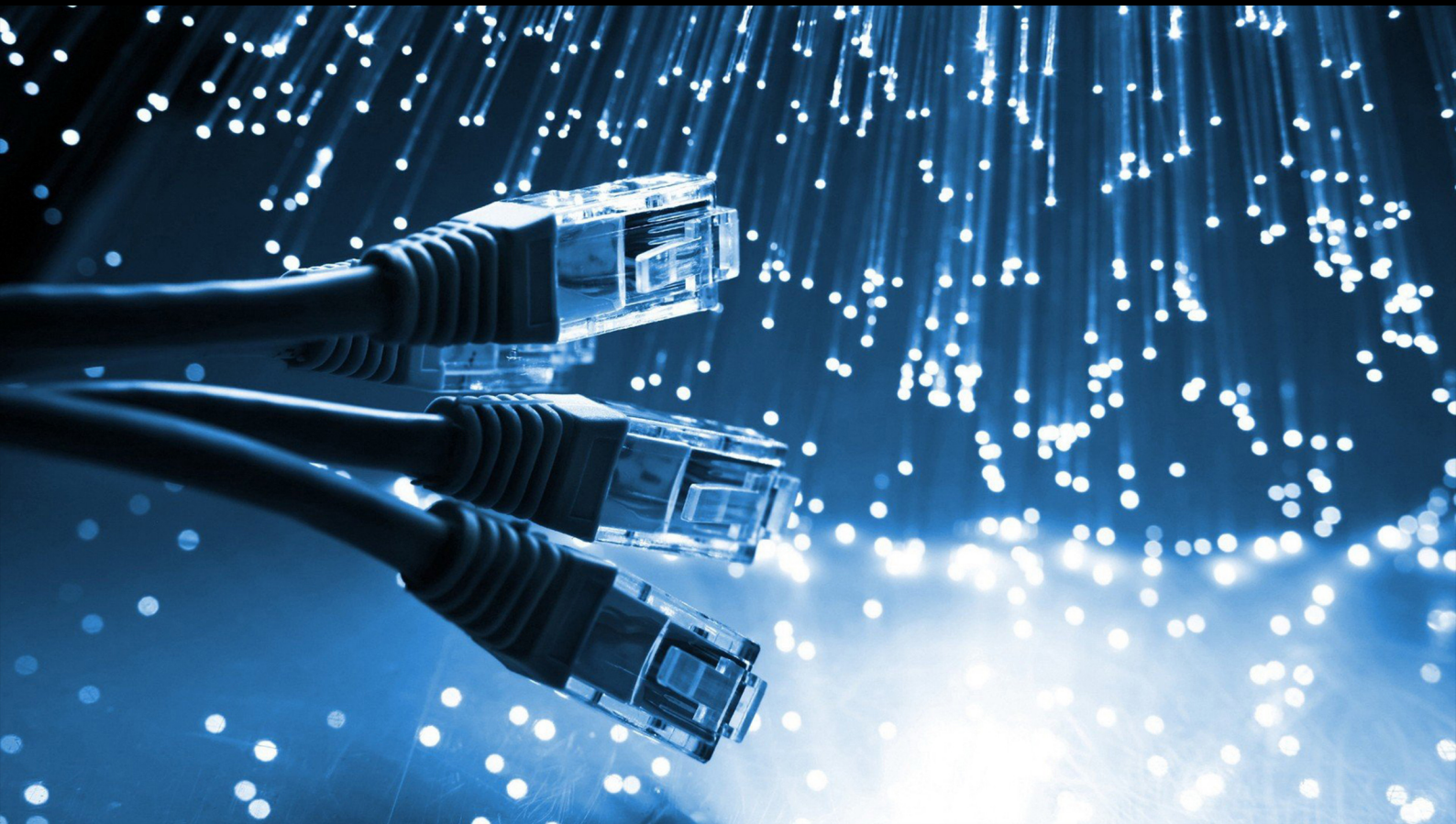


JCSI

Journal of Computer Sciences Institute

Volume 4/2017



Institute of Computer Science
Lublin University of Technology

jcsi.pollub.pl

ISSN: 2544-0764

Redakcja JCSI

e-mail: jcsi@pollub.pl
www: jcsi.pollub.pl
Instytut Informatyki
Wydział Elektrotechniki i Informatyki

Politechnika Lubelska
ul. Nadbystrzycka 36 b,
20-618 Lublin

Redaktor naczelny:

Tomasz Zientarski
e-mail: t.zientarski@pollub.pl

Redaktor techniczny:

Beata Pańczyk,
e-mail: b.panczyk@pollub.pl

Recenzenci numeru:

dr inż. Sławomir Przyłucki
dr hab. inż. Jerzy Montusiewicz, prof. PL
dr inż. Grzegorz Koziół
dr Mariusz Dzieńkowski
dr inż. Piotr Muryjas
dr inż. Elżbieta Miłosz
dr Beata Pańczyk
dr inż. Jakub Smółka

Skład komputerowy:

Piotr Misztal
e-mail: p.misztal@pollub.pl

Projekt okładki:

Marta Zbańska

JCSI Editorial

e-mail: jcsi@pollub.pl
www: jcsi.pollub.pl
Institute of Computer Science
Faculty of Electrical Engineering and
Computer Science
Lublin University of Technology
ul. Nadbystrzycka 36 b
20-618 Lublin, Poland

Editor in Chief:

Tomasz Zientarski
e-mail: t.zientarski@pollub.pl

Assistant editor:

Beata Pańczyk,
e-mail: b.panczyk@pollub.pl

Reviewers:

Sławomir Przyłucki
Jerzy Montusiewicz
Grzegorz Koziół
Mariusz Dzieńkowski
Piotr Muryjas
Elżbieta Miłosz
Beata Pańczyk
Jakub Smółka

Computer typesetting:

Piotr Misztal
e-mail: p.misztal@pollub.pl

Cover design:

Marta Zbańska

Spis treści

1. APACHE HADOOP, PLATFORMA DO GROMADZENIA, PRZETWARZANIA I ANALIZY DUŻYCH ZBIORÓW DANYCH	
MATEUSZ GIL.....	70
2. MOŻLIWOŚCI WYKORZYSTANIA BEACONÓW W REKLAMIE	
KONRAD ŁYDA, MAREK MIŁOSZ.....	76
3. PORÓWNANIE SZYBKOŚCI DZIAŁANIA WYBRANYCH FUNKCJI SKRÓTU I ALGORYTMÓW SZYFROWANIA	
DAWID GÓRNIAK, PIOTR KOPNIAK.....	82
4. PORÓWNANIE MECHANIZMÓW BEZPIECZEŃSTWA POPULARNYCH SYSTEMÓW OPERACYJNYCH URZĄDZEŃ MOBILNYCH	
MICHAŁ BURYTA, PIOTR KOPNIAK.....	87
5. POJĘCIE, MODELE I METRYKI JAKOŚCI OPROGRAMOWANIA – PRZEGLĄD	
YULIIA HOROBETS, MAREK MIŁOSZ.....	92
6. PORÓWNYWANIE ALGORYTMÓW IDENTYFIKACJI FIKSACJI ORAZ SAKKAD W ZAPISIE OKULOGRAFICZNYM DLA POTRZEB DIAGNOSTYKI MEDYCZNEJ	
MARCIN IGNATIUK, PRZEMYSŁAW ANTONI PASIECZNY, MALGORZATA PLECHAWSKA-WÓJCIK, MAGDALENA BORYS.....	99
7. ANALIZA MOŻLIWOŚCI WYKORZYSTANIA PLATFORMY MOODLE DO REALIZACJI PROCESU KSZTAŁCENIA NA KIERUNKU INŻYNIERIA SYSTEMÓW BEZPIECZEŃSTWA WEWNĘTRZNEGO W POLITECHNICE LUBELSKIEJ	
SŁAWOMIR BRONISZ, ELŻBIETA MIŁOSZ.....	104
8. OCHRONA DANYCH OSOBOWYCH W UJĘCIU PRZEPISÓW PAŃSTWA POLSKIEGO A ORGANU GENERALNEGO INSPEKTORA OCHRONY DANYCH OSOBOWYCH	
DAMIAN HARASIM.....	108
9. TEST-DRIVEN DEVELOPMENT JAKO NARZĘDZIE OPTIMALIZACJI PROCESU WYTWARZANIA OPROGRAMOWANIA NA PLATFORMIE JEE	
GRZEGORZ SOCHACKI, BEATA PAŃCZYK.....	112
10. PORÓWNANIE WYBRANYCH BIBLIOTEK DO TWORZENIA GRAFICZNEGO INTERFEJSU INTERNETOWYCH APLIKACJI JEE	
PIOTR PAWEŁ SUWAŁA, BEATA PAŃCZYK.....	117
11. ANALIZA PORÓWNAWCZA SYSTEMÓW WSPOMAGAJĄCYCH PROTOTYPOWANIE INTERFEJSÓW	
ARTUR ŁASOCHA, MAREK MIŁOSZ.....	122
12. PORÓWNANIE ALGORYTMÓW WYSZUKIWANIA GOOGLE I YANDEX	
OLEH BIHUN, MAREK MIŁOSZ.....	128
13. PORÓWNANIE MOŻLIWOŚCI I CECH WSPÓŁCZESNYCH LINUXOWYCH SYSTEMÓW PLIKÓW: EXT4, XFS, BTRFS	
BARTŁOMIEJ KOSSAK, MACIEJ PAŃCZYK.....	131
14. AKTUALNE NARZĘDZIA WYTWARZANIA OPROGRAMOWANIA NA PLATFORMIE JEE	
PAWEŁ OZDOBA, BEATA PAŃCZYK.....	137

Contents

1. APACHE HADOOP, PLATFORM FOR THE COLLECTION, PROCESSING AND ANALYSIS OF LARGE DATA SETS	
MATEUSZ GIL.....	70
2. POSSIBILITIES OF USING BEACONS IN ADVERTISING	
KONRAD ŁYDA, MAREK MIŁOSZ.....	76
3. COMPARING THE SPEED OF THE SELECTED HASH AND ENCRYPTION ALGORITHMS	
DAWID GÓRNIAK, PIOTR KOPNIAK.....	82
4. COMPARISON OF THE SECURITY MECHANISMS OF POPULAR OPERATING SYSTEMS FOR MOBILE DEVICES	
MICHAŁ BURYTA, PIOTR KOPNIAK.....	87
5. THE CONCEPT, MODELS AND METRICS OF SOFTWARE QUALITY – AN OVERVIEW	
YULIIA HOROBETS, MAREK MIŁOSZ.....	92
6. COMPARISON OF FIXATION AND SACCADIC IDENTIFICATION ALGORITHMS IN THE EYE TRACKING RECORDINGS FOR MEDICAL DIAGNOSIS	
MARCIN IGNATIUK, PRZEMYSŁAW ANTONI PASIECZNY, MALGORZATA PLECHAWSKA-WÓJCIK, MAGDALENA BORYS.....	99
7. THE ANALYSIS OF USAGE POSSIBILITIES OF THE MOODLE PLATFORM TO EXECUTE THE TRAINING PROCESS IN AN "ENGINEERING OF INTERNAL SECURITY SYSTEMS" FIELD OF STUDY AT LUBLIN UNIVERSITY OF TECHNOLOGY	
SŁAWOMIR BRONISZ, ELŻBIETA MIŁOSZ.....	104
8. PERSONAL DATA PROTECTION IN TERMS OF PROVISIONS OF THE POLISH STATE AND THE AUTHORITY OF THE GENERAL FOR PERSONAL DATA PROTECTION	
DAMIAN HARASIM.....	108
9. TEST-DRIVEN DEVELOPMENT AS A TOOL TO OPTIMIZE THE JEE PROGRAMMING	
GRZEGORZ SOCHACKI, BEATA PAŃCZYK.....	112
10. COMPARISON OF SELECTED GRAPHICAL WEB INTERFACE LIBRARIES USED IN JEE APPLICATIONS	
PIOTR PAWEŁ SUWAŁA, BEATA PAŃCZYK.....	117
11. COMPARATIVE ANALYSIS OF SUPPORT SYSTEMS FOR INTERFACES PROTOTYPING	
ARTUR ŁASOCHA, MAREK MIŁOSZ.....	122
12. COMPARISON GOOGLE'S AND YANDEX'S SEARCH ALGORITHMS	
OLEH BIHUN, MAREK MIŁOSZ.....	128
13. COMPARISON OF FEATURES AND CAPABILITIES OF MODERN LINUX FILE SYSTEMS EXT4, XFS, BTRFS	
BARTŁOMIEJ KOSSAK, MACIEJ PAŃCZYK.....	131
14. THE CURRENT SOFTWARE TOOLS FOR THE JEE PLATFORM	
PAWEŁ OZDOBA, BEATA PAŃCZYK.....	137

Apache Hadoop, platforma do gromadzenia, przetwarzania i analizy dużych zbiorów danych

Mateusz Gil*

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule przedstawiono możliwości wykorzystania platformy Hadoop w zarządzaniu wielkimi zbiorami danych. Na podstawie dostępnych źródeł przedstawiono rozwój wydajności aplikacji. Dodatkowo zostały opisane organizacje, które dzięki wdrożeniu tego oprogramowania odniosły sukces w świecie Internetu.

Słowa kluczowe: Hadoop; Big Data; Analiza danych

*Autor do korespondencji.

Adres/ e-mail: mateusz.gil5@gmail.com

Apache Hadoop, platform for the collection, processing and analysis of large data sets

Mateusz Gil*

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents the possibilities of using Hadoop platform to manage large data sets. The development of application performance has been shown based on available sources. Additionally, the article describes the organizations that have been successful in the Internet thanks to the implemented software.

Keywords: Hadoop; Big Data; Analysis of the data

*Corresponding author.

E-mail address: mateusz.gil5@gmail.com

1. Wstęp

W wyniku prac prowadzonych przez Douga Cuttina w projekcie Apache Nutch, została opracowana nowa platforma Apache Hadoop, która w styczniu 2006 roku została przeniesiona, jako odrębny projekt open source w tej organizacji. Dzięki darmowemu dostępowi tego produktu wiele firm zdecydowało się na tworzenie własnych hurtowni danych. Analiza pozwala wykorzystywać dane od strony spostrzeżeń, które w rezultacie pomagają podejmować lepsze decyzje oraz podsuwają strategiczne posunięcia biznesowe. Celem niniejszego artykułu jest przedstawienie progresu rozwoju platformy Hadoop, dostępnych modułów zintegrowanych z oprogramowaniem, korzyści wynikające z używania Apache Hadoop [1].

2. Charakterystyka platformy Hadoop

Firma Yahoo w lutym 2006 roku zaczęła rozwijać projekt Hadoop na potrzeby zastosowań produkcyjnych. Rezultaty prac zostały przedstawione w 2008 roku w artykule „Yahoo! Launches World's Largest Hadoop Production application”. Zawarto w nim informacje o tym, że dane zastosowane w cyklu produkcyjnym generowane były za pomocą klastra Linux mającego 10.000 rdzeni, opartego na platformie Hadoop. Gromadzenie tych danych przez komponent WebMap zaczynało się od każdej strony indeksowanej przez

Yahoo!. Tworzyło bazę danych wszystkich stron i witryn w Internecie oraz gromadziło informacje dotyczące każdej strony i witryny.

Wykorzystywane w tym czasie zasoby były bardzo duże:

- wielkość bazy produkcyjnej wynosiła ponad 300TB skompresowanych danych,
- ilość rdzeni wykorzystywanych do każdego zadania wynosiła ponad 10 tysięcy,
- przestrzeń dyskowa używana w klastrze produkcyjnym to ponad 5 petabajtów [2].

2.1. Zakres możliwości platformy

Możliwości tej platformy wykorzystane zostały przez znany na całym świecie magazyn. „New York Times” postanowił opublikować za darmo wszystkie artykuły od 1851-1922 w formacie PDF, ich łączna wartość to 11 milionów dolarów. Wygenerowanie takiej ilości artykułów w wersji PDF, które składają się z wielu mniejszych obrazów TIFF zajmuje bardzo dużo pracy. Do przetworzenia 4 terabajtów archiwów wykorzystano chmurę obliczeniową Amazon EC2. Cały proces przetwarzania zajął niecałe 24 godziny przy użyciu 100 maszyn [3]. W maju 2008 roku Owen O'Malley z firmy Yahoo! przedstawił rekordowy czas sortowania terabajta przez Hadoop. Klastrer, który został wykorzystany do sortowania to:

- 910 węzłów,

- 4 rdzenie Xeon 2.0 GHz dla każdego węzła,
- 8 GB RAM dla każdego węzła,
- 1 Gb Ethernet,
- Red Hat Enterprise Linux Server.

Sortowanie zakończyło się po 209 sekundach (3,48 minuty) [4].

W tym samym roku firma Google opublikowała kolejny rekord sortowania terabajta danych, dzięki wykorzystaniu zaimplementowanego modelu MapReduce, który wynosił 67 sekund dzięki wykorzystaniu 1000 węzłów. Sortowanie 1 petabajta danych zajęło 6,02 godziny przy wykorzystaniu 4000 węzłów klastra. W kwietniu 2009 roku Yahoo! poinformowało o kolejnym zmniejszeniu czasu sortowania jednego terabajta danych do 62 sekund. Klaster użyty do wykonania operacji to:

- 4000 węzłów,
- 2 czterordzeniowe procesory Xeon 2.5 GHz dla każdego węzła,
- 8GB RAM dla węzła,
- 1 Gb Ethernet,
- Red Hat Enterprise Linux Server.

W każdym roku dzięki większym możliwościom sprzętowym czasy sortowania były skracane. Z najnowszych danych opublikowanych w 2013 roku przez firmę Yahoo! wynika, że przy wykorzystaniu Hadoopa sortowanie 102,5 terabajtów danych zajęło 4,328 sekundy, czyli 1,42 TB na minutę [7].

2.2. Cechy charakterystyczne

Głównymi cechami charakterystycznymi dla Hadoopa są:

- elastyczność w przetwarzaniu danych,
- łatwa skalowalność,
- odporność na błędy,
- szybkość przetwarzania danych,
- opłacalność rozwiązania,
- rozbudowany ekosystem.

Elastyczność w przetwarzaniu danych to możliwość zarządzania dużą ilością filmów, grafik i tekstu w formach strukturalnych i niestructuralnych. Obsługa niestructuralnych danych stanowiła dla wielu organizacji wyzwanie. W tym przypadku większość danych była ignorowana ze względu na brak odpowiedniej technologii, aby je analizować [6].

Łatwa skalowalność jest ogromnym atutem platformy. Nowe węzły mogą być bardzo łatwo dodane do systemu, dzięki czemu rozbudowa przestrzeni dyskowej wraz z rosnącymi potrzebami nie stanowi problemów dla istniejących systemów i programów.

Odporność na błędy Hadoop umożliwia dzięki przechowaniu danych w Hadoop Distributed File System, gdzie są one automatycznie odtworzone dodatkowo w dwóch miejscach. Jeśli jeden lub dwa węzły ulegną uszkodzeniu, plik dostępny jest na trzecim węźle. Daje to bardzo wysoki poziom odporności na uszkodzenia. Poziom replikacji danych jest konfigurowalny, co sprawia, że Hadoop jest niezwykle niezawodny w przechowywaniu danych. Nawet, jeśli cały węzeł zostanie utracony, system automatycznie odtworzy go w innej lokalizacji i będzie kontynuował przetwarzanie danych [8, 9].

Szybkość przetwarzania danych, których czas trwania w tradycyjnych procesach ETL może trwać nawet wiele tygodni, ponieważ załadowanie dużej ilości danych wymaga przeanalizowania tych danych w czasie rzeczywistym. Hadoop jest bardzo dobry do wysokonakładowego przetwarzania wsadowego, ponieważ pozwala na wykonywanie przetwarzania równoległego. Może wykonywać przetwarzanie nawet 10 razy szybciej niż na pojedynczym serwerze lub wątku mainframe [8, 9].

Apache Hadoop został stworzony, żeby dać formom możliwość dostępu do wielkiej ilości danych poprzez Internet. Według analityków koszt systemu zarządzania danymi w tym sprzętu, oprogramowania i innych wydatków to około 1000 dolarów za jeden terabajt. Jest bardzo dobrze dostosowany do analitycznych programistów oraz małych i dużych organizacji. Wyposażony jest w zestaw narzędzi i technologii potrzebnych do wytwarzania i przetwarzania danych. Ekosystem Hadoopa pochodzi z różnych projektów między innymi:

- MapReduce,
- Hive,
- HBase,
- Heca,
- HCatalog,
- Apache Pig.

W miarę rozwoju rynku dodawane są kolejne nowe narzędzia i technologie [9].

3. Mechanizmy działania aplikacji

Platforma Hadoop składa się z wielu komponentów, które usprawniają tworzenie i zarządzanie wielkich zbiorów danych. Wszystkie rozwijane są w fundacji Apache, której celem jest wspieranie projektów Open Source.

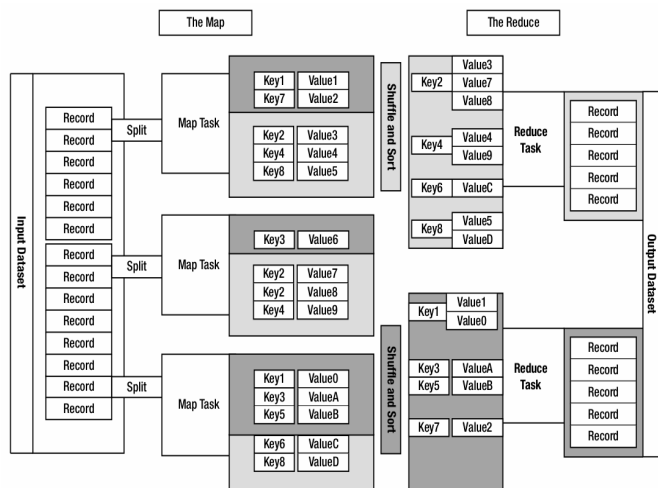
3.1. Komponent MapReduce

Pierwszym z podstawowych mechanizmów, na których opiera się platforma Hadoop to model MapReduce, który wykorzystywany jest do przetwarzania dużych zbiorów danych w sposób równoległy. W tym modelu Hadoop umożliwia uruchamianie programów w wielu językach programowania, między innymi są to: Java, Ruby, Python. Dokonanie analizy dużych zbiorów danych jest możliwe dla każdego, kto posiada wystarczającą liczbę węzłów [1]. Podstawowym założeniem jest zasada „dziel i rządź”, gdzie jeden problem dzielony jest na kilka poszczególnych podzadań. Takie podejście jest bardzo wydajne, gdyż podzadania te wykonywane są równolegle. W idealnym przypadku zadanie, którego wykonanie zajęłoby 1000 minut, może być przetwarzane w ciągu 1 minuty poprzez 1000 równoległych podzadań [10, 11]. Algorytm MapReduce zawiera dwa podstawowe zadania map i reduce. Rysunek 1 przedstawia sposób działania MapReduce.

Map to krok przetwarzania danych wejściowych. Pobiera on dane wejściowe w postaci plików lub katalogu przechowywanego w HDFS, a następnie rozkłada je na mniejsze zestawy danych. Plik wejściowy przekazywany jest do funkcji mapowania linia po linii. Kolejnym krokiem jest przesłanie podzbiorów do węzłów. W tym etapie dane są sortowane i przydzielane do węzłów, które mogą dokonać kolejnego porcjowania danych lub przekształcić i zwrócić na

wyjście odpowiedź danego zapytania. Dane uzyskane na wyjściu w etapie mapowania można również poddawać kompresji w trakcie zapisu na dysku, co przyspiesza zapis na dysk oraz zmniejsza ich wielkość.

Reduce zbiera wszystkie odpowiedzi z węzłów a następnie scala w jeden wynik tworząc nowy zestaw wyjściowy, który zostanie zapisany w HDFS [10, 11].



Rys. 1. Schemat modelu MapReduce [11].

Hadoop w modelu MapReduce dostarcza informacje o środowisku, na którym operacja jest wykonywana oraz pozwala sprawdzić nazwę pliku, który jest w trakcie przetwarzania. Każda operacja może zweryfikować, w której próbie jest wykonywana.

Powolne wykonywanie operacji może być powodem problemów sprzętowych bądź nieprawidłowej konfiguracji. Hadoop namierza operacje wykonujące się wolniej niż w czasie prognozowanym na początku. Następnie uruchamiany zostaje awaryjny egzemplarz identycznej operacji. Proces opisany w poprzednim zdaniu to wykonywanie *spekulacyjne operacji*. Procedury nie są duplikowane i uruchamiane w kolejnym wątku, natomiast rozpoczynane są dla małego fragmentu operacji, tych, które wykonują się znacznie dłużej w stosunku do założonego czasu średniego. Po zakończeniu operacji sukcesem wszystkie duplikaty są zamykane. Jeżeli zostanie uruchomiony duplikat, a operacja główna zakończy się szybciej, dodatkowy proces zostanie zakończony. Analogicznie rozwiązywany jest problem, gdy duplikat pierwszy zakończy operację, wtedy zamykana zostaje pierwotna procedura [1].

3.2. Komponent HDFS

Kolejnym z podstawowych elementów jest Hadoop Distributed Files System (HDFS), czyli rozproszony system plików. Interfejs był wzorowany na systemie plików Unix. Zaprojektowany został do przechowywania wielkich zbiorów danych rozproszonych w sieciach maszyn, co wiąże się programowaniem sieciowym. Głównym zadaniem jest uzyskanie niezawodnego systemu plików, który po awarii węzłów zachowa wszystkie dane. Dzięki możliwości przechowywania danych na wielu serwerach zasób dyskowy może rosnąć wraz z zapotrzebowaniem. Przykładowo klastr

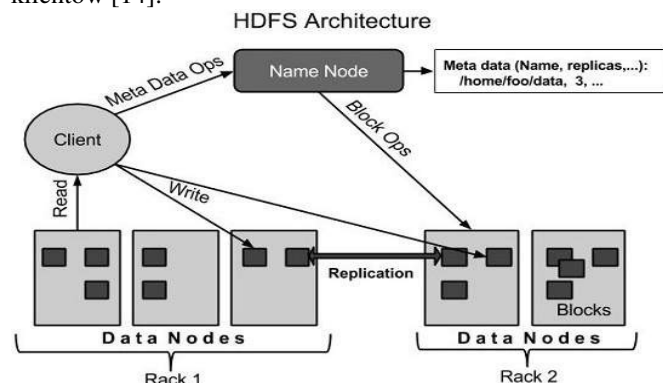
firmy Yahoo! zaalokowany jest na 40 000 serwerach i przechowuje 40 petabajtów danych [1].

W tym systemie plików wszystkie serwery są w pełni połączone i komunikują się ze sobą za pomocą protokołów TCP. HDFS nie opiera się na mechanizmach ochronnych RAID, natomiast tak, jak w Google File System zawartość plików jest powielana na wielu węzłach danych [13].

HDFS przechowuje oddzielnie metadane systemu plików i danych aplikacji. Architektura działania przedstawiona jest na rysunku 2. Odróżniane są dwa rodzaje węzłów: węzeł nazw oraz węzeł danych.

W rekordzie nazw przechowywane jest drzewo systemu plików oraz metadane plików i katalogów. Zawartość pliku jest podzielona na kilka bloków, zazwyczaj 128 megabajtowych. Każdy blok plików jest niezależnie powielany na dysku lokalnym. Metadane przechowywane są w dwóch plikach na dysku lokalnym: *drzewo nazw* i *dziennik zmian* [15].

Każda transakcja zainicjalizowana przez klienta rejestrowana jest w dzienniku. Plik jest opróżniany, a następnie synchronizowany przed wysłaniem do klienta. W tym przypadku punkt kontrolny nie jest konwersowany. Zmiana punktu kontrolnego następuje podczas restartu systemu lub poprzez działania administratora systemu. Podczas uruchamiania węzeł nazw inicjuje obraz przestrzeni nazw z punktu kontrolnego, który następnie zostaje odtworzony. Nowy punkt kontrolny i plik dziennika są zapisywane do katalogów przed rozpoczęciem obsługi klienta [14].



Rys. 2. Architektura Hadoop Distributed Files System [15].

Dla zwiększenia niezawodności nadmiarowe kopie punktów kontrolnych przechowywane są na wielu niezależnych woluminach lokalnych i zdalnych serwerach NAS. Podczas gdy węzeł nie będzie mógł zapisać dziennika do jednego z katalogów, zostaje on automatycznie usunięty z listy folderów przechowywania. Węzeł nazw przetwarza żądania jednocześnie dla wielu klientów. Zapisywanie transakcji na dysk jest w tym przypadku problemem, ponieważ inne wątki muszą czekać aż zainicjowana procedura zostanie zakończona. Dla zoptymalizowania procesu zapisuje się kilka transakcji jednocześnie.

Podczas normalnej pracy systemu, węzeł danych wysyła informację o dostępności i prawidłowym działaniu, co trzy sekundy do węzła nazw, aby potwierdzić czy komunikacja jest prawidłowa. Jeśli podczas weryfikacji nie będzie prawidłowego połączenia między nimi i nie zostanie nawiązane w ciągu dziesięciu minut, węzeł nazw zmienia status węzła danych na wycofany z eksploatacji. Przy

sprawdzaniu połączenia między węzłami sprawdzane są również informacje o całkowitej pojemności, ilość pamięci w użyciu oraz liczby transferów w toku. Te dane są wykorzystywane w podejmowaniu decyzji i alokacji bloków oraz równoważenia obciążenia w węzłach użytkownika.

Polecenia sprawdzające działanie systemu są bardzo istotne dla utrzymania integralności systemu. W związku z tym ważne jest utrzymanie wysyłania nawet dużej ilości impulsów do węzła nazw. Może on odbierać i przetwarzać tysiące informacji na sekundę bez wpływu na inne operacje [13, 14].

Dostęp użytkownika do systemu plików odbywa się za pomocą klienta HDFS. Podobnie do większości Hadoop Distributed File System obsługuje operacje odczytu, zapisu i usuwania plików oraz operacje tworzenia i usuwania katalogów. Użytkownik za pomocą klienta osiąga dostęp odwołując się do plików i katalogów ścieżkami łącząc się z węzłem nazw oraz węzłami danych. Konsument nie otrzymuje informacji, że metadane i przechowywane pliki znajdują się na różnych serwerach. W przeciwieństwie do standardowych systemów plików HDFS zapewnia interfejs AIP, który uwidacznia lokalizację ciągu bloku plików. Dzięki temu MapReduce może zaplanować zadania do miejsca, gdzie znajdują się dane, poprawiając wydajność odczytu [1, 13].

HDFS służy do niezawodnego przechowywania dużych plików między węzłami. Replikuje bloki plików w celu odporności na uszkodzenia. Komunikacja między dwoma węzłami danych w różnych instancjach jest zazwyczaj wolniejsza niż w obrębie jednego środowiska. Aplikacja może określić liczbę powtórzeń replikowanego pliku. Wartość można określić podczas tworzenia pliku lub w trakcie działania systemu. W otworzonym pliku może zapisywać tylko jeden klient, natomiast odczytywać z niego dane może wielu klientów. Węzeł nazwy podejmuje wszystkie decyzje dotyczące replikacji [13, 16]. Nie wszystkie bloki danych mogą być równomiernie rozłożone na węzłach danych. System plików wspiera ich równoważenie za pomocą różnych modeli. Jeden może przenieść bloki na inny węzeł automatycznie, gdy ilość wolnego miejsca na węzle danych spadnie poniżej ustalonego progu. Inny model może dynamicznie tworzyć dodatkowe repliki i zrównoważenia bloków w klastrze, jeśli wystąpi niespodziewany wzrost zapotrzebowania na dany plik.

Jedną z najczęstszych przyczyn wykorzystywania równoważenia jest dodanie nowych węzłów danych w klastrze. Podczas alokacji nowych bloków, węzeł nazwy analizuje różne parametry przed wyborem węzłów danych, są to między innymi:

- jednolite rozłożenie danych w węzłach,
- zmniejszenie przekroju sieci I/O,
- zapobieganie utracie danych podczas awarii.

Funkcja równoważenia klastra HDFS jest jednym z mechanizmów wykorzystywanych do utrzymania integralności danych. Kolejnym sposobem jest sprawdzanie sum kontrolnych zawartości plików. Przechowywane są one w oddzielnych, ukrytych plikach w tej samej przestrzeni nazw, które znajdują się w dzienniku transakcji każdego węzła [16].

3.3. Komponent Apache YARN

Do zarządzania zasobami klastra w systemie Hadoop wykorzystywany jest Apache YARN (ang. *Yet Another Resource Negotiator*). Jest to system do zarządzania aplikacjami rozproszonymi. Łączy w sobie wspólne usługi Hadoop: podstawowe biblioteki, HDFS i Map Reduce. YARN składa się z menadżera, który jest jeden na cały klastr oraz menadżera węzłów odpowiadającego za monitorowanie i uruchamianie kontenerów. Zarządca decyduje o kolejności przetwarzania zasobów wśród wszystkich aplikacji. Współpracuje z menadżerem węzłów oraz zarządcą aplikacji. Menadżer węzłów przyjmuje zapytania od węzła menadżera zasobów oraz zarządza dostępnymi zasobami w jednym węzle. Jest odpowiedzialny za kontenery i monitorowanie wykorzystania ich zasobów np. CPU, pamięci RAM, zasobów dyskowych, monitorowanie sieci. Zarządca aplikacji wykonuje operacje w kontenerze i zwraca wynik do klienta lub żąda zasobów, jeśli są niewystarczające od menadżera zasobów [17, 18].

3.4. Komponent Apache HBase

Apache HBase jest to jeden z modułów platformy Apache Hadoop. Umożliwia wykonywanie operacji odczytu i zapisu w czasie rzeczywistym w ogromnych zbiorach danych. Celem tego projektu jest hosting dużych tabel, posiadających miliardy wierszy i miliony kolumn. Jest to rozproszona, nie relacyjna baza danych. Pliki przechowywane są w systemie HDFS. HBase to rodzaj bazy NoSQL, co oznacza, że nie jest RDBMS, który obsługuje SQL. Od strony technicznej HBase jest tak naprawdę bardziej magazynem danych niż bazą danych, ponieważ brakuje wielu funkcji, które można znaleźć w SQL, takich jak indeksy pomocnicze, wyzwalacze, itp. Istnieje wiele funkcji, które posiada, takim przykładem może być liniowe i modułowe skalowanie. RDBMS też można skalować, jednak tylko do pewnego momentu, w szczególności uzależnione jest to od wielkości jednej bazy danych [26, 28]. HBase spójnie zapisuje i odczytuje dane. Doskonale nadaje się do zadań takich jak agregacja. Tabele są replikowane w klastrze za pomocą regionów, które można automatycznie podzielić i ponownie przekazać wraz z przyrostem bazy danych. Usługa zintegrowana jest z rozproszonym systemem plików HDFS. Obsługuje masowe przetwarzanie poprzez MapReduce. Dostępny jest łatwy w obsłudze interfejs API oraz komunikację REST. Zapewnia wbudowaną stronę web dla analizy operacyjnej. Moduł ten nie jest dobrym rozwiązaniem dla każdej bazy danych, głównym wymogiem jest duża liczba danych zaczynając od setek milionów lub miliardów wierszy. Drugim warunkiem jest praca bez dodatkowych funkcji, które zapewnia RDBMS. Aplikację zbudowaną przed RDBMS nie można przenieść na HBase poprzez zmianę sterownika JDBC. Trzecim wymaganiem jest odpowiednia ilość sprzętu, pozwalająca na uruchomienie klastra [28]. HDFS jest to system rozproszonych plików, który jest doskonały do przechowywania dużych plików. Jednak nie jest to system ogólnego przeznaczenia, oraz nie zapewnia szybkiego wyszukiwania poszczególnych rekordów w plikach. HBase jest oparty na systemie HDFS i zapewnia szybkie wyszukiwanie i aktualizację dużych tabel [27].

3.5. Komponent Apache Pig

Apache Pig jest kolejną aplikacją wchodzącą w rodzinę Apache Hadoop. To narzędzie jest używane do analizowania dużych zbiorów danych. Można wykonywać wszystkie operacje manipulacji danych w usłudze Hadoop. Implementowane są dzięki językowi wysokiego dostępu Pig Latin.

Za pomocą tego języka programiści w łatwy sposób mogą wykonywać implementację bez pisania złożonych kodów w języku Java. Czas rozwoju skraca się prawie 16 razy [28].

Pig jest językiem skryptowym, służącym do przetwarzania dużych zbiorów danych. Jest pomocny programistom tworzącym zapytania, ponieważ pozwala w trakcie pisania badać struktury danych. Największą zaletą jest możliwość wykonania przykładowego skryptu na zbiorze danych, dzięki temu można zweryfikować czy skrypt nie posiada błędów, które mogą wyniknąć w trakcie przetwarzania [1].

4. Zastosowanie Hadoop przez duże firmy

Największą firmą używającą tego oprogramowania jest Facebook. Według publikacji z 2010 roku, którą zamieścił jeden z inżynierów Dhrub Borthakur wielkość klastra hurtowni danych wzrasta bardzo szybko w przeciągu ostatnich lat. Wykorzystywane są różne elementy oprogramowania open source między innymi Apache Hadoop, Apache Hive, Apache Thrift HBase. Apache Hadoop w hurtowni danych Facebooka używany jest w trzech typach systemów: w magazynie danych, analityce internetowej oraz jako magazyn dla rozproszonej bazy danych i tworzenia kopii zapasowych baz danych MySQL. Klastry hurtowni danych są w dużym stopniu systemami przetwarzania wsadowego, gdzie nacisk kładzie się przede wszystkim na skalowalność i wydajność. Model blokowy węzła może być skalowany do 30 petabajtów, możliwie największego pojedynczego klastra Hadoop. Największe skupisko maszyn obejmuje trzy tysiące maszyn. W tym magazynie przetwarzanych jest około 2 petabajtów dziennie dla zadania skanowania. Rozproszona baza danych Apache HBase używana jest przez jedną z aplikacji do przechowywania wiadomości w usłudze Hadoop.

Kolejny przypadek użycia to zbieranie dzienników aktywności w czasie rzeczywistym z serwerów sieci web, które są przesyłane strumieniowo bezpośrednio do klastrów Hadoop. Trzeci przypadek użycia platformy to archiwizacja baz danych MySQL. Rozwiązanie umożliwia szybkie tworzenie kopii zapasowych i odzyskiwanie danych, które przechowywane są w systemie plików HDFS. Zmniejsza to koszty utrzymania i wdrażania archiwizacji petabajtów danych [19].

4.1. Użycie platformy Hadoop przez portal Facebook

W raporcie opublikowanym w 2015 roku, ilość użytkowników korzystających z Facebooka w przeciągu ostatnich 5 lat wzrosła do ponad 1,2 miliarda osób. W raporcie „*The Social Media Marketing Industry*” stwierdzono, że Facebook jest największą marketingową platformą społeczną.

Każdego dnia wymienianych jest 10 miliardów wiadomości, 4,5 miliarda „polubień”. 350 milionów dodawanych jest nowych zdjęć. Dzięki temu Facebook posiada mnóstwo informacji o każdym użytkowniku. Poza

firmą Google, Facebook jest jedyną firmą, która posiada wysoki poziom wiedzy o swoich użytkownikach. Coraz większa liczba osób korzystających z portalu wymaga inwestowania przez firmę w sprzęt, który będzie umożliwiał gromadzenie, przechowywanie i analizę tych danych. Dzięki wykorzystaniu śledzenia „ciasteczek”, jeśli użytkownik jest zalogowany, a jednocześnie przegląda strony internetowe, Facebook może śledzić odwiedzane strony. Jedną z najnowszych funkcji rozpoznawania twarzy i przetwarzania obrazu, pozwala śledzić użytkowników w całej sieci i na innych kontach Facebooka poprzez wykorzystanie danych obrazu, który jest przechowywany wraz z danymi udostępniania dla użytkowników.

Badania przeprowadzone przez naukowców z Cambridge University i Microsoft Research pokazują, że dzięki analizie naciśniętych przycisków „Lubię to!” bardzo dokładnie można przewidzieć u użytkowników orientację seksualną, zadowolenie z życia, poziom inteligencji, stabilność emocjonalną, religię, używanie alkoholu oraz narkotyków, status związku, wiek, płeć, rasę, poglądy polityczne oraz wiele innych [20].

Od lipca 2012 roku największy klaster HDFS z wykorzystaniem platformy Hadoop przez firmę Facebook zajmuje ponad 100 petabajtów miejsca na dysku. Jeden z kilku klastrów obsługiwany przez tę firmę obejmuje ponad cztery tysiące maszyn. Statystyki Facebooka podają, że dziennie powstaje:

- 20 TB skompresowanych nowych danych,
- 3 PB skompresowanych skanowanych danych [21].

4.2. Użycie aplikacji Hadoop przez stronę Twitter

Kolejnym gigantem społecznościowym wykorzystującym Hadoop jest firma Twitter, która wymaga dużych żądań dotyczących przechowywania i przetwarzania danych. Platforma zastosowana jest przede wszystkim do przechowywania skompresowanych danych. Bardzo dobry współczynnik kompresji równoważny jest z prędkością wykonywania w Hadoop. Zadania są zazwyczaj powiązane z operacjami wejścia-wyjścia dysków, algorytmy kompresji gzip i bzip2 są tak intensywne, że są ograniczane przez procesor [22].

4.3. Wykorzystanie Hadoop na potrzeby LinkedIn

Następną platformą społecznościową skupiającą wielu użytkowników oraz wykorzystującą skomplikowane algorytmy do analizy danych jest LinkedIn. Duża nie tylko pod względem przychodów lub ilości użytkowników, ale również pod względem ilości danych. Przetwarza tysiące zadań każdego dnia. Śledzi każde działanie użytkowników. Portal używa danych z profili użytkowników i wykonanych działań do budowy obrazu użytkownika oraz jego połączeń. Platforma wie, w jaki sposób łączyć użytkowników, gdzie dany użytkownik powinien ubiegać się o pracę oraz potrafi łączyć umiejętności użytkownika z rówieśnikami żeby znaleźć wymarzoną pracę. Apache Hadoop stanowi integralną część warunków technicznych w LinkedIn. Od 6 maja 2016 zespół stanowi 407 wykwalifikowanych pracowników odpowiadających za platformę Hadoop. Wszystkie dane pochodzące z innych źródeł są przenoszone z innej lokalizacji do scentralizowanego systemu, a następnie przetwarzane i analizowane przez Hadoop [23].

4.4. Funkcjonowanie wyszukiwarki Yahoo w oparciu o rozwiązanie aplikacji Hadoop

Jednym z największych amerykańskich portali jest Yahoo. To jedna z najlepszych po Google wyszukiwarka w świecie Internetu. Hadoop wykorzystywany jest w czasie rzeczywistym. Klaster stanowi 42000 węzłów w 20 klastrach YARN z 600 petabajtami danych w systemie HDFS. Platforma używana jest do blokowania około 20,5 miliardów wiadomości dziennie, sprawdza i wprowadza do swojego serwera pocztowego. Dzięki Hadoop zdolność wykrywania spamu wzrosła na serwerach Yahoo. Głównymi użytkownikami usługi Hadoop Yahoo są analitycy oraz naukowcy, którzy masowo korzystają z zaawansowanych technologii analizy, w celu optymalizacji procesów biznesowych. We własnych systemach Yahoo przechowuje 140 petabajtów danych, które w Hadoop występują w trzech egzemplarzach, co daje ponad 400 petabajtów pamięci na utrzymanie systemów [24, 25].

5. Wnioski

W tym artykule ukazane zostało znaczenie platformy Hadoop w dzisiejszym świecie bardzo szybkiego wzrostu ilości danych. Wykorzystanie systemu ułatwia przechowywanie i przetwarzanie informacji za pomocą klastrów komputerowych. Celem zastosowania tej technologii była maksymalizacja wydajności analizy danych. Przetworzenie terabajtów rekordów, które w ogromnym tempie rosną wymagają rozdystrybuowania ich do różnych komputerów. Apache Hadoop pomaga w dzieleniu informacji na grupy oraz rozpropagowanie ich do węzłów. Po rozproszeniu i przetworzeniu danych następuje ich złączenie w całość. Ta technika jest to algorytm MapReduce. Badając i analizując materiały źródłowe oraz znane portale można zaobserwować znaczny rozwój platformy. Obecny sposób wykorzystania systemu informatycznego Hadoop przedstawiony na przykładzie portalów społecznościowych Facebook oraz Twitter. Stale rosnąca liczba użytkowników generuje coraz większe zapotrzebowanie przechowywania informacji. Ponadto rozwój każdego z portali oraz nowe funkcje implementowane przez twórców mogą świadczyć o niezbędności narzędzia jakim jest Hadoop.

Literatura

- [1] Tom White, Hadoop. *Kompletny przewodnik. Analiza i przechowywanie danych*, wydawnictwo Helion.
- [2] Yahoo! Launches World's Largest Hadoop Production Application, <https://developer.yahoo.com/blogs/hadoop/yahoo-launches-world-largest-hadoop-production-application-398.html>, [10.12.2016].
- [3] Self-Service, Prorated Supercomputing Fun!, http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/?_r=0, [1.12.2016].
- [4] TeraByte Sort on Apache Hadoop, <http://sortbenchmark.org/YahooHadoop.pdf>, [11.12.2016].
- [5] Sorting 1PB with MapReduce, <https://googleblog.blogspot.com/2008/11/sorting-1pb-with-mapreduce.html>, [5.12.2016].
- [6] Winning a 60 Second Dash with a Yellow Elephant, <http://sortbenchmark.org/Yahoo2009.pdf>, [7.12.2016].
- [7] GraySort and MinuteSort at Yahoo on Hadoop 0.23, <http://sortbenchmark.org/Yahoo2013Sort.pdf>, [15.12.2016].
- [8] Big Data And Hadoop – Features And Core Architecture, <http://www.dataintegration.ninja/big-data-and-hadoop-features-and-core-architecture/>, [15.12.2016].
- [9] Big Data: Hadoop to przyszłościowe rozwiązanie w dziedzinie Buisness Intelligence, <https://www.oracle.com/assets/fy16q3-one-pl-web-2861776-pl.pdf>, [9.12.2016].
- [10] Garry Turkington, Hadoop Beginner's Guide, Packt Publishing Ltd., [9.12.2016].
- [11] Hadoop – MapReduce, https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm, [9.12.2016].
- [12] Jason Venner, Pro Hadoop, Apress, [9.12.2016].
- [13] HDFS Architecture, <http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>, [20.12.2016].
- [14] The Hadoop Distributed File System, <http://www.aosabook.org/en/hdfs.html>, [20.12.2016].
- [15] Introduction to Hadoop Distributed File System Version 1.0 and 2.0, <http://www.informit.com/articles/article.aspx?p=2460260&seqNum=>, [20.12.2016].
- [16] An introduction to the Hadoop Distributed File System, <https://www.ibm.com/developerworks/library/wa-introhdhfs/>, [20 grudzień 2016].
- [17] Apache Hadoop YARN, <http://hadoop.apache.org/docs/stable2/hadoop-yarn/hadoop-yarn-site/YARN.html>, [20.12.2016].
- [18] Apache Hadoop YARN – concepts and applications, <http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/>, [20 grudzień 2016].
- [19] Looking at the code behind our three uses of Apache Hadoop, <https://www.facebook.com/notes/facebook-engineering/looking-at-the-code-behind-our-three-uses-of-apache-hadoop/46821193919/>, [21.12.2016].
- [20] How facebook is Using Big Data – The Good, the Bad, and the Ugly, <https://www.simplilearn.com/how-facebook-is-using-big-data-article>, [20.12.2016].
- [21] How Facebook users Hadoop and Hive, <http://www.hadooppoint.com/facebook-uses-hadoop-hive/>, [2015].
- [22] Hadoop at Twitter, <https://blog.twitter.com/2010/hadoop-at-twitter>, [22.12.2016].
- [23] How LinkedIn uses Hadoop to leverage Big Data Analytics, <https://www.dezyre.com/article/how-linkedin-uses-hadoop-to-leverage-big-data-analytics/229>, [22.12.2016].
- [24] How Yahoo using Hadoop in real time, <http://www.hadooppoint.com/how-yahoo-using-hadoop-in-real-time/>, [22.12.2016].
- [25] Yahoo and Hadoop: In it for the long term, <http://www.informationweek.com/database/yahoo-and-hadoop-in-it-for-the-long-term/d/d-id/1104866?>, [26.12.2016].
- [26] What is Apache Hadoop, <http://hadoop.apache.org/What+Is+Apache+Hadoop%3F>, [26.12.2016].
- [27] What is Apache HBase, <http://hortonworks.com/apache/hbase/>, [26.12.2016].
- [28] Apache HBase Reference Guide, <http://hbase.apache.org/book.html#arch.overview>, [27 grudzień 2016].

Możliwości wykorzystania beaconów w reklamie

Konrad Łyda*, Marek Miłośz

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Beacon to emiter sygnału Bluetooth Low Energy zawierającego zakodowany identyfikator. Odbiorcą takiego sygnału jest zwykle urządzenie mobilne typu smartfon czy tablet. Zainstalowana na urządzeniu mobilnym aplikacja może dostarczać jej użytkownikowi dodatkowych funkcjonalności. Technologia ta umożliwia wytwarzanie różnych systemów, w tym używanych w reklamie sprzedaży lub usług. W pracy przedstawiono prototyp systemu, który pozwala na wsparcie strategii sprzedaży sklepów wielkopowierzchniowych opartej o zjawisko „webroomingu”, gdzie klient wybiera produkty i dodaje je do swojej listy przez Internet, a następnie odbiera je w rzeczywistym sklepie. Aplikacja mobilna będzie wspomagać klienta w tworzeniu listy zakupów, wskazania miejsc składowania interesujących towarów, optymalizację drogi dojścia z wykorzystaniem agentów mobilnych. Przetestowano wpływ odległości oraz sieci Wi-Fi na działanie aplikacji.

Słowa kluczowe: beacon; aplikacje mobilne; marketing

* Autor do korespondencji.

Adresy e-mail: konrad.lyda@gmail.com, m.milosz@pollub.pl

Possibilities of using beacons in advertising

Konrad Łyda*, Marek Miłośz

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. Beacon is a device, that broadcasts its identifier via Bluetooth Low Energy signal to nearby mobile devices, like smartphones or tablets. A special application installed on user's device serves additional functionalities, that improves general user experience. This technology allows companies to develop advanced systems, that can enhance and support advertising campaigns of products and services. In this paper there is a description of the prototype of the system supporting selling special strategy for a large retail store. This strategy, called webrooming, is based on the assumption, that user is choosing the products via special webpage, and afterwards collects the order on-site, at the store. The mobile part of the system is supporting the user in creating shopping lists, showing the nearest products during shopping trip and by path optimization based on mobile agents' signal. The impact of distance and Wi-Fi network on operation of application was tested.

