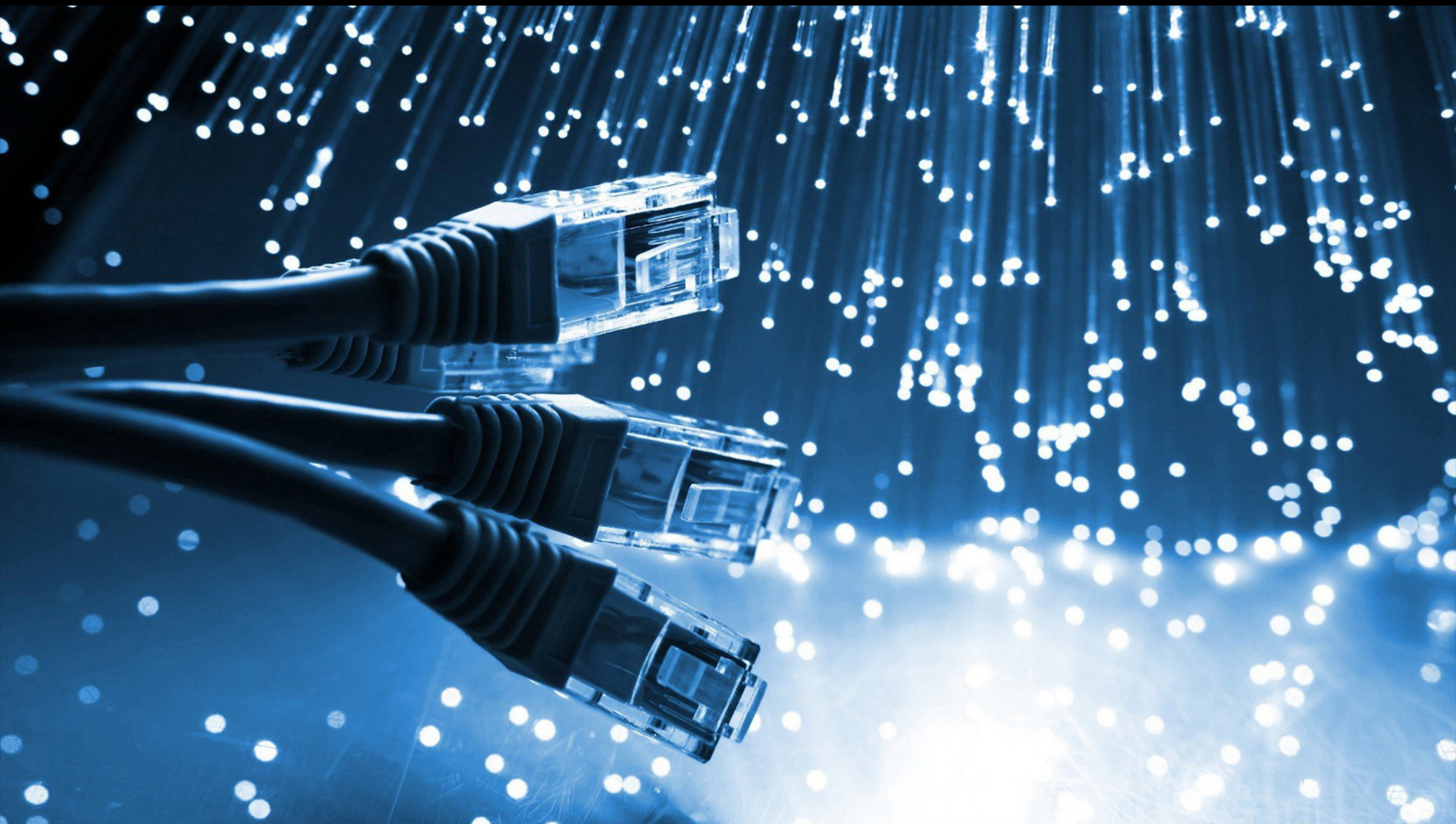


JCSI

Journal of Computer Sciences Institute

Volume 1/2016



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 Mariusz Dzieńkowski
dr Edyta Łukasik
dr inż. Piotr Kopniak
dr inż. Piotr Muryjas
dr Beata Pańczyk
dr inż. Małgorzata Plechawska-Wójcik
dr inż. Maria Skublewska-Paszkowska
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:

Mariusz Dzieńkowski
Edyta Łukasik
Piotr Kopniak
Piotr Muryjas
Beata Pańczyk
Małgorzata Plechawska-Wójcik
Maria Skublewska-Paszkowska
Jakub Smółka

Computer typesetting:

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

Cover design:

Marta Zbańska

Spis treści

1. INTEGRACJA BIG DATA I BUSINESS INTELLIGENCE JAKO INNOWACYJNE ROZWIĄZANIE WSPOMAGAJĄCE FUNKCJONOWANIE NOWOCZESNYCH ORGANIZACJI	
ŁUKASZ BIELAK, PIOTR MURYJAS.....	6
2. PORÓWNANIE WYDAJNOŚCI WYBRANYCH TECHNOLOGII TWORZENIA USŁUG SIECIOWYCH W ASPEKcie ZASTOSOWAŃ W APLIKACJACH INTERNETOWYCH	
ARTUR GDULA, MAŁGORZATA PLECHAWSKA-WÓJCIK.....	14
3. METODA REJESTRACJI I WIZUALIZACJI RUCHU URZĄDZENIA MOBILNEGO	
MACIEJ KULIG, JAKUB SMOŁKA.....	20
4. WYKORZYSTANIE USER EXPERIENCE W PROCESIE POPRAWY GUI APLIKACJI	
BARTŁOMIEJ KULTYS, MAREK MIŁOSZ.....	25
5. ANALIZA MOŻLIWOŚCI WYKORZYSTANIA ARCHITEKTURY SOA DO INTEGRACJI SYSTEMÓW INFORMATYCZNYCH	
PAWEŁ WARĘCIAK, PIOTR KOPNIAK.....	30
6. ZBADANIE WYDAJNOŚCI APLIKACJI INTERNETOWYCH Utworzonych z WYKORZYSTANIEM SPRING MVC ORAZ JAVASERVER FACES	
NAZAR KOVALCHUK, EDYTA ŁUKASIK.....	34
7. ANALIZA PORÓWNAWCZA NARZĘDZI DO BUDOWY PROTOTYPÓW INTERFEJSÓW	
STANISŁAW LIPSKI, DR INŻ. MAREK MIŁOSZ.....	38
8. ANALIZA PORÓWNAWCZA JAKOŚCI SERWISÓW INTERNETOWYCH POLSKICH BANKÓW	
MICHAŁ OSIŃSKI, MAREK MIŁOSZ.....	44
9. ANALIZA WYDAJNOŚCI APLIKACJI WEBOWEJ WE FRAMEWORKU ANGULARJS	
YURII VAVRIN, EDYTA ŁUKASIK.....	50
10. METODYKA TWORZENIA OPROGRAMOWANIA A JAKOŚĆ KODU – ANALIZA PORÓWNAWCZA DWÓCH PRZYPADKÓW	
BARTŁOMIEJ ZALEWSKI, MAREK MIŁOSZ.....	54