Keywords: beacon; mobile applications; marketing

*Corresponding author.

E-mail addresses: konrad.lyda@gmail.com, m.milosz@pollub.pl

1. Wstęp

W roku 2014 analitycy zbadali, że liczba aktywnych urządzeń mobilnych używanych przez ludzkość jest większa, niż populacja ziemi [1]. Coraz więcej codziennych czynności, wykonywanych przez ludzi, przenosi się do sfery wirtualnej. Użytkownicy urządzeń mobilnych poszukują informacji, wiadomości, oglądają telewizję, filmy, zamawiają jedzenie, robią zakupy. Rynek smartfonów w ciągu ostatnich 10 lat bardzo się rozwinął. Dzięki powstaniu systemów Android oraz iOS obecnie większość telefonów komórkowych obecnych na rynku, to właśnie urządzenia wyposażone w mobilne systemy operacyjne. Szacuje się, że w 2016 roku 31% populacji ludzkości będzie aktywnymi użytkownikami smartfonów. Jest to blisko 2,3 miliarda osób, które korzystają z urządzeń mobilnych [2].

Sklepy online stały się bardzo często pierwszym miejscem, od którego klient rozpoczyna poszukiwania pożądanego produktu. Pomimo tego, sklepy tradycyjne mogą osiągnąć bardzo duży wzrost sprzedaży wykorzystując digitalizację i podłączenie punktów sprzedaży (ang. Point Of Sale, POS) do sieci. Eksperti z firmy Deloitte szacują, że sprzedaż z wykorzystaniem technologii mobilnych w tradycyjnych

sklepach w roku 2016 dwukrotnie przekroczy sprzedaż w e-sklepach [3].

Coraz więcej sklepów detalicznych wdraża do swoich strategii marketingowych urządzenia zwane beaconami. W 2016 roku badacze z Retail System Research sprawdzili, że prawie ćwierć sklepów detalicznych zdecydowała się na wdrożenie tych urządzeń i jest zadowolona z ich wdrożenia, a 4 na 10 z nich odpowiedziało, że są one wartościową inwestycją [4], [5]. Jednakże, z tych samych badań wynika, że prawie jedna czwarta badanych sklepów nie ma w planach implementacji rozwiązania tego typu w swoich lokalizacjach.

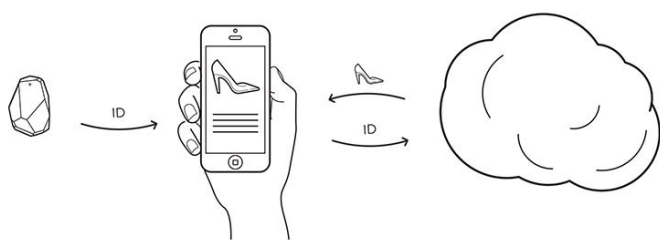
Powyższe badania wskazują za zasadne przeanalizowanie obecnych na rynku rozwiązań oraz zbadanie zaprogramowanego prototypu takiej aplikacji pod kątem praktycznego wykorzystania beaconów w symulowanych fizycznych lokalizacjach w celu wspomagania decyzji zakupowych klienta i poprawianiu jego odczuć względem tradycyjnych form nabywania towaru. Przy wykorzystaniu zbudowanego prototypu aplikacji komunikującej się z sensorami, sprawdzona zostanie możliwość wykorzystania technologii w sklepach wielkopowierzchniowych oraz dokładność określania odległości użytkownika od nadajnika.

2. Technologia beaconów

W najbardziej ogólnym znaczeniu beacon oznacza urządzenie, które ma za zadanie w pewien sposób przykuć uwagę do zadanego miejsca [6]. W połączeniu z różnymi wskaźnikami może on również przekazywać istotne informacje i w ten sposób beacons wykorzystywane np. na lotniskach, na wysokich budynkach, czy w nawigacji, np. na morzu. W świecie IT mianem beacon określa się małe urządzenia, wykorzystujące technologię Bluetooth Low Energy do emitowania swojego identyfikatora do pobliskich urządzeń [7]. W takim też znaczeniu będzie słowo „beacon” będzie używane w niniejszym artykule.

Bluetooth Low Energy (określany również jako Bluetooth LE, BLE lub Bluetooth Smart) jest rodzajem transmisji Bluetooth opracowanym specjalnie na potrzeby Internetu Rzeczy (ang. Internet of Things, IoT) [8]. W porównaniu do standardowej transmisji Bluetooth, BLE charakteryzuje się bardzo niskim zużyciem energii, co umożliwia wykorzystanie tej technologii w urządzeniach, które są przygotowywane do działania przez długi okres czasu na ograniczonych źródłach energii, takich jak baterie [8].

Beacon emituje zakodowany w jego pamięci identyfikator, aby znajdujące się w pobliżu urządzenia były go w stanie odebrać i zidentyfikować. Urządzenie nie może jednak danych odbierać. W związku z tym samo w sobie nie zagraża prywatności użytkownika korzystającego z tej technologii [9]. Aby wejść w interakcję posiadacz smartfonu musi zainstalować odpowiednią aplikację, bądź w przypadkach niektórych urządzeń i platform – aktywować odpowiednią funkcję w telefonie. Aplikacja, po odczytaniu, że znajduje się w zasięgu danego beacona, może na podstawie jego ID wykonać pewne operacje, jak np. połączenie do Internetu i pobranie szczegółowych informacji o produkcie bądź wyświetlenie informacji o aktualnej promocji.



Rys. 1. Przykładowy schemat obrazujący działanie aplikacji po znalezieniu się w zasięgu beacona (źródło: [10])

Obecnie przy wytwarzaniu i obsłudze urządzeń nadających znaczniki poprzez Bluetooth Low Energy wykorzystywane są głównie dwa protokoły: iBeacon oraz Eddystone.

2.1. iBeacon

iBeacon to protokół opracowany przez firmę Apple. Urządzenia w tym standardzie nadają jeden typ pakietu zawierający identyfikator urządzenia, składający się z trzech elementów [11]:

- 16-bajtowy ciąg znaków (ang. Universally Unique Identifier, UUID) identyfikujący dużą grupę urządzeń.
- Major – 2-bitowy ciąg znaków pozwalający na określanie mniejszych podzbiorów urządzeń (np. w danej pojedynczej lokalizacji).
- Minor – 2-bitowy ciąg znaków pozwalający na identyfikację konkretnego urządzenia (np. beacon przy wejściu do sklepu).

2.2. Eddystone

Eddystone to protokół opracowany przez firmę Google [12]. W przeciwieństwie do standardu Apple, jest otwartym protokołem, a jego specyfikacja jest dostępna publicznie. Urządzenia zgodne z Eddystone są w stanie emitować trzy różne typy pakietów [12]:

- 1) Eddystone-UID/Eddystone-EID zawiera indywidualny ciąg znaków jednoznacznie identyfikujący urządzenie: 10-bitowy klucz, grupujący pewną ilość urządzeń w spójny zbiór (ang. Namespace) oraz 6-bitowy ciąg znaków wskazujący pojedynczy beacon (ang. Instance),
- 2) Eddystone-URL, pozwala na rozsiewanie adresu URL usługi, do której chcemy skierować użytkownika będącego w zasięgu beacona,
- 3) Eddystone-TLM, umożliwia odczytanie informacji diagnostycznych danego urządzenia, takich jak stan baterii, temperatura czy liczba wysłanych pakietów od ostatniego restartu.

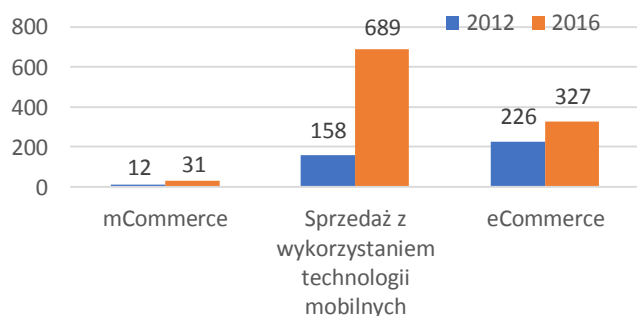
3. Zastosowanie beaconów w marketingu

Beacons mogą być wykorzystane w wielu różnych obszarach działalności komercyjnej. W reklamie, mogą służyć do wyświetlania powiadomień o promocjach, specjalnych ofertach, odbywających się niedaleko wydarzeniach, itp. Mogą zostać wykorzystane do budowania trwałych relacji i tworzenia spersonalizowanych doświadczeń dla obecnych, lojalnych klientów. Zastosowanie sieci urządzeń nadających sygnał Bluetooth umożliwia dokładniejsze lokalizowanie użytkownika będącego wewnątrz budynku, aniżeli pozycjonowanie na podstawie sygnału GPS, który w środku placówek jest skutecznie tłumiony przez ściany. Przy zastosowaniu metody z kilkoma nadajnikami w każdym pomieszczeniu, użytkownika można lokalizować nawet z dokładnością 2 metrów [13].

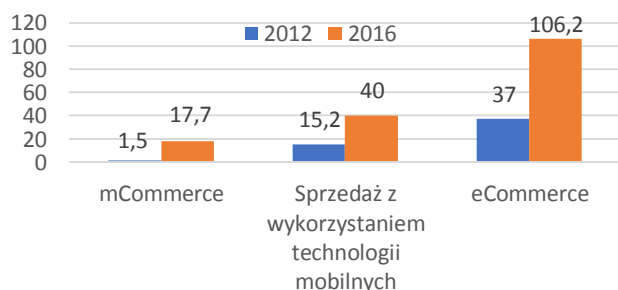
W raporcie „The dawn of mobile influence” [3] eksperci z firmy Deloitte szacują, że na terenie Stanów Zjednoczonych w roku 2016 sprzedaż w sklepach stacjonarnych, które wykorzystują technologie mobilne, zwiększy się ponad trzykrotnie, ze 158 miliardów \$ w 2012 do 689 miliardów \$ w roku 2016. Szacuje się również, że ponad dwukrotnie przekroczy ona sprzedaż poprzez sklepy online, którą to sprzedaż w 2016 szacuje się na 327 miliardów \$ [3].

Patrząc na ten sam raport, ale dotyczący terytorium Wielkiej Brytanii widać podobny trend wzrostowy w przypadku sklepów stacjonarnych, które zaadaptowały technologie mobilne. Jednak pomimo prawie trzykrotnego wzrostu, ze 15,2 miliarda £ w 2012 do 40 miliardów £ w 2016, dalej obszar ten pozostaje w tyle za sprzedażą w e-sklepach,

która wg szacunków w 2016 ma wynieść 106,2 miliarda £ [14].



Rys. 2. Szacowana przez Deloitte wielkość sprzedaży (w miliardach \$) w poszczególnych sektorach na terytorium Stanów Zjednoczonych (źródło: [3])



Rys. 3. Szacowana przez Deloitte wielkość sprzedaży (w miliardach £) w poszczególnych sektorach na terytorium Wielkiej Brytanii (źródło: [14])

3.1. Przykłady wdrożeń rozwiązań wykorzystujących beacony

Beacony mogą zostać wykorzystane w różnych rodzajach sklepów czy usług, co pozwala na szeroki wachlarz możliwości zastosowania tych urządzeń w marketingu i reklamie.

Sieć Target zdecydowała się wdrożyć beacony w celu powiadamiania swoich klientów o promocjach i rekomendacjach produktów, automatycznego sortowania listy zakupów na podstawie ścieżki klienta czy możliwości zgłoszenia potrzeby pomocy bezpośrednio z aplikacji telefonu [15].

Grupa Neiman Marcus wykorzystwała beacony w celu zapraszania użytkowników przechodzących nieopodal sklepów na specjalne wydarzenia odbywające się w salonach, jak np. porady modowe, informacje o pielęgnacji ubrań, specjalne pokazy, itp. [15]. Skorzystano z możliwości integrowania beaconów z aplikacją Passbook dostępną domyślnie w systemie iOS. Powodowało to, że kampania nie wymagała od użytkownika instalowania dodatkowego oprogramowania, ale ograniczona była do użytkowników iPhone'ów [16].

Również najważniejsza amerykańska liga baseballu, MLB, wdrożyła iBeacony. Użytkownicy, którzy przechodzą przez bramę jednego z 28 stadionów objętych wdrożeniem i mają aplikację na telefonie, otrzymują powiadomienie ze

specjalnymi ofertami, jak kupony na jedzenie, informacje o drużynach czy dostęp do powtórek wideo bezpośrednio na ekranie smartfona. Pomimo tego, że aplikacja jest jedna, każdy z zespołów może dostosować zachowanie aplikacji i wyświetlane treści do swojej bazy fanów [17].

4. Prototyp systemu

Do badań na potrzeby niniejszego artykułu opracowany został prototyp systemu, który pozwala na wsparcie strategii sprzedaży sklepów wielkopowierzchniowych opartej o zjawisko „webroomingu”, gdzie klient wybiera produkty i dodaje je do swojej listy przez Internet, a następnie odbiera je w sklepie fizycznym.

4.1. Aplikacja internetowa

Zamawianie produktów umożliwia przygotowana aplikacja internetowa. Pozwala ona na zalogowanie się na swoje konto klienta, a następnie z katalogu sklepu wybór interesujących produktów. Produkty dodawane są do listy zakupowej, która zapisywana jest na koncie użytkownika. Możliwe jest opatrzenie listy zakupów tytułem i opisem, aby w przyszłości móc ją ponownie wykorzystać.

Użytkownik ma możliwość dodawania produktów do listy bezpośrednio z ekranu przeglądu produktów, bez przeładowania aktualnej strony. Możliwe jest to dzięki zastosowaniu zapytań AJAX, które w tle wysyłają do serwera żądanie dodające dany produkt do aktywnej listy danego użytkownika.

4.2. Aplikacja mobilna

Za wsparcie klienta w sklepie odpowiedzialna jest aplikacja mobilna, która umożliwia zalogowanie się na konto użytkownika, a następnie zsynchronizowanie listy zakupów na urządzenie mobilne. W trakcie wizyty w sklepie, dzięki zastosowaniu beaconów przy odpowiednich regałach, aplikacja na bieżąco podczas przemieszczania się użytkownika sortuje listę zakupów na podstawie odległości do najbliższego regału z interesującym produktem, a także umożliwia wyświetlanie powiadomień, gdy użytkownik znajdzie się przy regale zawierającym produkt z listy.

4.3. Środowisko testowe

Do sprawdzenia prototypu aplikacji wykorzystano 4 beacony firmy Kontakt o nazwie Beacon. Charakteryzują się one wymiennymi bateriami, gdzie jedna powinna wystarczyć na 24 miesiące pracy urządzenia (dane producenta) [7]. Zasięg urządzenia oszacowany został na zakres 0 – 70 m. Beacony nadają sygnał o mocy w zakresie od -30dBm do 4dBm. Urządzenia są zabezpieczone poprzez użycie odpowiednich algorytmów szyfrujących oraz mechanizmu mieszania identyfikatorów urządzeń. Emitery mogą działać zarówno zgodnie z technologią iBeacon, jak również Eddystone [7].

W rozwiązaniu korzystano z beaconów ustawionych w tryb iBeacon. Urządzenia zostały rozmieszczone w linii prostej na pewnej powierzchni. Odstęp pomiędzy

poszczególnymi nadajnikami ustalono na 1,5 m, czyli minimalnej odległości pomiędzy regałami zapewniająca swobodne poruszanie się alejkami w sklepie samoobsługowym, w którym korzysta się z koszy lub wózków zakupowych [18]. W ten sposób zasymulowano rozmieszczenie urządzeń na pewnym obszarze sklepu wielkopowierzchniowego.

5. Rezultaty testowania prototypu - określanie odległości do znalezionej beacons. Wpływ sieci Wifi na odbierany sygnał.

Zastosowane bibliotek Estimote [19], które w przedstawionym rozwiązaniu odpowiadają za wszelkie operacje związane z odczytywaniem sygnałów z beaconów, dla każdego znalezionej urządzenia udostępniają wartość odległości przybliżonej (parametr Proximity), która może przyjąć cztery wartości [20] – rys. 4:

- Immediate.
- Near.
- Far.
- Unknown.



Rys. 4. Przedstawienie zakresów odległości od urządzenia dla poszczególnych wartości parametru Proximity [20]

Wartości tych można użyć do sortowania beaconów w zależności od odległości pomiędzy użytkownikiem a poszczególnymi urządzeniami, a następnie sortowania na tej podstawie listy zakupów. Jednak podczas testów aplikacji zauważono, że dosyć często, pomimo przebywania w odległości ok. 0,5 m od jednego z beaconów, a ok. 3,5 m od drugiego z nich, dla obydwu urządzeń odczytano wartość bądź Immediate. Odczytywane wartości są więc obarczone pewnym błędem.

Do określania odległości, w jakiej smartfon znajduje się od nadajnika, można również wykorzystać Odczytany Wskaźnik Mocy Sygnału (ang. Received Signal Strength Indicator, RSSI), który jest odczytaną przez odbiornik siłą sygnału nadanego przez nadajnik. Siła sygnału zależy od odległości pomiędzy urządzeniami oraz Moc Nadawania (ang. Broadcasting Power) nadajnika.

Do wyznaczania odległości za pomocą RSSI wykorzystywana jest jeszcze jedna wartość definiowana przez standard iBeacon. Każde z urządzeń zgodne z tym standardem ma fabrycznie skalibrowaną wartość zwaną „Measured Power”. Określa ona oczekiwaną wartość RSSI w odległości 1 metra od beacons. Odpowiednie połączenie tych dwóch wartości pozwala na obliczenie szacunkowej odległości od

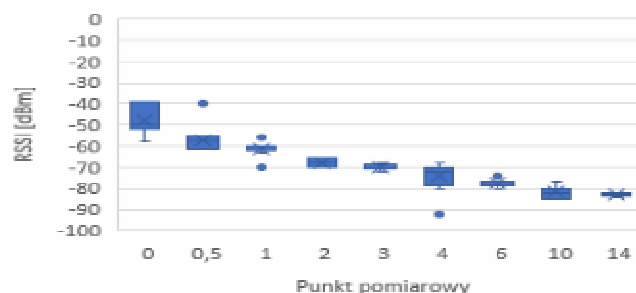
urządzenia. Szacunki te obarczane są jednak dosyć dużym błędem, a to za sprawą braku stabilności w odczycie wartości RSSI. Odczytana moc sygnału może być zakłócana przez przeszkody pomiędzy nadajnikiem i odbiornikiem, odbiciami fali od różnych obiektów czy nawet trzymaniem telefonu w różny sposób. Dodatkowo sygnał Bluetooth nadawany jest w paśmie 2,4 GHz, podobnie jak bardzo duża liczba sieci Wi-Fi. Występowanie sieci Wi-Fi w bliskiej odległości beaconów również może wpływać na odczyty RSSI na urządzeniu mobilnym.

W celu zbadania zależności pomiędzy odczytywaną wartością parametru przybliżonej odległości, wartością wskaźnika mocy sygnału RSSI a odległością od urządzenia przeprowadzono eksperyment. Wyznaczono 8 punktów pomiarowych w linii prostej w różnej odległości od jednego z beaconów, kolejno 0,5 m, 1 m, 2 m, 3 m, 4 m, 6 m, 10 m, 14 m. Do testu wykorzystano zmodyfikowaną wersję aplikacji mobilnej. Na ekranie programu udostępniono 8 przycisków do oznaczania aktualnego punktu pomiarowego. W tle następowało odczytywanie parametrów sygnału odbieranego z urządzenia, w tym przybliżonej odległość oraz RSSI, a następnie dane wyświetlano na ekranie oraz zapisywano jako kolejne linie w pliku tekstowym.

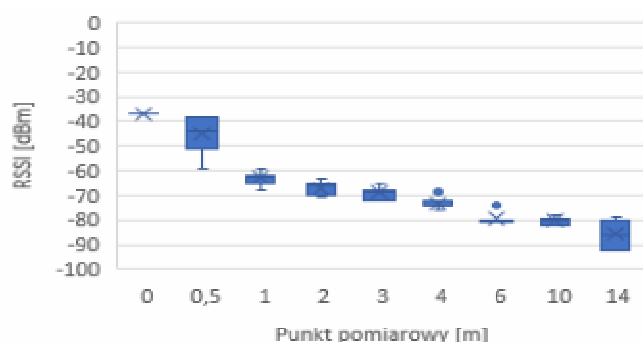
Badanie polegało na zebraniu kilku odczytów sygnału w każdym z punktów pomiarowych początkowo odchodząc od urządzenia, a następnie wracając do miejsca zamocowania beacons. Po znalezieniu się w każdym z wyznaczonych punktów wybierano przycisk z oznaczeniem odległości od źródła, a następnie zatrzymywano się przez chwilę, aby aplikacja zebrała kilka próbek odbieranego sygnału.

Aby dodatkowo sprawdzić wpływ zakłóceń sieci Wi-Fi znajdujących się w bliskim sąsiedztwie nadajników, test przeprowadzono dwukrotnie, z różnym umiejscowieniem beacons. Za pierwszym razem nadajnik umieszczono w miejscu, gdzie siła sygnału sieci Wi-Fi była poniżej -80 dBm. W drugim wariancie beacon umiejscowiono w bliskiej odległości (1m) od nadajnika sieci Wi-Fi.

Dane pozyskane z eksperymentu poddano obróbce statystycznej. Wyniki pomiarów w poszczególnych punktach pomiarowych przedstawiono na rys. 5 i 6.

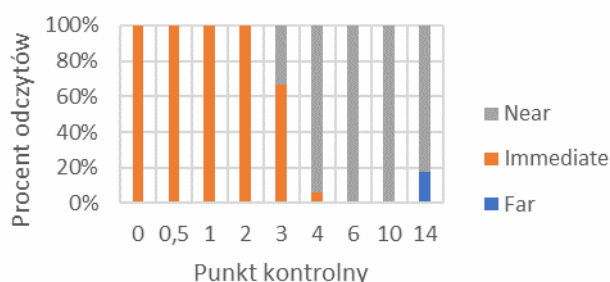


Rys. 5. Wykres prezentujący wartości RSSI w poszczególnych punktach pomiarowych

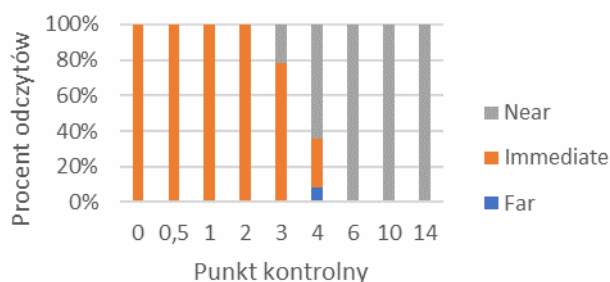


Rys. 6. Wykres prezentujący wartości RSSI w poszczególnych punktach pomiarowych – beacon blisko nadajnika Wi-Fi

Na podstawie zebranych danych sprawdzono również zależność pomiędzy odległością od źródła sygnału a odczytywanymi wartościami przybliżonej odległości (parametr Proximity) w danym punkcie kontrolnym – rys. 7 i 8. W przypadku testów w pobliżu źródła sygnału Wi-Fi można zauważyć, że nastąpiły pewne zakłócenia. Przykładem może być wystąpienie wartości „Far” w środkowej części wykresu (punkt kontrolny 4), a także nie wystąpienie wartości „Far” w znacznej odległości od źródła (punkt kontrolny 14).



Rys. 7. Procent odczytów danej wartości Proximity w poszczególnych punktach kontrolnych – test z dala od źródła Wi-Fi



Rys. 8. Procent odczytów danej wartości Proximity w poszczególnych punktach kontrolnych – test w pobliżu źródła Wi-Fi

6. Wnioski

Na podstawie zebranych danych stwierdza się, że odczytywanie dokładnych odległości od beaconów na podstawie jedynie sygnału RSSI jest obciążone dużym błędem i nieprzydatne w realnym stosowaniu. Potwierdza to rezultaty badań przedstawione w pracy [21]. Z drugiej jednak strony odczytywane wartości są wystarczające do rozróżniania źródeł

znajdującego się bliżej i dalej, np. beacon A jest bliżej niż beacon B, co jest wystarczające w zastosowaniu beaconów, jako źródeł przybliżonej lokalizacji użytkownika w sklepie.

W przypadku tworzonego prototypu dokładne określanie odległości nie jest niezbędne do zapewnienia poprawnego działania systemu. Szacunkowe przedziały wartości odległości przybliżonej są wystarczające, aby uszeregować listę zakupów. W czasie testów stwierdzono, że należałoby zastosować bardziej rozbudowane rozwiązanie sortujące, uwzględniające bardzo częste odczyty komunikatów z beaconów. Należałoby także wdrożyć w rozwiązaniu pewien system ograniczania niepotrzebnych przetasowań, aby zmniejszyć ilość wykonywanych operacji. Niemniej na potrzeby prototypu zastosowana metoda jest wystarczająca.

Stworzony na potrzeby pracy prototyp systemu składającego się z aplikacji internetowej oraz mobilnej, i wykorzystującego beacony, pozwolił na wskazanie pewnych wad, którymi obciążone są usługi korzystające z beaconów. Chodzi tu przede wszystkim o możliwe błędy odczytu rzeczywistej odległości czy wpływ sieci działających na podobnym paśmie częstotliwości na jakość odbieranego z beaconów sygnału. Z drugiej jednak strony, odpowiednie założenia co do budowanych systemów opartych o beacony, takie jak wzgląd na wspomniane wcześniej obciążenie błędami, pozwalają na stworzenie rozwiązań, które będą te ryzyka łagodzić. Przykładem sposobu na zwiększenia dokładności lokalizacji użytkownika może być zwiększenie liczby instalowanych beaconów, a następnie odpowiednie oprogramowanie, które stosując np. metodę triangulacji, będzie w stanie bardzo dokładnie wyznaczyć pozycję użytkownika. Innym sposobem może być takie dobranie funkcjonalności systemu, aby dokładna lokalizacja nie była konieczna, jak na przykład oferowanie użytkownikowi kuponów zniżkowych po wejściu do sklepu.

Kolejną zaletą stosowania beaconów jest możliwość monitorowania regionów nawet w przypadku wyłączonej aplikacji na urządzeniu końcowym użytkownika. Co więcej, są już rozwiązania takie jak Nearby Notifications [22], które nie wymagają nawet instalowania dedykowanych aplikacji. Beacon może rozsiewać informację z adresem URL lub krótkim tekstem, a urządzenie, które znajdzie się w zasięgu sygnału wyświetli użytkownikowi powiadomienie [22]. Funkcja monitorowania może w znaczący sposób wesprzeć tworzone systemy i ułatwić dotarcie do użytkownika w trakcie prowadzonych kampanii reklamowych.

Literatura

- [1] Boren Z. D.: There are officially more mobile devices than people in the world. The Independent, 2014. <http://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html> [19.12.2016].
- [2] <https://newzoo.com/insights/trend-reports/2016-global-mobile-market-report/> [21.11.2016].
- [3] <https://www2.deloitte.com/us/en/pages/consumer-business/articles/dawn-of-mobile-influence-retail.html> [17.11.2016].

- [4] Kilcourse B., Rowen S.: Mobile In Retail: The New Normal, 2016.
<http://www.rsrresearch.com/research/mobile-in-retail-the-new-normal?EID=62209992&CID=9390066>.
- [5] <https://www.emarketer.com/Article/Whats-Going-On-with-Beacons/1014218> [18.11.2016].
- [6] <https://en.oxforddictionaries.com/definition/beacon>
- [7] <https://store.kontakt.io/our-products/27-beacon.html> [3.12.2016].
- [8] <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy> [21.11.2016].
- [9] Laumonier H.: In-store mobile and beacons 101: What retailers can do with bluetooth and apps, 2015.
<http://blog.one2team.com/us/in-store-mobile-beacons-101>
- [10] <http://developer.estimote.com/ibeacon/> [21.11.2016].
- [11] <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf> [19.11.2016].
- [12] <http://developer.estimote.com/eddystone/> [21.11.2016].
- [13] Mazan F., Kovarova A.: A Study of Devising Neural Network Based Indoor Localization Using Beacons. *Computing and Information Systems Journal* 19, nr 1 (2015): 15–20.
- [14] <https://www2.deloitte.com/ie/en/pages/technology/articles/mobile-retail.html> [22.11.2016].
- [15] Senn A.: 10 Retailers Using Beacons Right, and How They Can Do Better, 2015.
<https://www.linkedin.com/pulse/10-retailers-using-beacons-right-how-can-do-better-alex-senn> [21.11.2016].
- [16] <https://developer.apple.com/support/app-store/> [13.12.2016].
- [17] Birkner C.: MLB's iBeacon Technology is A Marketing Home Run, 2015.
<https://www.ama.org/publications/MarketingNews/Pages/mlb-ibeacons.aspx> [21.11.2016].
- [18] Andrzejczak D. i inni.: Organizacja sprzedaży. Część 1. WSiP, 2010.
- [19] <https://github.com/aritchie/estimote-xplat> [09.12.2016]
- [20] <http://community.estimote.com/hc/en-us/articles/203356607-What-are-region-Monitoring-and-Ranging-> [10.12.2016].
- [21] Dong Q., Dargie W.: Evaluation of the reliability of RSSI for indoor localization. *ResearchGate*, 1–6, 2012.
https://www.researchgate.net/publication/261447858_Evaluation_of_the_reliability_of_RSSI_for_indoor_localization [22.11.2016].
- [22] Lewis P.: How Beacons Can Reshape Retail Marketing, 2016.
<https://www.thinkwithgoogle.com/articles/retail-marketing-beacon-technology.html> [21.11.2016].

Porównanie szybkości działania wybranych funkcji skrótu i algorytmów szyfrowania

Dawid Górniak*, Piotr Kopniak

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Dane to często najcenniejsza rzecz, jaką gromadzimy na naszych komputerach. Bez odpowiedniego zabezpieczenia danych algorytmami szyfrującymi, cenne dla nas informacje w przypadku dostania się w niepowołane ręce mogą zostać bezproblemowo wykorzystane. Artykuł przedstawia wybrane metody szyfrujące i funkcje skrótu dostępne w bibliotece Boucy Castle dla środowiska Java. Przedstawiona analiza dotyczy pomiaru prędkości generowania i weryfikacji 240 bitowej sygnatury dla algorytmów szyfrujących, natomiast dla funkcji skrótu analiza dotyczy szybkości działania funkcji. Z pośród badanych algorytmów szyfrowania i funkcji skrótu najszybsze okazały się AES i SHA1.

Słowa kluczowe: algorytmy szyfrujące; funkcje skrótu; Bouncy Castle

* Autor do korespondencji.

Adres/adresy e-mail: dawid.gorniak@pollub.edu.pl

Comparing the speed of the selected hash and encryption algorithms

Dawid Górniak *, Piotr Kopniak

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The data is often the most valuable thing that we collect on our computers. Without proper data security with encryption our valuable information may be illegally used by an unauthorised person. The article presents selected encryption methods and hash functions available in Boucy Castle library for Java programming language. The presented analysis applies to measurement of the speed of signature generation and verification. The signatures are for 240 bit encryption algorithms. In case of a hash function, the analysis refers to the speed of such functions. The fastest encryption algorithm and hash function from the research group were AES and SHA1.

Keywords: encryption algorithms; hash functions; Bouncy Castle

*Corresponding author.

E-mail address/addresses: dawid.gorniak@pollub.edu.pl

1. Wstęp

Postęp techniczny w dużej mierze wpływa na bezpieczeństwo prywatnych danych. Niezależnie od tego, do jakich celów wykorzystujemy komputer, na dysku twardym zazwyczaj znajduje się mnóstwo ważnych danych, które mogą dostać się w niepowołane ręce. Bez odpowiedniego szyfrowania, przechwycone dane przez osoby postronne mogą być bezproblemowo odczytane i wykorzystane. W związku z tym należy dbać o bezpieczeństwo informacji. Prywatne dane, chronione przez algorytmy szyfrujące, są w dużej mierze bezpieczne. Szyfrowanie to proces przekształcania informacji w taki sposób, aby osoba nieupoważniona nie mogła jej odczytać – zaufana osoba może odszyfrować daną wiadomość i przeczytać ją w pierwotnej formie. Istnieje wiele popularnych metod szyfrowania, deszyfrowania, jednak kluczem do bezpieczeństwa wcale nie jest odpowiednio zastrzeżony algorytm. Najważniejszą rzeczą jest utrzymanie w tajemnicy klucza szyfrowania[1].

Celem artykułu jest analiza wybranych algorytmów szyfrujących i funkcji skrótu pod względem szybkości. Do badań wykorzystano implementacje dostępne w bibliotece Bouncy Castle dla środowiska programistycznego Java. Dzięki odpowiedniemu skryptowi napisanemu w Javie

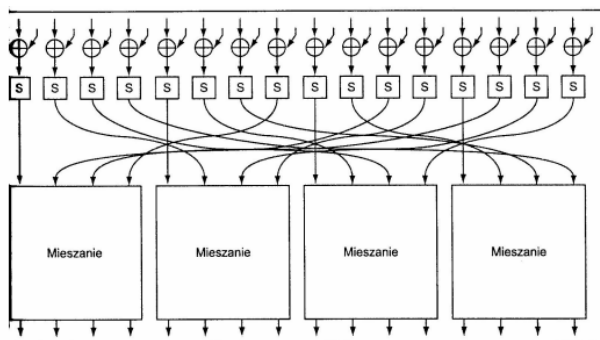
przeprowadzono testy pomiaru szybkości dla komunikatu o długości 240 bitów. Dla algorytmów szyfrujących przeprowadzono pomiar generowania i weryfikacji sygnatury. Natomiast dla funkcji skrótu zbadano szybkość działania funkcji haszującej. Uzyskane wyniki zostały przedstawione za pomocą wykresów, z których wysnuto wnioski.

2. Algorytmy szyfrujące

Do przeprowadzonych badań użyto wybranych algorytmów używanych do elektronicznego podpisu dokumentów oraz zaprezentowane zostaną schematy ich działania.

2.1. AES

Szyfr blokowy AES został przyjęty w 1997 roku przez Narodowy Instytut Samorządu Terytorialnego. Algorytm ten jest standardem USA używającym kluczy o rozmiarach 128, 192, 256 bitów i operuje na 128 bitowych blokach danych [2]. Na poniższym schemacie (Rys. 1) przedstawiona jest pojedyncza tura tego algorytmu.



Rys. 1. Struktura pojedynczej tury AES [2].

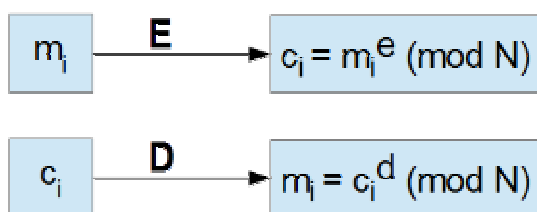
Na wejściu otrzymywany jest 16 bitowy tekst jawny, który poddawany jest metodzie XOR z 16 bitowym kluczem tury. Każdy z 16 bajtów wyjściowych składa się na tablicę S-Boksów. Istotą działania S-boksa jest podstawianie określonego bitu innym bitem zależnym od następnego i poprzedniego bitu. Następnie bity są grupowane dzięki funkcji mieszającej. Przebieg pojedynczej tury polega na przeprowadzeniu operacji XOR na kilku bitach wejściowych. Liczba tur szyfrowania zależna jest od rozmiaru klucza.

2.2. RSA

Nazwa algorytmu RSA wywodzi się od nazwisk trzech projektantów: Ron'a Rivest'a, Adi Shamir'a oraz Leonard'a Adleman'a. Asymetryczny blokowy algorytm wykorzystuje dwa powiązane klucze: publiczny i prywatny o długości od 1000 do 4000 bitów. Do zastosowania systemu RSA wymagane jest wygenerowanie pary kluczy, z którymi wiąże się poniższe zadania[3]:

- wybór dwóch dużych losowych liczb pierwszych p i q
- obliczenie wartości $n = pq$
- obliczenie dla n wartości funkcji Eulera: $\phi(n) = (p-1)(q-1)$
- wybranie liczby e ($1 < e < \phi(n)$) względnie pierwszej z $\phi(n)$
- wyszukanie liczby d : $d \equiv e^{-1} \pmod{\phi(n)}$

Głównym problemem jest wyszukanie liczb p i q . Liczby te należą do zbioru liczb pierwszych. W dzisiejszych czasach nie ma efektywnej metody do poszukiwania dużych liczb pierwszych, polegającej na wybraniu z dużego zbioru liczb p i q w celu uniemożliwienia ich odkrycia podczas kolejnych prób wyliczanych z wzoru $n = pq$. Jednak do tego celu stosuje się funkcję, która polega na wylosowaniu żądanej wielkości liczby nieparzystej i sprawdzeniu czy liczba zalicza się do zbioru liczb pierwszych.

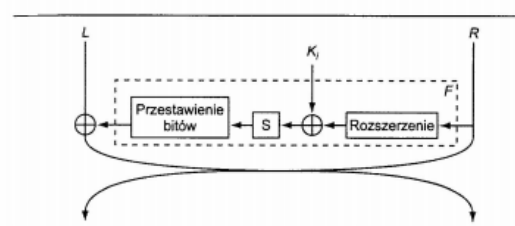


Rys. 2. Schemat szyfrowania i deszyfrowania w algorytmie RSA [3].

Klucz publiczny używany jest do szyfrowania wiadomości, która dzielona jest na fragmenty zamieniane na liczbę bitów. Takie liczby podnoszone są modulo n do potęgi e , co widoczne jest na powyższym rysunku (Rys. 2). Klucz prywatny wykorzystywany jest do deszyfrowania wiadomości składającej się z liczb mniejszych niż n . Liczby te mnożone są przez modulo n i potęgowane przez liczbę d (Rys. 2).

2.3. 3DES

Algorytm 3DES jest rozszerzeniem algorytmu DES. Zbudowany jest z trzech kolejnych rund DES. 3DES posiada dłuższy klucz, ale dziedziczy po DES właściwości dopełnienia oraz słabe klucze. 64 bitowy rozmiar bloku jest ograniczeniem tego szyfru, ponieważ wiąże się to z ograniczeniem szyfrowanych danych pojedynczym kluczem [4].



Rys. 3. Struktura pojedynczej tury algorytmu DES [4].

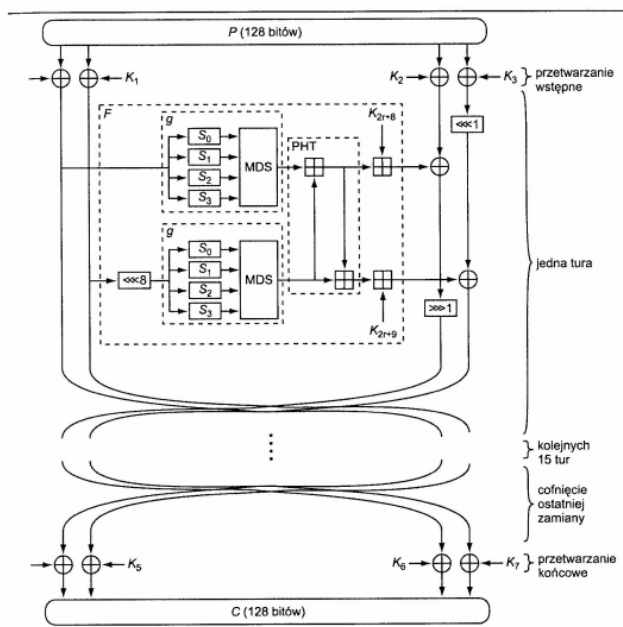
DES wykorzystuje 64 bitowy jawny tekst, który jest dzielony na dwie 32 bitowe części L i R . Szesnaście ponumerowanych od 1 do 6 tur, które tworzą nową parę zgodnie z K_i kluczem tury. Funkcja tury F wykonująca większość pracy oznaczona jest na rysunku za pomocą prostokąta narysowanego przerywaną linią. Początkowo za pomocą funkcji rozszerzającej zwiększana jest liczba bitów do 48 dla wartości R . Następnie przekształcany jest za pomocą metody XOR otrzymany ciąg z K_i 48 bitowym kluczem tury. Otrzymany wynik jest wejściem do S-boksu, składającego się z ośmiu tablic, z czego każda 6 bitowa tablica przekształcana jest na 4 bity. Dzięki temu ponownie wraca do 32 bitowej wielkości, która jest przekształcana przez funkcję mieszającą, a następnie poddawana z wartością L operacji XOR. Ostatecznie zmieniana jest wartość L i R .

2.4. Serpent

Algorytm Serpent pozwala jednocześnie szyfrować albo deszyfrować dane. Zalicza się do algorytmów bezpiecznych, skomplikowanych i wymagających dużej ilości zasobów sprzętowych przez jego blok deszyfrujący i szyfrujący. Działanie algorytmu serpent opiera się na 256 bitowym kluczu, który jest dzielony na osiem bloków 32 bitowych. Bloki te otrzymywane są rekurencyjnie z wartości k_i . Poprzez S-Boxy transformowane są wartości, które w dalszej części są scalane w wektory K_i stanowiące klucze dla pojedynczych rund szyfrowania i deszyfrowania. W skład rundy wchodzi operacja XOR danych z kluczem, przejście przez S-Box i transformacja liniowa. Proces deszyfrowania opiera się na bloku z kluczem K_i , który podlega operacji XOR. Pozostałe 32 rundy podają dane na wejściu S-Box-1 następnie stosują klucz K_i oraz odwrotną transformację liniową [5].

2.5. Twofish

Algorytm Twofish jest blokowym symetrycznym algorytmem szyfrującym, który operuje na 128 bitowych blokach danych i korzysta z kluczy z zakresu od 128 do 256 bitów. Poniższy schemat (Rys. 4) przedstawia działanie tego algorytmu.

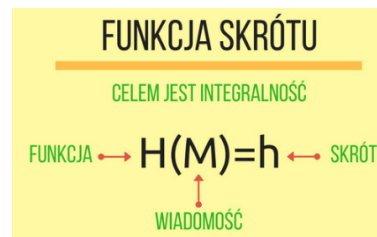


Rys. 4. Struktura Twofish [6].

Szyfrując, algorytm Twofish na wejściu otrzymuje 128 bitową wiadomość, którą dzieli na cztery 32 bitowe wartości. Funkcja F jest turą na którą składa się podwójne wywołanie funkcji g, funkcji PHT oraz dodania klucza. Rezultat funkcji F i pozostałe części danych poddawane są operacji XOR. Prostokąty z symbolami „<<<i>” i „>>>i>” oznaczają przesunięcia 32 bitowych wartości o zadaną liczbę bitów. Funkcja g zawiera cztery zależne od klucza S-boxy i funkcję liniową podobną do mieszającej funkcji AES. Na podstawie klucza wyliczana jest specjalnym algorytmem tablica S-boksów. Za mieszanie dwóch wyników funkcji g z wykorzystaniem 32 bitowej operacji odpowiada funkcja PHT. Algorytm Twofish posiada dodatkowe wstępne i końcowe przetwarzanie, które polega na dołączeniu informacji z klucza [6].

3. Funkcje haszujące

Podstawowym elementem systemów uwierzytelniania oraz sprawdzania integralności wiadomości są funkcje skrótu (ang. hash functions). Funkcja skrótu tworzy z wiadomości skrót o stałej długości nie zależnej od długości wejściowej wiadomości (Rys. 5). Występuje tutaj sytuacja zwana kolizją, która oznacza że istnieją wiadomości zwracające identyczne wartości funkcji skrótu. Takie sytuacje są wysoce niepożądane w zastosowaniach kryptograficznych [7].



Rys. 5. Funkcja skrótu [7].

3.1. MD5

Algorytm haszujący MD5 został opublikowany przez R. Rivesta bez podania argumentów, uzasadniających matematycznie, że może on pełnić funkcję dobrej jednokierunkowej funkcji skrótu. Wejściem algorytmu jest komunikat M o praktycznie dowolnej długości, a wyjściem 128 bitowy skrót, czyli wyciąg tego komunikatu. Przetwarzanie komunikatu na skrót odbywa się w 4 krokach, w których dane są reprezentowane za pomocą bloków 512 bitowych. Przetwarzanie komunikatu [8]:

- 1) gdy warunek $K \equiv 448 \pmod{512}$ jest spełniony, wiadomość dopełniana jest dodatkowymi bitami. K oznacza pierwotną długość komunikatu. W takiej sytuacji dopełnienie ma długość od 1 do 512 bitów.
- 2) do wiadomości dodawana jest liczba K, reprezentująca pierwotną długość komunikatu M. W przypadku gdy początkowa wartość komunikatu przekracza 264, wtedy poddana operacji modulo 264 zostaje liczba, która określa pierwotną długość komunikatu. W rezultacie tej czynności otrzyma się L 512-bitowych bloków Y_0, Y_1, \dots, Y_{L-1} , które można podzielić na 16 podbloki 32-bitowe, ponieważ $512=16 \cdot 32$.
- 3) operacje, realizowane przez algorytm MD5, polegają na zmianie wartości zmiennych D, C, B i A, zapamiętanych w 4 rejestrach 32 bitowych. Zmienne otrzymują wartości początkowe zapisane w systemie szesnastkowym: D = 76543210, C = fedcba98, B = 89abcdef, A = 01234567.
- 4) w tym kroku używa się czterech funkcji pomocniczych, których argumentami są trzy 32 bitowe słowa, a wyjściem jedno słowo 32 bitowe. Są to następujące funkcje:

$$F(X, Y, Z) = (X \text{ and } Y) \text{ or } ((\text{not } X) \text{ and } Z), (1)$$

$$G(X, Y, Z) = (X \text{ and } Z) \text{ or } (Y \text{ and } (\text{not } Z)), (2)$$

$$H(X, Y, Z) = X \text{ xor } Y \text{ xor } Z, (3)$$

$$I(X, Y, Z) = Y \text{ xor } (X \text{ or } (\text{not } Z)). (4)$$

W tym kroku stosuje się również 64 elementową tablicę T[1..64], której elementami są liczby całkowite zapisane w formie szesnastkowej. Niech $M[0, 1, \dots, N-1]$ oznacza 32 bitowe słowa uzupełnionego komunikatu, składającego się z L 512 bitowych bloków. Wobec tego $N = 16 \cdot L$, czyli każdy blok Y_q składa się z 16 słów $M[j]$. Proces przetwarzania w tym kroku obejmuje każdy blok 16 słowowy Y_q . Wyjściem algorytmu jest 128 bitowy skrót komunikatu M.

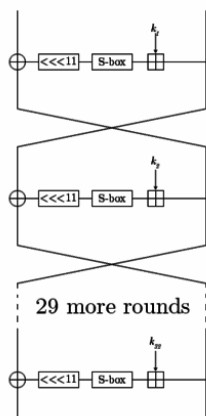
3.2. GOST

Algorytm GOST to szyfr blokowy opracowany w Związku Radzieckim. Operuje na 64 bitowych blokach,

a do szyfrowania wykorzystuje 256 bitowy klucz. Jego funkcja cykliczna składa się z dwóch kroków:

- dodawany jest 32 bitowy pod klucz modulo 232.
- wynik jest przetwarzany przez warstwę S-boxy i rotowany o 11 bitów w lewo.

Na poniższym schemacie (Rys. 6) jedna linia reprezentuje 32 bity [9].

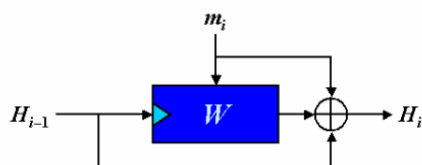


Rys. 6. Diagram GOST [9].

Podklucze są z góry wybierane w określonej kolejności. Plan kluczy jest prosty: dzieli klucz 256 bitowy na osiem 32 bitowych pod kluczy, a każdy pod klucz używany jest czterokrotnie w algorytmie. Pierwsze 24 rund używane jest w naturalnym porządku, a ostatnie osiem rund w odwrotnej kolejności.

3.3. Whirlpool

Funkcja skrótu Whirlpool przyjmuje wiadomość o dowolnej długości mniejszej niż 2^{256} bity i zwraca 512 bitowy skrót. Wiadomość jest dzielona na 512 bitowe bloki. Jeżeli wiadomość nie zostanie podzielona równomiernie, wtedy do ostatniego bloku jest dołączana jedynka i liczba zer, a na końcu dołączana jest długość oryginalnej wiadomości jako 256 bitowa liczba, tak aby ostatni blok uzyskał 512 bitów. Bloki te są kompresowane schematem Miyaguchi Preneel, a następnie poddawane są 10 rundom szyfru Whirlpool „W”, gdzie W wykorzystuje wynik poprzedniej operacji i obecnego bloku jako klucza. Wynikiem jest operacja XOR wyjścia W i dwóch wejść [9].

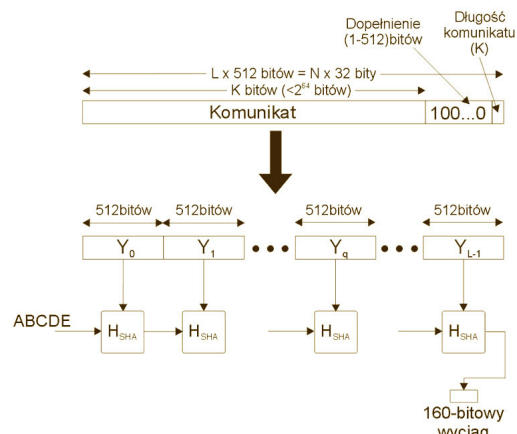


Rys. 7. Struktura Whirlpool. H_{i-1} jest wyjściem poprzedniego bloku, m_i jest wejściem bloku [9].

3.4. SHA

Secure Hash Algorithm to funkcja skrótu zaprojektowana przez National Security Agency i publikowana przez National Institute of Standards and Technology. Algorytm z informacji o rozmiarze 264 bitów generuje 160 bitowy skrót. SHA

wykonuje podstawowe operacje bitowe na 32 bitowych zmiennych podczas działania.



Rys. 8. Schemat działania algorytmu SHA [9].

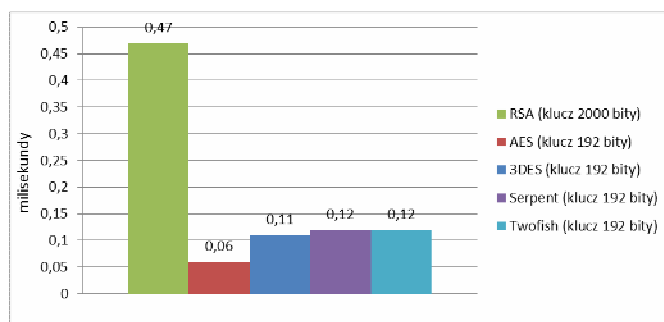
Działanie algorytmu SHA [9]:

- komunikat jest dopełniany w taki sposób, aby jego długość w bitach wynosiła do 448 mod 512.
- do długości wiadomości dodawana jest dodatnia 64 bitowa liczba.
- rezultatem tych kroków jest wiadomość o długości równej wielokrotności 512 bitów, którą można przedstawić jako sekwencję 512 bitowych bloków.
- stosowany jest bufor 160 bitowy, który można przedstawić jako pięć 32 bitowe rejestry. Bufor służy do przetrzymywania pośrednich i końcowych wartości funkcji haszującej.
- głównym etapem algorytmu SHA jest przetwarzanie komunikatu w 512 bitowych blokach.
- wynikiem L-tego przetworzenia 512 bitowego bloku jest 160 bitowy wyciąg.