Contents

1. INTEGRATION OF BIG DATA AND BUSINESS INTELLIGENCE AS INNOVATIVE SOLUTION SUPPORTING THE FUNCTIONING OF MODERN ORGANIZATIONS ŁUKASZ BIELAK, PIOTR MURYJAS.....	6
2. PERFORMANCE COMPARISON OF SELECTED WEB SERVICE DEVELOPMENT TECHNOLOGY IN WEB APPLICATIONS ARTUR GDULA, MAŁGORZATA PLECHAWSKA-WÓJCIK.....	14
3. A METHOD FOR LOGGING AND VISUALIZATION OF MOBILE DEVICE MOTION MACIEJ KULIG, JAKUB SMOLKA.....	20
4. USE OF USER EXPERIENCE IN THE PROCESS OF THE APPLICATION GUI IMPROVING BARTŁOMIEJ KULTYS, MAREK MIŁOSZ.....	25
5. ANALYSIS OF THE POSSIBILITIES OF USING SOA TO INTEGRATE IT SYSTEMS PAWEŁ WAREĆIAK, PIOTR KOPNIAK.....	30
6. EFFICIENCY TEST OF WEB APPLICATIONS CREATED USING SPRING MVC AND JAVASERVER FACES FRAMEWORKS NAZAR KOVALCHUK, EDYTA ŁUKASIK.....	34
7. COMPARATIVE ANALYSIS TOOLS FOR INTERFACES PROTOTYPING STANISŁAW LIPSKI, MAREK MIŁOSZ.....	38
8. COMPARATIVE ANALYSIS OF THE QUALITY OF POLISH BANKS INTERNET SERVICES MICHAŁ OSIŃSKI, MAREK MIŁOSZ.....	44
9. ANALYSIS OF WEB-APPLICATION WITH ANGULARJS FRAMEWORK YURII VAVRIN, EDYTA ŁUKASIK.....	50
10. METHODOLOGY OF SOFTWARE DEVELOPMENT VS. CODE QUALITY – A COMPARATIVE ANALYSIS OF TWO CASES BARTŁOMIEJ ZALEWSKI, MAREK MIŁOSZ.....	54

Integracja Big Data i Business Intelligence jako innowacyjne rozwiązanie wspomagające funkcjonowanie nowoczesnych organizacji

Łukasz Bielak*, Piotr Muryjas*

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

Streszczenie. Celem niniejszego artykułu jest przedstawienie możliwości integracji Business Intelligence(BI) i Big Data (BD). Na podstawie studiów literaturowych określono potencjalne korzyści i wady płynące z takiego rozwiązania. Ponadto, wymienione zostały korzyści jakie odniosły organizacje, które wdrożyły rozwiązania BI i BD. Praca obejmuje także autorski projekt systemu integrującego BI i BD w organizacji z sektora medycznego.

Słowa kluczowe: Big Data; Business Intelligence; Big Data Analytics

* Autor do korespondencji

Adresy E-mail: lukasz.bielak1@gmail.com, p.muryjas@pollub.pl

Integration of Big Data and Business Intelligence as innovative solution supporting the functioning of modern organizations

Łukasz Bielak*, Piotr Muryjas*

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

Abstract. The purpose of this article is to present the possibilities of integration of Business Intelligence (BI) and Big Data (BD). Based on literature studies identified the potential benefits and drawbacks coming from this solution. In addition the benefits of this solution were listed, based on case studies. The article also includes a proprietary system design that integrates BI and BD in the organization of the medical sector.

Keywords: Big Data; Business Intelligence; Big Data Analytics

*Corresponding author

E-mail addresses: lukasz.bielak1@gmail.com, p.muryjas@pollub.pl

1. Wstęp

Obecnie niemal każda organizacja musi dbać o swoją innowacyjność, aby być konkurencyjną na rynku. Nowoczesne oprogramowanie, takie jak systemy CRM czy ERP ułatwiają prowadzenie i zarządzanie nowoczesnych firm. Takie systemy gromadzą dane na temat sprzedaży produktów, przeprowadzonych transakcji, czy zachowań klienta. W sytuacji, gdy organizacja posiada zarchiwizowane dane z jakiegoś okresu, korzystna może okazać się ich analiza. Do tego celu wykorzystuje się rozwiązania klasy Business Intelligence (BI), które na podstawie danych wspomagają podejmowanie decyzji biznesowych, wykrywają potencjalne przekłamania i próby oszustw. Warunkiem poprawnego działania systemu BI są wiarygodne dane, pochodzące najczęściej z systemów transakcyjnych danej firmy. Bardzo powszechny w dzisiejszych czasach dostęp do urządzeń mobilnych opartych o system operacyjny jest przyczyną generowania dużej ilości danych. Co za tym idzie, mogą one okazać się użyteczne w kontekście analizy i wydobywania wiedzy. Wadą takiego rodzaju danych jest często brak określonej struktury i szybkość ich generowania. W tradycyjnym systemie BI może okazać się niemożliwe przeanalizowanie tak wielkiej ilości danych. Zjawisko przyrostu dużej ilości różnorodnych danych w szybkim tempie nazywane jest obecnie Big Data (BD). Potencjalnie korzystne dla organizacji posiadającej system klasy BI może okazać się połączenie go z Big Data. Celem niniejszej pracy jest

przedstawienie metod integracji Business Intelligence i Big Data, a także płynących z tego korzyści w oparciu o badania literaturowe i studia przypadków. Przytoczone też zostaną przypadki wdrożenia Business Intelligence i Big Data oraz ich wpływ na funkcjonowanie nowoczesnych organizacji. Ponadto, praca zawiera także autorski projekt systemu integrującego Business Intelligence i Big Data w organizacji z sektora medycznego.

2. Business Intelligence

Definicja zamieszczona w *IT Glossary* [20] określa terminem Business Intelligence (BI, system klasy Business Intelligence) aplikacje, infrastrukturę oraz najlepsze praktyki umożliwiające dostęp do danych i ich analizę. Celem omawianego rozwiązania jest optymalizacja i poprawa wydajności procesów biznesowych. Wang C. H. w artykule *A novel approach to conduct the importance-satisfaction analysis for acquiring typical user groups in business-intelligence systems* [13] podkreśla, że system BI ma za zadanie wspomaganie podejmowania decyzji. Podstawowym elementem architektury rozwiązań BI jest hurtownia danych(HD). W książce *Exam 70-463:Implementing a Data Warehouse with Microsoft® SQL Server® 2012* [11] jej autorzy Sarka D., Lah M. i Jerkic G. definiują hurtownię danych jako zcentralizowaną składnicę danych, która przechowuje scalone(z różnych systemów), oczyszczone, historyczne dane. Struktura HD ma zapewnić dużą