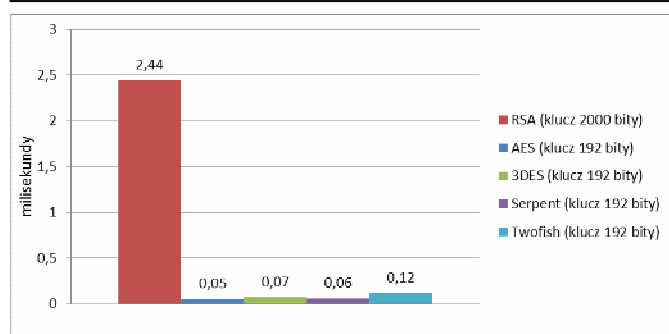
4. Wyniki

4.1. Algorytmy kryptograficzne

Wyniki szybkości generowania i weryfikacji sygnatury o długości 240 bitów dla algorytmów kryptograficznych: RSA, AES, 3DES, Serpent, Twofish są przedstawione na rysunkach 9,10.



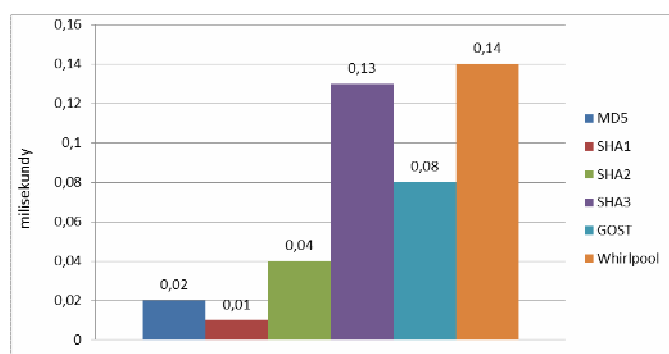
Rys. 9. Wynik szybkości generowania sygnatury.



Rys. 10. Wynik szybkości weryfikacji sygnatury.

4.2. Funkcje haszujące

Wyniki pomiarów szybkości działania funkcji skrótu dla MD5, SHA1, SHA2, SHA3, GOST, Whirlpool dla komunikatu 240 bitowego zostały przedstawione na poniższym rysunku 11.



Rys. 11. Wynik szybkości działania funkcji haszujących.

5. Wnioski

Analizując powyższe wykresy algorytmów szyfrujących można wywnioskować, że wygenerowanie 240 bitowej sygnatury najszybciej zachodzi w algorytmie AES generowanie trwało 0,06 milisekund. Natomiast RSA okazał się prawie ośmiokrotnie wolniejszy od najszybszego algorytmu, osiągając wynik 0,47 milisekund. Pozostałe algorytmy wypadły nie znacznie gorzej osiągając czas od 0,11 do 0,12 milisekund. W przypadku weryfikacji szybkość sprawdzania sygnatury dla algorytmu AES jest około czterdzieści osiem razy szybsza od metody RSA z czasem 2,44 milisekund. Nieco gorsze od AES z wynikiem 0,05 milisekundy okazały się pozostałe algorytmy, gdzie ich czasy

to: 3DES 0,07 milisekund, Serpent 0,06 oraz Twofish 0,12 milisekund. Z pośród analizowanych algorytmów szyfrujących algorytm AES okazał się najszybszym algorytmem w generowaniu i weryfikacji 240 bitowej sygnatury. Pamiętając o tym, że w dzisiejszych czasach dużą rolę odrywa bezpieczeństwo danych najbezpieczniejszym algorytmem z pośród badanych szyfrów jest szyfr RSA z kluczem 2000 bitowym, gdyż jest on szyfrem asymetrycznym wymagającym dwa klucze: klucza publicznego do wygenerowania sygnatury oraz klucza prywatnego do dekodowania danych.

Odnosząc się do analizy wykresu funkcji haszujących algorytm SHA1 okazał się najszybszą funkcją haszującą z czasem 0,01 milisekund, o 0,01 milisekund wolniejszym algorytmem jest algorytm MD5, o 0,02 milisekund wolniejszym od MD5 okazał się SHA2. Natomiast GOST uzyskał wynik 0,08 milisekund kwalifikując się na 4 miejsce co do prędkości generowania skrótu dla 240 bitowej wiadomości. Pozostałe algorytmy SHA3 oraz Whirlpool wypadły najgorzej osiągając kolejno czas 0,13 i 0,14 milisekund. W kwestii bezpieczeństwa funkcji haszujących, najodporniejszym na różnego rodzaju fałszerstwa jest funkcja mieszająca SHA1, gdyż jest ona tak zaprojektowana, aby obliczeniowo uniemożliwić uzyskanie wiadomości z danego skrótu wiadomości lub wystąpienia kolizji. Dzięki temu jakkolwiek zmiana dokonana w wiadomości utworzy inny skrót wiadomości.

Literatura

- [1] W. Stallings, Kryptografia i bezpieczeństwo sieci komputerowych, Helion, 2011.
- [2] Kryptografia i bezpieczeństwo sieci komputerowych, Helion, 2012.
- [3] Marcin Karbowski, Podstawy kryptografii RSA. Wydanie III, Helion, 2014.
- [4] A. Menezes, Kryptografia stosowana, WTN, 2005.
- [5] Serpent, [www.pl.wikipedia.org/wiki/Serpent_\(kryptografia\)](http://www.pl.wikipedia.org/wiki/Serpent_(kryptografia))
- [6] Twofish, www.pl.wikipedia.org/wiki/Twofish.
- [7] I. Damgard, A design principle for hash functions, Springer-Verlag, 1990.
- [8] MD5, www.pl.wikipedia.org/wiki/MD5
- [9] M. Jakobsson, Applied Cryptography and Network Security, Springer Verlag gmbh, 2014.

Porównanie mechanizmów bezpieczeństwa popularnych systemów operacyjnych urządzeń mobilnych

Michał Buryta*, Piotr Kopniak

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł dotyczy różnic w mechanizmach zabezpieczających trzech najpopularniejszych platform mobilnych, jakimi są Android, iOS i Windows Phone. Głównie skupiono się na odrębnościach związanych z dostępem do danych użytkowników, instalacją i uruchamianiem aplikacji, dostępem do Internetu, połączeniem sieciowym, obsługą chmury, tworzeniem kopii zapasowych, możliwości szyfrowania plików, stosowaniem podpisów elektronicznych oraz podłączaniem urządzenia do komputera.

Słowa kluczowe: bezpieczeństwo informacji; urządzenia mobilne; systemy operacyjne

* Autor do korespondencji.

Adres e-mail: buryta.michal@gmail.com

Comparison of the security mechanisms of popular operating systems for mobile devices

Michał Buryta*, Piotr Kopniak

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article concerns the differences in the mechanisms of securing the three most popular mobile platforms, which are Android, iOS and Windows Phone. Mainly focuses on the differences related to access to user data, installing and running applications, Internet access, network connectivity, cloud services, backup, file encryption capabilities, the use of electronic signatures and connect the device to your computer.

Keywords: information security; mobile devices; operating systems

*Corresponding author.

E-mail address: buryta.michal@gmail.com

1. Wstęp

Obecnie zdecydowana większość ludzi używa telefonów komórkowych wszędzie i do wszystkiego. Telefon pozwala na szybkie porozumiewanie się z innymi osobami w dowolnym miejscu, ale nie tylko. Telefony udostępniają wiele innych możliwości takich jak przeglądanie stron www, odczytywanie i wysyłanie poczty elektronicznej, wykonywanie operacji bankowych, używania telefonu zamiast karty płatniczej, przesyłanie danych po między różnymi urządzeniami. Smartfon przez to że udostępnia szereg powyższych usług, przechowuje dużo informacji o swoim użytkowniku. Do takich informacji zaliczają się między innymi numery kart kredytowych, piny, hasła do serwisów internetowych, dane osobowe, zdjęcia i filmy czy rozmowy głosowe zapisywane na urządzeniu. Wszystkie te możliwości oraz to, że samo urządzenie jest wartościowe sprawia, że telefony często są celem kradzieży. Wszystko to sprawia że bardzo istotną sprawą jest bezpieczeństwo, poufność, zachowanie prywatności danych przechowywanych na urządzeniu oraz samego telefonu. Problemem koniecznym do rozwiązania jest więc zapewnienie odpowiedniej ochrony bez nadmiernego ograniczenia użytkownika. Ma to

szczególne znaczenie również dlatego, że na platformach mobilnych jest trudniej rozpoznać zagrożenie. Sposobami na zapewnienie odpowiedniej ochrony telefonu są między innymi ograniczanie możliwości aplikacji przez uruchamianie ich w odrębnych izolowanych częściach pamięci, przydzielanie uprawnień tylko tych, które są niezbędne do poprawnego działania oraz świadome działanie użytkownika. Właśnie dla tego użytkownik powinien wiedzieć w jakim zakresie platforma jego telefonu oferuje bezpieczeństwo oraz na co mogą być wrażliwe jej mechanizmy zabezpieczające. Wiedza ta jest istotna nie tylko dla przedsiębiorstw, ale i każdego użytkownika telefonu

Niniejszy artykuł przedstawia różnice pomiędzy trzema najbardziej popularnymi platformami jakimi są Android, iOS i Windows Phone. Większość użytkowników wybiera obecnie urządzenia z systemem firmy Google. W pierwszym kwartale roku 2016 jest to ponad 80% z pośród wszystkich sprzedanych w tym roku. Wszystkich sprzedanych aparatów jest ponad 349 milionów w pierwszym kwartale roku 2016. Poniższa tabela przedstawia liczbę sprzedanych smartphonów oraz procentowy udział w rynku.

Tabela 1. Światowa sprzedaż smartphone dla użytkownika końcowego 1 kwartał roku 2016 [1]

System operacyjny	1Q16	1Q16 Udział w rynku (%)	1Q15	1Q15 Udział w rynku (%)
Android	293771.2k	84.1	264941.9k	78.8
iOS	51629.5k	14.8	60177.2k	17.9
Windows	2399.7k	0.7	8270.8k	2.5
Blackberry	659.9k	0.2	1325.4k	0.4
Others	791.1k	0.2	1582.5k	0.5
Suma	349251.4k	100	336297.8k	100

Tabela 2. Światowa sprzedaż smartphone dla użytkownika końcowego 2 kwartał roku 2016 [2]

System operacyjny	2Q16	2Q16 Udział w rynku (%)	2Q15	2Q15 Udział w rynku (%)
Android	296912.8k	86.2	271647.0k	82.2
iOS	44395.0k	12.9	48085.5k	14.6
Windows	1971.0k	0.6	8198.2k	2.5
Blackberry	400.4k	0.1	1153.2k	0.3
Others	680.6k	0.2	1229.0k	0.4
Suma	349251.4k	100	330312.9k	100

2. Cel, przedmiot badań pytania badawcze

Celem przeprowadzanych badań była analiza i porównanie mechanizmów zabezpieczających instalację i uruchamianiem oprogramowania, dostępu do danych, dostępem do telefonu i podczas łączenia się z komputerem. Przedmiotem analizy był poziom zabezpieczeń trzech platform mobilnych jakimi są Android firmy Google, iOS firmy Apple oraz Windows Phone firmy Microsoft. Główne pytania jakie były stawiane przed rozpoczęciem prac to, która platforma jest najlepiej zabezpieczona i czy ocena zabezpieczeń ulegnie zmianie w zależności od sposobu użytkowania telefonu oraz jego przeznaczenia.

2.1. Obszar badań

Wszystkie trzy platformy posiadają podobny zabezpieczenia, które różnią się stopniem zapewnianego bezpieczeństwa oraz wykonaniem. Częścią wspólną dla aplikacji uruchamianych w badanych systemach jest fakt,

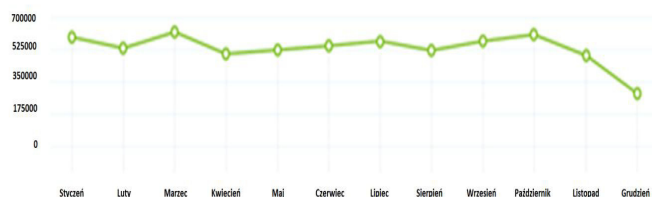
że każda z aplikacji jest uruchamiana na oddzielnej części pamięci i nie ma dostępu do innych aplikacji. Jest to tak zwana piaskownica, która ma za zadanie odizolować każdą aplikację na czas jej działania w przydzielonej części pamięci. Istnieją mechanizmy, które zapobiegają odwoływaniu się do tej samej komórki pamięci przez dwa różne programy gdy działają w tym samym czasie. Oznacza to, że aplikacja nie może się odwołać do pamięci zarezerwowanej dla innego programu przed zakończeniem działania programu, któremu przydzielona jest ta komórka pamięci. Z tego można wywnioskować że każda aplikacja musi być samodzielna i nie ma żadnego dostępu do danych zewnętrznych. Na szczęście nie jest tak. Aplikacje mogą używać danych pobranych z Internetu oraz przysyłać do innych aplikacji za pośrednictwem odpowiednich mechanizmów systemowych.

Kolejną częścią wspólną dla badanych platform związaną z aplikacjami jest miejsce z którego można je uzyskać. Każda z platform ma swoje własne miejsce, gdzie użytkownik może poszukiwać odpowiednich dla siebie programów oraz je pobierać na swój telefon. Dla iOS jest to Apple AppStore. Dla Androida jest to Google Play. Dla Windows Phone jest to Windows Phone Marketplace. Wszystkie firmy przed udostępnieniem aplikacji sprawdzają ją pod kontem złośliwego oprogramowania oraz poprawnością działania. Nie każdy może dodać aplikację do wybranej platformy. Każda firma wymaga aby programiści przed dodaniem pierwszej aplikacji założyli konto za pomocą którego będzie można dokładnie zweryfikować czy jest to aplikacja. Najtrudniej założyć konto programisty dla systemu iOS, ponieważ firma Apple bardzo restrykcyjnie podchodzi do sprawy tworzenia nowych kont. Apple oraz Google nie zezwalają na uruchomienie aplikacji na ich telefonach, które nie posiadają ważnych licencji wystawionych przez ich serwis z aplikacjami. W systemie Android istnieje metoda, która umożliwia instalację oraz uruchamianie aplikacji z poza Google Play. Android posiada prawdopodobnie najsłabszą metodę sprawdzania czy dodawana aplikacja może być złośliwa, ponieważ są wykrywane wirusy w Google Play[3]. Android posiada też możliwość usunięcia dowolnej aplikacji z telefonów użytkowników, która została ściągnięta z Google Play. Wirusy, którym udaje się przejść przez zabezpieczenia Androida i dotrzeć do użytkownika mogą próbować doinstalować do siebie dodatki z poza marketu co może być poważnym zagrożeniem bezpieczeństwa. Jednak od Androida w wersji 6.0 uprawnienia aplikacji weryfikowane są na bieżąco przy każdej próbie wykorzystania zabezpieczanych części systemu co uniemożliwia doinstalowanym automatycznie aktualizacjom aplikacji wykonywanie niebezpiecznych operacji bez wiedzy użytkownika. W większości platform nie ma systemowego menadżera plików. Co spowodowało, że można pobrać tego typu aplikację z odpowiedniego miejsca, która działa jak zwykły menadżer plików. Za pomocą menadżera plików można zainstalować na platformie Android aplikację z karty pamięci lub komputera.

Jedyną platformą na której dostępne są programy antywirusowe jest Android. Spowodowane jest to tym, że jako jedyna potrzebuje takich aplikacji. Natomiast już wszystkie platformy posiadają wszelkiego typu programy zabezpieczające przed phishingiem, niebezpiecznymi

stronami czy sprawującymi tak zwaną kontrolę rodzicielską. Dla wszystkich trzech platform istnieją aplikacje, które umożliwiają bezpieczne przechowywanie poufnych danych takich jak hasła, zdjęcia prywatne czy dokumenty na telefonie. Programy tego typu używają algorytmów takich jak AES czy BlowFish do szyfrowania danych. Algorytmy te zapewniają wysoki poziom bezpieczeństwa. Każda platforma posiada również aplikacje za pomocą której można odnaleźć telefon, zdalnie go zablokować lub też przywrócić fabryczne ustawienia i wyczyścić całą pamięć telefonu. Należy jednak wcześniej wprowadzić dane telefonu do tego typu usługi. Możliwość zaszyfrowania danych na telefonie została dodana do wszystkich systemów przez ich producentów. Do szyfrowania konieczne jest ustawienia hasła lub pinu. Szyfrowanie może znacznie spowolnić pracę telefonu. Jedną z form zabezpieczenia danych przed utratą jest tworzenie kopii zapasowej. Producenci badanych systemów mobilnych udostępniają aplikacje, które pozwalają na tworzenie kopii zapasowej na komputerze. Są to proste i wygodne rozwiązania. Niestety firma Microsoft bardzo ograniczyła dane jakie można zabezpieczyć przed utratą w ten sposób, natomiast Firma Google pozwala nawet bez dodatkowych aplikacji na tworzenie kopii zapasowej na komputerze. W przypadku przywracania plików systemowych mogą wystąpić problemy. Oczywiście kopię zapasową można automatycznie tworzyć w chmurze udostępnianej przez producenta. Takie rozwiązanie istnieje na wszystkich trzech platformach i jest bardzo wygodne. W tym przypadku najlepiej wypada Android, ponieważ udostępnia najwięcej miejsca oraz pozwala na zrobienie kopii zapasowej całego telefonu łącznie z aplikacjami. Apple niestety nie pozwala na tworzenie kopii zapasowej plików, których pochodzenie jest nieznane. Czyli wszystkich plików, które nie zostały stworzone za pomocą telefonu[4,5,6].

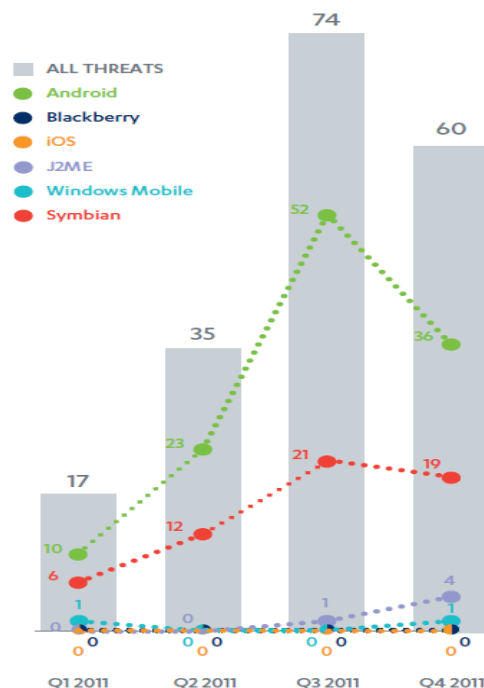
Jednym z najbardziej popularnych Trojanów, za pomocą których cyberprzestępcy próbują zabrać nasze pieniądze są tak zwane Trojany SMS. Taka szkodliwa aplikacja wysyła SMS'y premium i zapisuje nas do różnych usług. Takie aplikacje zostały wykryte ponad 6 mln razy na urządzeniach z Androidem.



Rys. 1. Liczba wykrytych Trojanów SMS Android w roku 2015 [7]

Jak widać na Rys. 1 liczba takiego oprogramowania spada. Jest to spowodowane coraz lepszymi programami antywirusowymi, ale także coraz większą popularnością Trojanów bankowych wśród cyberprzestępców[7]. Firma F-Secure w 2014 roku opublikowała raport bezpieczeństwa dotyczący urządzeń mobilnych w pierwszym kwartale tego roku. Okazuje się, że aż 99% nowych zagrożeń mobilnych dotyczy Androida; z 277 nowych złośliwych programów aż 275 jest wymierzona w platformę Firmy Google. Pozostałe dwa są wycelowane w iOS i Symbiana[8]. W poprzednich

latach sytuacja była podobna co przedstawiają poniższe wykresy przygotowane przez tą samą firmę.

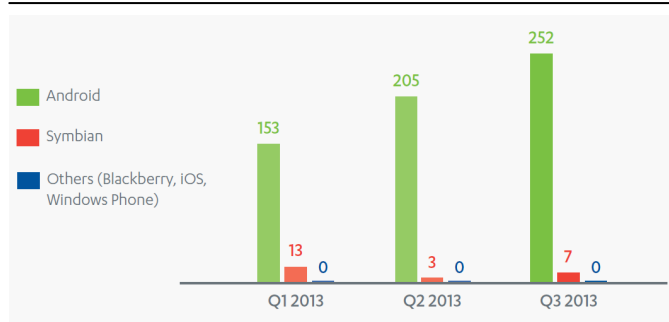


Rys. 2. Liczba wykrytego nowego złośliwego oprogramowania dla poszczególnych platform mobilnych w roku 2011 [9]

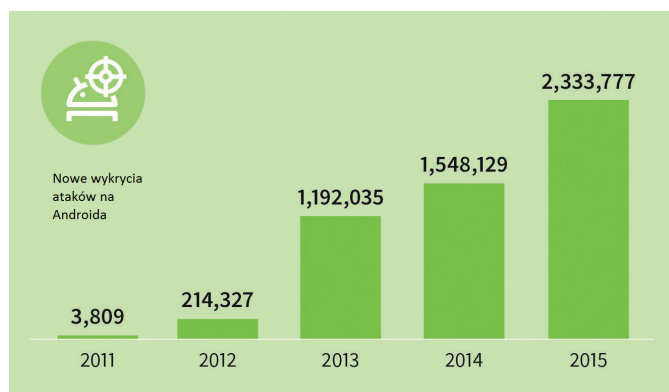


Rys. 3. Liczba wykrytego nowego złośliwego oprogramowania dla poszczególnych platform mobilnych w roku 2012 [10]

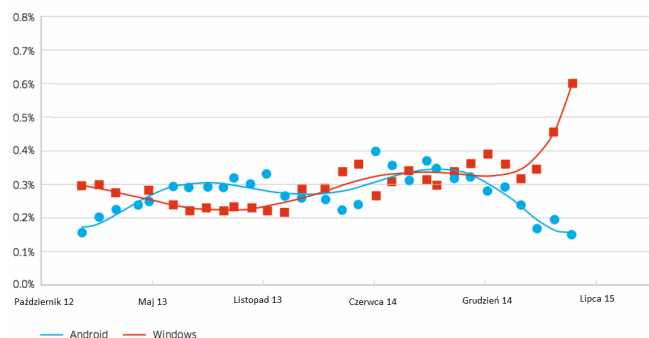
Na pierwszy rzut oka widać że najwięcej złośliwego oprogramowania pojawia się na platformę Android. Przyczynami tego zapewne jest to że na urządzenia te można instalować oprogramowanie z poza oficjalnego sklepu, platforma ta jest również najpopularniejszą co daje potencjalnie największą możliwą liczbę zainfekowanych telefonów.



Rys. 4. Liczba wykrytego nowego złośliwego oprogramowania dla poszczególnych platform mobilnych w roku 2013 [11]



Rys. 5. Liczba wykrytych ataków na Androida w poszczególnych latach [12]



Rys. 6. Procent zarażonych urządzeń na poszczególnych platformach [13]

3. Wnioski

Artykuł prezentuje ogólny zarys systemów zabezpieczających trzech najpopularniejszych platform mobilnych. Głównie pod kontem analizy i porównania mechanizmów zabezpieczających instalację i uruchamianiem oprogramowania, dostępu do danych, dostępu do telefonu i podczas łączenia się z komputerem. Ze względu na stosowaną przez firmy Apple oraz Microsoft kontrolę nad dodawanymi do ich sklepów aplikacjami, dużą liczbę zabezpieczeń oraz politykę firm nastawioną na bezpieczeństwo można stwierdzić, że obie mają bardzo wysoki poziom zabezpieczeń, który jest bardzo zbliżony do siebie ale wyższy od tego jaki posiadają urządzenia firmy Google. Najslabiej zabezpieczony jest Android.

Prawdopodobnie spowodowane jest to polityką firmy, która bardziej niż konkurencja stawia na możliwość personalizacji telefonu. Firma Apple i Microsoft stawiają bardzo dużą wagę na zabezpieczenie danych w telefonach, które posiadają ich mobilny system operacyjny. Kolejną przyczyną częstszego atakowania systemu Androida przez złośliwe oprogramowanie może być największa liczba użytkowników, która obecnie wynosi około 80% z pośród badanych platform mobilnych oraz to że wielu użytkowników posiada za małą świadomość o zagrożeniach. Co sprawia że na ten system pisane jest najwięcej złośliwego oprogramowania. Z czego natomiast wynika że najwięcej ludzi pracuje nad złamaniem zabezpieczeń Androida. Platforma Windows Phone posiada zabezpieczenia na poziomie zbliżonym do zabezpieczeń firmy Apple. Spowodowane jest to podobną polityką firmy związana z poziomem zabezpieczeń danych na telefonach z ich systemem, oraz tym że obie firmy posiadają bardzo podobne wymagania klientów co do zabezpieczeń danych. Fakt że Android ma najwięcej zarejestrowanych programistów aplikacji w serwisie Google Play, też sprawia, że Google ma najwięcej pracy przy sprawdzaniu i testowaniu nowych aplikacji. Przez co pracownicy mogą przeoczyć niektóre zagrożenia. Z faktu liczby programistów wynika też, że Android posiada zapewne najwięcej niedoświadczonych programistów, którzy popełniają błędy i piszą nieoptymalizowane aplikacje, które mogą wykorzystywać za dużo zasobów urządzenia. Przy wyborze systemu operacyjnego urządzenia należą zadać sobie kilka pytań. Czy będę używał telefonu do przechowywania bardzo ważnych i wrażliwych danych takich jak numery kart kredytowych, pin karty kredytowej czy hasła i loginy do portali internetowych. Jeżeli tak to najbezpieczniej jest wybrać urządzenie z systemem Windows Phone. Innym pytaniem jest czy nasz telefon powinien mieć możliwość szybkiego przywracania swoich poprzedniej wersji? Tego typu pytanie mogą zadać sobie na przykład programiści, którzy mają zamiar używać telefonu do testowania swoich aplikacji. Wszystkie platformy posiadają podobne rozwiązania problemów, które stawia przed nimi system operacyjny, użytkownicy oraz zagrożenia ze strony złośliwego oprogramowania. Wnioskowac z tego można że poziom zabezpieczeń podyktowany jest polityką firmy lub poziomem rozwoju danego mobilnego systemu operacyjnego. W dalszej pracy nad zabezpieczeniami platform mobilnych należałoby zbadać świadomość użytkowników na zagrożenia jakie występują przy używaniu telefonu. Określić w jakim kierunku zamierzają podążać badane systemy. Przebadać poziom wiedzy i umiejętności programistów, którzy piszą programy dla poszczególnych platform.

Literatura

- [1] Rob van der Meulen, Gartner Says Worldwide Smartphone Sales Grew 3.9 Percent in First Quarter of 2016, <http://www.gartner.com/newsroom/id/3323017> [5.01.2016].
- [2] Przegląd wirusów mobilnych – rok 2015, <http://news.drweb-av.pl/show/review/?i=9779> [11.01.2016].
- [3] Maciej Gajewski, 700 tysięcy złośliwych aplikacji w Google Play?, <http://www.chip.pl/news/bezpieczenstwo/wirusy/2013/08/700-tysiecy-zlosliwych-aplikacji-w-google-play> [13.01.2016].
- [4] Krystian Bezdziatny, iOS jest tak bezpieczny, że nawet NSA nie jest w stanie go zhakować,

- <http://komorkomania.pl/5870,ios-jest-tak-bezpieczny-ze-nawet-nsa-nie-jest-w-stanie-go-zhakowac> [11.01.2016].
- [5] Mobile Threat Report Q3 2013, https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q3_2013.pdf [9.01.2016].
- [6] Mobile Threat Report Q4 2012, <https://www.f-secure.com/documents/996508/1030743/Mobile+Threat+Report+Q4+2012.pdf> [9.01.2016].
- [7] Rob van der Meulen, Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016, <http://www.gartner.com/newsroom/id/3415117> [5.01.2016].
- [8] Mateusz Żołyniak, Android pod ostrzałem! Jak sprawdzić, czy twój telefon nie wykopuje bitcoinów, <http://komorkomania.pl/1349,jak-sprawdzic-czy-twoj-telefon-nie-wykopuje-bitcoinow> [5.01.2016].
- [9] Mobile Threat Report Q3 2012, <https://www.f-secure.com/documents/996508/1030743/Mobile+Threat+Report+Q3+2012.pdf> [9.01.2016].
- [10] Robert Hajduk, Szyfrowanie na WP8 (tylko) dla biznesu, <https://wpworld.pl/12778/szyfrowanie-na-wp8-tylko-dla-biznesu> [12.01.2016].
- [11] Adam Golański, Trzeba było uczyć się od Apple, czyli jak całodyskowe szyfrowanie Androida okazało się do niczego, <http://www.dobreprogramy.pl/Trzeba-bylo-uczyc-sie-od-Apple-czyli-jak-calodyskowe-szyfrowanie-Androida-okazalo-sie-do-niczego,News,74534.html> [12.01.2016].
- [12] G DATA Mobile Malware Report, http://pliki.gdata.pl/partner/materialy_prasowe/2016/02/MMWR_EN_Q42015.pdf [15.01.2016].
- [13] Dan Rowinski, Less Than 1% Of Smartphones Are Infected With Malware, <https://arc.applause.com/2015/09/21/android-malware-infection-rates> [16.01.2016].

Pojęcie, modele i metryki jakości oprogramowania – przegląd

Yuliia Horobets*, Marek Miłosz

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule przedstawiono analizę istniejących modeli i metryk pomiaru i oceny jakości oprogramowania. Do porównania zostały wybrane następujące modele jakości oprogramowania: McCalla, Boehma, Boeinga, FURPS, Dormey'a, ISO/IEC 9126, ISO/IEC 25000:2005 oraz metryki oprogramowania: SLOC, McCabe'a, Halsteada.

Słowa kluczowe: cykl życia; metryki jakości oprogramowania; modele jakości oprogramowania

*Autor do korespondencji.

Adres e-mail: Yuliia.horobets@gmail.com

The concept, models and metrics of software quality – an overview

Yuliia Horobets*, Marek Miłosz

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents analysis of existing models and metrics measurement and software quality assessment. For comparison were chosen following models of software quality: McCall, Boehm, Boeing, FURPS, Dormey, ISO / IEC 9126, ISO / IEC 25000: 2005 and software metrics: SLOC, McCabe, Halstead.

Keywords: software lifecycle; software quality metrics; models of software quality

*Corresponding author.

E-mail address: Yuliia.horobets@gmail.com

1. Wstęp

Celem artykułu jest próba zdefiniowania pojęcia jakości oprogramowania oraz przedstawienie przeglądu modeli i metryk, związanych z jakością oprogramowania. Modele i metryki jakości oprogramowania znajdują zastosowanie na różnych etapach cyklu życia oprogramowania: od wymagań po jakość kodu.

Metodą badawczą zastosowaną w niniejszym artykule jest systematyczny przegląd literatury przedmiotu i jej krytyczna ocena połączona z wyciąganiem wniosków.

Przegląd literatury był wykonywany w następujących krokach:

- badania literaturowe z zakresu inżynierii oprogramowania w celu ustalenia aktualnego stanu i sposobów poprawy procesu oceny jakości oprogramowania;
- analiza istniejących modeli i metryk pomiaru jakości oprogramowania;
- analiza dotycząca różnych modeli i metryk oceny jakości oprogramowania.

W trakcie przeglądu literaturowego wytypowano do opisu i analizy następujące modele jakości oprogramowania: McCalla, Boehma, Boeinga, FURPS, ISO/IEC 9126, ISO/IEC 25000:2005, SQuARE i Dromeya oraz statyczne metryki oprogramowania (kodu): SLOC, McCabe'a i Halsteada. Pierwsza grupa (modele) ocenia jakość oprogramowania z punktu widzenia użytkownika końcowego,

a druga z punktu technicznego – czyli programistów tworzących kod programu.

Modele i metryki dotyczą bowiem różnych etapów w cyklu rozwoju oprogramowania.

2. Pojęcie jakości oprogramowania

Pojęcie jakości w odniesieniu do produktu informatycznego jest trudne do sformułowania i stanowi przedmiot wielu badań teoretycznych i empirycznych. Trudność ta wynika nie tylko ze specyfiki samego produktu, ale także z procesu jego tworzenia. Zagadnienie jakości oprogramowania jeszcze bardziej komplikuje tempo zmian, jakie obserwuje się u użytkowników oprogramowania. Zmiany te powodują, że dobry produkt szybko traci wymagany poziom jakości. Wychodząc od dwóch aksjomatów jakości: jakość nie jest wielkością bezpośrednio mierzalną i jakość jest właściwością zbiorczą, proponuje się, w analogii do inżynierii jakości wyrobów, zastosowanie odpowiednich metod wartościowania, pozwalających na automatyzację wielokrotnie wykonywanych obliczeń, ułatwiających prowadzenie badań i formułowanie ocen jakości [1].

Okazuje się, że zarówno definicje jakości, jak i oceny jakości produktu nie zależą od tego, czy produkt jest materialny czy niematerialny. W obu przypadkach jakość nie jest bezpośrednio mierzalna i jest pojęciem wymagającym bliższego określenia. Dla każdego produktu programowego istotne są aspekty jakości, ustalone wcześniej dla produktów materialnych jako znamiona jakości. Określają one jakość

produktów, takich jak: pojazdy, narzędzia pracy, urządzenia pomiarowe, czy sprzęt gospodarstwa domowego, i programu komputerowego. Cechy jakości natomiast, w zależności od rodzaju produktu, mogą mieć różny poziom ważności [2].

Podobnie wygląda zagadnienie opisywalności cech składających się na jakość produktu. Najczęściej są to właściwości zbiorcze, wymagające dookreślenia. Dookreślanie polega na przedstawianiu cząstkowych kryteriów oceny jakości. W przypadku potrzeby uzyskania oceny obiektywnej, konieczna jest jej postać ilościowa. O ile dla produktów materialnych istnieją fizyczne miary, które łatwo jest zastosować do mierzenia atrybutów jakości, o tyle w przypadku produktu informatycznego zagadnienie to jest trudniejsze. Proste miary fizyczne, jak liczba bajtów czy linii kodu źródłowego programu, przedstawiające aspekt materialny produktu informatycznego, mają niewielkie znaczenie dla oceny jego jakości. Istotne aspekty jakości wymagają poszukiwania miar, adekwatnych do celu badania. Stąd też zasadniczy element wyróżniający podejście do jakości produktów materialnych i produktu informatycznego uważa się aspekt mierzalności. Jednakże nie z powodu jego znaczenia, ponieważ w obu przypadkach jest tak samo ważny, ale trudności w doborze właściwych miar.

Jakość oprogramowania to ogół funkcjonalności i cech oprogramowania, które mają wpływ na jego zdolność do zaspokojenia stwierdzonych lub domniemyanych potrzeb [3].

3. Modele jakości

Subiektywna interpretacja pojęcia jakości wymaga, aby dla otrzymania obiektywnego opisu przyjąć za podstawę uniwersalny model jakości. Celem analizy wybranych modeli, spośród wielu opisanych w literaturze przedmiotu, było określić najbardziej odpowiedni model dla zapewniania jakości oprogramowania na wszystkich etapach cyklu życia z kosztem najmniejszych zasobów.

W omawianych modelach (tab. 1) przyjęto zasadę określania składników jakości oprogramowania za pomocą złożonych charakterystyk oraz atrybutów zewnętrznych i wewnętrznych.

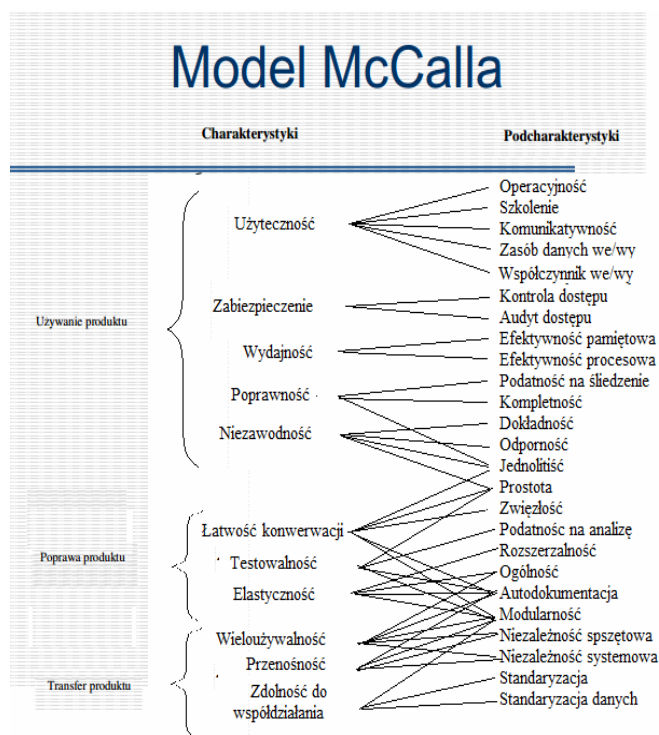
Tabela 1. Liczba charakterystyk modeli jakości oprogramowania

Model	Liczba charakterystyk/ podcharakterystyk
McCalla(1977)	11/25
Boehma(1978)	7/12
Boeinga(1987)	15/27
FURPS(1987)	5/26
Dromey	4/7
ISO/IEC 9126(2001)	6/27
ISO 2500	8/27

Pomiędzy nimi zachodzą relacje, które pozwalają określić wpływ atrybutów wewnętrznych na zewnętrzne, co w przypadku badania jakości oprogramowania może być wykorzystane np. do przewidywania jakości produktu końcowego na podstawie mierzonych atrybutów jakości wewnętrznej.

Model McCalla (1997) został opracowany na zlecenie Sił Zbrojonych USA, w celu poprawy jakości wytwarzanego oprogramowania [4].

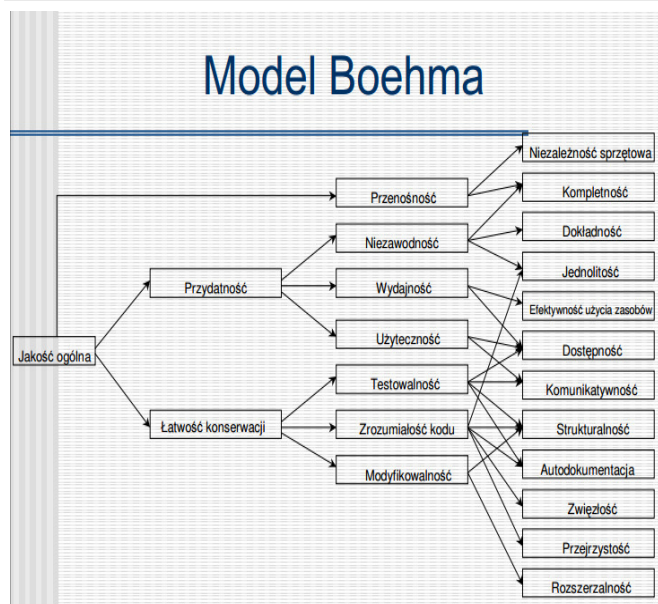
Oprogramowanie z perspektywy użytkownika jest badane ze względu na trzy wymiary: działanie, poprawianie oraz różne aspekty dotyczące zmian w środowisku. Są to tzw. zewnętrzne atrybuty jakości wysokiego poziomu, charakteryzujące się brakiem możliwości bezpośredniego mierzenia. Na tej podstawie wprowadzono 25 atrybutów niższego poziomu, który są nazwany kryteriami jakości. Jednak, że wiele z tych kryteriów jakości wymaga dalszej dekompozycji, w celu doprowadzenia do możliwości bezpośredniego ich mierzenia. Atrybuty mierzalne, nazywane metrykami jakości, znajdują się na czwartym poziomie tego drzewa (rys. 1).



Rys. 1. Model McCalla [4]

Ten model stosowany jest wyłącznie do oceny jakości produktu końcowego, ponieważ nie było zaproponowane podejście do stosowania tego modelu w innych etapach cyklu życia oprogramowania. Model trudno zastosowywać do nowoczesnych technologii i technik rozwoju oprogramowania. Metody ustalania właściwości wagi charakterystyk jakości są subiektywne, ponieważ są one oparte na wyborze konkretnych deweloperów. Wykorzystywane są liniowe metody, które nie biorą pod uwagę korelacji między charakterystykami. Główną wadą stosowania modelu McCall jest to, że każda metryka wpływa na ocenę innych czynników jakości.

W modelu Boehma [5] korzeń drzewa jest nazwany jakością ogólną. Składnikami tej jakości są użyteczność, pielęgnalność oraz przenośność (rys. 2).



Rys. 2. Model Boehma [5]

Punkt widzenia użytkownika został określony przez użyteczność, pozostałe dwa składniki sformułowano natomiast z punktu widzenia wytwórcy zainteresowanego łatwością modyfikowania i rozumienia kodu. Na poziomie pierwszym pojawiają się atrybuty zewnętrzne, zwane składnikami pośrednimi. Poziom następny zawiera 12 składników, nazwanych podstawowymi, które odpowiadają atrybutom wewnętrznym. i tutaj, podobnie jak u McCalla, dopiero czwarty poziom wskazuje na atrybuty mierzalne.

Model Boeinga [6] składa się z dwóch poziomów: na pierwszym określono 15 atrybutów zewnętrznych, a na drugim 27 atrybutów wewnętrznych (rys 3).

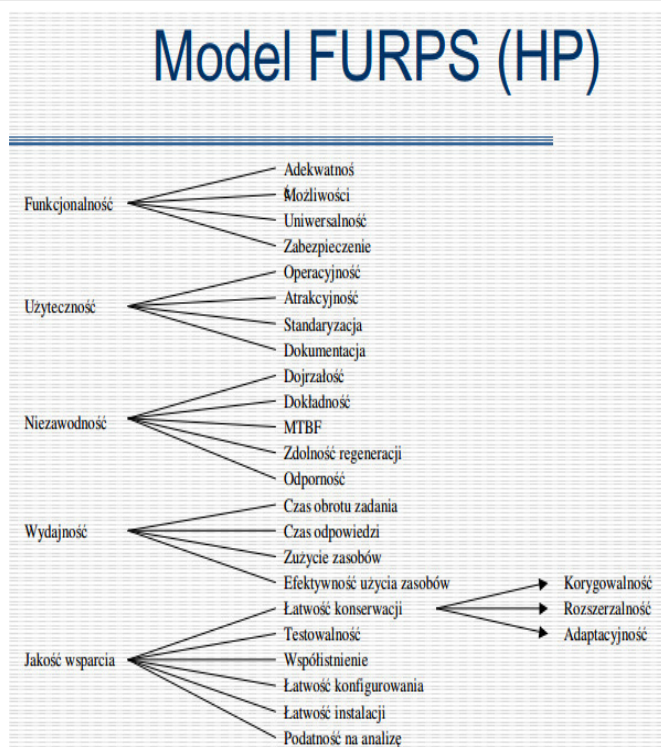
Nie wszystkie wewnętrzne atrybuty są bezpośrednio mierzalne, np. dokumentacja czy szkolenia, dlatego zastosowanie tego modelu wiąże się z ustaleniem mierzalnych atrybutów na kolejnym poziomie.

Model FURPS/FURPS+(1992) zaproponowany przez Roberta Grady'ego [7], składa się z pięciu charakterystyk (Funkcjonalność, Użyteczność, Niezawodność, Wydajność i Wsparcie).

FURPS zbudowany podobnie do modelu McCall i Boehm, ale w przeciwieństwie do nich składa się z dwóch warstw, pierwsza określa charakterystyki, a druga - związane z nimi atrybuty (rys. 4).

Ten model może być wykorzystany zarówno do oceny jakości, jak i specyfikacji wymagań funkcjonalnych i niefunkcjonalnych dla produktu.

Model FURPS + jest szeroko stosowany w tworzeniu oprogramowania oraz przy identyfikacji wymagań dla systemu w fazie rozwoju.



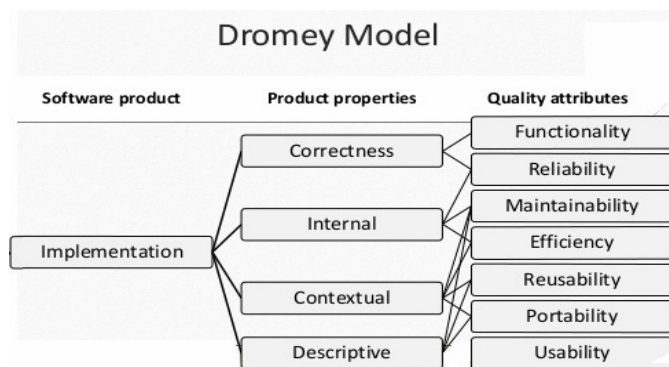
Rys. 3. Model Boeinga [6]



Rys. 4. Model FURPS [7]

Model zaproponowany przez Dromey'a w 1994 roku [8] i w 1998 koncentruje się na jakości produktu w oderwaniu od sposobu jego wytworzenia, co było pewnym przełomem w postrzeganiu jakości oprogramowania w latach 90'tych XX wieku. Model ma na celu zwiększenie wiedzy na temat relacji między atrybutami (charakterystyka) oraz atrybutów podrzędnych (sub-charakterystyka) jakości. Model Dromey'a definiuje dwie warstwy, atrybuty wysokiego szczebla

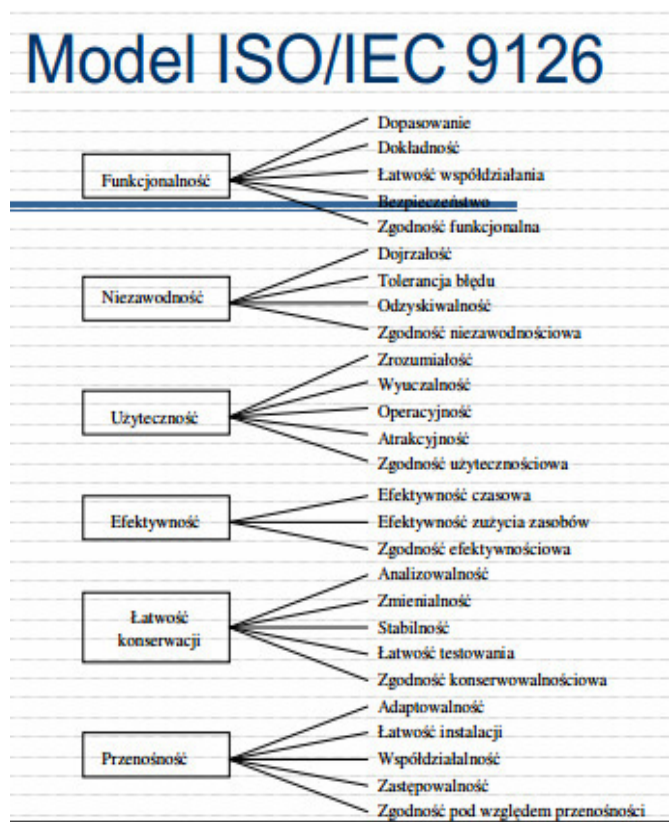
i atrybuty podległych (rys 5). Dlatego też model ten cierpi z powodu braku kryteriów mierzenia jakości oprogramowania.



Rys. 5. Model Dromey'a [8]

Jakość produktu, według autora, ma być jakością artefaktów związanych z tym produktem. Przez produkt w tym modelu nie był rozumiany wyłącznie produkt końcowy, ale także dokumentacja, projekt itd.

Model ISO/IEC 9126 [9], opracowany w 2001 roku, stanowi rozszerzenie modelu z roku 1991, opracowanego na bazie modelu McCalla oraz Boehma. Jego podstawą jest zestaw sześciu charakterystyk (rys.6). Wprowadzono trzy nowe pojęcia: jakość wewnętrzną, jakość zewnętrzną i jakość użytkową.

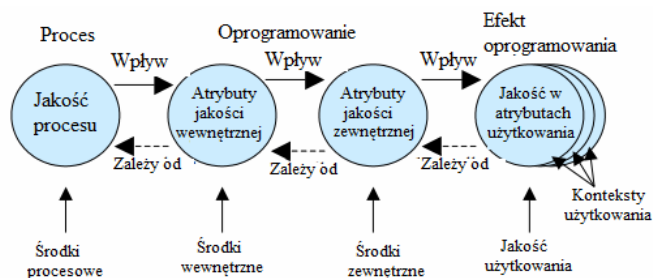


Rys. 6. Model ISO/IEC 9126 [9]

Dla każdej z nich określono odpowiednie zestawy spośród 27 podcharakterystyk. Zewnętrzne podcharakterystyki odzwierciedlają efekt użycia oprogramowania, stanowiącego

istotną część systemu komputerowego i są wynikiem jakości atrybutów wewnętrznych tego oprogramowania. Dla jakości użytkowej określono cztery charakterystyki, podkreślając jej uzależnienie od jakości wewnętrznej i zewnętrznej.

Model zdefiniowany w normie pokazuje również różne aspekty jakości dyskutowane we wcześniejszych modelach, czyli jakość produktu, jakość techniczną produktu, a jakość procesu, tak jak to przedstawiono na rysunku 7.



Rys. 7. Jakość w cyklu życia systemu [9]

Każdy z tych obszarów ma odmienne znaczenie i odmiennego interesariusza. Wewnętrzna i zewnętrzna jakość definiowana jest jako suma następujących charakterystyk: funkcjonalność (*functionality*), niezawodność (*reliability*), użyteczność (*usability*), wydajność (*efficiency*), łatwość utrzymania (*maintainability*), przenośność (*portability*).

Jakość użyteczna jest definiowana jest jako suma następujących charakterystyk: przydatność (*effectiveness*), produktywność (*productivity*), bezpieczeństwo (*safety*) i zdolność do zaspokojenia potrzeb (*satisfaction*).

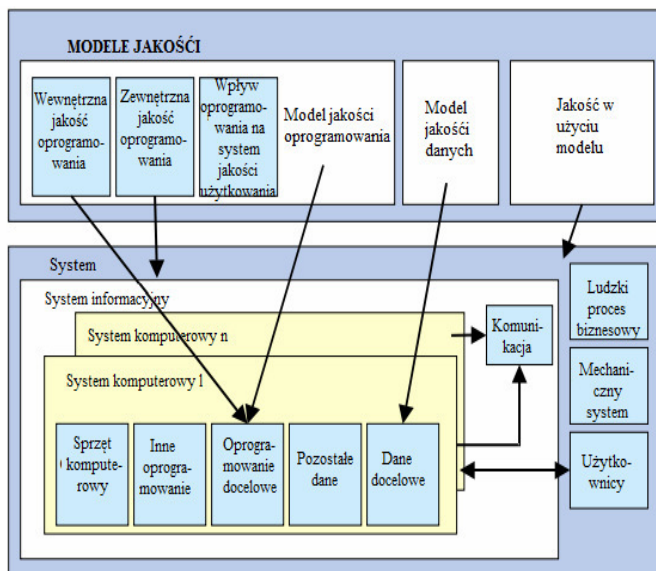
Te właściwości i atrybuty pozwalają systematycznie opisywać wymagania do oprogramowania, określając, które właściwości oprogramowania dla tej charakterystyki chcą widzieć osoby na pewnym etapie cyklu życia.

Model ISO/IEC 25000:2005. Model nazywany Drugą Generacją Standardów Jakości - SQuaRE (*Software product Quality Requirements and Evaluation*) [10] publikowany w serii norm ISO/IEC 25000.

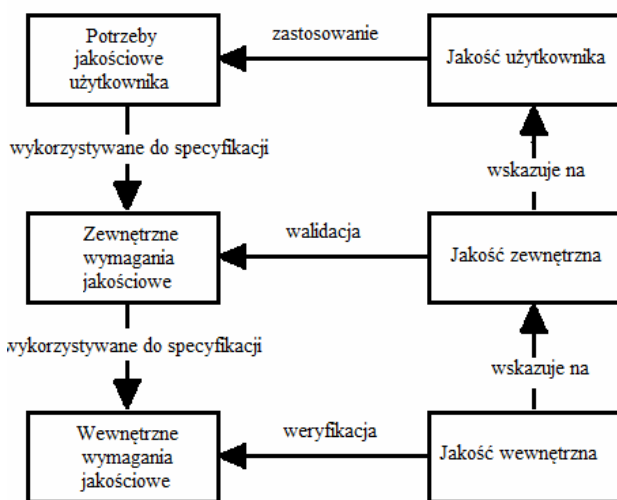
Na najwyższym poziomie modelu jakości oprogramowania pozostawiono podział na jakość wewnętrzną, zewnętrzną i użyteczną. Jakość wewnętrzna i zewnętrzna oprogramowania zdefiniowana jest analogicznie do definicji w ISO/IEC 9126:2001 [9], z tym że zmieniono charakterystyki wysokiego poziomu – zamiast sześciu w nowym modelu jest osiem: funkcjonalność (*functionality*), poufność (*security*), zgodność techniczna (*interoperability*), niezawodność (*reliability*), użyteczność (*usability*), wydajność (*efficiency*), łatwość utrzymania (*maintainability*) i przenośność (*portability*).

Jakość użyteczna oprogramowania, została zdefiniowana na nowo w modelu SQuaRE. Jej główne charakterystyki to: użyteczność (*usability in use*), zgodność z kontekstem (*context in use*), bezpieczeństwo (*safety in use*), poufność (*security in use*) i łatwość użycia (*adaptability in use*).

Modele jakości SQuaRE opisują zarówno oprogramowanie, jak i dane. Twórcy normy zauważają, że nie są to jedyne modele jakości dla bytu nazywanego „systemem”, lecz norma odnosi się tylko do wspomnianych modeli. Na rysunkach 8 i 9 pokazano mapowanie modeli jakości SQuaRE na model systemu, oraz cykl życia jakości w ramach rozwoju produktu jakim jest oprogramowanie.



Rys. 8. Odniesienie modeli SQuaRE do jakości elementów systemu [10]



Rys. 9. Model jakości w cyklu życia oprogramowania [10]

Spośród wszystkich istniejących modeli ten model najlepiej nadaje się do oceny jakości oprogramowania na wszystkich etapach cyklu życia.

Analiza porównawcza właściwości różnych modeli jakości najlepiej złożyć w formie tabeli 2.

Tabela 2. Analiza porównawcza modeli jakości oprogramowania [11]

Model \ Charakterystyka jakości	McCalla	Boehma	FURPS	ISO 9126	ISO 25010	Dromey
Poprawność					+	
Niezawodność	+	+	+	+	+	+
Skuteczność	+	+	+	+	+	+
Elastyczność	+			+	+	
Funkcjonalność			+	+	+	+
Ergonomia		+				
Integralność					+	
Zgodność	+			+	+	
Wsparcie	+	+	+	+	+	+
Możliwość zmiany		+		+	+	
Produktywność			+			
Przenośność	+	+		+	+	+
Stabilność				+	+	+
Skalowanie					+	
Bezpieczeństwo			+		+	
Dojrzałość					+	+
Testowalność	+	+		+	+	
Zrozumiałość		+	+	+	+	
Praktyczność	+		+	+	+	+

4. Metryki jakości oprogramowania

Metryki stanowią szybki i wygodny sposób oceny jakości oprogramowania. Znajomość wartości poszczególnych metryk i wielkości, które mierzą, pozwala oszacować złożoność systemu, koszt jego pielęgnacji czy elastyczność.

Metryka oprogramowania – miara pewnej własności oprogramowania lub jego specyfikacji [12].

Celem użycia metryk oprogramowania jest otrzymanie dokładnych wartości, które dotyczą wytwarzanych aplikacji. Metryki pozwalają na obiektywne spojrzenie na oprogramowanie i porównanie ze sobą poszczególnych jego elementów lub różnych produktów. Mierzenie jakości aplikacji oraz wydajności pracy bez danych liczbowych są bardzo trudne, a utrzymanie obiektywności jest niemal niemożliwe.

Obliczanie metryk jest istotnym ułatwieniem we wszystkich fazach procesu wytwarzania oprogramowania. Na tym etapie metryki są przydatne przede wszystkim dla klientów oraz projektantów. w fazie produkcji dużą rolę odgrywają metryki statyczne, pomagające w utrzymywaniu jakości kodu źródłowego. Korzystają z tego programiści, którzy dzięki metrykom mogą łatwo odnajdywać te miejsca w kodzie, które są potencjalnym źródłem nadmiernej złożoności, a co za tym idzie, powstających błędów. w fazie testów wykorzystywane są zarówno metryki statyczne, jak i dynamiczne, pozwalające na badanie przebiegu wykonania programu. Na tym etapie gromadzone są też dane liczbowe dotyczące wydajności czy niezawodności aplikacji. Wyniki testów są pożytecznym źródłem informacji dla wszystkich osób zaangażowanych w tym projekcie: kierowników, programistów i klientów [13].

W ocenie złożoności oprogramowania występują trzy główne grupy metryk:

- metryki rozmiaru;
- metryki logicznej struktury programu, czyli przepływu sterowania;
- metryki struktur danych.

Metryki rozmiaru są najbardziej proste, pozwalają na ocenę jakości kodu źródłowego i łączą się ściśle z analizą statyczną, dziedziną inżynierii zajmującą się badaniem struktury kodu źródłowego. Metryki te najbardziej przydatne są dla samych programistów i innych osób bezpośrednio zaangażowanych w proces powstawania oprogramowania. Pozwalają na bieżące śledzenie jakości kodu i zwrócenie uwagi na miejsca, które wymagają uproszczenia bądź szczególnie uważnego testowania.

Do tego typu metryk należy liczba linii kodu źródłowego - najprostsza metryka rozmiaru oprogramowania, oznaczana jako LOC (*lines of code*) lub SLOC (*source lines of code*). Wielkość ta daje ogólne pojęcie o skali programu, nie jest jednak wystarczająca do bardziej szczegółowych analiz. Wartość LOC jest bardziej zależna od użytego języka programowania – ten sam algorytm w języku wysokiego poziomu będzie miał wielokrotnie mniej linii kodu niż w assemblerze. Istotny jest też sposób zliczania linii kodu – ich liczba może się zdecydowanie zmieniać, zależnie od tego, czy wliczane są puste linie, komentarze czy linie zawierające jedynie nawiasy klamrowe bądź słowa kluczowe *begin* i *end*. Liczba linii z pominięciem wymienionych elementów jest określana jako eLOC (*effective lines of code*) [14].

Jedną z najbardziej podstawowych metryk złożoności jest CC (złożoność cyklomatyczna) zaproponowana przez Thomasa J. McCabe'a w 19760 [15]. Pierwotnie pomyślana była jako metryka dla programów strukturalnych, nadaje się jednak równie dobrze do programów obiektowych, dzięki czemu jest popularna także dzisiaj. Złożoność cyklomatyczna jest liczbą charakteryzującą funkcję lub metodę i odzwierciedlającą jej poziom skomplikowania.

Początkowo złożoność cyklomatyczna miała mierzyć liczbę niezależnych ścieżek przebiegu programu, co bezpośrednio przekłada się na łatwość jego przetestowania. Ponieważ skoki wstecz mogą spowodować nieskończony wzrost takich ścieżek, mierzy się liczbę prostych ścieżek bez uwzględniania cykli.

Złożoność cyklomatyczną dla danej funkcji można policzyć, mając do dyspozycji graf przebiegu programu. Wówczas wyraża się ona wzorem [15]:

$$v(G) = e - n + 2 \quad (1)$$

gdzie e - oznacza liczbę krawędzi, a n - liczbę wierzchołków w grafie. Inny wzór podaje prostą zależność CC od liczby węzłów, w których podejmujemy decyzje:

$$v(G) = d + 1 \quad (2)$$

gdzie d - to liczba węzłów decyzyjnych, w których podejmowana jest decyzja o charakterze binarnym (tak/nie). Taki wzór pozwala na szybkie wyznaczanie wartości CC – wystarczy tu zliczyć liczbę wystąpień słów kluczowych *if*, *while*, *for* czy *case*.

Niska wartość CC wskazuje na łatwość zrozumienia danej funkcji czy metody. Wartość powyżej 20 charakteryzuje złożony kod i wiąże się z dużym ryzykiem wystąpienia błędów.

Zaletami metryki CC są m.in. łatwość jej obliczenia i możliwość wskazania elementów aplikacji, które należy przeprojektować. Liczba rozgałęzień przepływu programu nie daje jednak pełnej informacji, ponieważ nie rozróżnia zagnieżdżonych i niezagnieżdżonych pętli oraz prostych instrukcji *case*, a także nie bierze pod uwagę skomplikowanych warunków w węzłach decyzyjnych. Istnieją modyfikacje definicji CC, które pozwalają zredukować lub wyeliminować te wady [15].

Złożoność Halsteada. Dana metryka dotyczy rozmiaru programu pozwalając na zdefiniowanie bardziej skomplikowanych metryk złożoności. Wyróżnia się wśród nich [16]:

- trudność (*difficulty*), czyli podatność na błędy, proporcjonalna do liczby unikatowych operatorów (n_1), a także do stosunku pomiędzy wszystkimi (N_2) a unikatowymi operandami (n_2):

$$D = \left(\frac{n_1}{2} \right) \left(\frac{N_2}{n_2} \right) \quad (3)$$

- poziom programu (*program level*), im niższy, tym bardziej aplikacja jest podatna na błędy:

$$L = 1/D \quad (4)$$

- wysiłek (*effort*) potrzebny do zaimplementowania programu, proporcjonalny do trudności oraz objętości:

$$E = V \cdot L \quad (5)$$

- czas (*time*) potrzebny do zaimplementowania, wyrażony w sekundach – współczynnik 18 został wyznaczony eksperymentalnie i jest modyfikowalny indywidualnie dla danego programisty:

$$T = E/18 \quad (6)$$

- szacunkowa liczba błędów (*number of delivered bugs*) – wartość ta jest zależna od użytego języka programowania, jest na przykład wyższa dla aplikacji w C++:

$$B = E^{\frac{2}{3}} / 3000 \quad (7)$$

Niniejsze wartości mimo oczywistej niedokładności stanowią cenną informację podczas etapu testowania aplikacji – przykładowo wartość B w porównaniu z rzeczywistą liczbą znalezionych błędów może dać wskazówki na temat efektywności przeprowadzonych testów.

5. Wnioski

W podsumowaniu opisanych modeli jakości stwierdza się występowanie cech wspólnych: punkt widzenia, złożoność pojęcia jakości, złożoność charakterystyk, problem mierzalności atrybutów, występowanie zależności między charakterystykami i podcharakterystykami. Zauważone różnice dotyczą natomiast: rodzaju specyfikowanych charakterystyk, liczebności charakterystyk na różnych poziomach, kwalifikacji atrybutów na wewnętrzne i zewnętrzne. Modele te stanowią wzorce, które dostarczają istotną pomoc przy określaniu zestawu charakterystyk jakości.

Podsumowując można stwierdzić, że modele ISO 9126 i ISO 25010 są najlepiej przystosowane do kontroli jakości, które mają najwyższe kryteria jakości i pozwalają ich obliczyć na każdym etapie cyklu życia oprogramowania.

Analiza wartości metryk dla tworzonego kodu programu pozwala w szybki i zgrubny sposób ocenić jakość programu, a przede wszystkim jego projektu. Dzięki temu można łatwo stwierdzić istnienie błędów w strukturalizacji danych lub nieprawidłowej faktoryzacji klas.

Literatura

- [1] ISO 8402:1994 - Quality management and quality assurance. 1994
- [2] Поморова О.В. Сучасні проблеми оцінювання якості програмного забезпечення, Радіoeлектронні і комп'ютерні системи. – 2013. - №5(64). – с. 319-327.
- [3] ISO/IEC 25000:2014 - Systems and software engineering. 2014
- [4] McCall J. A. i inni, Factors in Software Quality: Metric Data Collection and Validation. Final Technical Report. Vol. 1, National Technical Information Service, Springfield 1977.
- [5] Boehm B.W, Characteristics of Software Quality, TRW Series of Software Technology, Amsterdam, North Holland 1978
- [6] Deutsch M.S., Willis R.R.:Software Quality Engineering: A Total Technical and Management Approach, Prentice-Hall 1988
- [7] Grady R.B.: Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, Englewood Cliff. 1992
- [8] Dromey R.G.: A model for software product quality. IEEE transaction on Software Eginneering, 21(2), 1995, s 145-162
- [9] ISO/IEC 9126- - Software engineering. - Product quality. - Part 1: Quality model / ISO/IEC, 2001.
- [10] ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuARE) - System and software quality models / ISO/IEC. – 2011.
- [11] Жарко Е. Ф. Сравнение моделей качества программного обеспечения: аналитический подход, Институт проблем управления им. В. А. Трапезникова, Москва 2014.
- [12] Метрики //http://www.metrix.narod.ru/page1.htm.[27.09.2016]
- [13] Харченко О. Розробка та керування вимогами до програмного забезпечення на основі моделі якості – Вісник ТДТУ – 2009. Том 14. №1. – с. 201-207
- [14] Ледовских И. Метрики сложности кода. – 2012. – с. 22.
- [15] Т. McCabe, A Software Complexity Measure. IEEE Transactions on Software Enginnering, 1976.
- [16] Харченко В.С. Использование метрик Холстеда при оценке безопасности критического программного обеспечения 2003. – с.7.

Porównywanie algorytmów identyfikacji fiksacji oraz sakkad w zapisie okulograficznym dla potrzeb diagnostyki medycznej

Marcin Ignatiuk*, Przemysław Antoni Pasieczny, Malgorzata Plechawska-Wójcik, Magdalena Borys

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł przedstawia analizę porównawczą wybranych algorytmów identyfikacji ruchów oka. Analiza polega na porównaniu wyników klasyfikacji fiksacji oraz sakkad przez algorytmy I-VT oraz I-DT. Dla porównania algorytmów stworzona została aplikacja implementująca identyfikację danych z badań okulograficznych. Wyniki badań zostały przedstawione w formie wykresów.

Słowa kluczowe: Okulografia; I-VT; I-DT

* Autor do korespondencji.

Adres e-mail: general.91@o2.pl

Comparison of fixation and saccades identification algorithms in the eye tracking recordings for medical diagnosis

Marcin Ignatiuk*, Przemysław Antoni Pasieczny, Malgorzata Plechawska-Wójcik, Magdalena Borys

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents a comparative analysis of selected algorithms for eye movements identification. The analysis is based on a comparison of the classification of fixation and saccades using I-VT and I-DT algorithms. For the algorithms comparison the application that implementing the identification of eye tracking data has been developed. The results of the analysis are presented in the form of graphs.

Keywords: Eye Tracking; I-VT; I-DT

*Corresponding author.

E-mail address: general.91@o2.pl

1. Wstęp

Obecnie takie gałęzie przemysłu jak medycyna czy marketing szybko się rozwijają. Jest to spowodowane koniecznością sprostania coraz większym potrzebom czy oczekiwaniom rynku. Niezwykle ważną rolę zaczyna odgrywać tu dziedzina nauki zwana okulografią. Polega ona na analizowaniu wzroku ludzkiego za pomocą różnych technik badawczych. Dzięki okulografii możliwa jest m.in. obserwacja tzw. AOIs (*area-of-interests*), czyli obszarów zainteresowania. Wspomaga to marketing poprzez zrozumienie na czym skupia się uwaga klientów np. przy zakupie danego produktu. Medycyna również może mieć pożytek z korzyści jakie niesie eyetracking. Możliwe stało się bowiem m.in. badanie zależności między odpowiednimi ruchami oczu, a identyfikacją schorzeń mózgu. Występuje wiele technik pozwalających na badanie ruchów oczu począwszy od bezpośrednich obserwacji, przez używanie wideorejestracji, aż po badania EOG (elektrookulografia).

Celem niniejszego artykułu jest porównanie algorytmów identyfikacji ruchów oczu, opartych na metodach związanych z rejestracją wideo. Artykuł ma również na celu zbadanie słuszności tezy, iż wyniki działania algorytmów opartych na rozproszeniu (ang. *dispersion-based*) oraz prędkości (ang. *velocity-based*) są zbliżone.

2. Rodzaje ruchów gałki ocznej

Sakkady są to szybkie ruchy gałek ocznych przemieszczające obraz obserwowanego obiektu z obwodowej części siatkówki do jej centrum. Powoduje to, iż jest on lepiej widoczny [1]. Podczas tego ruchu oko porusza się z prędkością 30 - 700°/s. Istnieje zależność między amplitudą ruchu a jego prędkością. Gdy amplituda się zwiększa, prędkość ruchu także. Opóźnienie między bodźcem powodującym ruch a samym ruchem wynosi 150 - 250 ms. Podczas samego ruchu widzenie nie jest możliwe, podobnie jak zmiana jego rozpoczętej trajektorii. Sakkady są wyzwalane nie tylko poprzez bodźce wzrokowe, ale również przez dotyk czy słuch. Do sakkad zaliczane są również fazy oczopląsu podczas stymulacji przedsionkowej czy optokinetycznej, a także szybkie ruchy oczu podczas fazy snu REM. Ruchy sakkadowe są mimowolne, a za ich regulację odpowiadają wzgórki czworacze górne stanowiące podkorowy ośrodek wzroku. Ruchy sakkadowe pojawiają się w pierwszych tygodniach życia człowieka [2].

Fiksacja jest momentem pozornego utrzymania skupienia wzroku na danym punkcie. Reprezentują również większość czasu, w którym pobierana jest informacja wzrokowa. Fiksacja trwa około 200 - 300ms. Mimo, że oczy wydają się być wtedy nieruchome, wykonują mikroskopijne ruchy - niedostrzegalne gołym okiem [2]. Tymi ruchami są:

- mikrosakkady - amplituda ruchu do 0,1°, rola ruchów nie jest znana;

- mikrodryfy - amplituda ruchu jak u mikrosakkady, prędkość ruchu do 0,25°/s, zapobiega zanikaniu nieruchomego obrazu, związanego z adaptacją receptorów do oświetlenia;
- mikrodrżenia - amplituda wielokrotnie mniejsza niż przy mikrodryfie, częstotliwość ruchu 50 - 100 Hz, stała aktywność gałek ocznych przy fiksacji.

3. Badania eyetrackingowe

Badania eyetrackingowe mają swoje zastosowania w wielu dziedzinach nauki takich jak medycyna, informatyka, psychologia czy marketing. Na podstawie marketingowych badań eyetrackingowych można sprawdzić np. gdzie najczęściej skupiany jest wzrok klientów sklepów tak, aby następnie odpowiednio rozmieścić sprzedawane towary. Badania okulograficzne są bardzo pomocne również przy tworzeniu stron internetowych tak, aby zwiększyć użyteczność strony internetowej umieszczając w odpowiednich miejscach nawigację po stronie, czy też baner reklamowy. Badania eyetrackingowe można przeprowadzić na podstawie bezpośredniej obserwacji ruchów oczu, jak w przypadku podstawowych badań lekarskich, czy też z użyciem okulografów. Najczęściej stosowanym sposobem badań są pomiary z wykorzystaniem wideo – rejestratorów.

Stosowanie takiego sposobu badania pozwala na dogłębną analizę zarejestrowanego badania. Wyniki badań z wideo rejestracji przedstawiane są za pomocą:

- map cieplnych,
- map fiksacji,
- nagrań z naniesionymi markerami w punktach aktualnego skupienia wzroku,
- zestawień tabelarycznych i wykresów dostosowanych do rejonów zainteresowania [10].

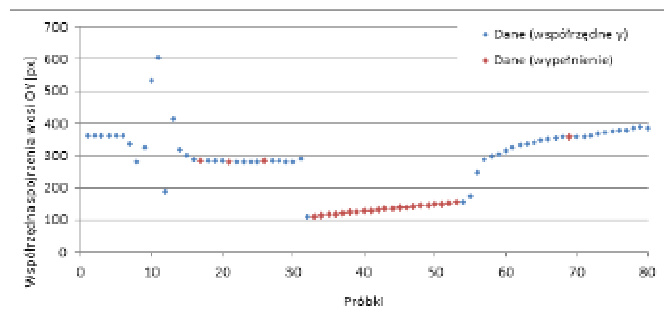
4. Obróbka danych eyetrackingowych

Przed analizą danych za pomocą algorytmów identyfikacji fiksacji oraz sakkad, powinny one zostać wstępnie zmodyfikowane. Jest to spowodowane występowaniem zakłóceń podczas pomiarów. Zakłócenia te - tzw. szumy, ujawniają się jako przypadkowe skoki wartości danych. Aby wyeliminować zakłócenia należy użyć filtru wygładzającego sygnał. Działa on na zasadzie usuwania wszelkich danych, których wartości - amplituda czy częstotliwość, wykraczają poza daną skalę. Filtr ten stosować należy na danych przed ich analizą. Alternatywą dla tego rozwiązania jest pogrupowanie pojedynczych fiksacji, dla których liczona jest tzw. średnia ruchoma. W ten sposób wygładzamy sygnał z danych o nadmiernych wartościach [8].

Kolejnym problemem podczas zapisu danych eyetrackingowych, są chwilowe ich braki spowodowane m.in. mruganiem czy wystąpieniem problemu podczas próby zapisu próbki. Innymi przyczynami mogą być momenty odwrócenia wzroku od eyetrackera, umieszczenie przeszkody między eyetrackerem, a oczami badanego, wady sprzętu lub opóźnienia systemu. Te czy inne przyczyny powstawania luk danych skutkują chwilowymi zapisami pustych wartości. Jeśli dana luka wystąpi podczas fiksacji, może ona być zaliczona jako sakkada [3]. Aby zapobiec takim przypadkom stosuje się

algorytmy, uzupełniające luki w danych. Algorytmy mogą realizować to zadanie m.in. poprzez:

- wypełnianie luk danymi będącymi wartościami liniowo rosnącymi / malejącymi między wiadomymi wartościami, przy których wystąpiła luka - tzw. interpolacja liniowa;
- wypełnienie luk danymi będącymi wartościami równymi średniej wiadomych wartości sygnału [4].



Rys. 1. Wykres prezentujący dane po działaniu funkcji wypełniania luk na fragmencie danych ze sprzętu SMI

Wykres zaprezentowany na rysunku 1 przedstawia działanie funkcji *GapFillin*, która odpowiada za wypełnianie pustych danych za pomocą interpolacji liniowej. Oznacza to zatem, wstawianie w miejsca luk wartości, które są wyznaczane na podstawie ilorazu różnicy położenia między ostatnimi znanymi próbkami do liczby brakujących próbek. Na zamieszczonym przykładzie uwidocznione są próbki oznaczone kolorem czerwonym, które zostały dodane w puste miejsca - fragment od 30-tej do 55-tej próbki.

5. Algorytmy identyfikacji fiksacji oraz sakkad

Według taksonomii Dario D. Salvucciego oraz Josepha H. Goldberga, algorytmy identyfikacji fiksacji możemy podzielić ze względu na cechy przestrzenne i czasowe. Klasyfikacja przez nich przedstawiona jest dobrym początkiem w drodze do porównań typowych algorytmów identyfikacji fiksacji. Jako cechy przestrzenne założone zostały trzy kryteria, które determinują trzy podstawowe typy algorytmów. Są nimi algorytmy identyfikacji oparte na szybkości (ang. *velocity-based*), oparte na rozproszeniu (ang. *dispersion-based*) oraz oparte na obszarze zainteresowania (ang. *area-based*). Biorąc pod uwagę dane przetwarzane algorytmami identyfikacji fiksacji, które zazwyczaj składają się ze współrzędnych punktów i są gromadzone w ustalonych jednakowych odstępach czasu, algorytmy *velocity-based* jako kryterium prędkości używają różnic między odległościami z kolejnych współrzędnych. Algorytmy *dispersion-based* oparte są na rozproszeniu, czyli odległości między współrzędnymi kolejnych próbek danych. Fiksacje - z racji na swój pozorny stan spoczynku wzroku, powinny być odzwierciedlone w danych, których współrzędne są skupione wokół jednego punktu. Trzeci rodzaj algorytmów *area-based* bazuje na tzw. obszarach zainteresowania wzroku. Składają się one na szczególne obszary, na którym skupia się badany tzw. AOIs (ang. *area of interest*). Oprócz wcześniejszych podziałów, wspomniani badacze wyznaczyli dodatkowe dwa kryteria podziałów jakimi są, podział ze względu na długość trwania oraz adaptacyjność. Długość trwania brana jest z typowego dla fiksacji czasu jej trwania tj. od 200 do 400

ms. Kryterium adaptacyjności mówi o tym, jaki wpływ na interpretację danych punktów, ma interpretacja punktów zbliżonych w czasie. Taksonomie tę dla wybranych algorytmów przedstawia tabela 1 [5].

Tabela 1. Podział algorytmów identyfikacji fiksacji i sakkad zgodnie z taksonomią Dario D. Salvucci oraz Josepha H. Goldberg

Kryteria		Algorytmy	
		I-VT	I-DT
przestrzenne	przemieszczenie	X	
	rozproszenie		X
	obszar		
tymczasowe	trwanie		X
	adaptacyjność		X

5.1. Algorytm I-VT

Algorytm I-VT (ang. *Velocity Threshold Identification*), czyli algorytm oparty na przemieszczeniu, pod względem implementacyjnym jest jednym z najprostszych. Algorytm ten z danego zbioru danych wydziela fiksacje i sakkady, bazując na przemieszczeniach od jednego punktu do następnego. Zazwyczaj, jako fiksacje traktuje się próbki o niskiej wartości przemieszczenia (prędkości) < 100 stopni na sekundę, a jako sakkady traktuje się próbki o wysokiej prędkości przemieszczenia > 300 stopni na sekundę. Działanie algorytmu I-VT zaczyna się od wyznaczenia prędkości przemieszczeń między kolejnymi próbkami. Zakładając wcześniej wspomnianą cechę danych eyetrackingowych tj. równą częstotliwość pomiarów, zamiast prędkości liczone będą zatem tylko odległości między punktami. W następnym punkcie algorytm I-VT - bazując na tej odległości, kwalifikuje dany punkt jako fiksację bądź sakkadę. Dzieje się to poprzez porównanie odległości między punktami do ustalonego progu (ang. *threshold*) [5]. Jeśli dana odległość jest poniżej danego progu, punkt oznaczany jest jako fiksacja, jeśli powyżej jako sakkada. Kolejnym etapem jest pogrupowanie punktów oznaczonych jako fiksacje w grupy fiksacji i eliminacja punktów sakkad. Końcowym etapem jest przyporządkowanie danej grupie fiksacji punktu, stanowiącego centroid tej grupy i sprowadzenie wyniku do postaci $\langle x, y, t, d \rangle$, gdzie x i y stanowią współrzędne, t jest czasem pierwszej fiksacji w grupie (numerem danej w danych), a d oznacza długość fiksacji (liczba danych składających się na grupę fiksacji). Zależnie od tego czy znamy częstotliwość próbkowania z jaką gromadzone były dane czy też nie, wspomniany wcześniej próg może być równy liczbie próbek opartej na tejże częstotliwości. Jeśli jednak nie mamy wiedzy na temat częstotliwości, a prędkości katowe między próbkami mogą być wyliczone, należałoby przyjąć próg o prędkości katowej 20 stopni na sekundę. Algorytm I-VT - jak wspomniano wcześniej, jest dość prosty w implementacji, szybki i z łatwością może być używany do identyfikacji fiksacji w czasie rzeczywistym. Czasem jednak generować może przekłamane wyniki w przypadkach, gdy

wartości przemieszczeń są bliskie progowi oraz gdy dodatkowo mamy do czynienia z szumem towarzyszącym pomiarom z eyetrackerów. W takich wypadkach zamiast otrzymania fiksacji będącą grupą wielu punktów, otrzymujemy wiele grup z pojedynczymi punktami lub ich małą ilością. Problem ten najczęściej dotyka identyfikacji danych uśrednionych czasowo. Metody, którymi się posługiwano by łagodzić tego typu zakłócenia to m.in. agregacja fiksacji, będących wynikami skupienia wzroku na konkretnym celu, bądź ustalenie minimalnego progu czasowego tak dla fiksacji, jak i dla sakkad [6][7].

5.2. Algorytm I-DT

Algorytm I-DT (ang. *Dispersion Threshold Identification*) oparty na rozproszeniu (dyspersji), w porównaniu do I-VT pomija fakt, iż punkty fiksacji - ze względu na niską wartość przemieszczenia, zbierają się w grupy. I-DT sam identyfikuje fiksacje jako grupy punktów mieszczących się w danym przedziale dyspersji czy maksymalnego oddzielenia. Ponieważ fiksacje zwykle trwają minimum 100 ms, algorytmy oparte na rozproszeniu często ustawiają próg (ang. *threshold*) na 100 - 200 ms, by zniwelować zmienność odczytów dla danych z różnych sprzętów. Przemieszczenie dwóch punktów liczone jest ze wzoru na odległość między punktami układu współrzędnych. Jak widać I-DT wymaga dwóch parametrów: progu dyspersji oraz progu długości. Tak jak dla I-VT próg dyspersji może mieścić się w przedziale od 0,5 do 1 °/s jeśli odległość oka od ekranu jest znana, bądź ustawiony może być odpowiednio w stosunku do częstotliwości próbkowania danych. Prog długości ustawiony powinien być na 100 - 200 ms w zależności od wymagań. I-DT podobnie jak I-VT również może być używany do identyfikacji w czasie rzeczywistym, jednak algorytm I-DT nie produkuje błędnych wyników wynikających z nakładania się wartości przemieszczeń na wartość progu, gdyż sam w sobie już wykorzystuje próg czasowy. Minusem tego algorytmu może być jednak zależność między poziomem przemieszczenia, a poziomem czasowym, która dla nieodpowiednio dobranych wartości może skutkować niewykryciem żadnej fiksacji [8][9].

6. Forma badania

Do badania wykorzystano dwa rodzaje eyetrackerów, mobilny oraz stacjonarny. Osobami poddanymi badaniom byli dwaj zdrowi mężczyźni w wieku 25 lat bez jakichkolwiek schorzeń wzroku czy pod wpływem jakichkolwiek leków. Forma badania różniła się w zależności od sprzętu. Dla okulo grafu stacjonarnego firmy Tobii przeprowadzono badanie psychomotoryczne. Polegało ono na śledzeniu wzrokiem kolejnych liczb w kolejności od najmniejszej do największej z przedziału od 1 do 25. Dla potrzeb badania przeprowadzono dziewięć - punktową kalibrację sprzętu za pomocą wbudowanych procedur z dołączonego oprogramowania Tobii Studio. W przypadku eyetrackera mobilnego firmy SMI badanie dotyczyło koncentracji podczas działań arytmetycznych. Zadaniem badanego była ocena poprawności wyświetlanych działań arytmetycznych. Przed badaniem przeprowadzono trzy - punktową kalibrację sprzętu z wykorzystaniem oprogramowania SMI. Do celów porównawczych algorytmów I-VT oraz I-DT służyły dane wygenerowane

przez oprogramowanie okulografów z minutowych wycinków badań. Liczba próbek danych z uwagi na różną specyfikację eyetrackerów różniła się. Wynosiła odpowiednio 3600 próbek dla urządzenia firmy SMI i 18000 próbek dla eyetrackera firmy Tobii. Przed uruchomieniem autorskiego oprogramowania do identyfikacji fiksacji oraz sakkad dane poddano wstępnej obróbce, która sprowadzała się m.in. do uśrednienia pozycji obu oczu od monitora czy do nadania każdej próbce kolejnego identyfikatora. W ten sposób dane do analizy zyskały jednolity format, numer próbki oraz uśrednione współrzędne oka x, y, z.

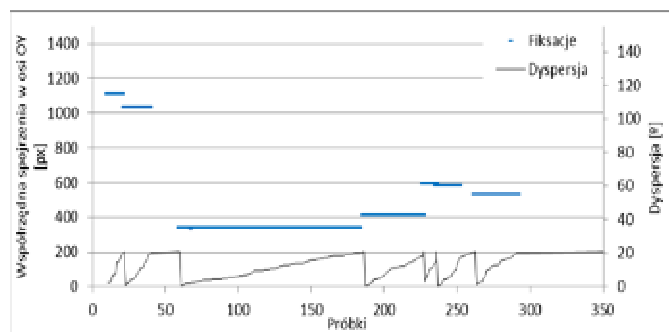
Urządzenia wykorzystane do badań

Tobii TX300, eyetracker bezkontaktowy typu Plug-and-play z dołączonym ekranem 23". Dane zbierane są z maksymalną częstotliwością 300Hz. Okulograf umożliwia ruchy głowy. System może być używany bez monitora co umożliwia rozbudowane oprogramowanie. Możliwa jest kalibracja systemu dla osiągnięcia najlepszych wyników. System może działać w oparciu o zewnętrzne dane np. EEG czy EKG. Tobii TX300 posiada dwie kamery umożliwiające obuooczne śledzenie metodą ciemnych źrenic.

SMI ETG 2.0 jest to mobilny, bezkontaktowy sprzęt do śledzenia ruchów oczu. Pod każdym szkłem zamontowana jest kamera, po jednej dla każdego oka oraz jedna kamera na środku między oczami do nagrywania tego na co aktualnie patrzy badany. Oprócz samych okularów eyetracker używa urządzenia nagrywającego, którym jest zmodyfikowany smartphone Samsung Galaxy S4 pozwalający na minimum 4 godziny nagrywania. Wraz ze sprzętem dostępne jest również oprogramowanie BeGaze firmy SMI. Do poprawienia jakości badań dostępna jest funkcja kalibracji.

7. Prezentacja wyników badań

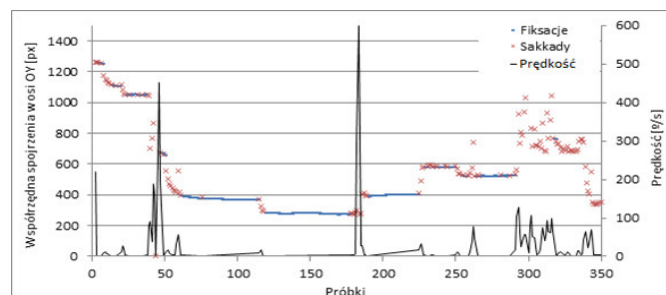
Wyniki przeprowadzonych badań zaprezentowane zostały na wykresach oraz histogramach. Wykresy przedstawiają występowanie fiksacji oraz sakkad na fragmencie liczącym 350 próbek. Histogramy prezentują częstotliwość występowania odpowiednio długich fiksacji w przedziałach czasowych.



Rys. 2. Wykres przedstawiający dane zidentyfikowane algorytmem I-DT dla danych z urządzenia SMI ETG 2.0

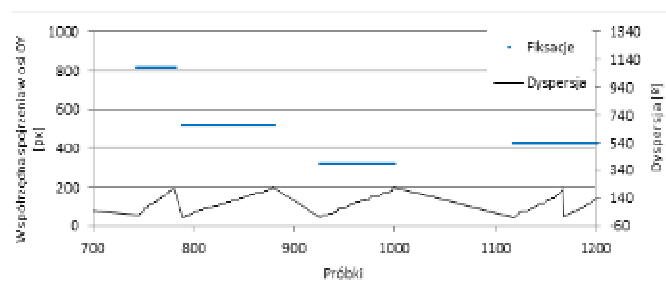
Wykres na rysunku 2 przedstawia fragment zidentyfikowanych fiksacji oraz zmieniającą się dyspersję - wyznaczoną przez aplikację z użyciem algorytmu I-DT dla urządzenia SMI. W obszarach pomiędzy fiksacjami - zaznaczonymi na niebiesko, zostały zidentyfikowane sakkady (nie oznaczone na wykresie) dla wartości dyspersji przekraczającej granicę 20° dla okna mniejszego niż 120 ms.

Jeżeli okno było dłuższe niż 120 ms, następował koniec fiksacji.

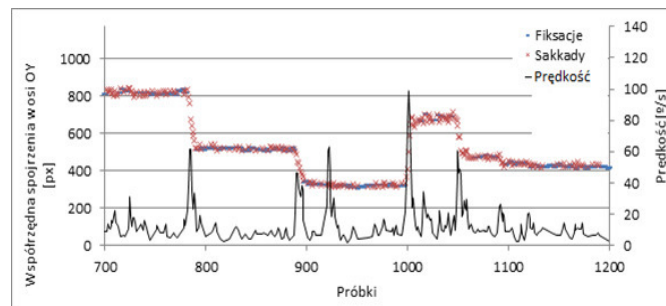


Rys. 3. Wykres przedstawiający dane zidentyfikowane algorytmem I-VT dla danych z urządzenia SMI ETG 2.0

Wykres na rysunku 3 przedstawia fragment zidentyfikowanych fiksacji, sakkad oraz zmieniającą się prędkość oka - wyznaczone przez aplikację z użyciem algorytmu I-VT. Algorytm wykrył znacznie więcej krótkich fiksacji i sakkad występujących pomiędzy nimi.

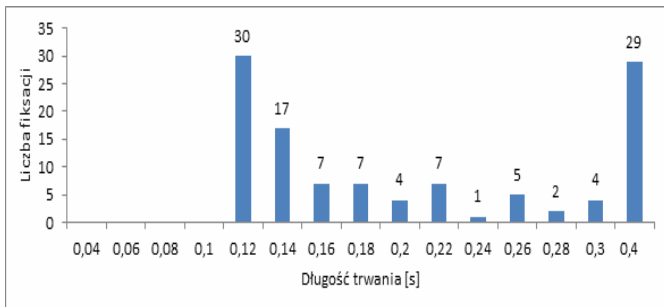


Rys. 4. Wykres przedstawiający dane zidentyfikowane algorytmem I-DT dla danych z urządzenia Tobii

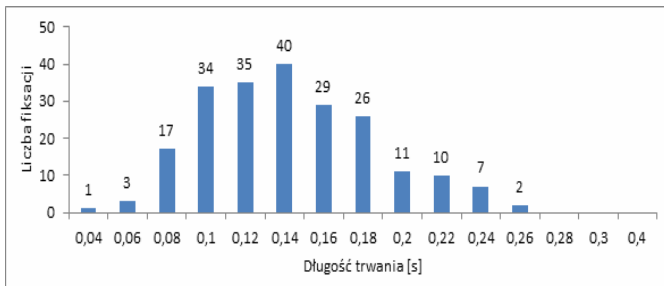


Rys. 5. Wykres przedstawiający dane zidentyfikowane algorytmem I-VT dla danych z urządzenia Tobii

Wykresy na rysunkach 4 oraz 5 przedstawiają wyniki działania algorytmów I-DT oraz I-VT dla danych z urządzenia firmy Tobii. Algorytm I-DT wykrył fiksacje, których łączny czas trwania wynosi ok. 32 sekund. Jest to o 3 sekundy więcej niż w przypadku użycia algorytmu I-VT gdzie czas ten wyniósł 29 sekund. Różnica ta z istoty działania algorytmów (identyfikacja fiksacji w oknach dla I-DT). Z całą pewnością można zauważyć podobieństwa między oboma wykresami jednakże, nie każdy fragment wykresu z I-DT odpowiada fragmentowi z wykresu I-VT. Przykładem jest fragment na odcinku od 1000-nej do 1050-tej próbki, gdzie algorytm I-DT nie zakwalifikował jako fiksacji żadnej próbki (okna). Jest to spowodowane nadmierną dyspersją - duże różnice w pozycjach kolejnych próbek.



Rys. 6. Histogram długości czasu trwania fiksacji dla identyfikacji algorytmem I-DT, dane z urządzenia Tobii



Rys. 7. Histogram długości czasu trwania fiksacji dla identyfikacji algorytmem I-VT, dane z urządzenia Tobii

Histogramy z rysunków 6 oraz 7 przedstawiają różnice w wynikach działania algorytmów I-DT i I-VT dla danych z urządzenia Tobii. Dla algorytmu I-VT większość fiksacji trwała od 0,08 do 0,26 sekundy. W przypadku I-DT czasy fiksacji zawierają się na przedziale od 0,12 do 0,3 sekundy (29 fiksacji miało czas większy niż 0,3 sekundy). Można zauważyć, iż zgodnie z działaniem algorytmu I-DT wykrył on więcej dłuższych fiksacji, a mniej krótszych w porównaniu do I-VT. Czasy trwania fiksacji dla obu algorytmów odpowiadają przyjętym zakresom dla czasu trwania fiksacji tj. 100 – 300 ms.

8. Wnioski

Wyniki badań przedstawione w artykule potwierdzają tezę, iż wyniki działania badanych algorytmów są zbliżone. Algorytmy I-DT oraz I-VT pomimo różnic w implementacji wyznaczyły fiksacje oraz sakkady w bardzo zbliżonych punktach. Podsumowując porównanie algorytmów stwierdzić można, iż każdy z nich ma swoje wady i zalety. Wybór odpowiedniego algorytmu należałoby dostosować do typu posiadanych danych (np. częstotliwość próbkowania) oraz oczekiwanej formy danych wynikowych. Gdy zależy nam na uśrednionych wynikach badania dobrym wyborem będzie algorytm I-DT grupujący dane punkty fiksacji w pojedyncze fiksacje, jeśli jednak zależy nam na zakwalifikowaniu każdej pojedynczej próbki jako sakkady czy fiksacji wybór powinien paść na algorytm I-VT. Wybór algorytmu może być również podyktowany złożonością implementacji czy prędkością jego wykonywania. Pod względem implementacji oba algorytmy znacznie się różnią co odbija się na prędkości ich. Pod uwagę należy również wziąć dobór odpowiednich parametrów, w przeciwnym wypadku zaburzone mogą zostać wyniki analizy.

Literatura

- [1] Krauzlis R.J. Eye movements. W: Squire L.R. (red.), „Fundamental neuroscience”, Wyd. 3. Elsevier, Amsterdam 2008.
- [2] K. Jadanowski, Sławomir Budrewicz, Ewa Koziorowska - Gawron, „Zaburzenia gałkoruchowe w chorobach ośrodkowego układu nerwowego”, Polski Przegląd Neurologiczny 2010.
- [3] User Manual — Tobii Studio Version 3.3.0 12/2014 Tobii Technology AB
- [4] Peter R Jones, „A computational reassessment of noise in eyetracking data: Modelling phenomena and ephemeral”.
- [5] Salvucci D.D., Goldberg J.H., „Identifying fixations and saccades in eye-tracking protocols. Proceedings of the 2000 symposium on Eye tracking research & applications”, ACM 2000.
- [6] Susan M.Munn Leanne Stefano Jeff B.Pelz, „Fixation - identification in dynamic scenes: comparing an automated algorithm to manual coding”.
- [7] Larsson L., Nystrom M., Stridh M., „Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit”, 2013.
- [8] Lopez, J. S. A., „Off-the-shelf Gaze Interaction”, 2009.
- [9] Van Rensbergen J., & De Troy A., „A reference guide for the Leuven dual-PC controlled Purkinje eyetracking system”, Leuven, Belgium, 2015.
- [10] <http://eyetracking.pl/> [11.2016]

Analiza możliwości wykorzystania platformy Moodle do realizacji procesu kształcenia na kierunku Inżynieria systemów bezpieczeństwa wewnętrznego w Politechnice Lubelskiej

Sławomir Bronisz*, Elżbieta Miłoś

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule przedstawiono wyniki badań dotyczących analizy możliwości wykorzystania platformy Moodle do realizacji procesu kształcenia na kierunku Inżynieria systemów bezpieczeństwa wewnętrznego w Politechnice Lubelskiej. Badania obejmowały analizę możliwości platformy Moodle oraz symulację komputerową realizacji przedmiotu Techniki Informacyjne.

Słowa kluczowe: Moodle; e-learning; nauczanie zdalne

*Autor do korespondencji.

Adres e-mail: slawek.bronisz@gmail.com

The analysis of usage possibilities of the Moodle platform to execute the training process in an "Engineering of internal security systems" field of study at Lublin University of Technology

Sławomir Bronisz*, Elżbieta Miłoś^a

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents the results of research on the analysis of the possibilities of using the Moodle platform for the implementation of the education process at the engineering systems of internal security in the Lublin University of Technology. The research involved an analysis of the possibility of Moodle and computer simulation of the subject of Information Technologies.

Keywords: Moodle; e-learning

*Corresponding author.

E-mail address: slawek.bronisz@gmail.com

1. Wstęp

Celem badań przedstawionych w artykule była weryfikacja, w jakim stopniu platforma Moodle nadaje się do realizacji procesu kształcenia na kierunku Inżynieria systemów bezpieczeństwa wewnętrznego (ISBW) w Politechnice Lubelskiej.

Tok badań obejmował następujące etapy:

- 1) Analiza przebiegu realizacji procesu dydaktycznego w sposób zdalny od strony wykładowcy, studenta, administratora.
- 2) Dopasowanie możliwości Moodle do tych potrzeb.
- 3) Weryfikacja tych możliwości na przykładzie wybranego przedmiotu - symulacja komputerowa działań wykładowcy i studentów.
- 4) Ocena badań – weryfikacja hipotezy badawczej.

Głównymi problemami badań są:

- 1) Czy za pomocą platformy wykładowca może weryfikować stopień opanowania materiału wykładowego przez studenta? Weryfikować, czy zapoznał się on z materiałem? Ustanawiać reguły,

w jakich ramach czasowych student może uzyskać dostęp do konkretnego zasobu?

- 2) Czy za pomocą platformy student może otrzymywać informacje o aktualnościach i wydarzeniach związanych z danym przedmiotem?
- 3) Czy za pomocą platformy możliwy jest obustronny kontakt wykładowcy i studenta?

Poszukiwanie odpowiedzi na postawione pytania pozwoli zweryfikować następującą hipotezę badawczą:

"Możliwości platformy Moodle w pełni zaspokajają wymagania realizacji procesu dydaktycznego na kierunku ISBW w Politechnice Lubelskiej".

2. Badania literaturowe

Zdalne nauczanie jest niewątpliwie jedną z najszybciej rozwijających się dziedzin powiązanych z edukacją. Współczesna technologia a szczególnie Internet, umożliwia pokonanie bariery odległości w kontakcie z edukacją, oraz naukę w dowolnym czasie i indywidualnym tempie [1].

Początki stosowania metod kształcenia na odległość szacuje się na XVIII wiek w Stanach Zjednoczonych. Wówczas odbywało się to w sposób korespondencyjny.

Za początek uznaje się rok 1728, kiedy to Caleb Phillipps zamieścił ogłoszenie w „The Boston Gazette” dotyczące korespondencyjnego kursu nauki szybkiego pisanie. Dalszy rozwój przyspieszony został po pojawieniu się nowych rozwiązań technologicznych w postaci radia i telewizji. Rozwój tej formy kształcenia szczególnie szybko ewoluował w krajach dużych powierzchniowo takich jak Australia, gdzie w latach 30 dozwolona była edukacja szkolna przy użyciu radia [2].

Kolejny rozwój technologiczny związany był z pojawieniem się komputerów. Zapoczątkowało to interakcję ucznia z nauczycielem. Internet sprawił, że ta forma kształcenia stała się bardziej powszechna [3].

Początki kształcenia na odległość w Polsce datuje się na 1776 rok. Edukacja ta odbywała się w formie rzemieślniczych kursów korespondencyjnych prowadzonych, przez Uniwersytet Krakowski [4].

Nauczanie na odległość z założenia jest sposobem nauczania wolnym od ograniczeń związanych z czasem i miejscem uczenia. Umożliwia to indywidualizację szybkości uczenia się oraz realizowanie nauki w przyjaznych okolicznościach środowiskowych [5, 6]. Taka forma kształcenia wprowadza w miejsce bezpośredniego kontaktu nauczyciela i ucznia kontakt pośredni [7].

Politechnika Lubelska wprowadza zdalne nauczanie na Wydziale Elektrotechniki i Informatyki na nowym kierunku inżynierskim na studiach I stopnia o profilu ogólnoakademickim - ISBW. Kierunek ISBW realizowany jest w innowacyjnej formie e-learningowej z wykorzystaniem metod i technik kształcenia na odległość, zgodnie z warunkami rozporządzenia Ministra Nauki i Szkolnictwa Wyższego z dnia 25.09.2007 (ze zm.) *w sprawie warunków, jakie muszą być spełnione, aby zajęcia dydaktyczne na studiach mogły być prowadzone z wykorzystaniem metod i technik kształcenia na odległość* (Dz. U. 2011.246.1470) [8].

Zajęcia prowadzone są w trybie niestacjonarnym. Wykorzystanie metod i technik kształcenia na odległość zakłada rezygnację z tradycyjnych wykładów przeprowadzanych w siedzibie uczelni oraz wymaga od studentów aktywnej pracy z systemem e-learningowym. Wszystkie wykłady, które stanowią 60% zajęć, prowadzone są w formie e-learningowej. Umieszczone są one na platformie Moodle i uzupełnione testami/quizami dla studentów. Wszystkie zajęcia praktyczne, czyli ćwiczenia i laboratoria, które stanowią 40% zajęć, a także część konsultacji, wszystkie zaliczenia i egzaminy, realizowane są w siedzibie uczelni podczas 4 zjazdów na każdy semestr. Zajęcia zaplanowane są na sobotę i niedzielę. Konsultacje w siedzibie uczelni odbywają się w piątki, ponadto istnieje możliwość konsultacji w formie elektronicznej z użyciem wbudowanego w platformę Moodle komunikatora lub poczty e-mail [9].