wydajność odczytu rekordów, przez co wykorzystuje się głównie trzy schematy logiczne przy jej projektowaniu, gwiazdy, płotka śniegu i konstelacji. Ładowanie danych do HD odbywa się cyklicznie i jest realizowane przez proces ETL, w którego skład wchodzi takie elementy jak ekstrakcja (ang. *extract*), transformacja (ang. *transform*) i ładowanie (ang. *load*) danych. Relatywnie, najbardziej skomplikowana jest transformacja, w której skład wchodzi zmiana struktury danych, a także zagregowanie określonych wartości transakcyjnych. R. Kimball w książce *The Data Warehouse Toolkit Third Edition* [5] zwraca uwagę, że budowa procesu ETL pochłania najwięcej czasu przy budowaniu systemu Business Intelligence. Elementem BI są także bazy analityczne OLAP (ang. *Online Analytical Processing*), które są zoptymalizowane pod kątem kwerend i raportowania, przez co możliwe są do wykonania operacje, które ze względów wydajnościowych nie byłyby możliwe w tradycyjnej hurtowni danych. W literaturze pojawiają się propozycje implementacji OLAP w chmurze obliczeniowej [14]. Kolejnym opisywanym w tej pracy elementem BI jest dźwignia danych (ang. *Data Minig*). Proces ten, zgodnie z definicją proponowaną przez firmę ORACLE [26] polega na odkrywaniu zależności niewidocznych dla badacza, na podstawie zbiorów danych. W literaturze pojawiają się zastosowania Data Miningu w medycynie [1] i systemach ostrzegania o katastrofach naturalnych [7]. Najwyższą warstwą w systemach klasy BI jest warstwa prezentacji danych. Obecnie coraz częściej do tego celu wykorzystywane są narzędzia typu Self Service Business Intelligence, których działanie ma być jak najbardziej intuicyjne dla użytkownika. Taka filozofia pozwala na korzystanie z zawartości baz danych osobom, które nie posiadają specjalistycznej wiedzy. Dzięki aplikacji Self Service BI, można się w relatywnie prosty sposób podłączyć do źródła danych i przy pomocy metody przeciągnij i upuść zbudować wykres czy raport.

2.1. Kierunek rozwoju BI

Jak pokazuje raport wydany przez organizację TDWI, *TDWI Best Practices Report - Business Driven Business Intelligence and Analytics. Achieving Value through Collaborative Business/IT Leadership* [12] obecnie, najbardziej wymagającym elementem, z punktu widzenia utrzymania BI jest warstwa technologiczna. W jej skład wchodzi między innymi procesy ETL i hurtownia danych, a utrzymaniem tych elementów zajmują się osoby ze specjalistyczną wiedzą techniczną. Zupełnie inaczej wygląda obsługa warstwy prezentacji, z którą bez problemu radzą sobie analitycy i pozostałe osoby pracujące z narzędziami tego typu. Co za tym idzie widoczny jest trend rozwoju intuicyjnych narzędzi dla osób nieposiadających specjalistycznej wiedzy, który Firma *Tableau*, producent oprogramowania klasy BI, wymienia w publikacji *Top 10 business intelligence trends for 2016* [23].

2.2. Korzyści z wdrożenia systemu klasy BI

Wiele organizacji wdraża u siebie rozwiązania klasy BI, co pokazują Hannula M. i Pirttimäki V. w publikacji *Business intelligence empirical study on the top 50 Finnish companies* [4]. Artykuł opierał się na ankietach wypełnionych przez 50

fińskich firm mających najlepsze wyniki finansowe w 2002 roku. Głównymi powodami, dla których organizacje decydowały się na wdrożenie systemu BI były:

- Wspieranie podejmowania decyzji i planowanie;
- Wiedza o własnym biznesie i podejmowanie akcji;
- Bycie konkurencyjnym na rynku pracy;
- BI jest trendem na rynku biznesowym;

Znaczna większość organizacji uznała, że wdrożenie systemu BI przyniosło korzyści, co więcej 1/3 respondentów udzieliła odpowiedzi, że nakłady na tego typu systemy wzrosną znacznie. Natomiast 44% uważa, że wzrosną trochę. Korzyści z wdrożenia rozwiązania klasy BI odniósł także Santander Consumer Bank, studium przypadku zostało opisane w publikacji *Studium przypadku z wdrożenia systemu do budżetowania InForum BI Studio w Santander Consumer Bank* [22]. Za główne zalety systemu są uważane:

- Efektywność, proces tworzenia budżetu angażuje mniejsze zasoby ludzkie, sprzętowe i licencyjne;
- Elastyczność i skalowalność;
- Automatyzacja pracy, wydajność pracy, kontrola kompletności, spójności i jakości danych źródłowych;
- Szybkość i łatwość pracy, raporty są budowane w czasie wielokrotnie krótszym niż poprzednio;
- Samodzielność pracy analityków - nie jest potrzebna wsparcie ze strony IT;
- Intuicyjny interfejs;

3. Big Data

Intensywny rozwój techniki w ostatnich latach sprawia, że na rynku pojawia się coraz więcej urządzeń mobilnych bazujących na systemie operacyjnym. Takie rozwiązanie, w połączeniu licznymi czujnikami obecnymi w telefonach, czy tabletach wpływa na wzrost liczby aplikacji, które w założeniach mają ułatwić życie użytkownikowi. Konsekwencją tego zjawiska jest generowanie bardzo dużych ilości danych, stanowiących doskonałe podłoże do wydobywania z nich informacji, a ostatecznie wiedzy. Na uwagę zasługuje fakt, iż w omawianym przypadku, klasyczny model przechowywania danych oparty o bazę relacyjną nie sprawdzi się i konieczne staje się inne podejście do tego zagadnienia. Nowoczesne składnice danych powinny obsługiwać dane nieustrukturyzowane, pochodzące z różnych źródeł, a także oferować szybkie rezultaty zapytań i możliwość analizy, takie rozwiązanie określane jest jako Big Data(BD). Mach-Król M. w publikacji *ANALIZA I STRATEGIA BIG DATA W ORGANIZACJACH* [6], zauważa, że obecnie nie ma ściśle określonej definicji terminu Big Data, zjawisko to jest opisywane za pomocą charakterystyk. Do opisu Big Data według Mach-Król M. wykorzystane są 3 główne właściwości:

- **Volume** (pol. objętość) cecha związana ze stale rosnącą ilością danych, która jest znacznie większa niż w klasycznych rozwiązaniach opartych o relacyjne bazy danych.
- **Velocity** (pol. szybkość) poprzez sieć internet stale przesyłane są dane, a przepustowość łącz musi być ciągle zwiększana. Wynika to z faktu wielu aplikacji mobilnych,

które w relatywnie krótkim czasie generują dane i transferują je do serwerów, bądź innych użytkowników.

- **Variety** (pol. różnorodność) właściwość, która określa strukturę danych. W Big Data gromadzone dane są w postaciach ustrukturyzowanych (systemy CRM, ERP itd.), semi-ustrukturyzowanych (Email, facebook, twitter) i nieustrukturyzowanych (obrazy, dźwięki, filmy).

Wymieniona też zostaje 4 cecha **Veracity** (pol. wiarygodność), która pojawiła się niedawno i paradoksalnie wiąże się z niepewnością danych. Firma SAS Institute, zajmująca się wytwarzaniem oprogramowania z zakresu między innymi Business Intelligence, oprócz 3 wyżej wymienionych głównych cech BD wymienia w publikacji *Big Data - What it is and why it matters* [17] jeszcze dwie dodatkowe:

- **Variability** (pol. zmienność) ilość generowanych danych może rosnąć lub maleć w danym okresie, wpływ na to mają na przykład wydarzenia społeczne takie jak igrzyska olimpijskie czy finały konkursów artystycznych (duża liczba komentarzy i zdjęć na portalach społecznościowych).
- **Complexity** (ang. złożoność) dane pochodzą z różnych źródeł, a ich integracja i oczyszczanie jest skomplikowane i trudne.