W programie studiów znajdują się przedmioty obowiązkowe i obieralne, które stanowią 30% punktów ECTS. Student wybiera jeden z dwóch przedmiotów

będących w modułach obieralnych. Platforma powinna umożliwiać ten wybór jeden semestr wcześniej. Program obejmuje przedmioty zawierające tylko wykłady, a także takie, które składają się z wykładów i ćwiczeń. Do każdego przedmiotu prowadzone są konsultacje w siedzibie uczelni i na platformie [10].

3. Charakterystyka platformy Moodle

Moodle jest liderem wśród platform e-learningowych. Tak wysoką pozycję zawdzięcza szerokim możliwościom, ciągłemu rozwojowi oraz darmowej formie licencjonowania. Nazwę platformy można rozwinąć w języku angielskim, jako *Modular Object-Oriented Dynamic Learning Environment*. Znakomicie opisuje ona istotę tego środowiska. Modułowa architektura umożliwia elastyczne dostosowywanie platformy dołączając dodatkowe funkcjonalności. Od początku wykorzystuje język PHP, ale nie jest to jedyna ciekawa technologia. Moodle korzysta także z MySQL, jQuery, Ajax i wielu innych rozwiązań. Otwarty kod platformy pozwala na wprowadzanie modyfikacji zarówno w wyglądzie graficznym jak również w zakresie funkcjonalnym.

Platforma Moodle pozwala na pracę z pojedynczymi studentami, jak również z wieloma tysiącami osób. Wdrożenia obejmują m.in. szkoły, począwszy od podstawowych po szkolnictwo wyższe, ale także organizacje wojskowe oraz rządowe i duże przedsiębiorstwa. Moodle jest wykorzystywana w sposób mieszany, łącząc edukację tradycyjną z nauczaniem na odległość, ale również do pracy w trybie całkowicie zdalnym.

Moodle jest bardzo rozbudowaną platformą. Większość opcji jest konfigurowalna przez administratora lub inne osoby z podwyższonymi uprawnieniami. Do dyspozycji jest kilka ról dla użytkowników oraz możliwe jest też tworzenie nowych. Domyślnie nowy użytkownik otrzymuje rolę Student, która pozwala na branie udziału w kursach. Role z wyższymi uprawnieniami to m.in. Prowadzący, który może prowadzić kurs, ale nie może go edytować; Autor kursu, który może stworzyć i edytować kursy; Menedżer, który wspomaga administratora w zarządzaniu kursami i użytkownikami oraz Administrator, który może wykonywać dowolne czynności. Platforma zawiera wiele funkcjonalności, które umożliwiają udostępnianie treści wykładowych, prowadzenie różnego rodzaju testów sprawdzających, przechowywanie plików oraz kontakt z prowadzącym i innymi studentami.

4. Przebieg badań

Badania obejmowały instalację platformy Moodle na serwerze z systemem operacyjnym Debian 8.4, analizę możliwości przygotowania materiałów dydaktycznych różnych typów, możliwości sterowania przebiegiem procesu nauczania oraz komunikacji dwustronnej między wykładowcą a studentem. W przeprowadzonych badaniach wykonana została symulacja komputerowa działań wykładowcy i studentów na przykładzie przedmiotu Techniki Informacyjne. Przedmiot składa się z 15 godzin wykładów i 15 godzin laboratoriów. Wykonana została realizacja kursu w zakresie od tworzenia kategorii i samego kursu, po

wypełnianie treści, tworzenie aktywności oraz ocenianie studentów. Realizowany kurs pokazany na rysunku 1 dotyczył edytorów tekstu i zawierał:

- wykład w formie aktywności, w której student po każdej stronie musi odpowiedzieć na pytanie dotyczące omawianego materiału,
- wykład w formie pliku PDF - student ma dostęp do tego pliku dopiero po zaliczeniu aktywności wykładu z pytaniami,
- plik multimedialny zawierający film szkoleniowy pokazujący jak krok po kroku wykonać jakieś ćwiczenie,
- zadanie zaliczeniowe - student ma za zadanie pobrać pliki ćwiczenia i wykonać na nich wymienione w poleceniach czynności, po wykonaniu musi umieścić pliki na platformie w celu sprawdzenia przez prowadzącego,
- zadanie zaliczeniowe składające się z 9 pytań - w celu sprawdzenia szerszego zakresu możliwości Moodle, każde z pytań oparte zostało na innej aktywności. Zawiera m.in. klasyczny test, pytanie otwarte oraz pytanie polegające na graficznym wskazywaniu poprawnych odpowiedzi.

Edytor tekstu



Rys. 1. Widok przygotowanej części kursu

Symulacja komputerowa dotyczyła również procesu nauczania - działań studentów w postaci realizacji zamieszczonych przez wykładowcę aktywności, jak również działań wykładowcy, oceniającego stopień opanowania materiału przez studentów. Informację o wynikach zaliczenia aktywności przez jednego ze studentów przedstawiono na rys. 2.

Twoje oceny - Student Trzeci

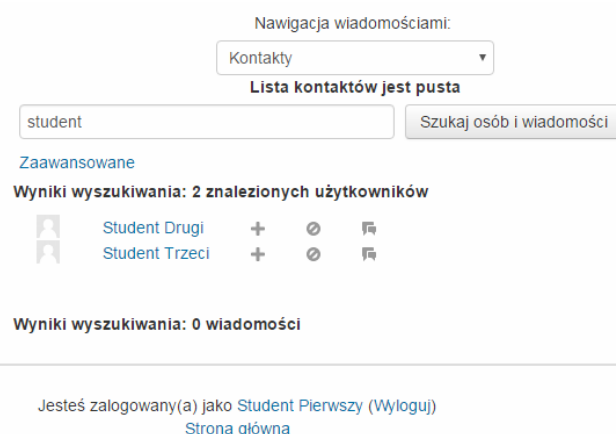
Oceniany element	Waga	Ocena	Zakres	Procentowo	Informacja zwrotna
Technologie informacyjne					
Wykład 2 - Edytor tekstu	48,78 %	5	0-100	100,0 %	
Lab 2. Zadanie	48,78 %	4	0-100	85,0 %	Rozwiązanie poprawne z niewielkimi błędami w punktach. 5 i 6.
Zaliczenie części drugiej	2,44 %	4,5	0-5	94,1 %	
W całym kursie	-	4,5	0-205	92,5 %	

Rys. 2. Widok arkusza ocen dla studenta

Za pomocą platformy symulowano również możliwości dwustronnej komunikacji pomiędzy użytkownikami platformy: wykładowcą i studentami z wykorzystaniem narzędzi takich jak: forum aktualności, kalendarz, nadchodzące wydarzenia, czat i e-mail. Przykładowe możliwości komunikacji w procesie kształcenia przedstawiono na rys. 3, 4.



Rys. 3. Komunikacja wykładowca - student



Rys. 4. Przesyłanie wiadomości pomiędzy użytkownikami platformy

5. Wyniki badań

Na podstawie przeprowadzonej analizy możliwości platformy Moodle pokazano, w jakim stopniu może być ona wykorzystana do realizacji procesu kształcenia:

- wykładowca może udostępniać materiał wykładowy studentom za pomocą zasobów typu plik i strona oraz aktywności typu lekcja
- osoba prowadząca może sprawdzić stopień opanowania materiału wykładowego przez studenta za pomocą aktywności typu zadanie oraz quiz,
- wykładowca ma możliwość weryfikacji czy student zapoznał się z materiałem wykładowym przy pomocy aktywności typu lekcja,
- autor kursu do każdego zasobu lub aktywności może nałożyć ograniczenie, w jakich ramach czasowych będą one dostępne dla studentów,
- student może otrzymywać informacje o aktualnościach i wydarzeniach związanych z przedmiotem za pomocą

forum aktualności oraz kalendarza i bloku nadchodzące wydarzenia,

- możliwy jest obustronny kontakt wykładowcy i studenta za pomocą wbudowanego w platformę systemu wiadomości oraz funkcjonalności typu czat.

Przeprowadzona symulacja komputerowa realizacji kursu (wybranych zajęć z przedmiotu Technologie Informacyjne) pokazała możliwe działania wykładowcy i studentów podczas realizacji procesu kształcenia.

Proces pozyskania przez studenta efektów kształcenia w zakresie wiedzy, umiejętności i kompetencji społecznych może być w dużym stopniu wspomagany aktywnościami na platformie Moodle.

Przeprowadzona analiza i symulacja prowadzenia wybranego przedmiotu pokazały, że możliwości platformy Moodle w pełni zaspokajają wymagania realizacji procesu dydaktycznego na kierunku ISBW w Politechnice Lubelskiej. Tym samym postawiona hipoteza badawcza okazała się słuszna.

6. Wnioski

Przeprowadzone badania wykazały, że dużą zaletą e-learningu jest wysoka elastyczność, która pozwala na wybór czasu i miejsca nauki. Zalety te są szczególnie ważne dla studentów, którzy pracują zawodowo. Teoretyczne przygotowanie do zajęć praktycznych, pozyskanie wiedzy z danego przedmiotu może być realizowane w formie pracy samodzielnej studenta, na podstawie dobrze przygotowanych przez wykładowcę różnorodnych aktywności na platformie Moodle. Studenci mają wolność wyboru czasu nauki, ale muszą dotrzymywać terminów oddawania zadań i zaliczania aktywności na platformie. Zajęcia praktyczne realizują tradycyjnie w formie zjazdów w siedzibie uczelni, podczas których mogą spotkać się z wykładowcami i kolegami z roku. Platforma Moodle jest cennym narzędziem dydaktycznym i z pewnością takie narzędzia będą rozwijane w przyszłości w coraz większym stopniu.

Literatura

- [1] Galwas B.: Otwarte uniwersytety, otwarte zasoby edukacyjne, otwarty dostęp do wiedzy, [W:] Postępy e-edukacji, praca zbiorowa pod red. Zespołu Ośrodka Kształcenia na Odległość OKNO PW, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2008.
- [2] Szablowski S.: E-learning dla nauczycieli, WO FOSZE, Rzeszów 2009.
- [3] Brzóska P.: Moodle dla nauczycieli i trenerów, Helion, Gliwice 2016.
- [4] Historia e-learningu na świecie i w Polsce, dostęp: <http://www.heuristic.pl/blog/e-learning/Historia-e-learningu-na-swiecie-i-w-Polsce;161.html> [27.12.2016].
- [5] Jukes I., McCain T., Crocket L.: Understanding the Digital Generation: Teaching and Learning in the New Digital Landscape, CreateSpace Independent Publishing Platform, Kanada 2010.
- [6] D'Antoni S.: The Virtual University. Models & Messages. Lessons from case studies, UNESCO, Paryż 2006.
- [7] Goban-Klas T.: Uczenie na odległość: formy i problemy. [W:] Okoń-Horodyńska E.: Nauczanie na odległość – nowa szansa dla edukacji, Śląskie Wydawnictwa Naukowe, Tychy 1999.
- [8] Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 25.09.2007 r. w sprawie warunków, jakie muszą być spełnione, aby zajęcia dydaktyczne na studiach mogły być prowadzone z wykorzystaniem metod i technik kształcenia na odległość, Dz.U. 2007 Nr 188 poz. 1347
- [9] Miłosz E., Juszczak M.: Kierunek „Inżynieria systemów bezpieczeństwa wewnętrznego” jako odpowiedź Politechniki Lubelskiej na wyzwania i potrzeby bezpieczeństwa narodowego, [W:] Wybrane aspekty bezpieczeństwa narodowego; [Red:] Skrobotowicz Grzegorz, Maciąg Kamil, Lublin Wydawnictwo Naukowe TYGIEL sp. z o. o., 2016, s. 50-61.
- [10] Kierunki studiów, Program studiów na kierunku: Inżynieria systemów bezpieczeństwa wewnętrznego, dostęp: <http://weii.pollub.pl/pl/kandydaci/kierunki-studiow> [03.01.2017].

Ochrona danych osobowych w ujęciu przepisów państwa Polskiego a organu Generalnego Inspektora Ochrony Danych Osobowych

Damian Harasim*

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł opisuje oraz rozwija zagadnienie danych osobowych oraz ich współczesnej ochrony. Przedstawia organ Generalnego Inspektora Ochrony Danych Osobowych, pracy Biura Inspektora oraz prezentuje przykładowe wyniki ich pracy. Dla artykułu zostało przeprowadzone badanie ankietowe polegające na zbadaniu małej grupy osób w celu potwierdzenia słuszności działalności instytucji.

Słowa kluczowe: GODO; dane osobowe; ochrona

*Autor do korespondencji.

Adres e-mail: damian.harasim@pollub.edu.pl

Personal data protection in terms of provisions of the Polish state and the authority of the General for Personal Data Protection

Damian Harasim*

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article describing and expand issue personal data and their contemporary protection. Present apparatus the Inspector General for Personal Data Protection and show example results of their work. For the article was conducted survey consisting in examining the small Group of people in order to confirm the rightness activity of the institutions.

Keywords: GODO; personal data, protecting

*Corresponding author.

E-mail address: damian.harasim@pollub.edu.pl

1. Wstęp

W ostatnich latach mogliśmy zaobserwować wzmożone wysiłki pojedynczych osób lub całych grup hackerskich wykradających głównie bardzo cenne informacje, na które w głównej mierze składały się chronione prawnie dane osobowe. Niejednokrotnie pojedyncze wycieki zawierały nie tylko dane teleadresowe, ale i również numery kont bankowych czy kart kredytowych. W 2016 roku, słynny serwis Yahoo oficjalnie poinformował, że w 2014 jego serwery padły ofiarą ataku a do sieci wyciekły informacje o około 500 milionach kont użytkowników zawierających takie dane jak: loginy, adresy e-mail, numery telefonów, daty urodzin czy kryptograficznie zabezpieczone hasła [1]. Są to oczywiście zjawiska bardzo niebezpieczne dla osób, których dane zostały wykradzione, gdyż w skrajnych przypadkach mogą prowadzić do defraudacji pieniędzy z kont bankowych.

2. Generalny Inspektorat Ochrony Danych Osobowych.

Polska posiada w swoich aparatach państwowych organ, jakim jest Generalny Inspektor Ochrony Danych Osobowych, w skrócie GODO. W jego skład wchodzi pięć departamentów:

- 1) Departament Rejestracji
- 2) Departament Prawny
- 3) Departament Inspekcji
- 4) Departament Informatyki
- 5) Departament Skarg

Samo GODO, najprościej ujmując, jest to jednoosobowe stanowisko, a osoby na nie są powoływane na 4 letnią kadencję przez Sejm Rzeczypospolitej Polskiej. Dotychczas urząd ten sprawowało sześciu Inspektorów, obecnie jest to Pani Edyta Bielak-Jamaa, wybrana 22 kwietnia 2015 roku. Inspektor realizuje swoją pracę poprzez Biuro Generalnego Inspektora Danych Osobowych a obowiązki Inspektora znajdują się w art. 12 ustawy o ochronie danych osobowych z 1997 roku [2]. Poza tym, Generalny Inspektor zajmuje się prowadzeniem ogólnokrajowego, jawnego rejestru zbioru danych osobowych zgłoszonych do rejestracji, art. 42 ust. 1 powyższej ustawy.

2.1. Dane osobowe

Za dane osobowe w rozumieniu litery prawa, uważa się każdą informację jaka może umożliwić identyfikację lub prowadzić do umożliwienia identyfikacji osoby fizycznej, przy czym należy zaznaczyć, iż nie istnieje prawnie określony zbiór, jakie dane mogą być danymi osobowymi. I nie chodzi tutaj o jednolitą informację, na przykład konkretnie identyfikujący wiersz w bazie danych a również o relacje pomiędzy tabelami w bazie, których złożenie prowadzi do identyfikacji podmiotu. Samo przetwarzanie takich danych również jest szeroko pojętym procesem. Na szczęście jego definicja została zawarta w art. 7 ust. 2 ustawy o ochronie danych osobowych i kryją się pod tym jakiejkolwiek operacje wykonywane na danych osobowych.

Biuro Generalnego Inspektora opublikowało na swojej stronie internetowej [3], szereg „odpowiedzi na pytania” gdzie znajdują się wytyczne mogące pomóc w zrozumieniu tego problemu. I tak, możemy się dowiedzieć iż administrator danych do przetwarzania danych osobowych zgodnie z przepisami „musi spełniać przynajmniej jeden z warunków decydujący o tym, że takie działanie jest legalne”, dopełnił obowiązku zgłoszenia zbioru danych do rejestracji GIODO (poza wyjątkami z art. 43 ust.1 pkt.1-11 ustawy o ochronie danych osobowych), stosował środki techniczne oraz organizacyjne zapewniające przetwarzanym danym odpowiedni poziom bezpieczeństwa, dopełnił obowiązku informacyjnego, dokonał szczególnej staranności w celu ochrony interesów osób, których dane dotyczą oraz respektował prawa osób, których dane dotyczą [4]. Powyższy opis stanowiska, będący wyjaśnieniem jednego z artykułów ustawy, jest jednym z wielu jakie może znaleźć w interpretacjach Głównego Inspektora.

Wracając do danych osobowych, warto zwrócić uwagę na ich interpretację przez GIODO. Zostało wcześniej powiedziane, iż nie istnieją żadne ustawowo spisane zbiory jakie dane są danymi osobowymi a więc ich interpretacja może być różna, w zależności od poziomu jej zrozumienia.

GIODO na swojej stronie internetowej umieścił zbiór kilku przykładowych danych osobowych, jakie On uznaje (a niektóre jedynie w pewnych sytuacjach) za dane osobowe. I tak możemy za dane osobowe wg. GIODO uznać:

- 1) Numer IP
- 2) Numer PESEL
- 3) Numer telefonu
- 4) Adres e-mail
- 5) Numer rachunku bankowego

Na tle powyższego zestawienia ciekawie prezentuje się numer VIN pojazdu [5]. Art.6 ustawy o ochronie danych osobowych stwierdza, iż za informację osobową nie uważa się danych, dzięki którym określenie osoby fizycznej wymaga nadmiernych kosztów, czasu lub działań. I tak, dla przeciętnego człowieka taki numer nie stanowi informacji na podstawie której można zidentyfikować konkretną osobę. Jednakże już dla administratora danych jakim będzie w tym wypadku minister właściwy do spraw administracji publicznej, numer VIN stanowi jedną z wielu powiązanych ze sobą informacji jakie posiada wraz z tym numerem. I to właśnie dla niego, taka składowa będzie uznawana za daną osobową, gdyż nie wymaga to od niego wyłożenia nadmiernych kosztów na jej uzyskanie.

2.2. Biometria

W kontekście danych osobowych warto wspomnieć o coraz popularniejszej metodzie autoryzacji – biometrii. Chociaż jest to dość nowa metodyka dążąca do zbierania fizycznych danych, również podlega ustawie o ochronie danych osobowych ponieważ w myśl art. 6 ust. 2, zbiera ludzkie cechy fizyczne.

Dotychczas sposób dostępu do komputera, telefonu czy konta bankowego polegał na ustaleniu pary login + hasło, pojedynczego numeru PIN, profilu zaufanego bądź podaniu zestawu danych w czasie wizyty w oddziale banku lub infolinii. Biometria pozwoliła na znaczne uproszczenie tego

procesu, wystarczy nam odcisk kciuka przyłożonego do czytnika, jak np. w telefonach marki Apple czy Samsung zawierających dedykowane czytniki linii papilarnych. Jednak taki poziom wygody może nie koniecznie iść z bezpieczeństwem danych. Jak bardzo może być to niebezpieczne zjawisko, dowiedzieliśmy się w 2014 roku w czasie konferencji zatytułowanej „Chaos Communication Congress”. W jej czasie, Jan Krissler jedynie na podstawie zwykłych zdjęć dokonał rekonstrukcji odcisku palca niemieckiej minister obrony Ursuli Gertrud von den Leyen [8]. Jest to jeden z wielu obecnie przykładów słabości zabezpieczenia dostępu poprzez odcisk palca. A przecież dotykając przeróżnych codziennych przedmiotów, odciski pozostają wszędzie. Na szczęście biometria nie stoi w miejscu i ewoluuje. Do dzisiaj doczekaliśmy się np. biometrii „Finger Vein” wykorzystującej unikalny dla każdej osoby, układ naczyń krwionośnych znajdujących się wewnątrz jej palca. Rozwiązanie zostało zastrzeżone patentem US7526111 B2. Jest to nie zależne od wieku ułożenie naczyń na których wpływ może mieć jedynie choroba. Z takiego rozwiązania coraz częściej korzystają instytucje bankowe, a przykładem może być bank BPH który od stycznia 2013 roku korzysta z tej technologii do uwierzytelniania klientów banku.

GIODO w swoim raporcie [7] cytuje bardzo ciekawą opinie Grupy Roboczej ds. ochrony osób fizycznych w zakresie przetwarzania danych osobowych. Twierdzi ona, iż wraz z rozwojem biotechnologii rodzi się niepokój związany ze zmniejszeniem wrażliwości ludzi, będącego wynikiem ekspansji w życiu codziennym tego rozwiązania. I trudno się tej opinii nie dziwić. Rosnąca akceptacja społeczna jest motorem napędowym większości innowacji na rynku. Jednakże filarem danych biometrycznych jest ich niezmiennosc, co jest przeciwieństwem zabezpieczeń znanym nam dotychczas. Polityki bezpieczeństwa korporacji kładą dość wysoki nacisk na częstą zmianę haseł oraz ich różnorodność (na przykład blokada ponownego użycia haseł z ostatnich 12 miesięcy).

3. Badanie

Badanie przeprowadzone w ramach tej pracy, polegało na udowodnieniu następującej hipotezy: „Przeciętna osoba jest w stanie rozpoznać poprawnie prawidłowe dane osobowe opierając się wyłącznie na treści artykułu ustawy.” Zostało ono przeprowadzone dnia 16 grudnia oraz 4 stycznia na terenie kampusu Politechniki Lubelskiej oraz ankiety internetowej, pośród studentów wydziału Elektrotechniki i Informatyki. Łącznie w badaniu brało udział 16 przypadkowo wybranych osób. Pytania zostały oparte o wcześniej zaopiniowane dane osobowe oraz ogólne przez Generalnego Inspektora. Miało to na celu możliwe zanegowanie postawionej wcześniej hipotezy.

Ankieta była całkowicie anonimowa oraz zawierała 8 pytań, na których odpowiedzi należało oznaczyć obrysowując obwódką bądź przekreślając poprawną odpowiedź. Ankieta zawierała krótki opis wstępu wraz z instrukcją wypełnienia, przytoczony art. 6 ustawy o ochronie danych osobowych (Dz. U. z 2002 r. Nr 101, poz. 926 ze zm.) oraz następujące pytania:

- 1) Płeć
- 2) Czy pracowałeś/łaś lub nadal pracujesz?

- 3) Czy miałeś styczność z pojęciem „danych osobowych”?
Czy Twoim zdaniem za dane osobowe:
4) Uznaje się numer telefonu?
5) Uznaje się numer rachunku bankowego?
6) Nie uznaje się numeru IP?
7) Nie uznaje się numeru karty miejskiej?
8) Uznaje się numer VIN pojazdu?

Uczestnicy do wyboru mieli następujące odpowiedzi. Dla pytania pierwszego: „Mężczyzna” lub „Kobieta”, dla pytania drugiego oraz trzeciego: „Tak” lub „Nie” oraz na pytania od czwartego do ósmego: „Tak”, „Nie”, „Zależy”, gdzie pod odpowiedzią „Zależy” zgodnie z opisem zawartym w ankiecie, należało rozumieć dane, jakie przytacza art. 6 ust. 3. Pytanie 5 oraz 6 zostały celowo sformułowane w kontekście negacji by lepiej sprawdzić zrozumienie ustawy.

3.1. Wyniki badania

Legenda: Pytanie 1: M – „Mężczyzna”, K – „Kobieta”; dla pytań 2-8: T – „Tak”, N – „Nie”, Z – „Zależy”

Tabela 1. Wyniki ankiety

	Pytanie	Pytanie	Pytanie	Pytanie	Pytanie	Pytanie	Pytanie	Pytanie
Osoba 1	M	T	T	T	T	N	T	N
Osoba 2	K	T	T	Z	Z	Z	Z	N
Osoba 3	M	T	T	T	T	N	N	N
Osoba 4	M	T	T	T	T	T	T	N
Osoba 5	M	N	N	Z	Z	T	N	N
Osoba 6	M	N	T	T	Z	T	T	N
Osoba 7	M	N	N	T	T	T	T	N
Osoba 8	M	N	T	T	T	T	T	N
Osoba 9	M	N	T	T	T	T	T	N
Osoba 10	M	T	T	T	T	T	T	N
Osoba 11	M	T	T	T	T	N	N	T
Osoba 12	M	N	T	T	T	N	T	N
Osoba 13	M	N	T	T	T	N	T	N
Osoba 14	M	N	T	T	T	N	T	N
Osoba 15	M	N	T	T	T	T	T	N
Osoba 16	M	N	T	T	Z	Z	T	N

Otrzymane wyniki nie dają nam konkretnej odpowiedzi na postawioną wcześniej hipotezę, ale mimo wszystko rezultaty są ciekawe. 37,5% badanych aktualnie pracuje lub wcześniej pracowało i każdy z nich spotkał się do tej pory z pojęciem danych osobowych. Jednakże już tylko 80% uczestników znajdujących się w grupie niepracujących, miało styczność z tym pojęciem. Jest to zaskakujący rezultat, nawet pomimo tak małej grupy ankietowej. Badania były przeprowadzane na wydziale uczelni technicznej, na którym nie powinno być ono dla nikogo obce. Można wysnuć hipotezę, iż konkretnych przepisów czy ich interpretacji, młodzi ludzie uczą się dopiero w czasie swoich początków kariery zawodowej, pomimo, iż

są studentami i powinni na własną rękę zdobywać wiedzę z takich zagadnień.

Daje nam to obraz mniej więcej następujący obraz sytuacji. Wiedza społeczeństwa wciąż rośnie, zdecydowanie można stwierdzić, iż każdy współczesny człowiek jest bardziej doinformowany i świadomy otaczającego go świata od swoich rodziców. Dostęp do informacji dzięki powszechności telewizji, radia oraz Internetu ma ku temu niezaprzeczalny wpływ. Jednak opierając się na grupie ankietowej, ludzie nie zdają sobie sprawy z pojęcia danych osobowych. Można uznać to za błahostkę, jednakże ciężko uznać, iż wynika to z ludzkiego pojęcia na ten temat. O ile numer telefonu oraz rachunek bankowy został jednoznacznie uznany za daną osobową co jest w pewnym stopniu logicznie uzasadnione, gdyż o wadze tych informacji jesteśmy informowani już na etapie podpisywania umów czy to z bankiem czy operatorem sieci telefonicznej. Tym bardziej, iż od 2017 roku, każda karta SIM musi być zarejestrowana, co wynika z ustawy z dnia 10 czerwca 2016 r. o działaniach antyterrorystycznych (Dz.U. 2016 poz. 904) [10], a użytkownicy telefonów komórkowych, poza oferowanymi profitami wiążącymi się z przeniesieniem numeru, mają też obowiązek wyrazić zgodę na przetwarzanie danych osobowych – a co z tym się wiąże – zapoznania się z tym, co jest uważane przez operatora za dane osobowe, zwyczajowo w paragrafach zawierających politykę prywatności. Odmienne prezentuje się numer IP, który jest uważany za daną ogólną. GIODO w swoim opracowaniu pt. „ABC zagrożeń bezpieczeństwa danych osobowych w systemach teleinformatycznych” [6,9] stwierdza, iż sam numer IP jest elementem, jaki może posłużyć do identyfikacji osoby fizycznej. Na przykład zestawiając abonenta internetowego z informacją o abonencie telefonicznym otrzymujemy dane osobowe wprost identyfikujące osobę. Trzy z pośród 11 osób jednoznacznie uznało adres IP za daną osobową i tylko jedna opowiedziała zgodnie z opinią GIODO, iż zależy od sytuacji. Podobnie ma się sytuacja z numerem VIN gdzie 100% osób udzieliło odpowiedzi innej, niż ta zaopiniowana przez Generalnego Inspektora.

4. Wnioski

Jest złudną nadzieją, iż dane przetrzymywane na serwerach różnych instytucji są w pełni zabezpieczone przed nieautoryzowanym dostępem osób trzecich. Należy mieć na uwadze, iż badanie zostało przeprowadzone dla małej liczby osób co może prowadzić do wyciągnięcia mylnych wniosków w stosunku do ogółu społeczeństwa. Ankieta ukazała problem jakim jest ogólny brak wiedzy z zakresu ochrony danych osobowych, z rozpoznaniem informacji będących danymi chronionymi a ogólnymi. Instytucje publiczne jak i prywatne radzą sobie z tym problemem na wiele sposobów od wprowadzenia wewnętrznych umów poufności po wdrażanie polityki bezpieczeństwa informacji. Jednakże istnieją odgórne aparaty państwowe stojące na straży poufności informacji oraz służące pomocą w precyzowaniu przepisów prawa. GIODO jest dla nas szczególnie ważnym organem a jego obowiązki z roku na rok rosną z powodu rozwoju branży technologicznej. Przepisy prawne są konstruowane w sposób dość ogólny, by sprawdzać się w jak największej ilości przypadków co powoduje nie rzadko problemy z jego interpretacją lub powoduje błędy w interpretacji. GIODO staje na straży poprawnego rozumienia ustaw w swoim zakresie kompetencji.

Dzięki niemu i jego mocy opiniowania ustaw osoby fizyczne jak i podmioty mogą korzystać z jasno uzasadnionych wy tłumaczeń często powielanych problemów interpretacyjnych.

Literatura

- [1] Yahoo Security Notice September 22, 2016, <https://help.yahoo.com/kb/account/sln28092.html> [27.12.2016]
- [2] Ustawa z dnia 29 sierpnia 1997 r. o ochronie danych osobowych, <http://isip.sejm.gov.pl/DetailsServlet?id=WDU19971330883> [27.12.2016]
- [3] Odpowiedzi na pytania dotyczące ustawy o ochronie danych osobowych, <http://www.giodo.gov.pl/317/j/pl/> [27.12.2016]
- [4] Jakie warunki musi spełnić administrator danych, aby przetwarzać dane osobowe zgodnie z ustawą o ochronie danych osobowych, http://www.giodo.gov.pl/317/id_art/3201/j/pl/ [27.12.2016]
- [5] Czy numer vin pojazdu stanowi dane osobowe w rozumieniu ustawy o ochronie danych osobowych?, http://www.giodo.gov.pl/319/id_art/2836/j/pl/ [27.12.2016]
- [6] Czy adres ip komputera należy do danych osobowych?, http://www.giodo.gov.pl/319/id_art/2258/j/pl/ [27.12.2016]
- [7] Raport o ochronie danych osobowych GIODO, http://www.giodo.gov.pl/plik/id_p/10383/j/pl/ [27.12.2016]
- [8] Rekonstrukcja odcisku palca niemieckiej minister obrony Ursuli Gertrud von den Leyen, video z konferencji, <https://www.youtube.com/watch?v=VVxL9ymiyAU> [27.12.2016]
- [9] Abc zagrożeń bezpieczeństwa danych osobowych w systemach teleinformatycznych, http://www.giodo.gov.pl/plik/id_p/1560/j/pl/ [27.12.2016]
- [10] Ustawa z dnia 10 czerwca 2016 r. o działaniach antyterrorystycznych, <http://isap.sejm.gov.pl/DetailsServlet?id=WDU20160000904> [27.12.2016]
- [11] Sakowska-Baryła M., Kontrolowanie przez GIODO przetwarzania danych osobowych, „Kontrola Państwowa” 2016, nr 2, s. 28-46
- [12] Przywora B., Z problematyki kontroli przetwarzania i ochrony danych osobowych przez Generalnego Inspektora Ochrony Danych Osobowych. Jawność i jej ograniczenia. Zadania i kompetencje (red. Szmulik B., Szpor G.), C.H. Beck, Warszawa 2015, Wyd. I, stron 400, ISBN 978-83-255-7664-6.
- [13] Kawecki M., Generalny Inspektor Ochrony Danych Osobowych jako centralny organ administracji państwowej, Przegląd Prawa Technologii Informacyjnych ICT Law Review z 2013 r. Wyd. I.
- [14] Byrski J., Tajemnica prawnie chroniona w działalności bankowej, Wydawnictwo C.H. Beck, Warszawa 2010, stron 417, ISBN: 978-83-255-1892-9
- [15] Krzysztofek M., Tajemnica bankowa i ochrona danych osobowych w praktyce bankowej, Wyd. LexisNexis, Warszawa 2010, stron 324, ISBN: 9788376203249

Test-Driven Development jako narzędzie optymalizacji procesu wytwarzania oprogramowania na platformie JEE

Grzegorz Sochacki*, Beata Pańczyk

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule poruszono temat korzyści płynących z zastosowania metodyki Test-Driven Development. Badania przeprowadzono z wykorzystaniem autorskiej aplikacji na platformę Java Enterprise Edition. Badana metodyka została porównana ze standardowym podejściem pisania testów.

Słowa kluczowe: Test-Driven Development; Java; testy jednostkowe; testowanie

*Autor do korespondencji.

Adres e-mail: g.sochacki91@gmail.com

Test-Driven Development as a tool to optimize the JEE programming

Grzegorz Sochacki*, Beata Pańczyk

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article is about the benefits of using the Test-Driven Development methodology. Tests were carried out on the author's application based on Java Enterprise Edition platform. The investigated methodology was compared with the standard approach to writing tests.

Keywords: Test-Driven Development; Java; unit tests; testing

*Corresponding author.

E-mail address: g.sochacki91@gmail.com

1. Wstęp

W procesach wytwarzania oprogramowania stosuje się wiele technik pozwalających na uzyskanie kodu dobrej jakości. Jednym z najpopularniejszych sposobów jest pisanie testów jednostkowych [1, 2]. Testy te mają za zadanie podzielić kod aplikacji na małe fragmenty funkcjonalności, które będą testowane niezależnie od reszty aplikacji. W programowaniu obiektowym, jedna klasa testowa przypada zazwyczaj na jedną klasę aplikacji a każdą metodę publiczną testuje przynajmniej jedna metoda testowa.

Standardowym podejściem w testach jest pisanie ich dla istniejących już funkcjonalności. W takim podejściu nie ma nic złego pod warunkiem, że do napisanego kodu istnieje możliwość napisania testu jednostkowego. Często próba napisania testów kończy się niepowodzeniem, ponieważ logika jest rozproszona pomiędzy wieloma klasami lub sama architektura kodu nie pozwala na poprawne przetestowanie. Wówczas pozostaje jedynie żmudny proces refaktoryzacji kodu (który bez napisanych testów nie zawsze jest bezpieczną operacją), lub zaniechanie testów.

Aby zapobiec tworzeniu kodu, który nie da się przetestować, wprowadzono metodykę programowania sterowaną testami (ang. *Test-Driven Development*) [3]. Metodyka ta odwraca kolejność operacji stosowanych w klasycznym podejściu. Najpierw pisze się testy do programowanej funkcjonalności, a następnie implementację.

Proces pisania testów i implementacji powtarzany jest do zakończenia programowania funkcjonalności. W momencie zakończenia programista powinien dokonać refaktoryzacji kodu, co dzięki testom wychwyci ewentualne naruszenie logiki aplikacji.

Używanie Test-Driven Development (TDD) spotyka się zazwyczaj ze sprzecznymi opiniami. Często wdrożenie metodyki do projektu zostaje zaniechane ze względu na obawy związane z dodatkowym narzutem czasu. W przypadku małych projektów brak testów nie stanowi dużego problemu. Ciężko natomiast odpowiednio utrzymać duży projekt, w którym pracuje wielu programistów, bez odpowiednio testowanego kodu.

W niniejszym artykule przedstawiono porównanie standardowego testowania z podejściem Test-Driven Development. Analizy dokonano podczas implementacji nowych funkcjonalności w aplikacji testowej opartej o platformę Java Enterprise Edition [4]. Do porównania obu podejść posłużyły obserwacje zarejestrowane podczas pisania kodu oraz struktura otrzymanego kodu. Testy wykonane zostały przez programistę, który posiadał już niezbędną wiedzę konieczną do realizacji postawionych zadań, zarówno w podejściu tradycyjnym jak i TDD.

Tezę badawczą jest:

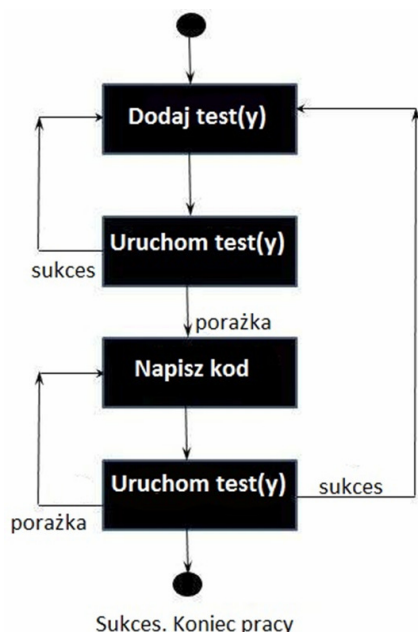
TDD jest narzędziem, które przyspiesza proces wytwarzania dobrej jakości kodu na platformie JEE.

2. Metodyka TDD w aplikacjach JEE

Platforma testowa pozwala na implementację funkcjonalności, tworzonych za pomocą metodyki TDD. Ocena korzyści zastosowania podejścia była oparta o wpływ TDD na proces tworzenia i jakość kodu.

W klasycznym podejściu pisania testów, testy pisane są do istniejącego już kodu. Odwrócenie tej zależności dla wielu programistów wydaje się zaprzeczeniem idei pisania testów jednostkowych. Istnieje obawa, że wdrożenie takiego podejścia może w początkowej fazie przynieść straty w postaci dłuższego czasu wytwarzanego kodu.

Samo w sobie podejście TDD oparte jest na prostym algorytmie pracy, który każdy, nawet początkujący programista powinien bez problemu zrozumieć. TDD wyróżnia 4 główne fazy pisania kodu. Rysunek 1 przedstawia podstawowy schemat pracy w TDD.



Rys. 1. Schemat pracy w metodyce Test-Driven Development

3. Realizacja badań

3.1. Platforma testowa

Do badań stworzono aplikację napisaną w języku Java 8. Jest to aplikacja internetowa, która została oparta o komponenty platformy Java Enterprise Edition. Aplikacja jest prostym systemem pozwalającym na zarządzanie klientami w przedsiębiorstwie. Aplikacja działa na serwerze WildFly w wersji 9.0.1. Dane o klientach przechowywane są w relacyjnej bazie danych uruchomionej na serwerze PostgreSQL. Do implementacji wykorzystano następujące technologie [5]:

- EJB 3,
- JAX-RS,
- Hibernate.

Zestaw technologii nie jest przypadkowy. Jest to podstawowy stos technologii, umożliwiający napisanie API funkcjonalnej aplikacji. Zapewnia on sprawne zarządzanie warstwą danych systemu oraz komunikację z zewnątrz poprzez interfejs REST [6].

W celu napisania odpowiednich testów jednostkowych, wykorzystano technologie TestNG [8] oraz Mockito [7]. Pierwsza z technologii jest frameworkiem wspierającym testowanie w aplikacjach Java. Umożliwia pisanie zarówno testów jednostkowych, jak i integracyjnych. Bibliotekę Mockito wykorzystano do tworzenia niezbędnych atrap obiektów.

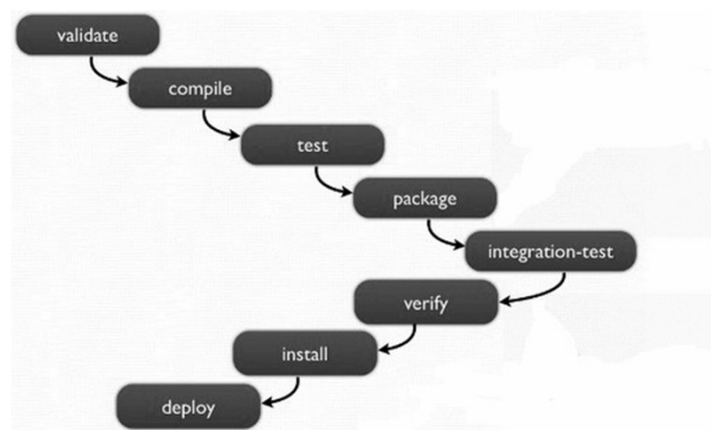
Do sprawnego poruszania się po projekcie zastosowano narzędzie developerskie Eclipse w wersji Neon, obsługujące większość języków programowania. W środowisku zainstalowano wtyczkę JbossTools, potrzebną do sprawnego integracji aplikacji z serwerem WildFly.

3.2. Struktura projektu

Automatyzacja budowania projektu użytego do badań jest zapewniona dzięki narzędziu Maven. Narzędzie to służy do budowania wielomodułowych projektów w języku Java. W pliku *pom.xml* definiowane są cele, które mają zostać osiągnięte po kompletnym zbudowaniu projektu [8]. Do przykładowych celi zalicza się kompilację plików źródłowych, uruchomienie testów, zbudowanie pliku wdrożeniowego.

Proces budowania projektu został podzielony na cykle, które odpowiadają za konkretne zadania (Rysunek 2):

- validate – weryfikacja poprawności projektu,
- compile – kompilacja kodu źródłowego,
- test – uruchomienie testów jednostkowych,
- package – zbudowanie pliku wdrożeniowego,
- integration-test – uruchomienie projektu w środowisku testowym,
- verify – weryfikacja paczki wdrożeniowej,
- install – umieszczenie paczki w lokalnym repozytorium,
- deploy – umieszczenie paczki w zdalnym repozytorium.



Rys. 2. Proces budowania aplikacji za pomocą narzędzia Maven [9]

3.3. Badane obiekty

Badanie przeprowadzono podczas implementacji nowych funkcjonalności w obu podejściach. Pojawiło się ryzyko, że programowanie dwa razy tych samych funkcjonalności może niekorzystnie wpłynąć na wyniki. Aby temu zapobiec, ustalono tygodniowy odstęp pomiędzy badaniami. W każdym przypadku wykorzystywano osobne klasy. Pomiędzy kodem obu podejść współdzielone są tylko klasy niezależne od testowanej w danym momencie metody. Dla wiarygodności testów, kolejność badanych technik była stosowana na przemian. Testy zostały przeprowadzone na implementacji trzech przykładowych metod biznesowych:

- obliczenie salda klienta,
- naliczanie rabatu dla klienta,
- naliczenia prowizji pracownika.

Na wyniki badań złożyły się dwa czynniki:

- informacje na temat parametrów pracy zarejestrowane podczas implementacji w danym podejściu,
- otrzymany kod.

Głównym elementem, na który należało zwrócić uwagę, był czas. Dla każdego z podejść rejestrowano czas poświęcony na pisanie kodu i pisanie testów. Rejestrowano także liczbę błędów w kodzie, jakie wykryły testy.

Otrzymany kod został poddany analizie w celu określenia jego jakości. Posłużyły do tego metryki CK. Metryki te pozwalają na ocenianie wykorzystanych mechanizmów programowania obiektowego.

4. Wyniki badań

4.1. Analiza implementacji

Pierwszą z badanych metod była metoda służąca do naliczania salda klienta. Logika metody nie była skomplikowana, lecz testy wymagały uwzględnienia wielu szczególnych przypadków (Przykład 1). Na potrzeby testów operacje prowadzone na danych zostały przeniesione z poziomu bazy danych na poziom kodu. Czas poświęcony na napisanie kodu nie różnił się bardzo pomiędzy dwoma podejściami. W przypadku TDD skrócił się natomiast czas pisania testów jednostkowych. W przypadku podejścia standardowego można zauważyć większą liczbę błędów popełnionych w kodzie. W przypadku TDD błędów jest znacznie mniej ze względu na implementację logiki w mniejszych fragmentach i ciągłe jej testowanie. Wyniki przedstawiono w tabeli 1.

Przykład 1. Fragment kodu pierwszej metody testowej

```
@Override
public BigDecimal calculateSaldo(Customer customer) {
    BigDecimal saldo = BigDecimal.ZERO;
    List<Obligation> obligations =
        obligationService.getObligationForClient(customer);
    List<Payment> payments = new ArrayList<Payment>();
    if(obligations == null || obligations.isEmpty()){
        return null;
    }
}
```

```
for(Obligation obligation : obligations){
    Calendar calendar = prepareFirstPenaltyDate(obligation);
    Date firstCashPenaltyDate = calendar.getTime();
    calendar = prepareSecondPenaltyDate(obligation);
    Date secondCashPenaltyDate = calendar.getTime();
    Date today = new Date();

    if(obligation.getPayments() != null){
        payments.addAll(obligation.getPayments());
    } else if
    (TimeUnit.DAYS.convert(today.getTime() -
        firstCashPenaltyDate.getTime(), TimeUnit.MILLISECONDS)
        <= 0){
        saldo = saldo.subtract(obligation.getAmountToPay().
            multiply(BigDecimal.valueOf(0.1)));
    } else if (TimeUnit.DAYS.convert(today.getTime() -
        secondCashPenaltyDate.getTime(),
        TimeUnit.MILLISECONDS) <= 0){
        saldo = saldo.subtract(obligation.getAmountToPay().
            multiply(BigDecimal.valueOf(0.3)));
    }
    saldo = saldo.subtract(obligation.getAmountToPay());
}

saldo = calculatePayments(saldo, payments);
return saldo;
}
```

Tabela 1. Wyniki badania pierwszej metody

	Standardowe podejście	Podejście TDD
Kolejność przeprowadzonego badania	1	2
Czas pisania kodu potrzebnego do implementacji metody [min]	50	45
Czas pisanie testów [min]	40	28
Liczba wykrytych błędów przez testy	5	1

Drugą metodą (Przykład 2) była implementacja logiki odpowiedzialnej za naliczenie rabatu dla klienta. Ze względu na dużo większą złożoność zadania niż w pierwszym przypadku, czas poświęcony na badanie znacznie się wydłużył. Złożona logika i wiele przypadków testowych wymagały większej liczby testów. Ponownie czas potrzebny na implementację logiki był zbliżony do siebie w obu przypadkach. Mniej czasu poświęcono na pisanie testów w podejściu TDD. Testy w obu przypadkach wykryły kilka błędów krytycznych. Większą liczbę błędów można zauważyć w standardowym podejściu. Wyniki zawarte w tabeli 2 przemawiają nieznacznie na korzyść TDD.

Przykład 2. Nagłówek drugiej metody testowej

```
public Integer calculateDiscount(Customer customer) {}
```

Badanie metody trzeciej (Przykład 3) polegało na naliczeniu prowizji dla pracownika. Złożoność metody była nieco mniejsza niż w przypadku poprzedniej. Testy wykryły błędy w operacjach matematycznych przeprowadzanych na kwotach prowizji. Czas poświęcony na napisanie logiki był znacznie dłuższy w przypadku standardowego podejścia.

Pisanie testów w obu podejściach zajęło podobną ilość czasu. Zestawienie wyników przedstawiono w tabeli 3.

Przykład 3. Nagłówek trzeciej metody testowej

```
public Provision calculateProvision(Contract contract){}
```

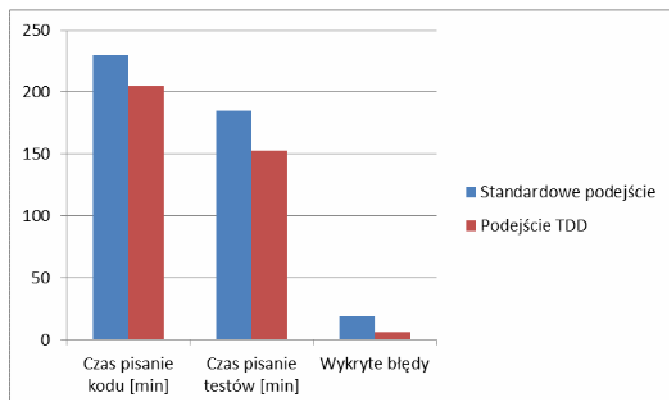
Tabela 2. Wyniki badania drugiej metody

	Standardowe podejście	Podejście TDD
Kolejność przeprowadzonego badania	2	1
Czas pisania kodu potrzebnego do implementacji metody [min]	110	105
Czas pisanie testów [min]	100	85
Liczba wykrytych błędów przez testy	8	3

Tabela 3. Wyniki badań trzeciej metody

	Standardowe podejście	Podejście TDD
Kolejność przeprowadzonego badania	1	2
Czas pisania kodu potrzebnego do implementacji metody [min]	70	55
Czas pisanie testów [min]	45	40
Liczba wykrytych błędów przez testy	6	2

Zebrane z trzech badań wyniki wskazują jako faworyta podejście TDD. Nie jest to jednak znacząca przewaga. Można uznać jednak, że wdrożenie TDD na pewno nie wpływa negatywnie na czas realizacji projektów. Na rysunku 3 przedstawiono porównanie wyników z trzech badań.



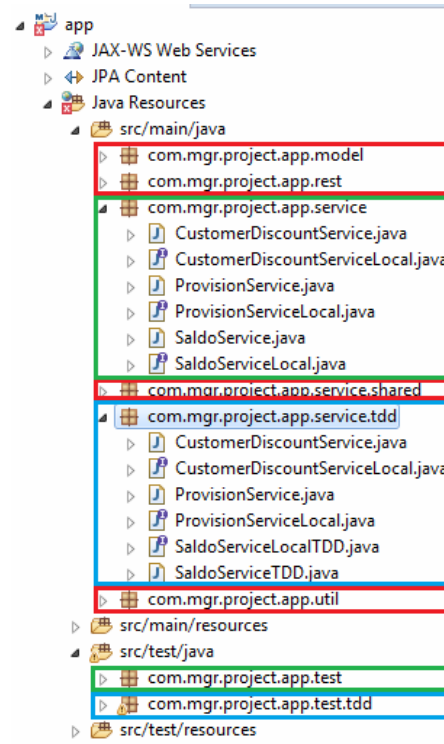
Rys. 3. Zestawienie wyników analizy implementacji

4.2. Analiza jakości kodu

Po oddzieleniu od siebie kodu obu podejść możliwa była jego analiza. Z analizy wykluczono metody i klasy wbudowane w język Java.

Na rysunku 4 przedstawiono strukturę projektu z podziałem na zasoby wykorzystane w badanych podejściach:

- kolor czerwony – zasoby współdzielone pomiędzy dwoma metodami,
- kolor zielony – standardowe podejście,
- kolor niebieski – podejście TDD.