3.1. Narzędzia i usługi do organizacji danych

Duże ilości danych, które w szybkim tempie zwiększają swoją objętość wymagają innego podejścia do ich przetwarzania. Według *Big Data for Dummies* [8], narzędzia i usługi do organizacji danych stanowią ekosystem technologii służących do integracji danych i przygotowania ich do dalszej analiz, w jego skład wchodzi:

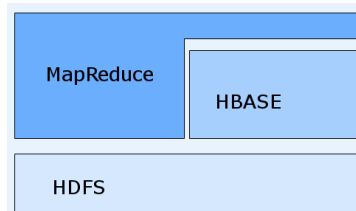
- rozproszony system plików zapewniający skalowalność i dużą przestrzeń dyskową,
- usługi serializacji niezbędne do zdalnego wywoływania procedur w systemie,
- usługi do koordynacji procesów rozproszonych,
- narzędzia ETL,
- usługi do zarządzania harmonogramem wykonywania procesów.

W *Big Data for Dummies* [8] autorzy zwracają uwagę na platformę Hadoop, jako przykład zbioru produktów do organizacji danych. Zgodnie z definicją podaną przez producenta w *Welcome to Apache Hadoop* [24] jest to framework, pozwalający na rozproszone przetwarzanie dużych zbiorów danych w klastrach komputerowych, z wykorzystaniem prostych modeli programowania. Zaprojektowany z myślą o skalowalności od jednego komputera do tysięcy maszyn. W tym rozwiązaniu, bardziej niż stosowanie elementów sprzętowych o wysokiej dostępności liczy się redundancja węzłów, aby w razie awarii, inna maszyna przejęła zadania uszkodzonej. Praca *Turning Small Analytics on Big Data: Data Partitioning and secondary indexes in Hadoop ecosystem* [10] prezentuje architekturę logiczną Hadoopa (rysunek 1), zawiera ona 3 główne elementy składowe: HDFS, HBase i MapReduce.

Rozproszony system plików HDFS (ang. *hadoop distributed*) - został zaprojektowany do przechowywania dużych plików na wielu maszynach. Ponadto, każdy plik jest podzielony na bloki, a te są replikowane na różne maszyny, przez co ryzyko awarii systemu spada.

HBase - jest to nierelacyjna, rozproszona, wersjonowana baza danych typu klucz-wartość. Umożliwia skalowanie horyzontalne (im więcej sprzętu, tym wydajniej działa).

MapReduce - framework programistyczny wykorzystywany przy tworzeniu rozwiązań obliczeń rozproszonych. Zasada działania opiera się na tworzeniu par klucz-wartość (*Map*), a następnie redukowaniu nieistotnych, z punktu widzenia zapytania wyników. Na rysunku 2 przedstawiony został przykład algorytmu MapReduce. Celem, jest uzyskanie zsumowanych wartości dla rekordów, których nazwy kontynentów są w zbiorze ('Europe', 'Africa'). Idąc od góry, dostępne są dwa zbiory danych zawierające klucz, nazwę kontynentu oraz wartość liczbowa. Pierwszym krokiem, jest zmapowanie wartości, oznacza to, że do każdej nazwy kontynentu (zadeklarowanego wcześniej) przypisana zostaje liczba (*Map*). Zadanie to jest wykonywane na obu zbiorach równolegle. Kolejnym krokiem jest scalenie zbiorów (*Merge-Sort*). Przedostatnim etapem, jest zastosowanie funkcji redukcji (*Reduce*), która w tym wypadku, sumuje wartości, na przykład, wiersz <AFRICA,12,10> po redukcji będzie miał postać <AFRICA,22>. Ostatnim krokiem jest zwrócenie wyników, czyli w tym przypadku wierszy <AFRICA,22> i <Europe, 15>



Rys. 1. Architektura logiczna Hadoopa na podstawie [10]

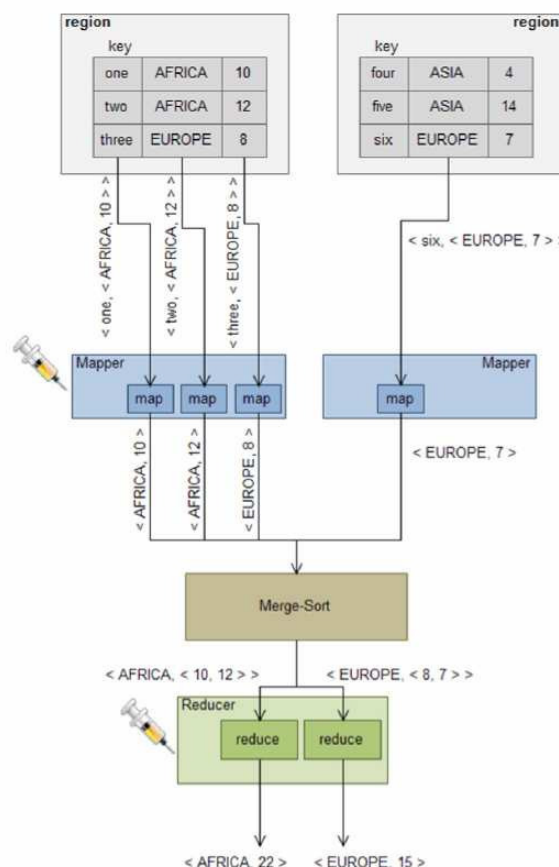
Dzięki wykorzystaniu opisywanych w tym rozdziale narzędzi, możliwa jest techniczna implementacja systemu obsługującego Big Data.

3.2. Big Data w organizacjach

Rosnąca popularność Big Data powoduje, że coraz więcej organizacji interesuje się rozwiązaniami tego typu. *International Institute for Analytics* opublikował raport *Big Data in Big Companies* [3] który został przygotowany na podstawie wywiadów przeprowadzonych w 2013 roku z dużymi organizacjami. Głównym celem było określenie, jak Big Data dostosowało się do już działających środowisk analitycznych, a także składujących dane. Jak zostało wspomniane wcześniej, 3 cechy określające Big Data to objętość, różnorodność i szybkość. Raport pokazuje, że najbardziej istotna dla firm jest różnorodność, która umożliwia analizę danych z różnych źródeł. Co może

potencjalnie przynieść zysk. Z uwagi na ciągły rozwój technologii, źródła danych takie jak strony internetowe czy logi są coraz bardziej powszechne i niosą ze sobą informację. Przykładem jest firma United Healthcare, która wykorzystuje Big Data do takich czynności jak wspieranie planowania leczenia. Co więcej, do kontroli jakości usług wykorzystywane są tam nagrania rozmów telefonicznych zmieniane na tekst i poddawane ostatecznie analizie z wykorzystaniem przetwarzania języka naturalnego. Jak przyznaje organizacja, kiedyś takie rozwiązanie nie byłoby możliwe, jednak obecne metody analizy pozwalają na określenie czy klient wyraża się pozytywnie, czy negatywnie o firmie. Dane typu Big Data i ich analiza we współczesnych organizacjach przyczyniają się do redukcji czasu, jaki kiedyś był potrzeby do wykonania pewnych zadań optymalizacyjnych. Firma Macy's, która jest właścicielem sieci galerii handlowych w Stanach Zjednoczonych informuje, że proces optymalizacji cen dla 73 milionów produktów został skrócony z 27 do 1 godziny. Jest to rezultat implementacji klastra Hadoop i zastosowania obliczeń rozproszonych, dodatkową korzyścią, jest redukcja kosztów sprzętu o 70%, co wynika z możliwości skalowania horyzontalnego Hadoopa. Firma UPS, zajmująca się przewozem przesyłek i logistyką dane na temat transakcji i tras przewozów gromadzi od 1980 roku. Samochody, którymi poruszają się kurierzy posiadają urządzenia zbierające informacje o dziennej wydajności (szybkość, kierunek, hamowanie). Optymalizowanie tras na podstawie takich danych doprowadziło do zaoszczędzenia 8.4 milionów galonów paliwa w 2011 roku, co w konsekwencji przyczyniło się do redukcji kosztów. Planowane jest także wdrożenie takiego rozwiązania dla floty lotniczej. Oprócz redukcji kosztów i czasu, powstają też całkiem nowe zastosowania wielkich zbiorów danych. Przykładem tego są oferty konstruowane na podstawie zgromadzonych danych (Data-Based offering). Firmy takie jak T-Mobile, Sprint, Verizon Wireless działające w branży komunikacyjnej, na podstawie zachowań użytkowników urządzeń mobilnych i ich lokalizacji przygotowują oferty dostosowane do regionów. Netflix jest jedną z największych firm oferujących wypożyczanie filmów przez media strumieniowe. System rekomendacji video jaki jest oferowany użytkownikom, również opiera się o Big Data. Powyższe rozwiązania są nie tylko korzyścią dla organizacji w postaci większych zysków ale także dla klientów, gdyż proponowana jest im oferta zbliżona do potrzeb, może się to przekładać na wzrost zadowolenia z usług. Podobnie jak w przypadku Business Intelligence, jednym z zadań Big Data jest wspomaganie podejmowania decyzji biznesowych, jednak w drugim przypadku wykorzystywane są także dane nieustrukturyzowane. Organizacje wykorzystujące do tego celu Big Data, korzystają z logów aplikacji webowych, rozmów telefonicznych, czy korespondencji mailowych. Amerykańskie banki Wells Fargo, Bank of America i Discover przyznały, że z uwagi na wiele kanałów, jakimi można dokonać zakupu usługi, monitorują postępowania swoich klientów. Dzięki temu możliwe jest lepsze zrozumienie grup społecznych decydujących się na skorzystanie z usługi, a w konsekwencji optymalizację procesów sprzedaży. Organizacje starają się zwiększać swoje zasoby ludzkie związane z zaawansowaną analizą danych

(Data Science). Najchętniej poszukiwane są osoby z dobrymi umiejętnościami Data Science, manipulowania danymi typu Big Data i IT umożliwiającymi pisanie skryptów. Pracodawcy zwracają uwagę, że najbardziej istotną kwestią jest umiejętność prezentacji danych w formie graficznej, czy narracyjnej. Bardziej liczy się przekazanie wyników badań w sposób przystępny niż znajomość narzędzi. GE amerykański konglomerat, działający między innymi w branżach energetycznych i produkcji maszyn, dla osób nowozatrudnionych na stanowisko Data Scientist ma przygotowany wewnętrzny program szkoleń.



Rys. 2. Algorytm MapReduce, źródło : Turning Small Analytics on Big Data: Data Partitioning and secondary indexes in Hadoop ecosystem [10]

4. Big data Analytics i integracja Business Intelligence i Big Data

Firma SAS Institute definiuje Big Data Analytics (BDA) jako badanie wielkich zbiorów danych w celu odkrycia ukrytych wzorców, korelacji i innych zależności (*Big Data Analytics - What it is and why it matters*) [16]. Z kolei IBM w publikacji *What is Big Data Analytics* [25] określa BDA jako użycie zaawansowanych technik analitycznych na wielkich i różnorodnych zbiorach danych, włączając w to dane ustrukturyzowane i nieustrukturyzowane, a także pakiety i strumienie. Taka analiza pomaga badaczom i użytkownikom biznesowym na trafniejsze i szybsze podejmowanie decyzji z wykorzystaniem Big Data. Zaawansowanie techniki analizy takie jak analiza tekstu, uczenie maszynowe, analiza predykcyjna, data mining, statystyka i przetwarzanie języka naturalnego skutkują znacznie trafniejszym i szybszym

decyzjom biznesowym. Podobne techniki stosuje się także w przypadku systemów klasy Business Intelligence. Jak podają autorzy *Big Data for Dummies* [8] kluczowe różnice między dwoma systemami, BI i BDA to przede wszystkim :

- dane typu Big Data mogą pochodzić z niepewnych źródeł,
- dane typu Big Data mogą być nieoczyszczone,
- dane typu Big Data mogą mieć niski współczynnik sygnału do szumu,
- Big Data Analytics umożliwia analizę w czasie rzeczywistym.

J., Hurwitz, A., Nugent, F., Halper, i M., Kaufman w *Big Data for Dummies* [12] dzielą metody analizy danych na 4 podgrupy:

- Prosta analiza, która składa się z takich metod jak
 - wycinanie (ang. slicing and dicing) polegająca na zawężaniu zbiorów danych do interesujących użytkownika na przykład tylko dane z 2016 roku,
 - proste monitorowanie zmian w czasie rzeczywistym,
 - identyfikacja anomalii.
- Zaawansowana analiza, składająca się z:
 - modelowania predykcyjnego pomagającego w obliczaniu prawdopodobieństwa zdarzeń w przyszłości. Zastosowaniem może być wykrywanie, czy transakcja kartą kredytową może być próbą oszustwa,
 - analizy tekstu,
 - innych statystycznych algorytmów takich jak segmentacje i mikrosegmentacje,
 - data miningu.
- Analiza operacyjna używana na przykład przy dobieraniu odpowiedniego produktu dla klienta.
- Monetyzacja danych (ang. *Monetizing analytics*) polega na zamianie zgromadzonych informacji na zysk w postaci pieniądza, wykorzystywane między innymi do odpowiedniego rozlokowania punktów sprzedaży na podstawie wiedzy o klientach.

4.1. Self Service BI jako narzędzie BDA

Podobnie jak w systemach klasy Business Intelligence, zgromadzone dane należy przedstawić w formie, która przenosi jak najwięcej treści dla użytkownika końcowego. Z tego względu, niektóre aplikacje typu Self Service BI umożliwią podłączenie się do Hadoopa, bądź baz typu *NoSQL*. Producenci narzędzi BI firmy *Tableau* i *Microsoft* zaimplementowali w swoich produktach możliwość podłączenia się do źródeł typu Big Data (*Connecting to Hadoop Hive, Data sources in Power BI Desktop*) [18, 19], a także z bazami *NoSql* (*Tableau & MongoDB: Visual Analytics on JSON at the Speed of Thought*) [23]. Dzięki temu, możliwe staje się zintegrowanie ze sobą danych typu Business Intelligence i Big Data. Konsekwencją tego jest relatywnie proste i intuicyjne tworzenie kokpitów menedżerskich, a także raportów przedstawiających dane z różnych źródeł. Na przykładzie tych dwóch wymienionych produktów zauważyć można, że aplikacje typu Self Service BI sprawdzają się także jako aplikacje Big Data Analytics.