Rys. 4. Podział projektu ze względu na kod wykorzystywany w badanych podejściach

Metryki CK, które miały za zadanie ocenić jakość kodu mierzyły następujące parametry:

- RFC – metryka mierząca odpowiedzialność klasy. Na jej wartość składa się liczba publicznych metod i jej podklas;
- CBO – metryka określająca powiązania pomiędzy obiektami. Uwzględnia liczbę odwołań pomiędzy klasami np. poprzez parametry metod;
- WMC – miara złożoności w klasie, składa się z ważonej liczby metod w klasie;
- LCOM3 – miara (wartość z przedziału $<0,1>$) spójności metod wewnątrz klasy.

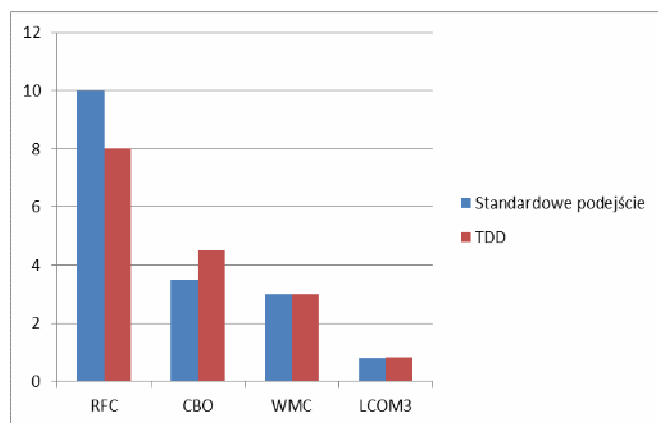
Analizę kodu przeprowadzono na klasach zawartych w katalogu `src/main/java`. To w tym katalogu znajduje się główna implementacja logiki. Do analizy nie wykorzystano klas z testami, ponieważ głównym celem badań był kod odpowiedzialny za logikę aplikacji.

Wartości metryk dla otrzymanego kodu przedstawione zostały w tabeli 4.

Tabela 4. Wartości metryk CK

	Standardowe podejście	TDD
RFC	10	8
CBO	3,5	4,5
WMC	3	3
LCOM3	0,8	0,82

Wyniki obu podejść są bardzo zbliżone do siebie i trudno jednoznacznie określić przewagę któregoś z nich. Graficzne zestawienie wyników przedstawiono na rysunku 5.



Rys. 5. Wyniki porównania wartości metryk

5. Wnioski

Wyniki badań wskazują na to, że zastosowanie techniki Test-Driven Development wpłynęło korzystanie na proces

tworzenia kodu. Wbrew obawom, czas jaki należy przeznaczyć na zaprogramowanie funkcjonalności, nie wydłużył się. Fakt ten korzystnie wpływa na ocenę z biznesowego punktu widzenia.

Jakość kodu otrzymanego w badaniach utrzymuje się na wysokim poziomie. Na jakość kodu metodyka TDD nie ma bezpośredniego wpływu. Główną rolę odgrywają tutaj testy jednostkowe. To one zapewniają bezpieczeństwo oraz zachowanie odpowiednich struktur w pisany kodzie. Dzięki zastosowaniu badanej techniki można uniknąć sytuacji, w której napisany kod nie jest możliwy do przetestowania. Zastosowanie Test-Driven Development samo w sobie wymusza pokrycie testami praktycznie całej aplikacji, dzięki czemu z góry można założyć, że otrzymany kod będzie dobrej jakości.

Oczywiście największą rolę w programowaniu odgrywa programista. Wiele zależy od jego osobistych upodobań i przyzwyczajęń. Sama metodyka nie zastąpi doświadczenia i kreatywności w programowaniu.

Literatura

- [1] Pressman R.S., Software Engineering: A Practitioner's Approach 8th Edition, McGraw-Hill Higher Education, 2015.
- [2] <http://www.testerzy.pl> [07.03.2017]
- [3] Farcic V, Garcia A., *Test-Driven Java Development*, Packt Publishing, 2015.
- [4] Mil M., *Java EE 7 Development with Wildfly*, Packt Publishing, 2014.
- [5] <https://docs.oracle.com/javaee/7/> [07.03.2017]
- [6] Burke B., *RESTful Java with JAX-RS 2.0*. 2nd Edition, O'Reilly Media, 2013.
- [7] <http://site.mockito.org/> [07.03.2017]
- [8] <http://junit.org> [03.01.2017]
- [9] <https://cooldevstuff.wordpress.com/tag/maven-jetty-plugin-2/>, [10.02.2017]

Porównanie wybranych bibliotek do tworzenia graficznego interfejsu internetowych aplikacji JEE

Piotr Paweł Suwała*, Beata Pańczyk

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W środowisku Java Enterprise Edition podstawową technologią tworzenia graficznego interfejsu użytkownika (GUI) aplikacji internetowych jest JavaServer Pages (JSP). Możliwości oferowane przez strony JSP nie są jednak wystarczające, co spowodowało powstanie wielu rozszerzeń wspomagających programistę w szybkim tworzeniu nowoczesnych GUI. W artykule przedstawiono wyniki porównania bibliotek: JavaServer Faces, Rich Faces i Prime Faces.

Słowa kluczowe: JavaServer Faces; PrimeFaces; RichFaces; GUI

* Autor do korespondencji.

Adres e-mail: suwcio@gmail.com

Comparison of selected graphical web interface libraries used in JEE applications

Piotr Paweł Suwała*, Beata Pańczyk

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The basic technology for creating a graphical user interface (GUI) for Java Enterprise Edition web applications is JavaServer Pages (JSP). However, the capabilities offered by JSP are not sufficient, which has resulted in many extensions that assist the programmer in quickly creating modern GUI. The paper presents the results of comparison JavaServer Faces, Rich Faces and Prime Faces libraries.

Keywords: JavaServer Faces; PrimeFaces; RichFaces; GUI

*Corresponding author.

E-mail address: suwcio@gmail.com

1. Wstęp

Bardzo istotną rzeczą podczas produkcji oprogramowania jest zapewnienie atrakcyjnego graficznego interfejsu użytkownika (GUI). Z punktu widzenia klienta to nie tylko funkcjonalność, ale w dużym stopniu nowoczesny wygląd i intuicyjny interfejs wpływa na pozytywny odbiór aplikacji.

Java Enterprise Edition (JEE) [1, 2, 3] jest rozbudowaną platformą programowania, umożliwiającą, między innymi, tworzenie aplikacji internetowych. W jej skład wchodzi wiele technologii pozwalających na szybkie i zautomatyzowane tworzenie oprogramowania. Dodatkowe rozszerzenia są tworzone i udostępniane przez firmy zewnętrzne (np. cieszący się popularnością Spring czy trochę już przestarzały Struts).

W środowisku JEE powstało wiele rozwiązań wspomagających budowę interesujących GUI. Najbardziej znanym sposobem obrazowania danych, są strony JavaServer Pages (JSP) [4]. Składają się one ze zwykłych znaczników HTML oraz znaczników specyficznych dla JSP, które odpowiadają za dynamiczne generowanie treści wynikowego dokumentu HTML.

Strony JSP okazały się znakomitą podstawą dla innych technologii generowania widoku. Bardzo dobrym przykładem jest tu JavaServer Faces (JSF) [5, 6]. Potrzeby rynku wymusiły rozwój kolejnych rozszerzeń umożliwiających tworzenie bardziej atrakcyjnych i nowoczesnych interfejsów użytkownika. Między innymi są to PrimeFaces [7] oraz RichFaces [8].

2. Cel badań

Celem badań była analiza funkcjonalności oraz porównanie najbardziej popularnych bibliotek ułatwiających proces tworzenia graficznego interfejsu użytkownika internetowych aplikacji JEE.

Tezę badawczą sformułowano następująco: *Dla aplikacji internetowych tworzonych w środowisku JEE można budować atrakcyjne graficznie i nowoczesne interfejsy użytkownika.*

3. Aplikacje testowe

Do badań zostały utworzone trzy aplikacje testowe o identycznej funkcjonalności, różniące się jedynie technologią zastosowaną do budowy interfejsu graficznego. Są to odpowiednio JavaServer Faces (JSF), PrimeFaces

i RichFaces. Wybór tych bibliotek został podyktowany ich popularnością.

W aplikacjach, poza bibliotekami do tworzenia GUI, wykorzystano:

- Java 7,
- serwer bazy danych MySQL 5.6.11 [9],
- phpMyAdmin,
- technologię ORM – Hibernate 4.2.2.

Testy zostały przeprowadzone na komputerze z systemem operacyjnym Windows 8, Intel Core i5-3210M 2 x 2.50 GHz, 8 GB RAM.

Aplikacja jest formą bloga umożliwiającą zarządzanie postami oraz użytkownikami. Stronę główną aplikacji testowej zrealizowanej w oparciu o JSF przedstawiono na rysunku 1.



Rys. 1. Strona główna aplikacji testowej (JSF)

4. Analiza porównawcza bibliotek

Do porównania trzech wybranych bibliotek analizowano 7 kryteriów:

- liczbę komponentów,
- łatwość implementacji,
- podstawowe możliwości,
- funkcje niewystępujące w innych bibliotekach,
- dostęp do dokumentacji, wsparcie,
- wydajność,
- popularność.

4.1. Liczba komponentów

Liczba dostępnych kontrolki GUI została określona na podstawie witryny każdej biblioteki. Witryny te prezentują wygląd, implementację oraz sposób zastosowania każdej kontrolki (ang. showcase).

W Tabeli 1 znajduje się liczba kategorii oraz komponentów dla poszczególnych bibliotek.

Tabela 1. Liczba kategorii i komponentów w poszczególnych bibliotekach

	JSF	PrimeFaces	RichFaces
Liczba kategorii	13	15	12
Liczba komponentów	81	177	68

W przypadku liczby komponentów najlepiej wypada biblioteka PrimeFaces. Posiada ich najwięcej co z założenia pozwala na budowanie bardziej różnorodnych oraz ciekawych interfejsów użytkownika.

4.2. Łatwość implementacji

Każdą z bibliotek można zaimplementować w aplikacji na dwa sposoby. Pierwszym z nich jest pobranie biblioteki w tzw. pliku JAR i umieszczenie go w katalogu zawierającym biblioteki w aplikacji. Drugi sposób polega na automatycznym ściąganiu biblioteki bezpośrednio do projektu przy pomocy narzędzia Maven.

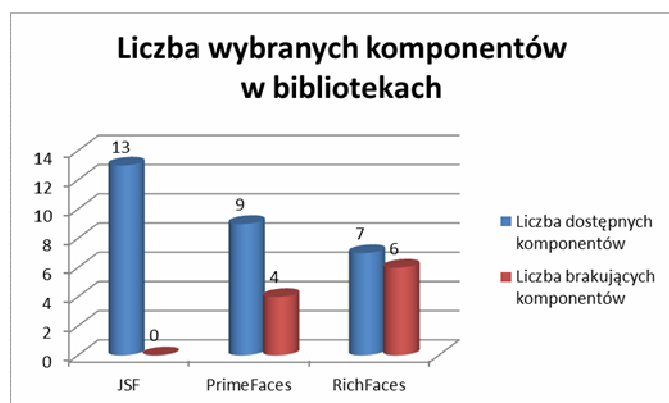
4.3. Podstawowe możliwości

Każda z bibliotek posiada pewną liczbę komponentów, dzięki którym można stworzyć podstawowy interfejs użytkownika aplikacji webowej. Są to na przykład podstawowe kontrolki do wyświetlania i wprowadzania tekstu, lista rozwijana czy prezentacja tabelaryczna danych. W tabeli 2 przedstawiono wybrane znaczniki oraz wskazano ich dostępność w każdej z bibliotek.

Tabela 2. Dostępność podstawowych kontrolki w poszczególnych bibliotekach

Kontrolki	JSF	PrimeFaces	RichFaces
Przycisk polecenia	h:commandButton	p:commandButton	a4j:commandButton
Hiperłącze polecenia	h:commandLink	p:commandLink	a4j:commandLink
Tabela danych	h:dataTable	p:dataTable	rich:dataTable
Tekst wejściowy	h:inputText	p:inputText	brak
Plik wejściowy	h:inputFile	p:uploadFile	rich:fileUpload
Obraz graficzny	h:graphicImage	p:graphicImage	brak
Tekst wyjściowy	h:outputText	brak	brak
Panel siatka	h:panelGrid	p:panelGrid	brak
Wybór jednej wartości	h:selectOneMenu	brak	rich:select
Wybór pozycji	f:selectItem	brak	brak
	f:selectItems	brak	brak
Wiadomość i oraz Wiadomości	h:message	p:message	rich:message
	h:messages	p:messages	rich:messages

Rysunek 2 obrazuje liczbę dostępnych oraz brakujących komponentów w poszczególnych bibliotekach.



Rys. 2. Liczba dostępnych oraz brakujących komponentów GUI (z grupy podstawowych kontrolki)

Biblioteka JSF zawiera większość podstawowych komponentów. Jednak warto zaznaczyć, że komponenty z biblioteki PrimeFaces wyglądają o wiele bardziej atrakcyjnie graficznie. Ich liczba jest nieco mniejsza, ale brakujące znaczniki można zastąpić tymi z JSF. Najslabiej prezentuje się biblioteka RichFaces, która posiada najmniejszą liczbę znaczników i przegrywa pod względem wyglądu z PrimeFaces.

4.4. Funkcje niewystępujące w innych bibliotekach

JavaServer Faces

JSF udostępnia użytkownikom tzw. system szablonów. Pozwala on na stworzenie szkieletu strony WWW, a następnie tworzenia konkretnych stron na podstawie utworzonego szablonu.

Implementacja biblioteki JSF Mojarra [6] zawiera standardową bibliotekę znaczników Java. Funkcja ta jest jeszcze pozostałością po JSP i pozwala na obsługę podstawowych funkcji Java na stronach JSF.

PrimeFaces i RichFaces

Przewagą bibliotek PrimeFaces oraz RichFaces są większe możliwości tworzenia interfejsu użytkownika w oparciu o dodatkowe komponenty rozszerzające podstawowe funkcje (np. rozwijany panel, galeria, zegar, komponent pozwalający na wydruk zawartości innego komponentu, komponenty do tworzenia wykresów itp.). W tym wypadku dużo lepiej wypada PrimeFaces, która pozwala na osiągnięcie ciekawszych rezultatów poprzez ogromną różnorodność znaczników.

4.5. Dokumentacja i wsparcie społecznościowe

Każda z prezentowanych bibliotek posiada dokumentację, wsparcie (np. w postaci forum lub FAQ) oraz społeczność zorganizowaną wokół danego frameworka. Tabela 3 prezentuje elementy dostępne w dokumentacji poszczególnych bibliotek.

Tabela 3. Elementy dostępne w dokumentacji w poszczególnych bibliotekach

	JSF	PrimeFaces	RichFaces
Instalacja	nie	tak	tak
Konfiguracja	nie	tak	tak
Opis kontrolki	tak	tak	tak
Przykłady	nie	tak	nie

Tabela 4 prezentuje rodzaje wsparcia dla poszczególnych bibliotek.

Tabela 4. Możliwości wsparcia w poszczególnych bibliotekach

	JSF	PrimeFaces	RichFaces
FAQ	tak	tak	tak
Forum	tak	tak	tak
Blog	nie	tak	tak
Get started	tak	tak	tak
Screencasts	nie	nie	tak

W tabeli 5 przedstawiono dostępność poszczególnych bibliotek na portalach społecznościowych.

Tabela 5. Dostępność na portalach społecznościowych

	JSF	PrimeFaces	RichFaces
Facebook	tak	tak	nie
Twitter	tak	tak	tak

Po przeanalizowaniu dostępnych materiałów na stronach poszczególnych bibliotek można stwierdzić, że najgorsze wsparcie posiada JSF. Najlepiej pod tym względem wypada biblioteka PrimeFaces, która oferuje szeroką dokumentację, przykłady zastosowań oraz wsparcie ze strony społeczności skupionej wokół technologii.

4.6. Wydajność

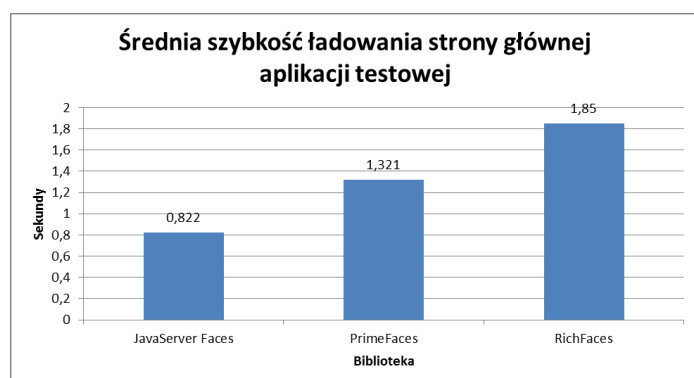
Mimo szybkich i wydajnych serwerów, to jak dany framework wpływa na czas generowania odpowiedzi na żądania klienta, jest jednym z najważniejszych kryteriów oceny. Wydajność bibliotek została zmierzona poprzez sprawdzenie całkowitego czasu ładowania aplikacji testowej przez przeglądarkę internetową. Posłużyła do tego aplikacja webowa PageScoring [10]. Jako serwer dla aplikacji testowych wykorzystano platformę OpenShift [11] w wersji darmowej. Udostępnia ona kontener dla aplikacji opartych o JEE.

Dla każdej biblioteki osobno, w aplikacji testowej wykonano 10 pomiarów prędkości ładowania strony głównej. Wyniki zostały przedstawione w tabeli 6.

W celu lepszego zobrazowania wydajności aplikacji testowej na rysunku 3 przedstawiono wykres prezentujący średnie czasy pomiaru.

Tabela 6. Wyniki pomiarów szybkości ładowania strony głównej aplikacji

Numer pomiaru	JSF	PrimeFaces	RichFaces
1	0,75	1,35	2,05
2	0,67	1,33	1,53
3	0,63	1,57	3,81
4	0,69	1,5	1,79
5	0,63	1,31	1,87
6	1,44	1,19	1,4
7	0,85	1,27	1,39
8	0,69	1,27	1,39
9	1,04	1,22	1,58
10	0,83	1,2	1,69
Średni czas [s]	0,822	1,321	1,85



Rys. 3. Średnia szybkość ładowania strony głównej aplikacji testowej

Na wykresie wyraźnie widać, że aplikacja testowa z biblioteką JSF ładuje się najszybciej. Nieco wolniejszy czas ładowania został odnotowany przy testach z biblioteką PrimeFaces. Najwolniej strona ładowała się gdy została użyta biblioteka RichFaces.

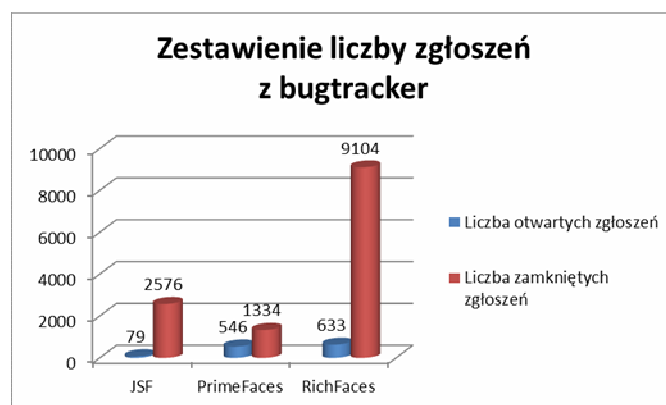
4.7. Popularność

W celu zbadania popularności poszczególnych bibliotek, zastosowano dwa kryteria. Jest to:

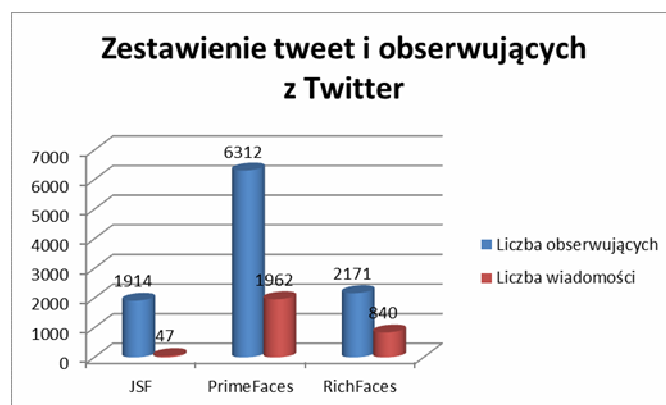
- liczba aktualnie otwartych zadań na odpowiednich platformach typu 'bugtracker' (Rys. 4);
- liczba osób obserwujących stronę projektu oraz liczba wiadomości na portalu społecznościowym Twitter (Rys. 5).

Odnośnie pierwszego kryterium, tylko JSF i RichFaces używają tej samej platformy, ale służy ona jedynie do raportowania zgłoszeń o wykrytych problemach. Darmowa wersja PrimeFaces posługuje się platformą, która jest repozytorium, a dodatkowo pozwala na tworzenie zgłoszeń.

Z przedstawionych danych można wywnioskować, że największą popularnością cieszy się PrimeFaces. Świadczy o tym także duże zainteresowanie na portalu Twitter. Z liczby zadań zgłoszonych na portalach typu 'bugtracker' wynika, że intensywne prace trwają nad bibliotekami PrimeFaces oraz RichFaces. JSF nie jest już tak intensywnie rozwijany, stąd liczba zgłoszeń nie jest duża.



Rys. 4. Zestawienie liczby zgłoszeń z bugtracker



Rys. 5. Zestawienie z portalu Twitter

4.8. Ocena bibliotek - podsumowanie

W tabeli 7 przedstawiono próbę oceny poszczególnych kryteriów w skali od 1 do 5. Nie stosowano wag dla poszczególnych kryteriów.

Tabela 7. Ocena bibliotek według wybranych kryteriów w skali 1 - 5

	JSF	PrimeFaces	RichFaces
Liczba komponentów	3	5	3
Łatwość implementacji	5	5	5
Podstawowe możliwości	5	4	3
Funkcje wyróżniające bibliotekę	3	5	3
Dokumentacja, wsparcie, społeczność	3	5	4
Wydajność	5	4	3
Popularność	3	5	4
Średnia ocena końcowa	3,86	4,71	3,57

Porównując wyniki z tabeli 7 widać, że najlepiej prezentuje się biblioteka PrimeFaces, następnie JavaServer Faces, na końcu zaś RichFaces.

5. Wnioski

Niewątpliwym atutem, który został zauważony podczas porównywania, jest to, że biblioteki nie wykluczają się nawzajem. W jednym projekcie istnieje możliwość używania komponentów z każdej analizowanej biblioteki. Pozwala to na ogromną elastyczność i dowolność stosowanych rozwiązań.

Z przeprowadzonej analizy wynika, że najlepszym wyborem do budowy w pełni funkcjonalnych aplikacji webowych z bogatym i przyjaznym interfejsem użytkownika jest połączenie możliwości JSF z PrimeFaces. JSF udostępnia podstawowe funkcje, niezbędne do utworzenia szkieletu strony internetowej, natomiast efektywniejszych wizualnie komponentów dostarcza PrimeFaces.

Jeśli programista musi się ograniczyć do jednego rozwiązania, to najlepszym wyborem jest JSF. Dwie pozostałe biblioteki nie oferują wszystkich podstawowych znaczników, bez których utworzenie kompletnej strony, nie jest możliwe.

Najmniej zachęcającą biblioteką okazała się RichFaces. O ile w pewnych aspektach (np. dokumentacja, wsparcie, funkcje wyróżniające) jest lepsza od JSF to jednak i tutaj PrimeFaces jest lepszym wyborem.

Literatura

- [1] <http://www.oracle.com/technetwork/java/javasee/overview/index.html>, [10.04.2017].
- [2] E. Jendrock, R. Cervera – Navaroo, I. Evans, D. Gollapudi, K. Haase, W. Markito, C. Srivathsa, "Java EE 6 Zaawansowany przewodnik", Wydanie IV, Helion, 2013.
- [3] K. Rychlicki – Kicior, "Java EE 6 Programowanie aplikacji WWW", Wydanie II, Helion, 2015.
- [4] <http://www.oracle.com/technetwork/java/javasee/jsp/index.html>, [10.04.2017].
- [5] <http://www.oracle.com/technetwork/java/javasee/javaxserverfaces-139869.html>, [10.04.2017].
- [6] Mojarra JavaServer Faces – Oracle's open source implementation of the JSF standard, <https://javaserverfaces.java.net/>, [20.02.2017].
- [7] PrimeFaces Ultimate UI Framework for Java EE, <http://www.primefaces.org/>, [20.02.2017].
- [8] RichFaces The next – generation JSF component framework by JBoss!, <http://richfaces.jboss.org/>, [20.02.2017].
- [9] K. Yank, „PHP i MySQL Witryna WWW oparta na bazie danych”, Wydanie III, Helion, 2005.
- [10] <http://www.pagespeeding.com/website-speed-test/>, [20.02.2017].
- [11] <https://www.openshift.com/>, platforma testowa dla aplikacji JEE, [20.02.2017].

Porównanie wybranych bibliotek do tworzenia graficznego interfejsu internetowych aplikacji JEE

Piotr Paweł Suwała*, Beata Pańczyk

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W środowisku Java Enterprise Edition podstawową technologią tworzenia graficznego interfejsu użytkownika (GUI) aplikacji internetowych jest JavaServer Pages (JSP). Możliwości oferowane przez strony JSP nie są jednak wystarczające, co spowodowało powstanie wielu rozszerzeń wspomagających programistę w szybkim tworzeniu nowoczesnych GUI. W artykule przedstawiono wyniki porównania bibliotek: JavaServer Faces, Rich Faces i Prime Faces.

Słowa kluczowe: JavaServer Faces; PrimeFaces; RichFaces; GUI

* Autor do korespondencji.

Adres e-mail: suwcio@gmail.com

Comparison of selected graphical web interface libraries used in JEE applications

Piotr Paweł Suwała*, Beata Pańczyk

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The basic technology for creating a graphical user interface (GUI) for Java Enterprise Edition web applications is JavaServer Pages (JSP). However, the capabilities offered by JSP are not sufficient, which has resulted in many extensions that assist the programmer in quickly creating modern GUI. The paper presents the results of comparison JavaServer Faces, Rich Faces and Prime Faces libraries.

Keywords: JavaServer Faces; PrimeFaces; RichFaces; GUI

*Corresponding author.

E-mail address: suwcio@gmail.com

1. Wstęp

Bardzo istotną rzeczą podczas produkcji oprogramowania jest zapewnienie atrakcyjnego graficznego interfejsu użytkownika (GUI). Z punktu widzenia klienta to nie tylko funkcjonalność, ale w dużym stopniu nowoczesny wygląd i intuicyjny interfejs wpływa na pozytywny odbiór aplikacji.

Java Enterprise Edition (JEE) [1, 2, 3] jest rozbudowaną platformą programowania, umożliwiającą, między innymi, tworzenie aplikacji internetowych. W jej skład wchodzi wiele technologii pozwalających na szybkie i zautomatyzowane tworzenie oprogramowania. Dodatkowe rozszerzenia są tworzone i udostępniane przez firmy zewnętrzne (np. cieszący się popularnością Spring czy trochę już przestarzały Struts).

W środowisku JEE powstało wiele rozwiązań wspomagających budowę interesujących GUI. Najbardziej znanym sposobem obrazowania danych, są strony JavaServer Pages (JSP) [4]. Składają się one ze zwykłych znaczników HTML oraz znaczników specyficznych dla JSP, które odpowiadają za dynamiczne generowanie treści wynikowego dokumentu HTML.

Strony JSP okazały się znakomitą podstawą dla innych technologii generowania widoku. Bardzo dobrym przykładem jest tu JavaServer Faces (JSF) [5, 6]. Potrzeby rynku wymusiły rozwój kolejnych rozszerzeń umożliwiających tworzenie bardziej atrakcyjnych i nowoczesnych interfejsów użytkownika. Między innymi są to PrimeFaces [7] oraz RichFaces [8].

2. Cel badań

Celem badań była analiza funkcjonalności oraz porównanie najbardziej popularnych bibliotek ułatwiających proces tworzenia graficznego interfejsu użytkownika internetowych aplikacji JEE.

Tezę badawczą sformułowano następująco: *Dla aplikacji internetowych tworzonych w środowisku JEE można budować atrakcyjne graficznie i nowoczesne interfejsy użytkownika.*

3. Aplikacje testowe

Do badań zostały utworzone trzy aplikacje testowe o identycznej funkcjonalności, różniące się jedynie technologią zastosowaną do budowy interfejsu graficznego. Są to odpowiednio JavaServer Faces (JSF), PrimeFaces

i RichFaces. Wybór tych bibliotek został podyktowany ich popularnością.

W aplikacjach, poza bibliotekami do tworzenia GUI, wykorzystano:

- Java 7,
- serwer bazy danych MySQL 5.6.11 [9],
- phpMyAdmin,
- technologię ORM – Hibernate 4.2.2.

Testy zostały przeprowadzone na komputerze z systemem operacyjnym Windows 8, Intel Core i5-3210M 2 x 2.50 GHz, 8 GB RAM.

Aplikacja jest formą bloga umożliwiającą zarządzanie postami oraz użytkownikami. Stronę główną aplikacji testowej zrealizowanej w oparciu o JSF przedstawiono na rysunku 1.



Rys. 1. Strona główna aplikacji testowej (JSF)

4. Analiza porównawcza bibliotek

Do porównania trzech wybranych bibliotek analizowano 7 kryteriów:

- liczbę komponentów,
- łatwość implementacji,
- podstawowe możliwości,
- funkcje niewystępujące w innych bibliotekach,
- dostęp do dokumentacji, wsparcie,
- wydajność,
- popularność.

4.1. Liczba komponentów

Liczba dostępnych kontrolki GUI została określona na podstawie witryny każdej biblioteki. Witryny te prezentują wygląd, implementację oraz sposób zastosowania każdej kontrolki (ang. showcase).

W Tabeli 1 znajduje się liczba kategorii oraz komponentów dla poszczególnych bibliotek.

Tabela 1. Liczba kategorii i komponentów w poszczególnych bibliotekach

	JSF	PrimeFaces	RichFaces
Liczba kategorii	13	15	12
Liczba komponentów	81	177	68

W przypadku liczby komponentów najlepiej wypada biblioteka PrimeFaces. Posiada ich najwięcej co z założenia pozwala na budowanie bardziej różnorodnych oraz ciekawych interfejsów użytkownika.

4.2. Łatwość implementacji

Każdą z bibliotek można zaimplementować w aplikacji na dwa sposoby. Pierwszym z nich jest pobranie biblioteki w tzw. pliku JAR i umieszczenie go w katalogu zawierającym biblioteki w aplikacji. Drugi sposób polega na automatycznym ściąganiu biblioteki bezpośrednio do projektu przy pomocy narzędzia Maven.

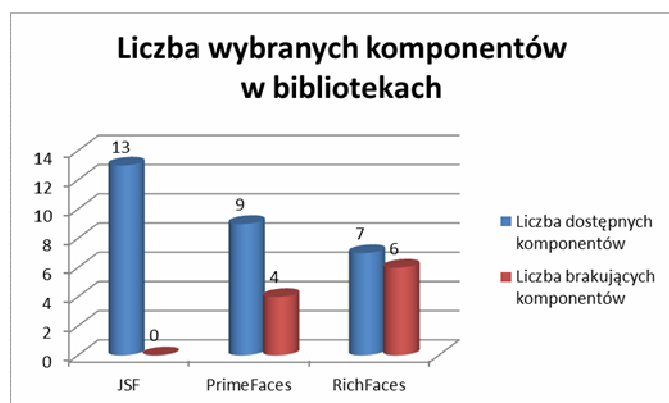
4.3. Podstawowe możliwości

Każda z bibliotek posiada pewną liczbę komponentów, dzięki którym można stworzyć podstawowy interfejs użytkownika aplikacji webowej. Są to na przykład podstawowe kontrolki do wyświetlania i wprowadzania tekstu, lista rozwijana czy prezentacja tabelaryczna danych. W tabeli 2 przedstawiono wybrane znaczniki oraz wskazano ich dostępność w każdej z bibliotek.

Tabela 2. Dostępność podstawowych kontrolki w poszczególnych bibliotekach

Kontrolki	JSF	PrimeFaces	RichFaces
Przycisk polecenia	h:commandButton	p:commandButton	a4j:commandButton
Hiperłącze polecenia	h:commandLink	p:commandLink	a4j:commandLink
Tablica danych	h:dataTable	p:dataTable	rich:dataTable
Tekst wejściowy	h:inputText	p:inputText	brak
Plik wejściowy	h:inputFile	p:uploadFile	rich:fileUpload
Obraz graficzny	h:graphicImage	p:graphicImage	brak
Tekst wyjściowy	h:outputText	brak	brak
Panel siatka	h:panelGrid	p:panelGrid	brak
Wybór jednej wartości	h:selectOneMenu	brak	rich:select
Wybór pozycji	f:selectItem	brak	brak
	f:selectItems	brak	brak
Wiadomość oraz Wiadomości	h:message	p:message	rich:message
	h:messages	p:messages	rich:messages

Rysunek 2 obrazuje liczbę dostępnych oraz brakujących komponentów w poszczególnych bibliotekach.



Rys. 2. Liczba dostępnych oraz brakujących komponentów GUI (z grupy podstawowych kontrolki)

Biblioteka JSF zawiera większość podstawowych komponentów. Jednak warto zaznaczyć, że komponenty z biblioteki PrimeFaces wyglądają o wiele bardziej atrakcyjnie graficznie. Ich liczba jest nieco mniejsza, ale brakujące znaczniki można zastąpić tymi z JSF. Najslabiej prezentuje się biblioteka RichFaces, która posiada najmniejszą liczbę znaczników i przegrywa pod względem wyglądu z PrimeFaces.

4.4. Funkcje niewystępujące w innych bibliotekach

JavaServer Faces

JSF udostępnia użytkownikom tzw. system szablonów. Pozwala on na stworzenie szkieletu strony WWW, a następnie tworzenia konkretnych stron na podstawie utworzonego szablonu.

Implementacja biblioteki JSF Mojarra [6] zawiera standardową bibliotekę znaczników Java. Funkcja ta jest jeszcze pozostałością po JSP i pozwala na obsługę podstawowych funkcji Java na stronach JSF.

PrimeFaces i RichFaces

Przewagą bibliotek PrimeFaces oraz RichFaces są większe możliwości tworzenia interfejsu użytkownika w oparciu o dodatkowe komponenty rozszerzające podstawowe funkcje (np. rozwijany panel, galeria, zegar, komponent pozwalający na wydruk zawartości innego komponentu, komponenty do tworzenia wykresów itp.). W tym wypadku dużo lepiej wypada PrimeFaces, która pozwala na osiągnięcie ciekawszych rezultatów poprzez ogromną różnorodność znaczników.

4.5. Dokumentacja i wsparcie społecznościowe

Każda z prezentowanych bibliotek posiada dokumentację, wsparcie (np. w postaci forum lub FAQ) oraz społeczność zorganizowaną wokół danego frameworka. Tabela 3 prezentuje elementy dostępne w dokumentacji poszczególnych bibliotek.

Tabela 3. Elementy dostępne w dokumentacji w poszczególnych bibliotekach

	JSF	PrimeFaces	RichFaces
Instalacja	nie	tak	tak
Konfiguracja	nie	tak	tak
Opis kontrolki	tak	tak	tak
Przykłady	nie	tak	nie

Tabela 4 prezentuje rodzaje wsparcia dla poszczególnych bibliotek.

Tabela 4. Możliwości wsparcia w poszczególnych bibliotekach

	JSF	PrimeFaces	RichFaces
FAQ	tak	tak	tak
Forum	tak	tak	tak
Blog	nie	tak	tak
Get started	tak	tak	tak
Screencasts	nie	nie	tak

W tabeli 5 przedstawiono dostępność poszczególnych bibliotek na portalach społecznościowych.

Tabela 5. Dostępność na portalach społecznościowych

	JSF	PrimeFaces	RichFaces
Facebook	tak	tak	nie
Twitter	tak	tak	tak

Po przeanalizowaniu dostępnych materiałów na stronach poszczególnych bibliotek można stwierdzić, że najgorsze wsparcie posiada JSF. Najlepiej pod tym względem wypada biblioteka PrimeFaces, która oferuje szeroką dokumentację, przykłady zastosowań oraz wsparcie ze strony społeczności skupionej wokół technologii.

4.6. Wydajność

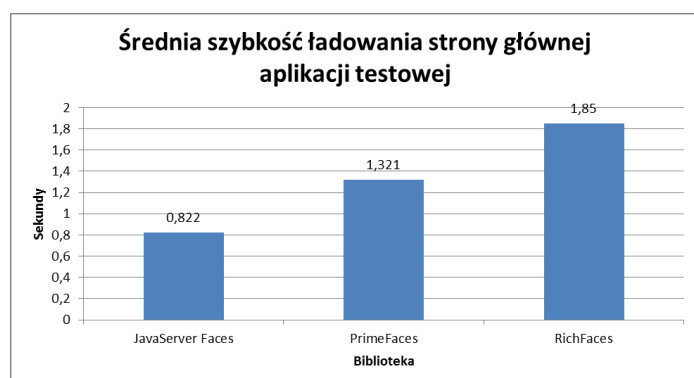
Mimo szybkich i wydajnych serwerów, to jak dany framework wpływa na czas generowania odpowiedzi na żądania klienta, jest jednym z najważniejszych kryteriów oceny. Wydajność bibliotek została zmierzona poprzez sprawdzenie całkowitego czasu ładowania aplikacji testowej przez przeglądarkę internetową. Posłużyła do tego aplikacja webowa PageScoring [10]. Jako serwer dla aplikacji testowych wykorzystano platformę OpenShift [11] w wersji darmowej. Udostępnia ona kontener dla aplikacji opartych o JEE.

Dla każdej biblioteki osobno, w aplikacji testowej wykonano 10 pomiarów prędkości ładowania strony głównej. Wyniki zostały przedstawione w tabeli 6.

W celu lepszego zobrazowania wydajności aplikacji testowej na rysunku 3 przedstawiono wykres prezentujący średnie czasy pomiaru.

Tabela 6. Wyniki pomiarów szybkości ładowania strony głównej aplikacji

Numer pomiaru	JSF	PrimeFaces	RichFaces
1	0,75	1,35	2,05
2	0,67	1,33	1,53
3	0,63	1,57	3,81
4	0,69	1,5	1,79
5	0,63	1,31	1,87
6	1,44	1,19	1,4
7	0,85	1,27	1,39
8	0,69	1,27	1,39
9	1,04	1,22	1,58
10	0,83	1,2	1,69
Średni czas [s]	0,822	1,321	1,85



Rys. 3. Średnia szybkość ładowania strony głównej aplikacji testowej

Na wykresie wyraźnie widać, że aplikacja testowa z biblioteką JSF ładuje się najszybciej. Nieco wolniejszy czas ładowania został odnotowany przy testach z biblioteką PrimeFaces. Najwolniej strona ładowała się gdy została użyta biblioteka RichFaces.

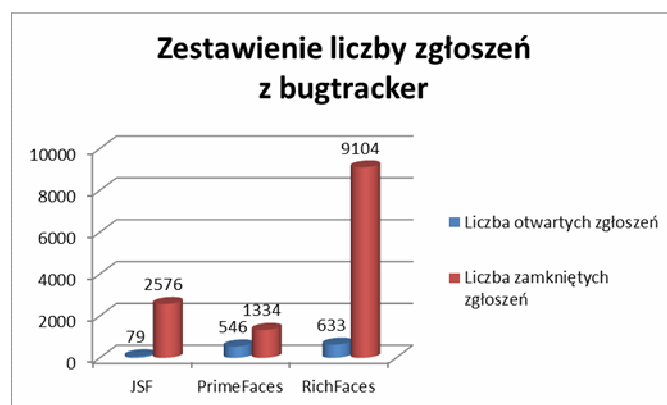
4.7. Popularność

W celu zbadania popularności poszczególnych bibliotek, zastosowano dwa kryteria. Jest to:

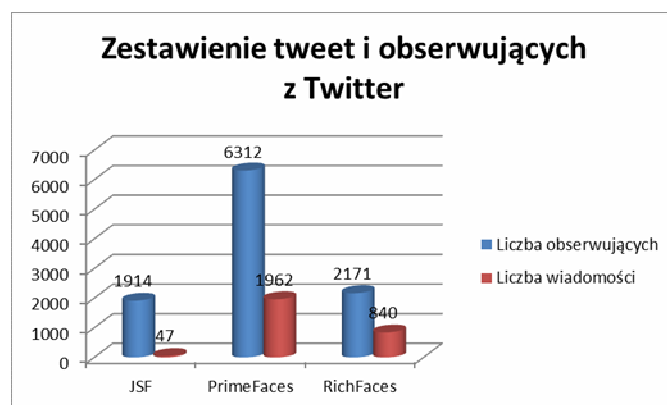
- liczba aktualnie otwartych zadań na odpowiednich platformach typu 'bugtracker' (Rys. 4);
- liczba osób obserwujących stronę projektu oraz liczba wiadomości na portalu społecznościowym Twitter (Rys. 5).

Odnośnie pierwszego kryterium, tylko JSF i RichFaces używają tej samej platformy, ale służy ona jedynie do raportowania zgłoszeń o wykrytych problemach. Darmowa wersja PrimeFaces posługuje się platformą, która jest repozytorium, a dodatkowo pozwala na tworzenie zgłoszeń.

Z przedstawionych danych można wywnioskować, że największą popularnością cieszy się PrimeFaces. Świadczy o tym także duże zainteresowanie na portalu Twitter. Z liczby zadań zgłoszonych na portalach typu 'bugtracker' wynika, że intensywne prace trwają nad bibliotekami PrimeFaces oraz RichFaces. JSF nie jest już tak intensywnie rozwijany, stąd liczba zgłoszeń nie jest duża.



Rys. 4. Zestawienie liczby zgłoszeń z bugtracker



Rys. 5. Zestawienie z portalu Twitter

4.8. Ocena bibliotek - podsumowanie

W tabeli 7 przedstawiono próbę oceny poszczególnych kryteriów w skali od 1 do 5. Nie stosowano wag dla poszczególnych kryteriów.

Tabela 7. Ocena bibliotek według wybranych kryteriów w skali 1 - 5

	JSF	PrimeFaces	RichFaces
Liczba komponentów	3	5	3
Łatwość implementacji	5	5	5
Podstawowe możliwości	5	4	3
Funkcje wyróżniające bibliotekę	3	5	3
Dokumentacja, wsparcie, społeczność	3	5	4
Wydajność	5	4	3
Popularność	3	5	4
Średnia ocena końcowa	3,86	4,71	3,57

Porównując wyniki z tabeli 7 widać, że najlepiej prezentuje się biblioteka PrimeFaces, następnie JavaServer Faces, na końcu zaś RichFaces.

5. Wnioski

Niewątpliwym atutem, który został zauważony podczas porównywania, jest to, że biblioteki nie wykluczają się nawzajem. W jednym projekcie istnieje możliwość używania komponentów z każdej analizowanej biblioteki. Pozwala to na ogromną elastyczność i dowolność stosowanych rozwiązań.

Z przeprowadzonej analizy wynika, że najlepszym wyborem do budowy w pełni funkcjonalnych aplikacji webowych z bogatym i przyjaznym interfejsem użytkownika jest połączenie możliwości JSF z PrimeFaces. JSF udostępnia podstawowe funkcje, niezbędne do utworzenia szkieletu strony internetowej, natomiast efektywniejszych wizualnie komponentów dostarcza PrimeFaces.

Jeśli programista musi się ograniczyć do jednego rozwiązania, to najlepszym wyborem jest JSF. Dwie pozostałe biblioteki nie oferują wszystkich podstawowych znaczników, bez których utworzenie kompletnej strony, nie jest możliwe.

Najmniej zachęcającą biblioteką okazała się RichFaces. O ile w pewnych aspektach (np. dokumentacja, wsparcie, funkcje wyróżniające) jest lepsza od JSF to jednak i tutaj PrimeFaces jest lepszym wyborem.

Literatura

- [1] <http://www.oracle.com/technetwork/java/javasee/overview/index.html>, [10.04.2017].
- [2] E. Jendrock, R. Cervera – Navaroo, I. Evans, D. Gollapudi, K. Haase, W. Markito, C. Srivathsa, "Java EE 6 Zaawansowany przewodnik", Wydanie IV, Helion, 2013.
- [3] K. Rychlicki – Kicior, "Java EE 6 Programowanie aplikacji WWW", Wydanie II, Helion, 2015.
- [4] <http://www.oracle.com/technetwork/java/javasee/jsp/index.html>, [10.04.2017].
- [5] <http://www.oracle.com/technetwork/java/javasee/javaxserverfaces-139869.html>, [10.04.2017].
- [6] Mojarra JavaServer Faces – Oracle's open source implementation of the JSF standard, <https://javaserverfaces.java.net/>, [20.02.2017].
- [7] PrimeFaces Ultimate UI Framework for Java EE, <http://www.primefaces.org/>, [20.02.2017].
- [8] RichFaces The next – generation JSF component framework by JBoss!, <http://richfaces.jboss.org/>, [20.02.2017].
- [9] K. Yank, „PHP i MySQL Witryna WWW oparta na bazie danych”, Wydanie III, Helion, 2005.
- [10] <http://www.pagespeeding.com/website-speed-test/>, [20.02.2017].
- [11] <https://www.openshift.com/>, platforma testowa dla aplikacji JEE, [20.02.2017].

Analiza porównawcza systemów wspomagających prototypowanie interfejsów

Artur Łasocha*, Marek Miłosz

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Analiza porównawcza narzędzi do prototypowania interfejsów użytkownika przeprowadzona została przy pomocy analizy wielokryterialnej. Przeprowadzone zostały testy wybranych narzędzi do prototypowania interfejsów. Rezultaty badań pozwoliły określić wartości zmiennych diagnostycznych, które posłużyły do wyboru najlepszego narzędzia.

Słowa kluczowe: interfejs oprogramowania; prototypowanie; analiza wielokryterialna

*Autor do korespondencji.

Adresy e-mail: artur@lasocha.pl, m.milosz@pollub.pl

Comparative analysis of support systems for interfaces prototyping

Artur Łasocha*, Marek Miłosz

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. Comparative analysis tools for prototyping of user interfaces was carried out using a multi-criteria analysis. They were carried out tests of selected tools for prototyping interfaces. The results of the research allowed to determine the values of diagnostic variables that were used to choose the best tool.

Keywords: software interface; prototyping; multi-criteria analysis

*Corresponding author.

E-mail addresses: artur@lasocha.pl, m.milosz@pollub.pl

1. Wstęp

Systemy informatyczne używane są przez człowieka praktycznie codziennie w działaniach prywatnych oraz zawodowych. Często wiąże się to z wielogodzinną pracą z oprogramowaniem lub innymi narzędziami informatyki. Wykorzystanie systemów informatycznych w obecnych czasach ułatwia wykonywanie codziennych zadań. Dostosowanie środowiska i narzędzi komputerowych do użytkownika zapewnia wzrost wydajności i jakości rezultatów pracy pracowników, czyli wzrost ich produktywności, a także wzrost efektywności w życiu codziennym.

Interfejs oprogramowania jest to narzędzie interakcji człowieka z komputerem. Dostosowanie interfejsów oprogramowania do mentalnych oraz fizycznych cech człowieka jest jednym z najistotniejszych celów projektowania systemów informatycznych. Interfejs oprogramowania skonstruowany w poprawny sposób umożliwia efektywnie wykorzystywać oprogramowanie oraz zmniejsza wysiłek człowieka w pracy z oprogramowaniem.

Ważnym elementem w cyklu wytwarzania oprogramowania, który wpływa na końcowy efekt odgrywa prototypowanie. Jest ono wykorzystywane w początkowej fazie wytwarzania oprogramowania. Prototypowanie polega na opracowaniu prototypu planowanego systemu lub jego części oraz zaprezentowaniu jego kolejnych wersji do oceny przez potencjalnych użytkowników.

Prototypowanie jest istotnym elementem projektowania iteracyjnego zorientowanego na użytkownika, gdzie osiągnięcie wysokich własności użytkowych produktu uzyskuje się poprzez stałą współpracę z potencjalnymi użytkownikami. Iteracyjność projektowania polega również na tym, że doskonalenie opracowanych rozwiązań następuje na podstawie wyników testów oraz opinii użytkowników, zaś wyniki testów wytyczają zakres oraz kierunek prac w etapie kolejnym.

Prototypami mogą być zarówno proste szkice rysunkowe – tzw. prototypy papierowe, jak i makiety samego interfejsu użytkownika, aż do niemal kompletnych rozwiązań obejmujących pełną funkcjonalność danego systemu [1].

2. Metoda analizy porównawczej

Przy wspomaganie procesu decyzyjnego, stosowana do oceny i wyboru najlepszego rozwiązania wśród analizowanych wariantów, wykorzystana została analiza wielokryterialna.

Istnieje wiele metod analizy porównawczej z wykorzystaniem wielu kryteriów – ich przegląd znajduje się w pracy [2]. Wybrane metody były stosowane do analizy porównawczej różnorodnych systemów informatycznych, takich jak: Framework php [3], serwisy bankowe [4], narzędzi RAD [5], a nawet gier komputerowych [6].

Przed przystąpieniem do analizy określone zostały zmienne diagnostyczne, czyli zestaw cech przedstawiających charakterystykę wybranych systemów. Dobór odpowiednich zmiennych diagnostycznych ma wpływ na wyniki badania, a ich nieodpowiednia selekcja może negatywnie wpłynąć na końcowe wyniki. Wyodrębnienie zmiennych diagnostycznych dla systemów do prototypowania interfejsów odbywa się w tym przypadku za pomocą kryteriów poza statystycznych, czyli merytorycznych oraz formalnych. Wybrane zostały cechy, które dla badanych systemów odgrywają ważną rolę do przeprowadzenia analizy porównawczej [7].

W analizie porównawczej zmienne traktować można w sposób współmierny lub zastosować wobec nich odpowiednie wagi różnicowe, których wartość stanowi o ważności danego kryterium. Wagi nie mogą mieć wartości ujemnych, a suma wartości zmiennych powinna być równa jeden.

Przeprowadzenie wielowymiarowej analizy porównawczej na wybranych zmiennych wiąże ze sobą konieczność dokonania transformacji w celu ujednolicenia badanych parametrów. Proces polegający na zmianie kryteriów zwany jest transformacją normalizacyjną. Jednym z wariantów normalizacji jest unitaryzacja, która polega na uzyskaniu zmiennych o znormalizowanym zakresie określonym przez różnicę pomiędzy wartościami maksymalnymi oraz minimalnymi. Wynikiem uzyskanej wartości jest zmienna w przedziale [0;1]. Ogólną formułę normalizacji prezentuje wzór:

$$Z_{ij} = \frac{x_{ij} - \min(x_{ij})}{\max(x_{ij}) - \min(x_{ij})} \quad (1)$$

gdzie: x_{ij} - oznacza obserwowaną wartość j-tej zmiennej w i-tym obiekcie, $\min(x_{ij})$ - oznacza minimalną wartość j-tej zmiennej w i-tym obiekcie, $\max(x_{ij})$ - oznacza maksymalną wartość j-tej zmiennej w i-tym obiekcie.