4.2. Zaawansowana analiza z użyciem platform BDA dostępnych na rynku.

Analiza predykcyjna i inne działy zaawansowanej analizy są obecnie najprężniej rozwijającym się przedstawicielem rynku analityki dla firm według raportu Gartnera [30]. Zgodnie z definicją przedstawioną w omawianym dokumencie, zaawansowana analityka to analizowanie wszystkich typów danych z wykorzystaniem wysokiej klasy metod ilościowych (statystyki opisowe i predykcyjne, data mining, uczenie maszynowe, symulacja i optymalizacja) dla uzyskania wiedzy, której nie udałoby się wydobyć z wykorzystaniem klasycznego BI (raporty i zapytania). Na rynku obecnych jest relatywnie dużo rozwiązań spełniających powyższe kryteria, co pokazuje rysunek 4.1 Autorzy raportu podkreślają, że środowisko zaawansowanej analizy powinno być kompletne i umożliwiać:

- dostęp do różnych źródeł danych wliczając w to tekst, logi, dane z sensorów, dane lokalizacyjne,
- przygotowanie, eksploracje i wizualizacje danych,
- rozwój i budowanie modeli analitycznych (na przykład modeli predykcyjnych),
- wdrażanie modeli i ich integracje z procesami i aplikacjami biznesowymi działającymi w firmie,
- wysoką wydajność i skalowalność zarówno dla rozwoju jak i wdrażania rozwiązań.

W czołówce rankingu znajduje się firma SAS, w [30] podkreślone zostaje, że posiada ona najszerszy zakres produktów zaawansowanej analizy. Ponad to, aplikacje takie jak Visual Analytics, Visual Statistics są ciągle udoskonalane, a obsługa jest coraz prostsza i bardziej intuicyjna. Klienci cenią w rozwiązaniach SAS jakość i elastyczność, co przekłada się na duże grono lojalnych klientów. Jedną z wad platformy SAS jest pokrywanie się funkcjonalności w produktach, przez co potencjalny klient może mieć problem w wyborze najlepszej opcji. Co więcej, dosyć duży odsetek użytkowników zgłaszał problemy z instalacją i aktualizacją oprogramowania. Gartner zwraca także uwagę, na drogi i nietransparentny system licencjonowania, co może być przyczyną wybierania przez organizacje innych rozwiązań.

Liderem w zakresie zaawansowanej analizy jest także firma IBM. Innowacyjna platforma *Watson*, która łączy przetwarzanie języka naturalnego i uczenie maszynowe aby przetwarzać ogromne zbiory nieustrukturyzowanych danych na wiedzę miała wpływ na to, że o firmie zrobiło się głośniejsze na rynku zaawansowanej analizy. Ponad to, oprogramowanie do analizy predykcyjnej SPSS jest dobrym produktem, który zapewnia duże możliwości, bez konieczności pisania kodu. Na pozycję firmy wpływa także jakość produktów i doświadczenie w różnych branżach. W raporcie zostaje zwrócona uwaga, na kiepską integrację produktów firmy, a także na to, iż konsultanci nie potrafią często dobrać odpowiedniego produktu dla klienta. Dodatkowo, słabe oceny wśród użytkowników zbierała także dokumentacja i mała ilość dostępnych szkoleń.



Rys. 3. Magic Quadrant dla platform zaawansowanej analizy [21]

Firma Microsoft i jej platforma zaawansowanej analizy osiągnęła najwyższy wskaźnik *Completeness of Vision*, wpływ na to miało rozwiązanie *Cortana Analysis Gallery* (CAG, obecnie Cortana Intelligence Gallery), które jest sklepem z rozwiązaniami analitycznymi dla platformy Azure. Istotne znaczenie ma też fakt, iż największy odsetek ankietowanych wybrał firmę Microsoft do własnych rozwiązań w przyszłości na podstawie harmonogramu (ang. *roadmap*). Klienci zwracają także uwagę na dobrą integrację z oprogramowaniem *open-source* takim jak R (język i platforma do obliczeń statystycznych) czy Python (język skryptowy). Dla niektórych respondentów, wadą rozwiązania typu CAG jest konieczność przechowywania danych w chmurze.

Na podstawie analizowanego raportu widoczny jest ciągły rozwój w zakresie zaawansowanej analizy. Istotny jest fakt, że organizacje wytwarzające oprogramowanie do tego celu mają różne wizje dotyczące przyszłości tego zagadnienia (Microsoft Cortana Intelligence Gallery, IBM WATSON). Widoczny jest także trend ograniczania konieczności pisania kodu do minimum i tworzenia intuicyjnych interfejsów użytkownika.

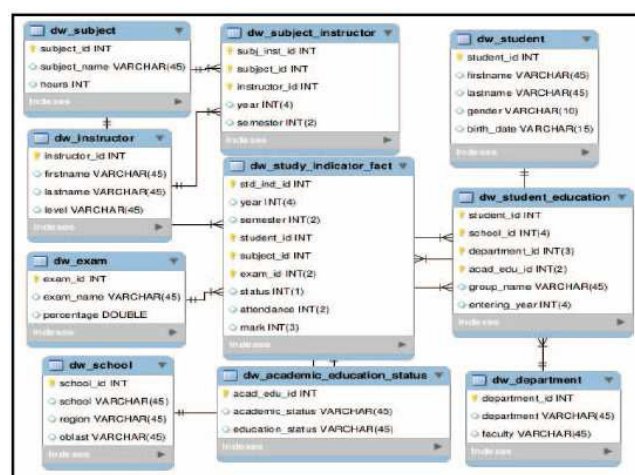
4.3. Możliwości integracji Big Data i rozwiązań Business Intelligence

Organizacje, posiadające dane typu Big Data i typu Business Intelligence mogą czerpać potencjalne korzyści z ich integracji. Jak zostało wspomniane w rozdziale 4.1, oba te rodzaje różnią się i mogą służyć do nieco innych celów. Dane typu Business Intelligence są ustrukturyzowane, oczyszczone i składowane w hurtowni danych, która jest zorientowana tematycznie. Dodatkowo, dane te pochodzą z pewnych źródeł i gromadzone są bardzo często latami, co za tym idzie, użytkownik, po wykonaniu zapytania do hurtowni danych

otrzymuje odpowiedź, na podstawie wysokiej jakości danych [12]. Ze względu na charakterystykę Big Data, dane te nie zawsze muszą być wysokiej jakości, a także mogą pochodzić z różnych źródeł i nie być odpowiednio oczyszczone, co może przyczynić się do większego prawdopodobieństwa wyciągnięcia nieprawdziwych wniosków. Jednym z rozwiązań tego problemu jest utworzenie hurtowni danych, która będzie przechowywać zarówno dane typu BD jak i BI. Takie podejście zaprezentowano w poprzednim rozdziale na rysunku 3.4. Qin H., Zhao Y., Qian Z., w artykule *On the Research of Data Warehouse in Big Data* [15] wymieniają potencjalne wyzwania, przed jakimi stoją projektanci takiej analitycznej bazy danych, są nimi:

- architektura danych i ich przetwarzanie,
- struktura bazy danych i infrastruktura systemu.

W publikacji *Data Warehouse on Hadoop Platform for Decision Support Systems in Education* [2] został przedstawiony przykład projektu i implementacji hurtowni danych z wykorzystaniem technologii platformy Hive. Umożliwia ona budowę hurtowni danych w środowisku Hadoop. Hive oferuje struktury znane z relacyjnych baz danych takie jak tabele, kolumny i partycje wspiera także indeksowanie i metody optymalizacyjne, z uwagi na HDFS (rozproszony system plików platformy Hadoop), zapytania wykonywane są w sposób równoległy [2]. Projekt przedstawiony w artykule *Data Warehouse on Hadoop Platform for Decision Support Systems in Education* [2] miał na celu przeniesienie danych z bazy relacyjnej opartej o MySQL do hurtowni danych zaimplementowanej z wykorzystaniem Hive. Dane pochodziły z uniwersytetu International Ataturk Alaturk University i dotyczyły edukacji na tejże uczelni. Do załadowania danych skorzystano z narzędzia Apache Sqoop, które zgodnie z dokumentem *Apache Scoop* [14], zostało zaprojektowane do transferu dużych ilości danych między bazami relacyjnymi, a platformą Hadoop. Schemat hurtowni danych wykorzystanej do projektu przedstawiony został na rysunku 3.



Rys. 4. Schemat hurtowni danych opartej na platformie Hive [2]

Autorzy projektu twierdzą, że spełnił on postawione na początku wymagania. Możliwa jest:

- wizualizacja danych,

- składowanie danych różnych typów w jednym miejscu,
- wykorzystanie danych do podejmowania decyzji.

Innym, mniej skomplikowanym podejściem jest wykorzystanie narzędzi Big Data Analytics i utworzenie różnych źródeł danych, analiza tych danych i przedstawienie ich na wspólnych kokpitach menedżerskich. To rozwiązanie można zastosować w przypadku posiadaniu hurtowni danych i baz NoSQL, gdy konieczne jest zestawienie takich danych na wykresach. Należy jednak pamiętać, że w tym wypadku znacznie ograniczane są możliwości analizy w odniesieniu do integracji we wspólnej hurtowni danych.

Potencjalne korzyści jakie niesie za sobą integracja danych typu BI i BD to przede wszystkim lepsze wspieranie podejmowania decyzji, gdyż opiera się ono nie tylko na niezwyfikowanych danych nieustrukturyzowanych, ale także na tych ustrukturyzowanych, gromadzonych latami z systemów ERP i CRM. Możliwe jest wykorzystanie prostych w obsłudze narzędzi Self Service BI do tworzenia raportów i kokpitów menedżerskich przez kadrę zarządzającą, a także skorzystanie z platform zaawansowanej analizy obsługiwanych przez analityków danych. Takie rozwiązanie, to niemalże wszystkie zalety systemów BI i BD dla organizacji takie jak: redukcja kosztów, redukcja czasu, poznanie klienta, optymalizacja produktu.

Proponujemy integrację danych typu Business Intelligence i Big data jest rozwiązaniem innowacyjnym, gdyż na rynku nie ma gotowych platform, które oferują takie możliwości. Co za tym idzie, konieczne jest indywidualne określenie potrzeb klienta i decyzja projektantów o połączeniu tych dwóch systemów, a ostatecznie implementacja. Istotny fakt odgrywają potencjalne problemy przy integracji takie jak dobór odpowiedniej Bazy danych, czy narzędzi do budowy procesu ETL.

5. Studium przypadku - wdrożenie Big Data Analytics do organizacji z sektora medycznego

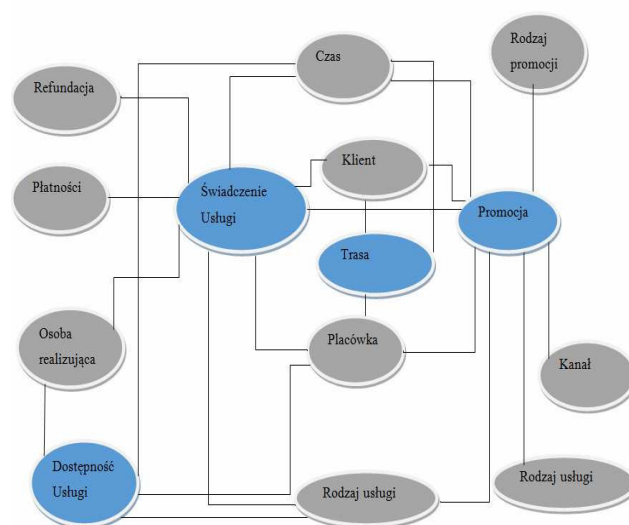
Autorski projekt systemu został wykonany dla firmy świadczącej usługi medyczne w zakresie badań laboratoryjnych, konsultacji lekarskich, oferującej także zabiegi operacyjne, wraz z możliwością pobytu w szpitalu pacjenta. Omawiana organizacja ma kilkanaście oddziałów w całym kraju i w różnych budynkach, niektóre z nich oferują wykorzystanie małych urządzeń działających w technologii Bluetooth w celu nawigacji wewnątrz budynku (Beacon) [15]. Takie rozwiązanie może posłużyć także do zbierania danych o drodze jaką przebył klient od wejścia, do punktu docelowego. Firma posiada własny system klasy CRM, który przechowuje dane na temat wizyt i pacjentów. Głównym celem wdrożenia jest zwiększenie rentowności poszczególnych usług medycznych o 20% w skali roku. Cele szczegółowe to:

- Zwiększenie rentowności funkcjonowania poszczególnych placówek medycznych o 20% w skali roku.
- Opracowanie strategii marketingowej promującej usługi medyczne.
- Wsparcie budowy regionalnych sieci przychodni.

- Wspieranie budowy strategii zwiększania dostępności usług medycznych.
- Opracowanie odpowiedniego oznaczenia drogi do recepcji w wynajmowanych budynkach.

5.1. Projekt systemu

Autorskie rozwiązanie, przygotowane na potrzeby niniejszej pracy składać się będzie z hurtowni danych, której schemat koncepcyjny przedstawiony został na rysunku 4, a miary i hierarchie wymiarów w tabeli 1 i 2. Dane pochodzą z istniejącego już systemu CRM, a także z urządzeń Beacon. Wizualizacja danych z hurtowni będzie odbywać się na raportach i kokpitach menedżerskich.



Rys. 5. Schemat koncepcyjny hurtowni danych -opracowanie własne

Tabela 1. Miary opisujące działalność organizacji - opracowanie własne

Świadczenie usługi medycznej	Cena jaką płaci klient
	Koszty jakie ponosi firma
	Zysk
	Jakość usługi - ocena klienta (0-5)
Promocja	Koszt promocji
	Liczba klientów nowych
	Liczba klientów stałych
	Wartość usług zrealizowanych w ramach promocji
	Zysk z promocji
Dostępność usługi	Liczba przewidywanych dni oczekiwania
	Liczba osób które zrezygnowały
Trasa	Czas przebycia trasy
	Liczba miniętych urządzeń Beacon

Oprócz klasycznej hurtowni danych, rozwiązanie przechowywać będzie dane z urządzeń Beacon w nierelacyjnej bazie *MongoDB*, która umożliwia przechowywanie dużych ilości nieustrukturyzowanych danych. Zasilanie hurtowni danych z systemu CRM i bazy *MongoDB* odbywać się będzie raz dziennie, w momencie najmniejszego obciążenia systemu. Za pomocą aplikacji typu Self Service BI możliwe jest utworzenie kokpitów menedżerskich które będą wizualizować odpowiednie dane i tym samym wspomagać podejmowanie decyzji w taki sposób, aby spełnione zostały określone na początku cele.