Następnie można przystąpić do uporządkowania zbioru poprzez zsumowanie wartości pośrednich w celu uzyskania nowego kryterium. Wartość nowego kryterium dla każdego z badanych elementów tworzone jest za pomocą wzoru 2, który dodatkowo uwzględnia wagę każdego kryterium [8]:

$$S_i = \sum_{j=1}^m Z_{ij} W_j \quad (2)$$

gdzie: Z_{ij} - oznacza znormalizowaną wartość kryterium cząstkowego j-tej zmiennej w i-tym obiekcie, W_j - oznacza wagę kryterium j-tej zmiennej.

3. Charakterystyka wybranych narzędzi do prototypowania interfejsów

Przy budowie prototypu interfejsu można skorzystać z szeregu dostępnych narzędzi, za pomocą których w prosty sposób można stworzyć makietę systemu. Do przeprowadzenia analizy porównawczej wybrane zostały cztery narzędzia wspomagające prototypowanie: Axure RP, Moqups, UXPin oraz Jumpchart.

3.1. Axure RP

Axure RP jest to narzędzie umożliwiające projektowanie prototypów dla aplikacji i stron internetowych. Narzędzie to ma bardzo szeroki wachlarz możliwości m.in. można tworzyć makiety systemu, prototyp aplikacji, diagram przepływu danych, czy też dokumentację. Dzięki polskiemu wsparciu jest łatwiejszy oraz przyjemniejszy w obsłudze. Dodatkową zaletą jest 30 dniowa wersja testowa, dzięki której zanim dokonamy zakupu możemy przetestować jego działanie.

Axure jest licencjonowane na rynku w dwóch pakietach: Standard i Axure RP PRO, których koszt wynosi odpowiednio \$289 oraz \$589 jednorazowo. Licencje są przypisane dla pojedynczej jednostki na dwóch komputerach oraz są bezterminowe. Produkt Pro udostępniany jest za darmo dla uczniów i nauczycieli.

Oprogramowanie jest raczej z średniej półki i można z niego korzystać na komputerach z systemem operacyjnym Windows oraz MacOS [9], ponieważ wymagana tu jest instalacja aplikacji na komputerze lokalnym, która łączy się z serwerem producenta. Niestety wersja, która została przetestowana była również w języku angielskim.

Okno aplikacji Axure jest podzielone na 7 głównych obszarów:

- Sitemap – hierarchiczna lista stron;
- Widgety;
- Masters (szablony wielokrotnego użytku, zbiory widжетów);
- Obszar projektowania;
- Uwagi strony i interakcje;
- Widget adnotacje i interakcje;
- Widget manager – wykaz wszystkich elementów na stronie.

Użytkownicy mogą tworzyć niestandardowe formanty łącząc istniejące widgety i przypisane działania w odpowiedzi na wydarzenia, takie jak: OnClick, OnMouseOver i onMouseOut lub gestów dotykowych. Na przykład panel interfejsu może mieć kilka różnych stanów, z których każdy jest aktywowany poprzez kliknięcie na elemencie takim jak przycisk zakładki.

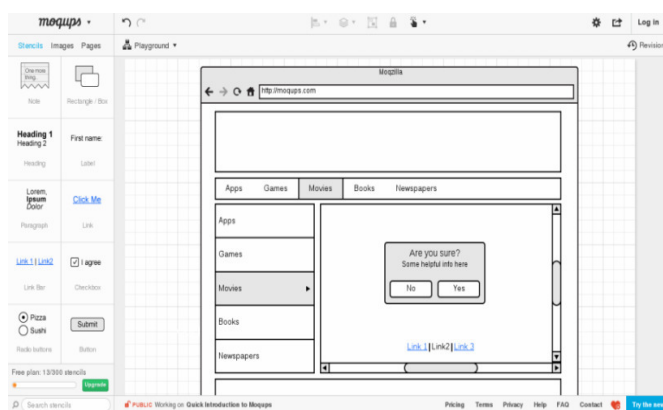
3.2. Moqups

Moqups jest to proste narzędzie do prototypowania interfejsów aplikacji i stron internetowych – rys. 1. Narzędzie jest udostępniane poprzez sieć Internet, dzięki czemu nie

trzeba instalować niczego na własnym komputerze lokalnym. Wystarczy jedynie dostęp do sieci Internet.

Po przejściu na stronę producenta <https://moqups.com/> znajduje się krótka, lecz prosta i zrozumiała instrukcja obsługi narzędzia. W sieci Internet dostępna jest darmowa wersja oprogramowania, która ma wiele ograniczeń w porównaniu z wersją płatną. Umożliwia ona jednak stworzenie prostego prototypu, np. aplikacji www.

Wersja płatna aplikacji jest licencjonowana w okresie miesięcznym oraz rocznym. W podstawowej wersji Standard jest do dyspozycji 1GB przestrzeni oraz można stworzyć do 10 projektów dla nieograniczonej ilości użytkowników. Cena za rok to 99 euro. Wersja Professional oferuje zaś 10GB przestrzeni dyskowej i możliwość tworzenia do 50 projektów za 199 euro rocznie. Najdroższy i największy pakiet Unlimited dostarcza 20GB przestrzeni oraz nieograniczoną ilość projektów w cenie 419 euro za rok [10].



Rys. 1. Prototyp aplikacji www – narzędzie Moqups

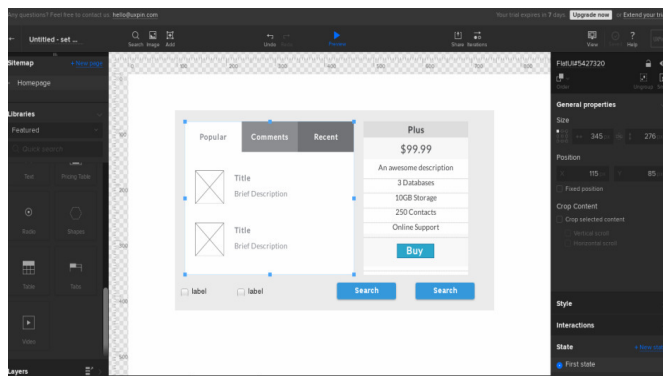
3.3. UXPin

UXPin pozwala tworzyć prototyp interfejsu stron www, jak i aplikacji na iphony, a także inne interfejsy. Dostępny jest poprzez sieć Internet, jak poprzednie narzędzie, więc nie jest zależny od posiadanego sprzętu i oprogramowania. Niewątpliwie jest to jego duża zaleta. Kolejną dość ciekawą zaletą jest możliwość testowania oprogramowania przez okres 7 dni bez żadnych opłat, co daje pogląd na testowane narzędzie. Cena narzędzia waha się od \$19 miesięcznie za licencję na nieograniczoną ilość projektów dla jednego użytkownika do \$98 miesięcznie za licencję na nieograniczoną ilość projektów dla 3 użytkowników. Dodatkowo można dokupić licencje dla każdego kolejnego użytkownika za kwotę \$49. Interfejs jest w języku angielskim, co niewątpliwie może stanowić utrudnienie dla niektórych użytkowników, ale ma bardzo prosty intuicyjny interfejs, który jest przejrzysty. Dzięki przejrzystości praca z tym oprogramowaniem jest dość przyjemna. Ciekawostką jest to, że UXPin jest to narzędzie polskiej firmy, która powstała w 2010 roku i z której produktu korzystają takie firmy jak Microsoft [11].

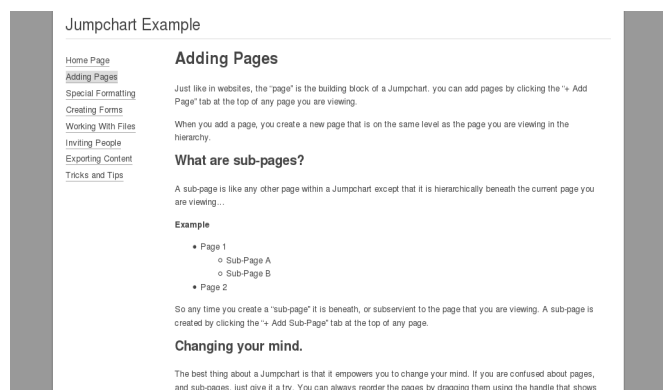
3.4. Jumpchart

Jumpchart służy do prototypowania interfejsów aplikacji oraz stron www. Narzędzie jest dostępne w sieci Internet, co sprawia że można korzystać z niego na każdym komputerze, który ma dostęp do sieci Internet – rys. 3. Niestety obsługa Jumpchart jest dość skomplikowana i nieintuicyjna, przez co używanie z tej aplikacji jest utrudnione. Pomoc odbywa się poprzez bazę wiedzy stworzoną przez producenta oprogramowania. Niestety jest ona w języku angielskim. Można również kontować się mailowo z pomocą techniczną, lecz ze względu na strefy czasowe oczekiwanie na odpowiedź może się wydłużyć.

Oprogramowanie posiada testową wersję, która jest darmowa oraz która oferuje możliwość stworzenia 1 projektu do 10MB i do 10 stron. Płatność może być tu realizowana miesięcznie, jak i rocznie, co daje większą elastyczność potencjalnym klientom. Najdroższa wersja oprogramowania oferuje stworzenie do 30 projektów na 100GB przestrzeni za kwotę \$400 rocznie [12].



Rys. 2. Narzędzie UXPin



Rys. 3. Narzędzie Jumpchart

4. Przedmiot analizy porównawczej i plan badań

Zagadnieniem poruszonym w danej pracy jest kwestia prototypowania oraz dedykowanych narzędzi do budowania prototypów interfejsów oprogramowania. Każde z analizowanych narzędzi posiada własny, indywidualny zestaw cech, które mogą wpływać na cały proces prototypowania, czy efekt finalny. Analizując narzędzia, można zweryfikować ich przydatność oraz wybrać odpowiednie oprogramowanie, które podoła wymaganiom projektu.

W pracy postawiona została następująca teza:

Właściwy dobór kryteriów oraz wag dokonany w trakcie analizy porównawczej jest istotny dla wyboru odpowiedniego narzędzia do budowy prototypu interfejsu na potrzeby konkretnego projektu.

Etapy analizy porównawczej w danej pracy obejmują następujące elementy:

- określenie kryterium analizy porównawczej dla badanych narzędzi do budowy prototypów interfejsu,
 - określenie parametrów technicznych,
 - określenie kryterium analizy jakości interfejsu oprogramowania,
- nadanie określonym kryteriom odpowiednich wag,
- przypisanie na podstawie własnych obserwacji poszczególnym cechom badanych narzędzi odpowiedniej ilości punktów,
- przedstawienie zsumowanych wyników dla badanych narzędzi na podstawie wzoru (2),
- zestawienie wyników uzyskanych z własnych obserwacji.

5. Analiza parametrów technicznych

Przy wyborze kryteriów analizy porównawczej zostały wzięte pod uwagę czynniki, które mogą wpłynąć na odpowiedni dobór oprogramowania do budowy prototypu interfejsu na potrzeby konkretnego projektu oraz które wpływają na efektywność przebiegu prototypowania. Parametry techniczne narzędzi przedstawione zostały w tab. 1. Poniższa tabela prezentuje również przypisane wagi, które wskazują, jakie parametry w większym lub mniejszym stopniu wpłyną na wyniki przeprowadzonej analizy, gdyż nie wszystkie parametry są tak samo istotne. Wagi zaproponowane oraz użyte w badaniu mają na celu jak najlepsze dopasowanie parametrów do analizy narzędzi, a tym samym określenie, które z analizowanych narzędzi jest najbardziej optymalne do budowy prototypu interfejsu.

Tabela 1. Parametry techniczne oprogramowania

Kryterium (waga)	Wskaźniki	Przypisane wagi
Dokumentacja (0,15)	Język angielski	0,5
	Dodatkowe języki >1	0,2
	Aktualność dokumentacji	0,3
Wsparcie techniczne (0,15)	Język angielski	0,3
	Dodatkowe języki >1	0,1
	Infolinia telefoniczna	0,2
	Chat	0,05
	Wsparcie e-mail	0,1
	Wsparcie 24 godziny na dobę	0,25
Uruchomienie na platformie (0,25)	Windows, MacOS	0,35
	On-line	0,65
Eksport prototypu (0,2)	HTML	0,5
	PDF, PNG	0,5
	Wszystkie	1
Budowanie prototypu na platformy mobilne (0,25)	Tak	1
	Nie	0

Ocena dokumentacji oraz wsparcia technicznego jest zadaniem dość trudnym, a jej wynik w dużym stopniu zależy od punktu widzenia oceniającego. W pracy zostały dołożone starania, aby jak najbardziej rzetelnie i obiektywnie ocenić czynniki takie jak wsparcie techniczne i dostępność dokumentacji, także w kontekście wielojęzyczności. Aktualna i dobrze przygotowana dokumentacja ułatwia pracę z narzędziem, a tym samym wpływa na efektywność procesu prototypowania. Ważną cechą wybraną do analizy porównawczej narzędzi jest również platforma, na której zostało uruchomione narzędzie. Do badań zostały więc wybrane między innymi narzędzia, które działają online, dzięki czemu nie są im straszne awarie sprzętowe oraz przestój w prototypowaniu. Wiele dostępnych narzędzi do prototypowania interfejsów rozpowszechniana jest z wbudowanymi narzędziami ułatwiającymi z ich korzystania. Niektóre narzędzia pozwalają na export projektów do jpg lub .png, inne zaś mają możliwość na odpalenie narzędzia w aplikacji mobilnej, co dzisiejszych czasach staje się podstawą każdego systemu do prototypowania interfejsu.

Dla każdego z analizowanych narzędzi do wybranych kryteriów przydzielone zostały punkty zgodnie z uwzględnionymi w tab. 1 kryteriami. System przydzielania punktów dokonany został na podstawie informacji zawartych na stronach internetowych producentów oprogramowania oraz na podstawie prób przeprowadzonych w trakcie przygotowania analizy. Zestawienie wyników zostało przedstawione w tab. 2.

Tabela 2. Analiza parametrów technicznych systemów do prototypowania interfejsów

Kryterium	Axure RP	Moqups	UXPin	Jumpchart
Dokumentacja	1	0,8	0,8	0,8
Wsparcie techniczne	0,4	0,4	0,6	0,4
Uruchomienie na platformie	0,35	0,65	0,65	0,65
Eksport prototypu	1	1	1	1
Budowanie prototypu na platformy	0	0	1	0
Suma wyników po normalizacji	0,5	0,54	0,82	0,54

6. Analiza jakości interfejsu oprogramowania

Tab. 3 prezentuje wybrane kryteria analizy jakości badanych systemów do prototypowania interfejsu. Tabela ta przedstawia również przypisane wagi, które wskazują, jakie parametry w większym lub mniejszym stopniu wpłyną na wyniki przeprowadzonej analizy. Istotnymi kryteriami, które otrzymały najwyższą wagę są: intuicyjność interfejsu i łatwość nauki jego obsługi. Ważnym kryterium jest również cena systemu, gdyż często od niej zależy decyzja o wyborze danego systemu. Wcześniejsza możliwość bezpłatnego testowania systemu daje pogląd na narzędzie, dzięki czemu można stwierdzić, czy jest ono odpowiednie do budowy prototypu interfejsu na potrzeby danego projektu.

Do ukazania jednoznacznych wyników analizy jakości interfejsu oprogramowania dla badanych systemów przypisane zostały punkty zgodnie z uwzględnionymi w tab. 3 parametrami. System naliczania punktów uwzględniał analizę każdego systemu pod kontem ustalonych kryteriów z uwzględnieniem poszczególnych progów punktowania. Punkty zostały przydzielone na podstawie własnych obserwacji z przeprowadzonych testów oprogramowania. Tab. 4 prezentuje przydzielone punkty oraz znormalizowaną sumę wyników.

Tabela 3. Kryteria analizy jakości systemów do prototypowania interfejsu

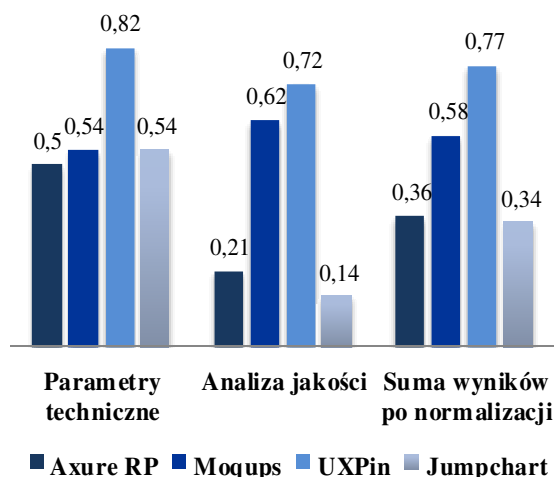
Kryterium (waga)	Wskaźniki	Przypisane wagi
Cena narzędzia (0,2)	Wysoka	0,16
	Średnia	0,22
	Wersja bezpłatna	0,37
	Darmowa wersja próbna	0,25
Intuicyjność interfejsu (0,3)	Duża	0,8
	Średnia	0,15
	Mała	0,05
Łatwość nauki obsługi interfejsu (0,25)	Łatwa	1
	Trudna	0
Stopień rozbudowy interfejsu (0,15)	Dodatkowe komponenty (>6)	0,65
	Dodatkowe komponenty (3-6)	0,3
	Dodatkowe komponenty (1-3)	0,05
Możliwość współdzielenia projektu z innymi użytkownikami (0,1)	Tak	1
	Nie	0

Tabela 4. Analiza jakości systemów do prototypowania interfejsu

Kryterium	Axure RP	Moqups	UXPin	Jumpchart
Cena narzędzia	0,47	0,59	0,41	0,41
Intuicyjność interfejsu	0,05	0,8	0,8	0,05
Łatwość nauki obsługi interfejsu	0	1	1	0
Stopień rozbudowy	0,65	0,05	0,3	0,3
Możliwość współdzielenia	0	0	1	0
Suma wyników po normalizacji	0,21	0,62	0,72	0,14

7. Rezultat analizy porównawczej narzędzi do prototypowania interfejsu

Oba główne kryteria objęte analizą (parametry techniczne i analiza jakości interfejsu oprogramowania) potraktowane zostały w sposób równoważny oraz przypisana została im jednakowa waga. Końcowy wynik obliczony na podstawie wzoru (2) przedstawia rys. 4.

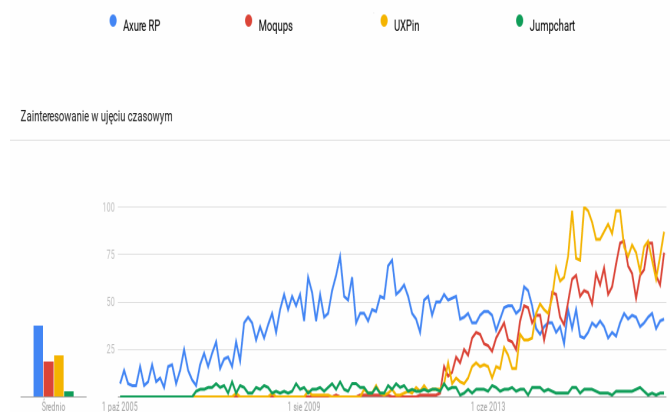


Rys. 4. Rezultat analizy porównawczej

Na podstawie przeprowadzonej analizy poszczególnych systemów pod względem parametrów technicznych i analizy jakości, oraz po zsumowaniu wyników, można jednoznacznie stwierdzić, odczytując z rys. 4, że narzędzie UXPin uzyskało najwyższą liczbę punktów. Głównym czynnikiem, który wpłynął na wynik analizy jest możliwość tworzenia prototypu na platformy mobilne. Drugim systemem, które otrzymało mniejszą liczbę punktów jest narzędzie Moqups. Narzędzie to jest bardzo łatwe w obsłudze i jako jedyne z analizowanych systemów posiada bezpłatną wersję, idealną przy tworzeniu prostych projektów. W przypadku tego narzędzia warto podkreślić, iż jego podstawowa wersja, która była analizowana, jest uboższa od wersji płatnej. W przypadku więc wzięcia pod uwagę płatnej wersji oprogramowania, która posiada większą ilość komponentów, narzędzie wypadłoby porównywalnie do oprogramowania UXPin.

Zgodnie z powyższym rysunkiem systemy Axure RP oraz Jumpchart otrzymały podobną ilość punktów. Oba systemy są mało intuicyjne, trudne w nauce obsługi. System Axure RP najgorzej wypadł pod względem cechy „uruchomienie na platformie”, gdyż jako jedyne do swojej obsługi wymaga instalacji na urządzeniach Windows oraz MacOS. Najlepiej jednak wypadł pod względem rozbudowy interfejsu, która w podstawowej wersji systemu posiada dużą ilość komponentów. Oprogramowanie Axure RP jako jedyne posiada również dokumentację oraz wsparcie techniczne w języku polskim, realizowane przez prywatnych pasjonatów tego oprogramowania. Narzędzie Jumpchart wypada zaś dość przeciętnie w przypadku wszystkich kryteriów, co w końcowym zestawieniu wyniku daje mu ostatnie miejsce.

Warto podkreślić, iż wyniki analizy badanych systemów do prototypowania interfejsów odzwierciedlają przedstawioną na rys. 5 popularność badanych systemów do prototypowania interfejsów w przeglądarce Google.



Rys. 5. Popularność badanych systemów do prototypowania interfejsów w przeglądarce Google

8. Wnioski

W pracy podjęto próbę analizy najistotniejszych cech systemów do prototypowania interfejsów oraz dokonano porównania wybranych narzędzi: Axure RP, Moqups, UXPin oraz Jumpchart. Zgodnie z przeprowadzonymi badaniami, a tym samym analizując badane aspekty, najlepiej w zestawieniu wypadło narzędzie UXPin. Posiada ono wiele ciekawych opcji, dzięki czemu staje się atrakcyjnym i najbardziej efektywnym narzędziem dla potencjalnego użytkownika.

Biorąc pod uwagę całokształt przeprowadzonej analizy, można przypuszczać stwierdzenie, że wybór odpowiedniego systemu zależy od rodzaju projektu oraz stawianych mu wymagań. Każdy nowo budowany prototyp wymaga indywidualnego podejścia w stosunku do procesu jego budowy. Stawiając odpowiednie warunki, możliwe jest wybranie właściwego systemu, który będzie w stanie sprostać oczekiwaniom projektu.

Postawioną w niniejszej pracy tezę „właściwy dobór kryteriów oraz wag dokonany w trakcie analizy porównawczej jest istotny dla wyboru odpowiedniego narzędzia do budowy prototypu interfejsu na potrzeby konkretnego projektu” można uznać więc za potwierdzoną.

Literatura

- [1] Sikorski M., Prototypowanie [online], [05.04.2016], Dostęp w serwisie HCI.pjwstk.edu.pl Koła Naukowego HCI – Human-Computer Interaction Polsko – Japońskiej Wyższej Szkoły Technik Komputerowych: <http://hci.pjwstk.edu.pl/index.php?page=prototypowanie>
- [2] Montusiewicz J., Wspomaganie procesów projektowania i planowania wytwarzania w budowie i eksploatacji maszyn metodami analizy wielokryterialnej. Politechnika Lubelska, 2012, s. 207.
- [3] Miłosz M., Zurkiewicz A., Selecting a php framework for a web application project – the method and case study. 9th International Technology, Education and Development Conference INTED, 2-4 March 2015, Madrid, Spain - 2015, pp. 1702-1712.
- [4] Osiński M., Miłosz M., Analiza porównawcza jakości serwisów internetowych polskich banków. Journal of Computer Sciences Institute, vol. 1, 2016, pp. 44-49.
- [5] Pasikova T., Miłosz E., Analiza porównawcza narzędzi RAD do wizualnego programowania w języku C++. Journal of Computer Sciences Institute, vol. 2, 2016, pp. 76-80.
- [6] Miłosz M., Dąbska A., New systematisation of computer games as a factor facilitating decision-making. Proceedings of ICERI 2014: 7th International Conference of Education, Research and Innovation, 2014, s. 1645-1654.
- [7] Cid-López A., Hornos M., Carrasco R., Herrera-Viedma E., Applying a linguistic multi-criteria decision-making model to the analysis of ICT suppliers' offers. Expert Systems with Applications, 57, 2016, 127-138.
- [8] Panek T., Statystyczne metody wielowymiarowej analizy porównawczej. Szkoła Główna Handlowa – Oficyna Wydawnicza, 2009.
- [9] Axure Software Solutions, Inc., [20.12.2015], <http://www.axure.com/buy>
- [10] S.C Evercoder Software S.R., Wireframe & UI Prototyping Tool • Moqups, [20.01.2017], <https://moqups.com/>
- [11] UXPin Sp. z o.o., UXPin - UX Design, Wireframing Tools, Prototyping Tools, [20.01.2017], <https://www.uxpin.com/>
- [12] Website Planning and Organization » Jumpchart, [05.05.2016], <https://jumpchart.com/>

Porównanie algorytmów wyszukiwania Google i Yandex

Oleh Bihun*, Marek Miłosz

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule przedstawiono rezultaty analizy porównawczej wyszukiwarek Google i Yandex. Analizę przeprowadzono pod kątem szybkości wyszukiwania oraz liczby znalezionych wyników. Przy pomocy metody kryteriów ważonych porównano Google i Yandex dla wyszukiwania w języku angielskim i rosyjskim. Analiza wielokryterialna pozwoliła wskazać wyszukiwarkę o szybszych i wydajniejszych algorytmach.

Słowa kluczowe: wyszukiwarka; porównanie; algorytmy wyszukiwania

* Autor do korespondencji.

Adres e-mail: oleh.bihun@pollub.edu.pl

Comparison Google's and Yandex's search algorithms

Oleh Bihun*, Marek Miłosz

^aInstitute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents the results of a comparative analysis Google's and Yandex's search algorithms. The analysis was made from the point of view of search speed and number of results. Google and Yandex were compared for search in Russian and English language, using the method of weighted criteria. Multi-criteria analysis allowed to find faster and efficient search engine.

Keywords: search engine; comparison; search algorithms

*Corresponding author.

E-mail address: oleh.bihun@pollub.edu.pl

1. Wstęp

Na dzień dzisiejszy zasoby sieci Internet zawierają około 10 miliardów stron [1], do których ma dostęp każdy użytkownik sieci. Zatem dla znalezienia potrzebnej informacji należy korzystać z zaawansowanych i nowoczesnych wyszukiwarek.

W przypadku dużej liczby informacji pojawia się problem jej wyszukiwania i filtrowania. W dzisiejszych czasach jest już nadmiar danych i im więcej ich pojawia się, tym trudniej coś znaleźć [2].

A z drugiej strony, obecnie nie można sobie wyobrazić współczesnego życia bez Internetu i bez wyszukiwarek, za pomocą których można odnaleźć konkretne informacje w tej prawie nieskończonej objętości danych. Internet jest obecnie wszędzie – i w laptopach i telefonach, ale kluczem do poszukiwania są wyszukiwarki.

Aktualność problemu można rozważać w kategorii konfrontacji pomiędzy dużą liczbą informacji w sieci a słabymi możliwościami szybkiego jej wyszukania [3]. Algorytmy wyszukiwarek, które pozwalają szybko uszeregować dane i wyświetlić użytkownikom potrzebne wyniki – odgrywają tu bardzo ważną rolę. Algorytm wyszukiwarki jest główną częścią, która gwarantuje prawidłowość wyświetlanych danych.

W ramach pracy zostały przeanalizowane dwie popularne wyszukiwarki – Google i Yandex.

2. Metody oceny wyszukiwarek

Istnieje wiele różnych metod analizy wielokryterialnej, przy czym niektóre z nich zostały opracowane już wiele lat temu, a niektóre powstały całkiem niedawno. Wśród nich można wyróżnić podstawowe techniki, które są dziś powszechnie znane i stosowane w nauce.

Metoda kryteriów ważonych (ang. *Weighting Metod*) [4] polega na transformacji funkcji wielowymiarowej na jednowymiarową, czyli na połączeniu wielu funkcji w jedną [4]. Otrzymaną w ten sposób funkcję optymalizuje się za pomocą standardowych metod optymalizacji funkcji jednowymiarowej. Główną wadą tej metody jest problem z doбором odpowiedniej wartości wag dla poszczególnych kryteriów. Dobór ten w znaczący sposób wpływa na jakość otrzymanego rozwiązania.

Metoda optymalizacji hierarchicznej (ang. *Hierarchical Optimization Metod*) [4] polega na zastosowaniu optymalizacji jednowymiarowej krok po kroku, z wykorzystaniem poszczególnych kryteriów.

Pomysł, zastosowany w algorytmie VEGA (VEGA: Vector Evaluated Genetic Algorithm) [5], polega z kolei na podziale kategorii na podkategorie o tej samej liczbie elementów. Selekcja w obrębie każdej kategorii odbywa się niezależnie (jedna podkategoria jest odpowiedzialna za konkretne kryteria).

3. Obiekt badań

Do analizy porównawczej wybrano dwie popularne wyszukiwarki – Google i Yandex. Google jest popularna na całym świecie i posiada największy udział w rynku [6].

Yandex - to popularna wyszukiwarka w Rosji i bardzo znana wśród osób korzystających z języka rosyjskiego [7]. Yandex jest popularna w takich państwach jak: Rosja, Ukraina, Kazachstan, Białoruś. By opracować metodę porównywania wyszukiwarek, trzeba poznać ogólną zasadę ich działania.

Wyszukiwarka ma skomplikowaną strukturę, ale oprócz web-serwera czy baz danych, główną częścią wyszukiwarek jest robot. Robot zajmuje się poszukiwaniem stron, pobieraniem ich do bazy danych i analizą [8]. Działa on za pomocą określonych algorytmów, które pozwalają mu analizować i szeregować dane [6]. To właśnie algorytmy odpowiedzialne za szeregowanie są obiektem badań [9]. Celem niniejszej pracy jest przeanalizowanie tych algorytmów oraz porównanie ich szybkości i wydajności.

4. Metoda badawcza

Główna teza pracy zawiera się w stwierdzeniu, że Google lepiej wyszukuje informacje w języku angielskim, a Yandex - w języku rosyjskim. Dlatego też zostały sformułowane dwie hipotezy badawcze:

H1: Google zwraca lepsze wyniki niż Yandex w języku angielskim.

H2: Yandex zwraca lepsze wyniki niż Google w języku rosyjskim.

Trzeba zauważyć, że trudno jest obiektywnie zmierzyć jakość wyszukiwania, czy jakość działania algorytmów [10]. Jest to związane przede wszystkim z niedostępnością danych o tych algorytmach – nie jest to bowiem informacja publiczna [11]. W związku z tym została opracowana metoda, która pozwoli na porównanie wyszukiwarek za pomocą konkretnych parametrów [12]. Trzeba zauważyć, że badania zostały przeprowadzone przy użyciu próbki losowej danych do zapytań, co pozwoliło w przejrzysty sposób ocenić i porównać systemy wyszukiwania.

Do porównania wybrano dwa kryteria, które zostały podzielone na dwie kategorie. W etapie finalnym wyszukiwarki są porównane za pomocą kryterium końcowego. Te kategorie to:

- 1) Jakość wyszukiwania w języku angielskim.
- 2) Jakość wyszukiwania w języku rosyjskim.

W etapie finalnym, kryterium końcowe powstaje jako suma powyższych. Wychodzi się bowiem z założenia, że każda z wyżej wymienionych kategorii w równym stopniu wpływa na ocenę końcową.

Analizując kierunek pracy podjęto decyzję o korzystaniu z uproszczonej metody ważonych kryteriów. Ocena przy pomocy metody ważonych kryteriów została podzielona na dwa etapy.

Na pierwszym etapie następuje przyznanie "wagi" dla określonych kryteriów lub kategorii. W ramach danej pracy można dołączyć określone wagi dla żądań, na przykład z różną częstotliwością. Celem pracy jest porównanie wyszukiwarek w aspekcie wyszukiwania dla każdego rodzaju zapytania, biorąc pod uwagę znaczenie każdego zapytania w zależności od użytkownika. Dlatego brano pod uwagę, że zapytania niskiej częstotliwości są tak samo ważne dla

użytkownika, jak i wysokiej częstotliwości. Zakłada się, że nawet rzadko zadawane pytania będą mieć znaczenie dla użytkownika. Na tym etapie przyznano jednakowe "wagi" dla wszystkich kategorii.

Na etapie drugim następuje obliczenie kompleksowych ocen. Ponieważ zostały przyjęte jednakowe "wagi" dla wszystkich kategorii, to obliczenia są realizowane według następującego wzoru [4]:

$$P = \sum_{k=1}^Q \left(\sum_{i=1}^N y_{ki} \frac{100}{N * m_k} \right) \quad (1)$$

gdzie:

- Q – liczba kategorii,
- N – liczba kryteriów w danej kategorii,
- y_{ki} – wartości i -tego kryterium w k -tej kategorii,
- m_k – maksymalna liczba punktów za kryterium.

5. Kryteria oceny wyszukiwarek

Kryteria oceny wyszukiwarek zostały dobrane tak, aby na podstawie uzyskanych w wyniku eksperymentu danych, można było w łatwy i obiektywny sposób je porównać.

Dla badania wyszukiwarek wykorzystano 500 najpopularniejszych słów kluczowych dla Google i Yandex. Przykładowe frazy prezentuje tabela 1.

Tabela 1. Frazy zastosowane do analizy wyszukiwarek

Frazy w języku angielskim	Frazy w języku rosyjskim
Hotels	Вк
Facebook	Авито
Facebook login	Фильмы онлайн
Amazon.com	Рамблер почта
Google earth	Яндекс почта
Scorpio	Русская весна
The hunsyellow pages	Телепрограмма
Map quest	Школьный портал

Ponieważ celem pracy było zbadanie aktualnych algorytmów, wykorzystano parametry, które są udostępniane przez same wyszukiwarki. To wyszukiwarka wyświetla użytkownikom czas i liczbę wyników. Te właśnie parametry stały się też kryteriami do oceny wyszukiwarek.

Jakość wyszukiwania w języku angielskim

Ta kategoria odpowiada za ocenę jakości wyszukiwania przy wprowadzaniu do wyszukiwarki słów kluczowych w języku angielskim. W celu oceny i opisu tej kategorii skorzystano z dwóch kryteriów:

- Czas wyszukiwania.
- Liczba zwracanych wyników.

Jakość wyszukiwania w języku rosyjskim

Ta kategoria odpowiada za ocenę jakości wyszukiwania przy wprowadzaniu słów kluczowych w języku rosyjskim. W celu oceny i opisu tej kategorii - skorzystano z tych samych kryteriów co w języku angielskim.

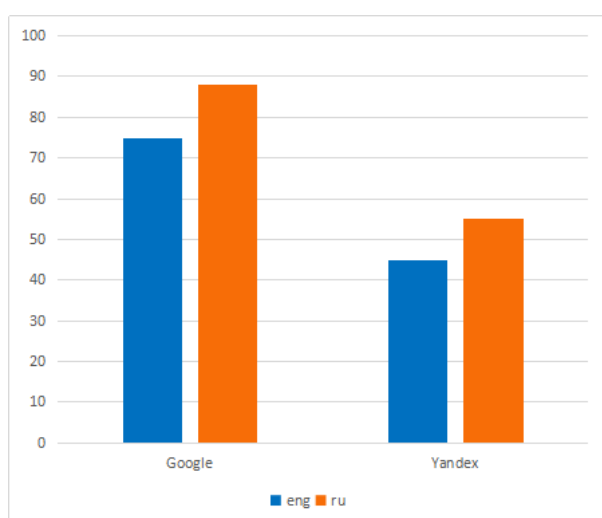
6. Rezultaty badań

Rezultaty badań można zaprezentować w postaci tabeli 2. Tabela prezentuje średni czas wyszukiwania t_{avg} oraz średnią liczbę zwracanych wyników n_{avg} dla wyszukiwania w języku angielskim (eng) i w języku rosyjskim (ru).

Tabela 2. Rezultaty badań wyszukiwarek

	Google		Yandex	
	eng	ru	eng	ru
$t_{avg}(sek)$	0.75	0.55	1.04	1.02
$n_{avg}(mln)$	326.6	572.9	40.8	171

Na podstawie tabeli 2 można stwierdzić, że Google ma lepsze parametry niż Yandex, co graficznie zaprezentowano w postaci diagramu – rys. 1.



Rys. 1. Porównanie Google i Yandex w kategoriach wyszukiwania w języku angielskim (eng) i języku rosyjskim (ru)

Google uzyskał lepszą ocenę niż Yandex w obydwu kategoriach. Rezultatem badań jest fakt, że wyszukiwarka Google zwraca więcej wyników w krótszym czasie podczas wyszukiwania zarówno w języku angielskim jak i rosyjskim.

Wynik końcowy prezentuje tabela 3.

Tabela 3. Ranking badanych wyszukiwarek

Pozycja	Wyszukiwarka	Kryterium końcowe
1	Google	81,5
2	Yandex	50

7. Wnioski

W ramach pracy przebadano i porównano parametry dla dwóch języków wyszukiwania, co pozwoliło sformułować wniosek, która wyszukiwarka zwraca lepsze wyniki i dla jakiego języka wyszukiwania.

Z otrzymanych wyników jednoznacznie można stwierdzić, że algorytmy wyszukiwarki Google są lepsze niż Yandex. Zatem, hipoteza H1 potwierdziła się, a hipoteza H2 – nie. Yandex zwraca średnio mniej wyników w dłuższym czasie.

Google korzysta z bardziej nowoczesnych i zaawansowanych algorytmów, niż Yandex, które pozwalają na szybsze wyszukiwanie informacji i zwracają większą liczbę wyników. Yandex jest w fazie ciągłego rozwoju,

również korzysta z nowoczesnych technologii, ale algorytmy przez nią stosowane, na chwilę obecną nie dorównują wyszukiwarce Google. Przyczyną tego mogą też być większe moce obliczeniowe Google (tj. lepsza warstwa techniczna wyszukiwarki). Dokonana analiza potwierdza tezę o bardzo zaawansowanych technologiach, stosowanych w Google.

W pracy zostały przeanalizowane tylko dwie wyszukiwarki. Ciekawym kierunkiem dalszych badań może być wskazanie innej metody oceny wyszukiwarek i ich algorytmów. Można też wybrać inne języki do porównania oraz inne wyszukiwarki, np. DuckDuckGo czy Ecosia.

Literatura

- [1] С. Симонович, Информатика. Базовый курс, СПб: «Питер» (2007), 85-88.
- [2] К. Маннинг, Введение в информационный поиск, Вильямс (2011), 115-117.
- [3] В. Белозеров, Эффективность систем Гугл и Яндекс для поиска учебного материала (2015), 254-255.
- [4] T. Borek, K. Cichosz, Wprowadzenie do optymalizacji wielokryterialnej, Akademia górniczo-hutnicza, Kraków, 2017
- [5] T. Lachowicz, Optymalizacja wielokryterialna decyzji w zagadnieniach bezpieczeństwa funkcjonowania podmiotu, Studia Ekonomiczne, Uniwersytet Ekonomiczny w Katowicach, no. 235 (2015), 144-145.
- [6] E. Schmidt, J. Rosenberg, How Google Works, 2014.
- [7] www.spravki.se-ua.net/yandex, Как работает яндекс [28.04.2017]
- [8] Z. Joyner, Slaying the Search Engine Dragon: Smarter, More Efficient Internet Searches, University of Georgia School of Law (2016), 11-15.
- [9] R. Meusel, P. Mika, R. Blanco, Focused Crawling for Structured Data, Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (2014), 1039-1048.
- [10] D. Nandy, H. Gohel, F. Kalyani, M. Khakhariya, Duckduckgo vs. Google search engine – comparative analysis, GRD Journals, Volume 2, Issue 1, 2016.
- [11] P. Chen, H. Xie, S. Maslov, S. Redner, Finding scientific gems with Google's PageRank algorithm, Journal of Informetrics, Boston, 2007

Porównanie możliwości i cech współczesnych Linuxowych systemów plików: ext4, XFS, Btrfs

Bartłomiej Kossak*, Maciej Pańczyk*

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule zostały porównane trzy nowoczesne systemy plików powszechnie stosowanych w systemach Linux: ext4, Btrfs i XFS. Została omówiona budowa i cechy funkcjonalne poszczególnych systemów plików. Systemy plików zostały porównane pod względem wydajności. Uwzględniono specyfikę dysków SSD i przystosowanie do pracy z nimi. Do testów funkcjonalności oraz wydajności został użyty system Ubuntu 15.10

Słowa kluczowe: system plików; ext4; btrfs; xfs

* Autor do korespondencji

Adresy e-mail: Bartek.kossak@gmail.com;m.panczyk@pollub.pl

Comparison of features and capabilities of modern Linux file systems ext4, XFS, Btrfs

Bartłomiej Kossak*, Maciej Pańczyk*

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. This article compares three modern Linux file systems: ext4, Btrfs and XFS. Their construction and functional features were discussed. File systems were compared in terms of performance. Additionally modern SSD drives were included in discussion. Ubuntu 15.10 distribution was used to test functionality and performance of these three Linux file systems.

Keywords: file system; ext4; btrfs; xfs

*Corresponding author

E-mail addresses: Bartek.kossak@gmail.com; m.panczyk@pollub.pl

1. Wstęp

System plików służy do kontroli sposobu zapisu i odczytu danych. Tworzy on warstwę, która pozwala operować na nazwach wszystkim powłokom znajdującym się nad nią. Zaś wszystkim warstwom znajdującym się pod nią operować na adresach fizycznych [1].

Prawie wszystkie systemy operacyjne posługują się systemami plików. W przypadku najbardziej popularnych systemów operacyjnych, takich jak Windows, Linux, czy MacOS mamy do wyboru po wiele systemów plików, w zależności od potrzeb. Znaczący rozwój technologii informatycznej na przestrzeni ostatnich lat zrodził dużą ilość problemów, z którą systemy plików muszą sobie poradzić. W systemach Linux wszystko tworzone jest w formie plików. Plik to zbiór powiązanych ze sobą informacji. Użytkownik widzi jedynie nazwę pliku, natomiast system Linux traktuje plik jako ciąg bajtów. Dobór systemu plików zależy od zastosowania. Inny będzie właściwy dla systemów o wysokich wymaganiach ciągłej i bezawaryjnej pracy, inny dla systemów, w których krytyczna jest wydajność a jeszcze inny dla systemów, w których istotne są możliwości migracji bądź zmiany pojemności w locie (jak np. w przypadku maszyn wirtualnych).

W związku ze znaczącym rozwojem technologii wzrosły wymagania związane z bezpieczeństwem danych oraz wydajnością systemów informatycznych. Zmiana nośników danych z magnetycznych na technologie półprzewodnikową spowodowała wzrost zapotrzebowania na całkiem nowe cechy systemów plików.

Linux oferuje szeroką paletę systemów plików. W artykule zostaną porównane trzy najpopularniejsze: ext4, XFS i Btrfs. Ext4 jest to jeden z najpopularniejszych systemów plików domyślny w wielu dystrybucjach Linux. System plików ext4 jest kontynuacją systemu plików ext, który był pierwszym stworzonym od podstaw systemem plików dla Linuxa [2]. XFS jest dostępny w Linuxie od dwudziestu lat, jest stale rozwijany przez duże koncerny informatyczne, pozwala na obsługę bardzo dużych plików i wolumenów [3]. XFS można uznać za współczesny system plików dla wydajnego sposobu radzenia sobie z dyskami w erze „big data”. Btrfs to najmłodszy system plików, który jest oferowany w Linuxie [4]. Jest stale rozwijany, posiada wiele nowych cech np. optymalizacyjnych, czy zabezpieczających, a wiele z nich jest jeszcze w fazie testów [4].

2. System plików

System plików odpowiada za porządkowanie, budowanie hierarchii oraz tworzenie struktury porządkującej plików.

Ponadto realizuje operacje wymagane do działania systemu operacyjnego m.in.:

- tworzy i usuwa pliki;
- otwiera pliki do odczytu i zapisu;
- wyszukuje dane w plikach;
- zamyka pliki;
- tworzy katalogi;
- listuje zawartość katalogów;
- usuwa pliki z katalogów [1].

Linux niemal od samego początku, z powodów historycznych (odziedziczona różnorodność wersji i producentów systemów UNIX) miał do dyspozycji więcej niż jeden system plików. Do ułatwienia implementacji nowych systemów plików stworzona została warstwa abstrakcji, nazwana VFS (Virtual File System). Warstwa VFS to interfejs łączący jądro i implementację systemu plików. Dzięki tej warstwie operacje zapisu lub odczytu z punktu widzenia programisty są takie same. Zadaniem VFS jest przetłumaczenie podstawowych poleceń na polecenia zgodne ze specyfikacją systemu plików [1].

Z plikami w systemach UNIX związane są tzw. i-węzły. Każdy plik w systemie Linux posiada swój własny i-węzeł. W każdym i-węźle znajdują się elementy struktury opisującej plik (z wyjątkiem danych i nazwy) oraz wskaźniki do bloków danych [1].

Za obszarem rozruchowym (może on być ew. pusty) znajduje się superblok zawierający atrybuty i metadane danego systemu plików [1]. Ostatnim z podstawowych elementów jest obszar systemu plików zawierający bloki danych.

3. EXT4

System ext4 jest kontynuacją systemu plików z rodziny ext. System ext jest pierwszą wersją najpopularniejszego systemu plików dostępnego w Linuxie.

W ext nazwa pliku wynosiła 255 znaków, maksymalny rozmiar partycji i pliku wynosił 2GB [2]. System posiadał jednak dużą wadę - użycie listowego przydziału bloków prowadziło do występowania niekorzystnego zjawiska jakim była nieciągłość obszarów danych na dysku twardym, czyli fragmentacja [1].

W ext2 sunięty został problem związany z fragmentacją plików, poprzez zastosowanie mechanizmu przydzielania bliskich bloków oraz prelokację [1,2]. Lokalizacja bloków z danymi odbywa się na dwa sposoby: bezpośrednio - adresy pierwszych bloków są zapisane bezpośrednio w i-węźle - oraz pośrednio - adresy bloków zapisane są w bloku, którego adres znajduje się w i-węźle [1]. Wprowadzony został również mechanizm naprawy systemu plików po awarii (fsck), który uruchamiany jest przy starcie systemu - naprawia szkody poprzez przywracanie uszkodzonych plików do odpowiedniego katalogu [5]. Skanowanie było mniej wydajne niż w przypadku następnych systemów plików z księgowaniem.

System plików ext3 nie różnił się budową od ext2. Najważniejszym elementem, który został wprowadzony do ext3 był mechanizm księgowania [6]. Księgowanie

zabezpiecza pliki przed utratą lub uszkodzeniem i polega na zapisaniu metadanych - informacji o danej - na początku do dziennika, a następnie zapisaniu danych na dysku. W systemie plików ext3 mamy do wyboru trzy tryby księgowania:

- journal - najbezpieczniejszy i najwolniejszy tryb tworzenia kroniki, tworzone są dane i meta dane. W tym sposobie występuje najmniejsza szansa na stratę danych ale wydłuża czas pracy dysku, ponieważ w trakcie tworzenia nowego pliku bloki danych muszą być zapisane w dzienniku dwa razy.
- ordered - najpopularniejszy tryb księgowania, w którym zapisywane są jedynie meta dane. Tryb zagwarantuje integralność systemu plików lecz w końcowych częściach plików mogą wystąpić nieprawidłowości w danych.
- writeback - do dziennika zapisywane są jedynie meta dane, operacja zapisu dokonywana jest zgodnie z harmonogramem, zapisywane są wtedy dane, a następnie metadane. Jest to tryb najszybszy lecz mniej bezpieczny, zezwala na modyfikację danych, których metadane zostały już zmodyfikowane, a dane nie zostały zapisane jeszcze na dysk [5].

Nową cechą wprowadzoną do ext3 jest indeksowanie katalogów. W ext3 odpowiada za nią funkcja hashdrees, dzięki której katalogi mają postać drzewa [5]. Katalogowanie jest potrzebne w przypadku posiadania dużej ilości plików w katalogach. Drzewiasta struktura znacznie skraca czas dostępu do szukanego pliku. System ext3 dopuszcza podział bloku na równe części, w których jest możliwość przechowywania plików, dzięki temu zmniejsza fragmentację w przypadku dużych ilości małych plików.

W systemie plików ext4 wielkość pojedynczego pliku została zwiększona do 16 TiB, oraz obsługa pojedynczych partycji do 1 EiB, liczba możliwych podkatalogów została zwiększona do 64 tysięcy [5].

Główną cechą różniącą ext4 od starszych wersji ext jest mechanizm ekstentów [5], który zastępuje adresowanie pośrednie bloków. Ekstent jest to ciągły zbiór bloków, w którym przechowywane są pliki, ekstenty przechowywane są w i-węźle zamiast wskaźników do bloków [5]. Dzięki ekstentom zwiększył się rozmiar pojedynczego pliku, zwiększona została wielkość pojedynczej partycji i zredukowany jest rozmiar meta danych. Redukcja wielkości meta danych znacznie przyspiesza prace na dużych plikach. Włączenie usługi ekstents umożliwia realizację prelokacji bloków, który odpowiada za ciągły przydział obszaru dla pliku.

Nową cechą ext4 jest opóźniona alokacja [6], technika ta opóźnia zapis danych na dysku przez przetrzymywanie ich w pamięci podręcznej. Opóźniona alokacja odbywa się w warstwie wirtualnej systemu plików. Rozszerzeniem opóźnionej alokacji jest alokacja wieloblokowa (mballoc). Zaletami tego rozwiązania jest zwiększenie wydajności w przypadku zapisu plików rosnących - połączenie jednoczesnej i wieloblokowej alokacji zwiększa szanse na to, że kolejne alokacje zgrupują się w jedną. Następną zaletą jest zmniejszenie fragmentacji danych dla plików rosnących, oraz duża szansa na brak alokacji dla plików tymczasowych [2].

Rozwiązanie posiada również wady, zwiększone jest ryzyko utraty danych podczas awarii zasilania.

W ext4 został usprawniony również mechanizm księgowania dzięki dodaniu sum kontrolnych dziennika [5]. Dziennik jest najintensywniej używanym obszarem na dysku, z tego powodu uszkodzenia w tym obszarze mogą doprowadzić do awarii całego systemu plików. Suma kontrolna to ciąg znaków, powstających w wyniku operacji matematycznych podczas przesyłania danych. W przypadku systemu plików, suma kontrolna wyliczana jest dla każdej transakcji i deskryptora grupy bloków [5]. Nową cechą, która została wprowadzona jest mechanizm barier. Odpowiada za wydawanie poleceń sterownikowi dysku zapisu danych w określonym porządku, co wzmacnia spójność danych na dysku [5].

4. XFS

XFS to wysokowydajny 64-bitowy system plików z obsługą księgowania. Obsługiwany jest w większości systemów operacyjnych Linux.

Projektując system plików XFS przyjęto następujące cele [1]:

- system plików powinien nadawać się do pracy w naukowych serwerach plików, komercyjnych serwerach obróbki danych oraz w serwerach mediów elektronicznych;
- musi odzyskać szybko sprawność po awarii, zapewnić dużą dostępność danych, utrzymywać dane dyskowe w spójnym stanie;
- musi obsługiwać 64-bitowe pliki;
- musi obsługiwać wydajnie pliki rozrzedzone – czyli pliki, które w dowolnym miejscu zawierają dziury (obszary, w których zapis nigdy się nie odbył, odczytywane są jako bloki zer);
- zwiększona wydajność dla obsługi małych plików, mniejszych niż 1KB;
- musi wydajnie obsługiwać duże katalogi;
- obsługa listy kontroli dostępu oraz funkcji POSIX 1003.6 – rozszerzenia zabezpieczeń systemu spełniające kryteria opublikowane przez Departament obrony USA;
- umożliwienie dynamicznej zmiany logicznych rozmiarów bloków [1,3].

System plików obsługuje pliki o wielkości 8 EiB oraz maksymalną wielkość partycji do 16 EiB. Długość nazwy pliku reprezentowana jest za pomocą ciągu 255 znaków z tablicy ASCII [18]. Rozmiar jednostki alokacji wynosi od 512KB do 1MB. Jak wszystkie systemy plików, XFS również pracuje pod kontrolą VFS. XFS jest dziennikowym systemem plików, co oznacza, że aktualizacja meta danych odbywa się szeregowo w obszarze dziennika zanim nastąpi rzeczywista aktualizacja [1]. W przypadku awarii, operacje zapisu można powtórzyć lub cofnąć za pomocą danych do przywrócenia systemu plików zawartych w dzienniku. Głównymi komponentami systemu XFS są:

- menedżer dziennika – wszelkie zmiany w meta danych są rejestrowane szeregowo w specjalnie wyszczególnionym do tego obszarze. Każdy system plików prowadzi

oddzielny dziennik, dzięki któremu możliwa jest szybka rekonstrukcja wadliwego systemu plików.

- menedżer pamięci podręcznej buforów – w pamięci podręcznej buforu przechowywane są bloki dysków, przechowuje się tu meta dane systemu pliku i dane plików
- menedżer rygli – realizuje ryglowanie plików użytkownika
- menedżer przestrzeni dyskowej – rozporządza alokacją przestrzeni dyskowej w systemie plików
- menedżer atrybutów – realizuje operacje na atrybutach systemów plików
- interfejs wywołań systemowych i VFS
- menadżer przestrzeni nazw – tłumaczy ścieżki dostępu na odwołania do plików [1].

System plików XFS jest podzielony na grupy alokacji jednakowej wielkości. Grupa alokacji jest to autonomiczna jednostka systemu plików, która zawiera osobne struktury potrzebne do zarządzania jej przestrzenią [1].

Grupy alokacji składają się z superbloku umieszczonego w lokalizacji 0, nagłówka grupy alokacji oraz danych wskazanych przez nagłówek grupy alokacji [1]. Nagłówek grupy alokacji składa się z następujących pól:

- magicznego numeru nagłówka alokacji;
- numeru wersji nagłówka grupy alokacji;
- numeru sekwencyjnego grupy alokacji;
- w zależności od używanej alokacji, jeśli jest alokacja z bitmapą – położenie oraz rozmiar bitmapy wolnych bloków, w przypadku alokacji z dwoma B-drzewami [33] – położenie korzenia obu B-Drzew;
- położenia (względnego numeru bloku) i-węzła, który zawiera tablicę i-węzłów [1].

Przestrzeń magazynowa plików w XFS reprezentowana jest na trzy sposoby [1]. Wybór metody zależy od rozmiaru i ciągłości pliku. Dla plików małych, dane przetrzymywane są w i-węźle. W przypadku plików średnich wykorzystywane są ekstenty (i-węzeł zawiera wskaźnik do danego ekstentu, który zawiera dane pliku). W przypadku plików dużych stosowane są B-Drzewa [3].

System XFS można aktywnie rozszerzać poprzez dodanie przestrzeni do bazowego woluminu, operację taką przeprowadza menadżer przestrzeni dyskowej [3]. XFS obsługuje mechanizm opóźnionej alokacji, który podczas operacji zapisu plik jest zapisywany do bufora pamięci podręcznej [3]. Dzięki tej metodzie zwiększona jest szansa, że plik zostanie zapisany w ciągłej grupie bloków, co zmniejsza problemy z fragmentacją i wydajnością.

System XFS obsługuje tzw. pliki rozrzedzone [1,3]. Plik rozrzedzony to typ pliku, który zezwala zapisywać dane w dowolnym miejscu bez zajmowania niezapisanych bloków pliku.

Dla zwiększenia wydajności XFS udostępnia bezpośrednią realizację operacji we/wy. XFS pozwala na niebuforowanie we/wy, ponieważ dane przekazywane są poprzez bufor aplikacji i dysku używającego DMA (umożliwiający zapis pełnego pasma we/wyz podstawowych urządzeń).

Cechą unikalną systemu XFS jest gwarantowana przepustowość operacji we/wy, za rezerwację odpowiedniej przepustowości odpowiada interfejs API [3]. Wykorzystywane jest to głównie do zastosowań w czasie rzeczywistym np. strumieniowe transmisje wideo.

XFS zapewnia również defragmentację plików podczas pracy, tzw. defragmentacja online oraz zmianę rozmiaru online, dzięki czemu system plików może zwiększać swój rozmiar tak długo jak jest dostępne wolne miejsce na urządzeniach fizycznych [3].

XFS obsługuje również mechanizm księgowania, który ma za zadanie zapewnienie spójności danych w przypadku awarii zasilania lub systemu [1,3]. System XFS zapewnia księgowanie meta danych, których zapis odbywa się w pierwszej kolejności przed tym, jak zostaną zaktualizowane bloki dysku. Pliki dziennika systemu są zapisywane w odrębnej części dysku, która nie jest wykorzystywana podczas normalnej pracy systemu. W XFS możemy wyróżnić dwa rodzaje dzienników: logiczny i fizyczny [1]. Logiczny opisuje w sposób zrozumiały dla człowieka jakie czynności zostały wykonywane, natomiast dziennik fizyczny przechowuje kopie bloków zmodyfikowanych podczas każdej operacji. Aktualizacje dzienników wykonywane są asynchronicznie, dzięki temu zwiększona jest wydajność systemu plików [1]. W przypadku awarii XFS potrafi odzyskać pliki, które nie zostały zapisane zgodnie z informacjami zawartymi w dzienniku. Odzyskiwanie odbywa się automatycznie w trakcie pierwszego uruchomienia systemu plików po awarii, szybkość odzyskiwania nie zależy od wielkości plików, a od operacji w nich dokonanych.

5. Btrfs

Kolejnym rozważanym formatem plików jest 64-bitowy Btrfs. W systemie Btrfs maksymalna wielkość pliku wynosi 16 EiB, przestrzeń nazw wynosi 255 znaków z tablicy ASCII, a liczba plików, która może zmieścić się w katalogu wynosi 264 [4].

Główne założenia systemu plików Btrfs to:

- wykorzystanie mechanizmu kopiowania przy zapisie do tworzenia migawek systemowych;
- możliwość odtworzenia stanu systemu z punktów w przeszłości,
- użycie sum kontrolnych w celu zapewnienia integralności danych;
- efektywniej wykorzystywana przestrzeń dyskowa po przez przyjazną dla użytkownika kompresję danych;
- możliwość pracy na wielu woluminach, pełna integralność z macierzami RAID;
- możliwość dynamicznej zmiany pojemności woluminu, za pomocą usuwania i dodawania urządzeń fizycznych [4].

System Btrfs w całości jest oparty o las B-drzew w zmienionym wariantcie. W systemie Btrfs zastosowano zmodyfikowaną wersję zwaną B+drzewami [4]. Zostały usunięte połączenia między sąsiadującymi liśćmi, zamiast tego została stworzona lista jednokierunkowa, która znacząco przyspieszała przechodzenie po zawartości struktury. Na rzecz systemu Btrfs struktura Drzew została wzbogacona

o kilka rozwiązań: zastosowano aktualizację drzewa od wierzchołka do liścia, usunięto powiązania między liśćmi, wprowadzono zliczanie dowiązań. Dzięki tego typu modyfikacjom operacje dodawania i usuwania elementów nie wymagają kopiowania całego drzewa, lecz tylko ścieżki prowadzącej do elementu poddanego modyfikacji. W systemie plików Btrfs drzewo jest wykorzystywane jako podstawowa. Struktura tego typu pozwala tworzyć dynamiczny system do przechowywania danych. Wszystkie informacje i struktury, które są częścią architektury systemu (wykluczając superblok), są przechowywane wewnątrz lasu drzew [6].

Jak już wspomniano system plików Btrfs opiera się na mechanizmie kopiowania przy zapisie [4]. Mechanizm kopiowania przy zapisie to technika optymalizacji w przypadku potrzeby pracy na dużej ilości danych. Zamiast rzeczywistego kopiowania pamięci, które wymaga dużych nakładów czasu i pamięci, zwracany jest jedynie wskaźnik do oryginalnych danych. Kopiowanie odbywa się wtedy, gdy zachodzi potrzeba ich modyfikacji. Najważniejszą zaletą jest brak wykonywania kopii pliku, który nie został zmodyfikowany [6]. W przypadku Btrfs funkcja COW daje możliwość skopiowania sektora dysku, podczas gdy dysk nadpisany jest nową porcją informacji [4]. Dzięki temu w łatwy sposób można tworzyć kopie zapasowe. Dotychczas, aby to zrobić należało kopiować fizycznie dane w inne miejsce na dysk.

Dzięki strukturze Btrfs nie jest wymagana fizyczna kopia. Wystarczy wykonać dodatkowe powiązanie w systemie plików, dzięki któremu dwa odrębne pliki będą wskazywać na ten sam blok danych fizycznie zapisanych na nośniku. Rozwiązanie jest nie tylko szybkie, ale również i oszczędne, ponieważ kopiowane są tylko informacje które są różne w pliku oryginalnym i jego kopii [4].

W jednym wolumenie Btrfs może znajdować się kilka podwoluminów [4]. Podwolumin może działać jako odrębny system plików. Podczas tworzenia podwoluminów nie jest wymagane partycjonowanie. Podwoluminy tworzone są z dostępnej przestrzeni dysku. Tak utworzony wolumen widziany jest jako katalog główny [4]. W połączeniu z mechanizmem kopiowania przy zapisie, można stworzyć kopię całego systemu plików bez utraty dużej ilości przestrzeni dyskowej. W tym przypadku zmianie ulegają jedynie połączenia w logicznych strukturach, a kopiowanie następuje dopiero, gdy plik oryginalny zostanie zmodyfikowany. W przypadku rozwiązań zdalnych i chmurowych, podczas aktualizacji danych nie ma konieczności kopiowania wszystkich danych. Kopiowane są jedynie sektory, które uległy modyfikacji [4]. Rozwiązania wprowadzone w Btrfs znacząco przyspieszają operacje na dużych plikach [4].

Wprowadzenie sum kontrolnych znacząco zwiększa bezpieczeństwo plików na mikro uszkodzenia. W Btrfs tworzona jest suma kontrolna dla każdego sektora dysku zawierającego informację. Podczas każdego odczytu suma kontrolna jest ponownie obliczana, a w przypadku zmiany wartości któregoś z bitów, suma kontrolna nie będzie się zgadzała z tą zapisaną wcześniej. W przypadku błędu, system Btrfs może automatycznie przywrócić informację,

zapobiegając tym samym trwałym uszkodzeniom pliku [4].

W systemie Btrfs wprowadzono szereg funkcji wspomagających zarządzanie partycjami bez potrzeby odmontowania systemu pliku [4]. Rozwiązanie wykorzystywane jest podczas dynamicznej zmiany rozmiaru wolumenu. Dzięki funkcji dynamicznej zmiany rozmiaru wolumenu jest możliwość zmian rozmiaru partycji przy działającym systemie.

Nową cechą wprowadzoną do Btrfs jest możliwość kompresji „w locie” [4]. W przypadku Btrfs za kompresję odpowiadają dwa algorytmy ZLIB (wolniejszy, większe ratio) i LZO (szybszy, mniejsze ratio) [4]. Mechanizm kompresji jest inteligentny, plik który nie został wcześniej skompresowany, zostaje automatycznie poddany kompresji. Dzięki funkcji kompresji nie tylko jest zmniejszona objętość, ale również poprawiona jest wydajność całego systemu plików [4].

Nową cechą wprowadzoną do Btrfs jest funkcja migawek [4]. Tradycyjne menedżery woluminów wymagają, by cały dysk logiczny był objęty migawką. Btrfs natomiast pozwala tworzyć migawki zarówno całych folderów, jak i pojedynczych plików znajdujących się w dowolnym miejscu na dysku [4]. Migawka jest bardziej użytecznym rozwiązaniem niż kopiowanie całego woluminu. W środowiskach produkcyjnych, można uniknąć wyłączania maszyny i zachować podzielność danych podczas archiwizacji całych plików lub folderów. Migawki można zaplanować zgodnie z harmonogramem, co znacznie ułatwia administratorom powrót do archiwalnej wersji pliku (przypadku modyfikacji lub skasowania), nie zatrzymując środowiska produkcyjnego [4].

Rynek macierzy danych stale się rozwija. Na przestrzeni ostatnich kilku lat popularność zyskały dyski zbudowane na bazie pamięci flash tzw. SSD (ang. solid-state drive). System Btrfs posiada szereg funkcji mających na celu sprawną pracę dysków SSD m.in. TRIM [4].

Prace nad Btrfs cały czas trwają. Niektóre funkcje, które zostaną dodane to: szybka weryfikacja systemu plików, dynamiczna weryfikacja systemu plików, równoważenie obciążenia danych w przypadku obsługi maszyn wielowątkowych, dodanie nowych algorytmów kompresji, czy funkcje związane z wydajnością transferu.

6. Badania

W testach została zbadana szybkość odczytu i zapisu dużej liczby małych plików oraz dużych plików.

6.1 Środowisko testowe

Do przeprowadzenia testów został użyty Laptop Dell Precision M4500 z dyskiem twardym HDD 500GB, o prędkości obrotowej 7200 obr/min oraz zainstalowanym systemem Ubuntu 15.10. Do przeprowadzenia testów użyto własnego prostego skryptu bash.

6.2 Przebieg badania

Przed przeprowadzeniem testów, należało wykonać poprawną konfigurację systemu Ubuntu. Aby uzyskane testy

były wiarygodne, zainstalowany został trzy razy system Ubuntu wraz z testowanym systemem plików.

Testy wydajności będą obejmować czas wykonywania operacji kopiowania przez system plików. Testy obejmą kopiowanie dużej ilości małych plików o rozmiarach 1KB, 1M, 1GB, 5GB, 10GB, 75GB. Do testów wykorzystano skrypt bash, który ma za zadanie generowanie plików, wykonywanie operacji kopiowania na danej partycji z folderu do folderu, zmierzenie czasu wykonania operacji, usunięcie plików. Poniżej został przedstawiony użyty skrypt bash do testowania wydajności systemu plikowego.

Przykład 1. Skrypt bash badający czas wykonywania operacji kopiowania w systemie plików

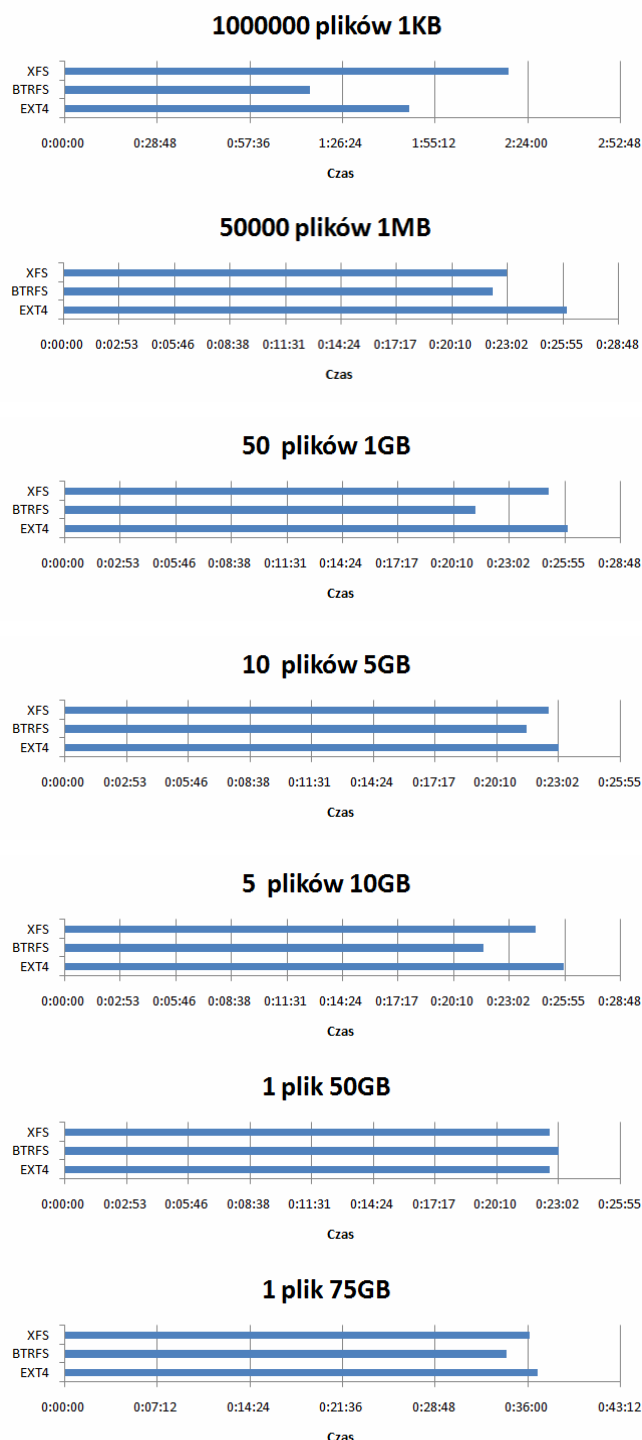
```
1. #!/bin/bash
2. NUM_FILES=10 #ilość plików
3. SIZE_FILE=500M #rozmiar pliku
4. FROM_COPY=from_copy
5. TO_COPY=to_copy
6. mkdir $FROM_COPY #folder tworzenia plików
7. mkdir $TO_COPY #folder do którego kopiowane są pliki
8. cd $FROM_COPY #przejdź do katalogu, w którym
   generowane będą pliki
9. echo "Tworzę pliki"
10. for i in $(seq $NUM_FILES); do
11. dd if=/dev/zero of=file_${i}bs=$SIZE_FILE count=1 &>
   /dev/null #funkcja generująca pliki
12. done
13. echo "Utworzyłem pliki"
14. cd ..
15. echo "Rozpaczynam kopiowanie"
16. echo "Kopiowanie $NUM_FILES plików o rozmiarze
   $SIZE_FILE zajęło: "
17. time(cp $FROM_COPY $TO_COPY -rf; sync)#pomiar
   czasu kopiowania
18. rm $FROM_COPY $TO_COPY -rf#kasowanie plików
19. sync
```

6.3 Prezentacja rezultatów badań

Testy wykazały, że najwydajniejszym systemem plików z omawianych jest system Btrfs. Na rysunku 1 zostały przedstawione wyniki testów. Uzyskał on najlepszy czas w sześciu z siedmiu przeprowadzonych testów. Z testów wynikało, że Btrfs gorzej radzi sobie z dużą ilością małych plików, lecz bardzo dobrze poradził sobie z plikami dużymi. Na drugiej pozycji znalazł się system XFS, otrzymał najlepszy wynik w jednym z sześciu przeprowadzonych testów. Czas wykonywania operacji przez system XFS w przypadku dużych plików był gorszy o około 10%. XFS najgorzej poradził sobie w przypadku operacji na dużej ilości małych plików, wynik uzyskany przez XFS był o 50% gorszy od wyniku uzyskanego przez system BTRFS. W teście drugim system XFS uzyskał zbliżony wynik do Btrfs, w pozostałych czterech testach wynik był od 3% do 6% gorszy od wyników uzyskanych przez Btrfs.

System Btrfs jest systemem plików, który służy do zapewnienia najwyższej wydajności. Wysoka wydajność obsługi plików w połączeniu z przystosowaniem do obsługi dysków SSD stanowią bardzo dobre połączenie dla rozwiązań wysoko wydajnościowych. System XFS mimo, że na rynku dostępny jest od ponad dwudziestu lat nie ustępuje znacząco wydajnościowo najnowszemu systemowi Btrfs. XFS od 2013 roku jest stale rozwijany i ma szanse w niedługim czasie

wyprzeć jeden z najpopularniejszych systemów jakim jest ext4. Ext4 powoli przestaje być domyślnym systemem plików w wielu dystrybucjach Linuxa. Jest mniej wydajny od pozostałych testowanych systemów plików oraz zajmuje najwięcej miejsca w przypadku zamontowania systemu plików na partycji.



Rys 1. Wyniki testów wydajnościowych poszczególnych systemów plików

6.4 Zestawienie cech funkcjonalnych

W tabeli 1 zawarte jest zestawienie cech funkcjonalnych porównywanych systemów plików.

7. Wnioski

Aktualnie najlepszy okazał się system Btrfs - jest najwydajniejszy, posiada szereg ciekawych funkcji oraz jest stale rozwijany. Btrfs idealnie nadaje się do rozwiązań domowych jak i niekomercyjnych. Na drugim miejscu uplasował się system XFS, który odbiega wydajnością od Btrfs, ale jest bardzo stabilny oraz wydajnie obsługuje systemy zawierające wiele małych plików. W połączeniu z kompresją w locie daje nieograniczone możliwości powiększania systemu plikowego. Najgorszy okazał się system ext4. Mimo, że łatwo można z niego odzyskać pliki oraz, że posiada kompresję w locie - tylko w przypadku zastosowania partycji GTP- jest to nadal bardziej zbiór ulepszeń niż nowy w sensie, zaprojektowany od podstaw system plików, który jest w stanie zaspokoić potrzeby ciągle rozwijającego się zapotrzebowania na rynku informatycznym

Tabela 1. Cechy funkcjonalne porównywanych systemów plików

Cecha funkcjonalna	EXT4	XFS	BTRFS
Programy testujące i naprawiające	posiada	brak	posiada
Mechanizm sum kontrolnych	posiada	brak	posiada
Mechanizm opóźnionej alokacji	posiada	posiada	posiada
Dynamiczne rozszerzanie rozmiaru woluminu	z pomocą LVM	tak	tak
Indeksowanie katalogów	posiada	brak	brak
Obsługa plików rozrzedzonych	nie	tak	nie
Obsługa plików 64-bitowych	nie	tak	tak
Migawki systemu	nie	nie	tak
Defragmentacja podczas pracy systemu plików	brak	posiada	Posiada
Mechanizm księgowania	tak	nie	tak
Konwersja na inny system plików	tak na ext3	brak	tak na ext4
Gwarantowana przepustowość danych	brak	posiada	Brak
Obsługa macierzy RAID	tak	tak	tak

Literatura

- [1] Bar M.: Linux systemy plików, Wyd. I Warszawa: RM 2002. ISBN 83-87-02948-1.
- [2] Seidel U.: System plików ext. „Linux Magazine”, 2013, nr 12, s. 63-66.
- [3] Oficjalna strona systemu XFS: http://xfs.org/index.php/Main_Page [06.06.2013]
- [4] Wiki systemu Btrfs: https://Btrfs.wiki.kernel.org/index.php/Main_Page [18.01.2016]
- [5] Wiki systemu ext4 https://ext4.wiki.kernel.org/index.php/Main_Page [03.09.2016]
- [6] Brendel J.C.: Porównanie Ext3, Ext4, XFS i Btrfs. „Linux Magazine”, 2014, nr 9, s.18- 21

Aktualne narzędzia wytwarzania oprogramowania na platformie JEE

Paweł Ozdoba*, Beata Pańczyk

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł ma charakter przeglądowy i opisuje aktualnie stosowane narzędzia wspomagające proces wytwarzania oprogramowania na platformę Java Enterprise Edition. W artykule dokonano wyboru omawianych narzędzi w oparciu o statystyki wykorzystania udostępnione na stronach internetowych. Przeanalizowano zintegrowane środowiska programistyczne, narzędzia do kontroli wersji kodu, narzędzia automatyzujące budowę oprogramowania oraz analizujące poprawność kodu. W niniejszej pracy wskazano najbardziej optymalne (według autora) narzędzia, wykorzystując je do budowy przykładowej aplikacji testowej w środowisku JEE.

Słowa kluczowe: JEE; środowiska programistyczne; repozytorium binarne; wersjonowanie; statyczna analiza kodu

*Autor do korespondencji.

Adres e-mail: PawelOzdoba07@gmail.com

The current software tools for the JEE platform

Paweł Ozdoba*, Beata Pańczyk

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. This article is an overview and describes the currently used tools supporting software development process for Java Enterprise Edition. The tools were selected based on the usage statistics provided by websites. Analysis includes integrated development environments, code revision tools, software automation tools, and code validation tools. This paper shows the most optimal (according to the author) tools, using them to build sample test application in JEE.

Keywords: JEE; integrated development environment; binary repository; versioning; static code analysis

*Corresponding author.

E-mail address: PawelOzdoba07@gmail.com

1. Wstęp

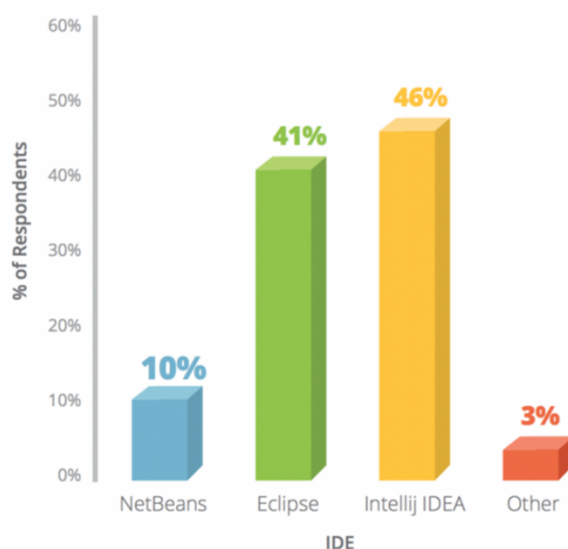
Szybki rozwój narzędzi programistycznych umożliwia efektywną pracę nawet z bardzo skomplikowanymi projektami. Wytwarzanie aplikacji Javy, z zachowaniem właściwego poziomu jakości, nie jest łatwym zadaniem. Jednak dzięki wykorzystaniu odpowiednich narzędzi programistycznych, można podnieść przewidywalność, niezawodność i efektywność procesu wytwarzania oprogramowania oraz sprawić, że projektowanie, kodowanie oraz wdrażanie wysokiej jakości aplikacji Java będzie maksymalnie uproszczone. Wdrażanie i stosowanie większości narzędzi programistycznych nie jest trudne, jednak wymaga wysiłku związanego z weryfikacją bieżących praktyk pod kątem zgodności z nowymi trendami. Dla każdego zadania związanego z kolejnym etapem wytwarzania aplikacji, istnieje przynajmniej kilka skutecznych narzędzi typu open source. Przybliżenie tych narzędzi oraz próba wskazania najbardziej optymalnego ich zestawu jest głównym celem niniejszego artykułu.

2. Narzędzia wspomagające proces tworzenia aplikacji JEE

2.1. Zintegrowane środowiska programistyczne

Najpopularniejsze środowiska programistyczne dla platformy Java Enterprise Edition to **Eclipse**, **NetBeans** i **IntelliJ**. Rysunki 1 i 2 przedstawiają popularność środowisk programistycznych dla Javy w roku 2016 na bazie portali *zeroturnaround.com* oraz *www.baeldung.com*. Rysunek 3

prezentuje udział poszczególnych IDE na przełomie lat 2012-2016.



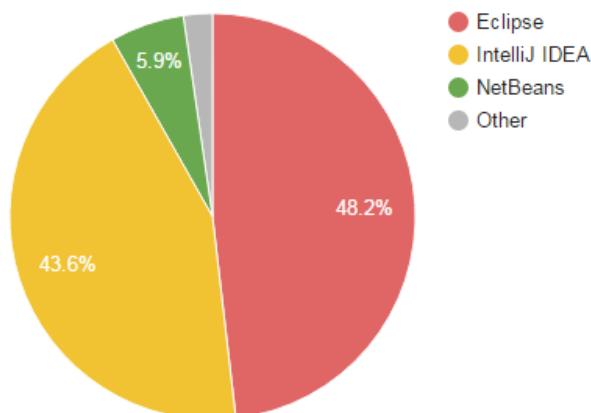
Rys.1. Popularność IDE dla Javy [1]

Porównując wartości liczbowe z rokiem 2012 (Rys. 3), widać wyraźny wzrost popularności IntelliJ, kosztem Eclipse..

2.2. Kontrola wersji kodu

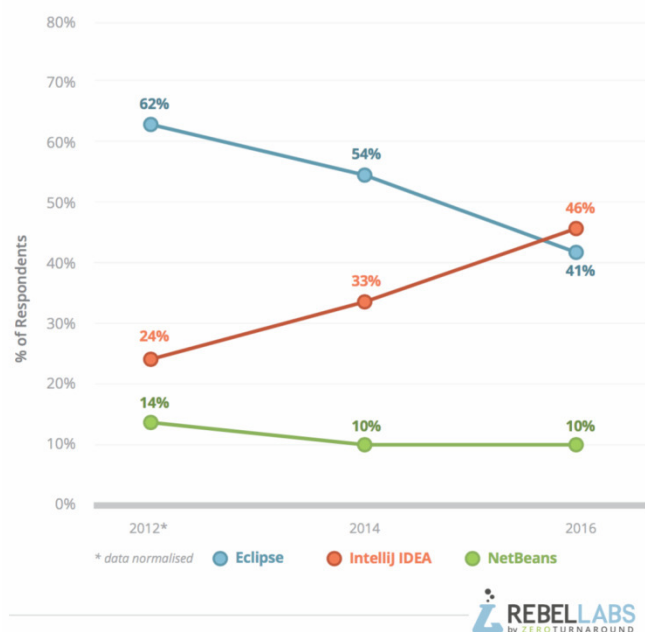
Systemy kontroli wersji (lub systemy zarządzania kodem źródłowym) pozwalają na przechowywanie wielu wersji plików, umożliwiając dostęp do ich poprzednich wersji.

Najpopularniejsze aktualne narzędzia wykorzystywane do kontroli wersji kodu to: **SVN** i **GIT** (Rys. 4).



Rys.2. Popularność IDE dla Javy [2]

Figure 3.3 IDE Usage Since 2012

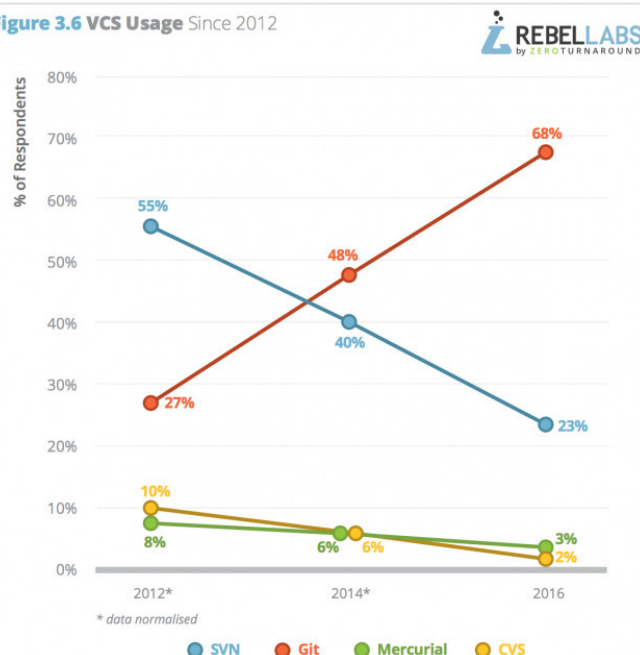


Rys.3. Popularność IDE w latach 2012-16 [3]

2.3. Automatyzacja budowy oprogramowania

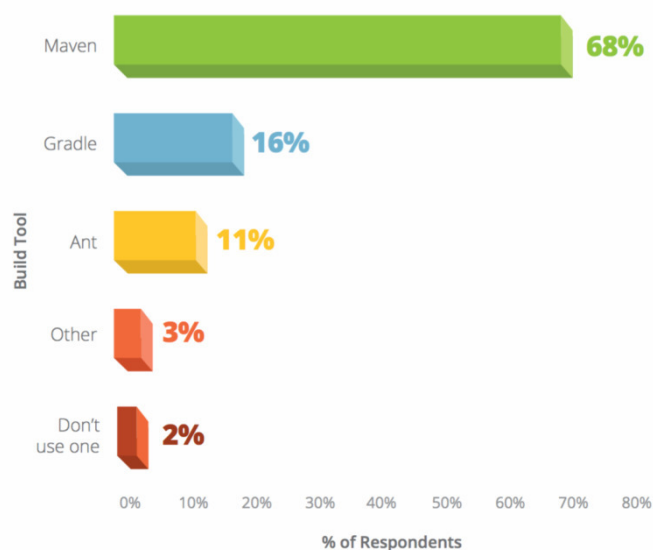
W prostych projektach testowanie i kompilowanie oprogramowania można wykonać za pomocą narzędzi dostępnych w IDE. Duży projekt, który zawiera więcej niż jeden plik wykonywalny, wymaga większej kontroli. Konieczne staje się wtedy wykorzystanie narzędzia do automatycznej kompilacji oprogramowania, które uprości zarządzanie strukturą i zależnościami projektu. Najpopularniejsze narzędzia automatyzujące budowę oprogramowania to: **ANT**, **MAVEN** oraz **GRADLE** (Rys. 5).

Figure 3.6 VCS Usage Since 2012



Rys.4. Popularność narzędzi kontroli wersji kodu [3]

Figure 1.12 Battle of the build tools



Rys.5. Popularność narzędzi automatyzacji oprogramowania [1]

2.4. Analiza statyczna poprawności kodu

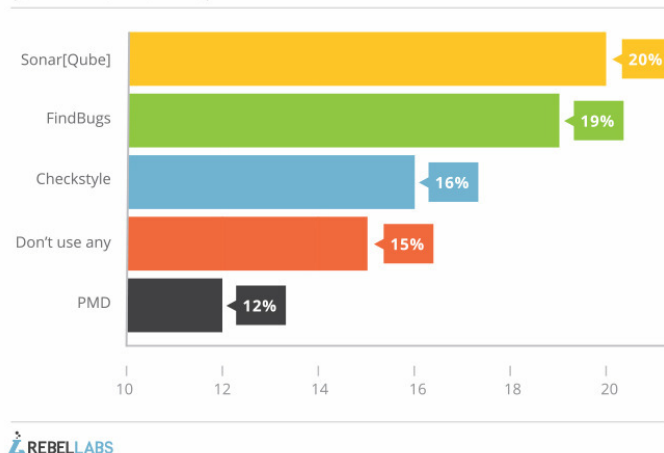
Napisanie elastycznego, łatwego w utrzymaniu i wysokiej jakości oprogramowania nie jest prostym zadaniem. Bardzo pomocne są wzorce programistyczne, zebrane w formie standardów, wskazujące dobre praktyki pisania kodu. Największe problemy występują, gdy programiści stosujący odmienne konwencje programowania, pracują nad tą samą aplikacją. W celu rozwiązania tych trudności powstało wiele narzędzi do analizy oprogramowania np. **SonarQube** czy **CheckStyle** (Rys. 6).

Analiza statyczna kodu obejmuje:

- analizę poprawności składni;

- błędy występujące przy danych wejściowych;
- sprawdzenie istnienia funkcji nieaktualnych (niebezpiecznych), przepełnienia bufora, wycieków pamięci;
- wykrycie stosowania zmiennych niezainicjalizowanych;
- szukanie nieużywanych fragmentów kodu, powtórzeń;
- podpowiedzi dotyczące poprawienia wydajności kodu;
- wskazówki co dostosowania standardów kodowania.

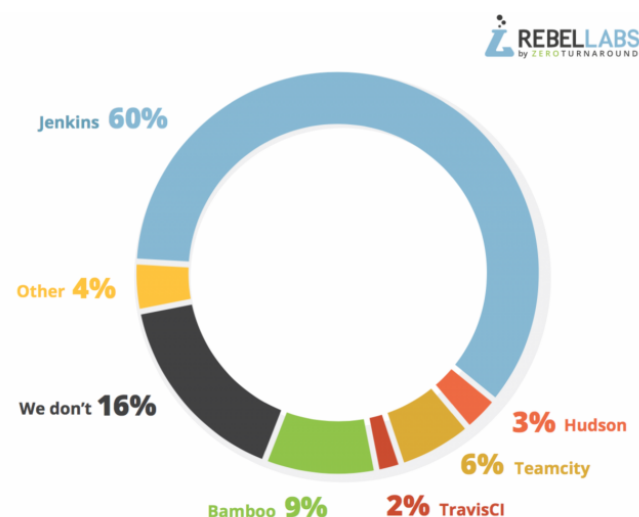
STATIC CODE ANALYSIS TOOL USAGE BY DEVELOPERS (SAMPLE SIZE: 2119)



Rys.6. Popularność narzędzi do analizy statycznej kodu [1]

2.5. Serwery automatyzujące procesy wytwarzania kodu

Ciągła integracja wymaga, aby za każdym razem, gdy programista wprowadzi jakąkolwiek zmianę, cały projekt został skompilowany i poddany różnokierunkowym testom zautomatyzowanym. Konieczna jest zatem stała dostępność działającego oprogramowania. Popularne narzędzia służące do ciągłej integracji to np.: **Jenkins**, **Bamboo**, **Hudson**, **Continuum**, **CruiseControl** (Rys 7).



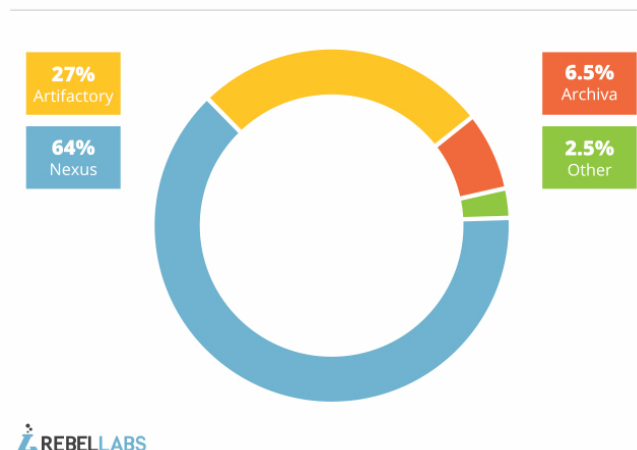
Rys.7. Popularne narzędzia do ciągłej integracji kodu [1]

2.6. Serwery przechowujące dystrybucje aplikacji

Repozytorium binarne jest to repozytorium oprogramowania dla pakietów, artefaktów i odpowiadających im metadanych. Może być używane do przechowywania

plików binarnych. Serwer przechowujący binaria (dystrybucje aplikacji) służy do optymalizacji przechowywania i pobierania plików binarnych, które są stosowane podczas etapu wytwarzania oprogramowania. Po usunięciu całego repozytorium artefaktów, można przywrócić wszystkie wartościowe informacje. Na rysunku 8 przedstawiono zestawienie najpopularniejszych narzędzi do przechowywania dystrybucji kodu.

Binary/artifact repository used*

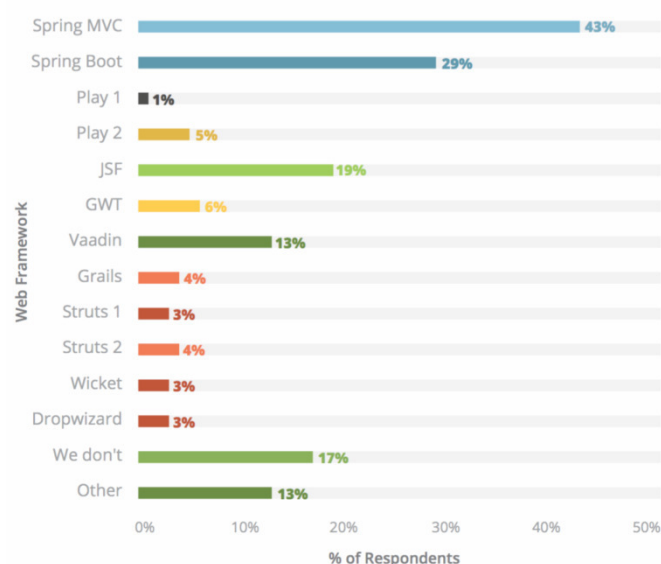


Rys.8. Popularność narzędzi do przechowywania dystrybucji kodu [1]

2.7. Frameworki do tworzenia aplikacji internetowych

Aplikacje internetowe w środowisku JEE można tworzyć z wykorzystaniem wielu dostępnych szkieletów programistycznych. Najpopularniejsze z nich zestawiono na rysunku 9.

Figure 1.16 War of the web frameworks



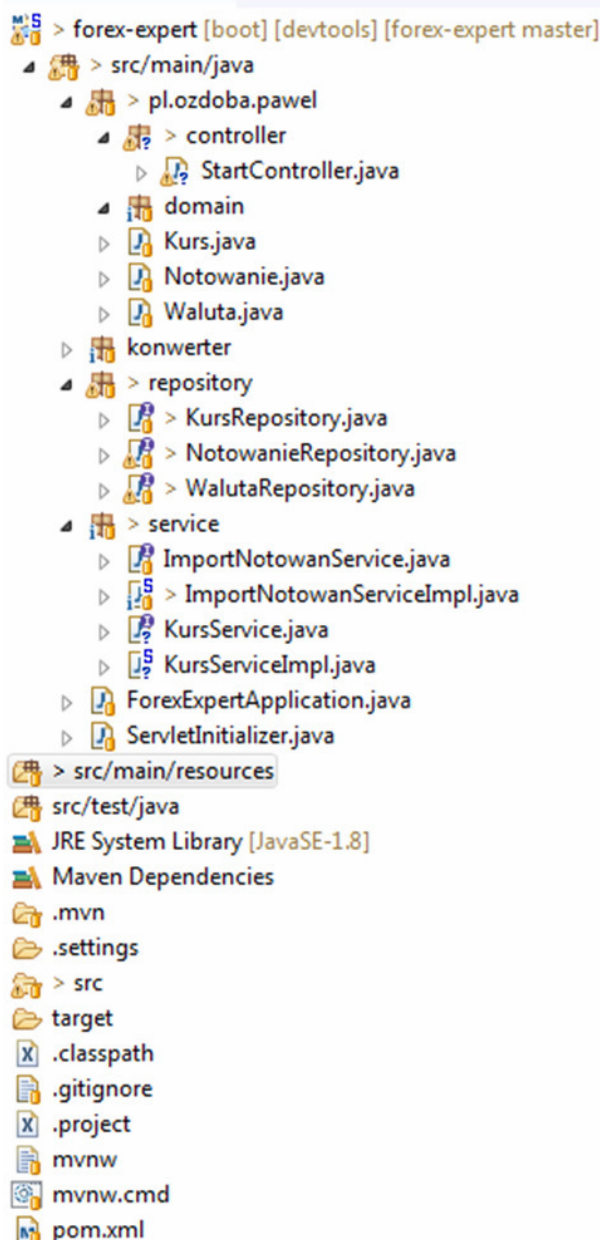
Rys.9. Popularność narzędzi do zarządzania zadaniami programistów [1]

3. Aplikacja testowa

Prosta aplikacja testowa pobiera ze strony [4] dane (wartość waluty, nazwę oraz datę notowania) za pomocą technologii REST i zapisuje je do bazy MySQL. Proces tworzenia przykładowej aplikacji internetowej „forex-expert” pokazano na przykładzie wykorzystania wybranego zestawu aktualnych i popularnych narzędzi:

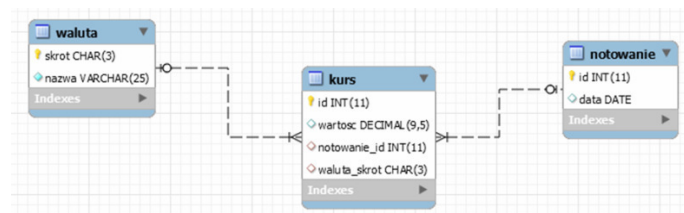
- Eclipse IDE (zintegrowane środowisko programistyczne),
- Maven (automatyzacja zarządzania projektem) [5],
- Spring Boot (szkielet aplikacji internetowej) [6],
- Git (system kontroli wersji kodu) i Bitbucket (zarządzanie repozytorium),
- SonarQube (statyczna analiza kodu),
- Jenkins (ciągła integracja),
- JFrog Artifactory (repozytorium binarne) [7].

Rysunek 10 przedstawia strukturę projektu Spring Boot [5] w Eclipse z podziałem na warstwy aplikacji.



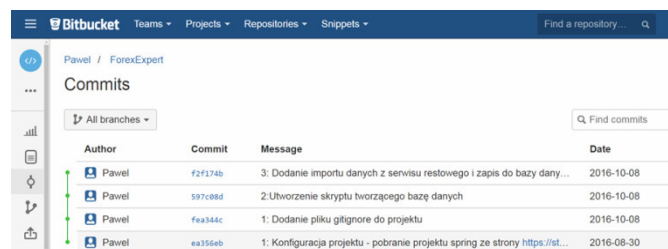
Rys.10. Struktura aplikacji testowej Spring Boot

W aplikacji zdefiniowano między innymi encje: *Kurs*, *Notowanie* i *Waluta*. Do operacji na bazie danych (Rys. 11) zastosowano technologię Hibernate.



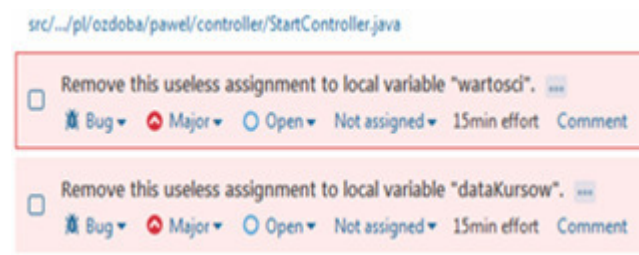
Rys. 11. Baza danych dla aplikacji testowej

BitBucket [8] został wykorzystany do zarządzania repozytorium GIT z wersjami projektów. Przykładowe zmiany repozytorium kodu GIT przedstawia rysunek 12.



Rys. 12. Bitbucket i zmiany repozytorium GIT

Przykład analizy jakości kodu za pomocą SonarQube przedstawia rysunek 13 (uwagi o nie używaniu zmiennych lokalnych) oraz rysunek 14 (informuje o procencie powielenia linii kodu).



Rys. 13. Analiza jakości kodu za pomocą SonarQubej – uwaga dotycząca zmiennych lokalnych

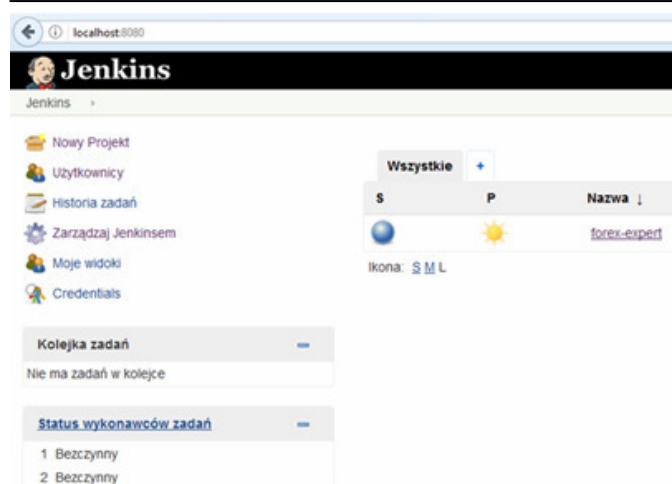
Duplicated Lines (%)

6.6%

	Duplicated Lines (%)	Duplicated Lines
src/main/java/pl/ozdoba/pawel/domain/Notowanie.java	30.0%	24
src/main/java/pl/ozdoba/pawel/domain/Kurs.java	26.1%	24
src/main/java/pl/ozdoba/pawel/repository/WalutaRepository.java	0.0%	0
src/main/java/pl/ozdoba/pawel/domain/Waluta.java	0.0%	0
src/main/java/pl/ozdoba/pawel/controller/StartController.java	0.0%	0

Rys. 14. Analiza jakości kodu za pomocą SonarQubej – uwaga o duplikacji kodu

Do prawidłowej pracy narzędzia Jenkins (wybranego do procesu ciągłej integracji [9]), niezbędne jest zintegrowanie go za pomocą Maven z testową aplikacją (Rys. 15).



Rys. 15. Integracja na bazie serwera Jenkins

4. Wnioski

Po przeanalizowaniu aktualnych narzędzi wykorzystywanych do wytwarzania oprogramowania na platformę Java EE, wskazano te, które cieszą się największą popularnością wśród programistów na konkretnych etapach wytwarzania oprogramowania. Wiele darmowych narzędzi pozwala tworzyć komercyjne oprogramowanie. Wszystkie

wybrane narzędzia mają rozbudowaną dokumentację oraz materiały dodatkowe pomagające w szybki sposób opanować podstawowe funkcjonalności. Dużą zaletą jest to, że większość z analizowanych narzędzi pozwala na integrację z innymi narzędziami.

Literatura

- [1] <https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016/>, [23.05.2017].
- [2] <http://www.baeldung.com/java-ides-2016>, [24.05.2017].
- [3] <https://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-2016-trends/>, [23.05.2017].
- [4] <http://api.fixer.io/latest?base=PLN>, [23.05.2017].
- [5] <https://maven.apache.org/> [25.04.2017].
- [6] Gutierrez F., Pro Spring Boot, Apress, 2016.
- [7] <https://www.jfrog.com/binary-repository/>, [24.05.2017].
- [8] <http://www.intenso.pl/atlassian/pl/atlassian-bitbucket>, [23.05.2017].
- [9] Humble J., Farley D.: Ciągłe dostarczanie oprogramowania. Automatyzacja kompilacji, testowania i wdrażania, Helion, 2015.