Tabela 2. Hierarchie wymiarów wykorzystane w projekcie - opracowanie własne

Klient	Województwo -> Miasto->Dzielnica ->Klient
Rodzaj usługi	Rodzaj usługi-> Podrodzaj usługi-> Usługa Specjalizacja usługi->Podspecjalizacja
Osoba Realizująca	Rodzaj wykonywanej pracy -> Typ personelu -> Nazwa stanowiska->Osoba Rodzaj Etatu -> Wymiar godzin
Płatności	Rodzaj -> Podrodzaj->Płatność
Placówka	Województwo -> Miasto->Dzielnica->Placówka
Czas	Rok->Kwartał->Miesiąc->Tydzień->Dzień->Godzina
Kanał Promocji	Rodzaj kanału-> Podrodzaj kanału -> Nazwa kanału
Rodzaje promocji	Rodzaj -> Podrodzaj->Promocja
Refundacja	Rodzaj->Podrodzaj->Refundacja
Lokalizacja	Województwo -> Miasto->Dzielnica

6. Wnioski

Głównym celem niniejszej pracy było określenie możliwości integracji danych typu Business Intelligence i Big Data. Cel ten został osiągnięty i na podstawie przeprowadzonych studiów literaturowych wyznaczono takie metody integracji jak:

- wspólna hurtownia danych,
- podłączenie narzędzia typu Big Data Analytics do źródeł danych BD i BI.

Na podstawie metod badawczych stwierdzono, że korzyścią z połączenia BD i BI będzie wiedza oparta na wysokiej jakości danych Bli wysokiej różnorodności BD. Zrealizowany zakres pracy wpłynie na kierunek badań w tym temacie, gdyż jak wynika z przeprowadzonego przeglądu literaturowego, nie jest on jeszcze popularny w środowisku naukowym. Co więcej, autorski projekt hurtowni danych integrującej w sobie dane Big Data i dane z systemu transakcyjnego może przynieść potencjalne korzyści w postaci zwiększenia rentowności usług medycznych w organizacji, która to rozwiązanie wdroży.

Istotnym wnioskiem jest także fakt, że producenci narzędzi do wizualizacji danych przykładają obecnie dużą uwagę do intuicyjności ich obsługi, przez co kadra zarządzająca potrzebuje mniejszego wsparcia ze strony działów IT. To przyczynia się do skrócenia czasu konstruowania kokpitów menedżerskich i raportów, a w konsekwencji szybszego i trafniejszego podejmowania decyzji. Ponad to, z przeanalizowanych raportów wynika, iż firmy użytkujące systemy klasy BI są zadowolone z korzyści jakie przyniósł firmie.

Literatura

- [1] Arslan A. K., Colak C., Sarihan M. E., *Different medical data mining approaches based prediction of ischemic stroke*, computer methods and programs in biomedicine 130, 2016
- [2] Bondarev A., Zakirov D., *Data Warehouse on Hadoop Platform for Decision Support Systems in Education*, 2015 Twelve International Conference on Electronics Computer and Computation (ICECCO)
- [3] Devenport T.H., Dyche J., *Big data in big companies*

- [4] Hannula M., Pirttimaki V., *Business intelligence empirical study on the top 50 Finnish companies*, Journal of American Academy of Business, Marzec 2003
- [5] Kimball R., Ross M. *The Data Warehouse Toolkit Third Edition*, Wiley, 2013
- [6] Mach-Król M., *ANALIZA I STRATEGIA BIG DATA W ORGANIZACJACH*, Studies & Proceedings of Polish Association for Knowledge Management 74, 2015
- [7] Praca zbiorowa, *A review on application of data mining techniques to combat natural disasters*, Ain Shams Engineering Journal, 2016
- [8] Praca zbiorowa, *Big Data for Dummies*, John Wiley & Sons, Inc., 2013
- [9] Praca zbiorowa, *Scalable real-time OLAP on cloud architectures*, J. Parallel Distrib. Comput. 79–80, 2015
- [10] Romero O, Herrero V., Abelló A., Ferrarons J., *Turning Small Analytics on Big Data: Data Partitioning and secondary indexes in Hadoop ecosystem*, Information Systems 54, 2015
- [11] Sarka D., Lah M., Jerkic G., *Exam 70-463:Implementing a Data Warehouse with Microsoft® SQL Server® 2012*, O'Reilly Media, Inc., 2012
- [12] Stodder D., TDWI Best Practises Report - Business Driven Business Intelligence and Analytics. Achieving Value through Collaborative Business/IT Leadership, tdwi.org, 3. kwartał 2014
- [13] *Apache Scoop*, <http://sqoop.apache.org/> [01.09.2016]
- [14] *Beacony i mikrolokalizacja*, <http://www.computerworld.pl/news/403124/Beacony.i.mikrolokalizacja.html> [02.08.2016]
- [15] *Big Data Analytics - What it is and why it matters*, http://www.sas.com/en_us/insights/analytics/big-data-analytics.html [01.07.2016]
- [16] *Big Data - What it is and why it matters*, http://www.sas.com/en_th/insights/big-data/what-is-big-data.html [22.07.2016]
- [17] *Connecting to Hadoop Hive*, <http://kb.tableau.com/articles/knowledgebase/hadoop-hive-connection> [16.08.2016]
- [18] *Data sources in Power BI Desktop*, <https://powerbi.microsoft.com/en-us/documentation/powerbi-desktop-data-sources/> [26.08.2016]
- [19] *It Glossary*, <http://www.gartner.com/it-glossary/> [21.07.2016]
- [20] *Magic Quadrant for Advanced Analytics Platforms*, <https://www.gartner.com/doc/reprints?id=1-2YEIILW&ct=160210&st=sb> [01.07.2016]
- [21] *Studium przypadku z wdrożenia systemu do budżetowania InForum BI Studio w Santander Consumer Bank*, <http://prnews.pl/analizy/studium-przypadku-z-wdrozenia-systemu-do-budzetowania-inforum-bi-studio-w-santander-consumer-bank-3118696> [03.07.2016]
- [22] *Tableau & MongoDB: Visual Analytics on JSON at the Speed of Thought*, <http://www.tableau.com/about/blog/2015/6/tableau-mongodb-visual-analytics-json-speed-thought-39557> [05.08.2016]
- [23] *Welcome to Apache Hadoop*, <http://hadoop.apache.org/> [01.09.2016]
- [24] *What is Big Data Analytics*, <https://www01.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html> [07.07.2016]
- [25] *What Is Data Mining*, https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/process.htm [29.07.2016]

Porównanie wydajności wybranych technologii tworzenia usług sieciowych w aspekcie zastosowań w aplikacjach internetowych

Artur Gdula*, Małgorzata Plechawska-Wójcik

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

Streszczenie. Artykuł przedstawia analizę porównawczą wybranych technologii tworzenia usług sieciowych. Analiza polega na porównaniu usług wykorzystujących protokół SOAP z usługami typu RESTful oraz szablonów wspomagających tworzenie usług w języku Java: Apache CXF, Spring WS, Jersey. W pierwszej kolejności analiza bazuje na metrykach statycznych porównując przykładowe aplikacje. W drugim etapie analiza skupia się na kwestii wydajności i porównuje wybrane technologie pod kątem czasu odpowiedzi usługi.

Słowa kluczowe: usługi sieciowe; wydajność; metryki

* Autor do korespondencji.

Adres E-mail: artur.gdula@pollub.edu.pl

Performance comparison of selected web service development technology in web applications

Artur Gdula*, Małgorzata Plechawska-Wójcik

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

Abstract. The article presents a comparative analysis of selected web service development technology. The analysis consists on a comparison of services using SOAP with RESTful web service and frameworks support the creation of web service in Java: Apache CXF, Spring WS, Jersey. In the first the analysis is based on a static metrics, compares the sample applications. In the second stage, the analysis raise the issue of performance, and compares the selected technologies in terms response time service.

Keywords: web services, performance; metrics

*Corresponding author.

E-mail address: artur.gdula@pollub.edu.pl

1. Wstęp

Usługi Web Service to współcześnie termin pojawiający się bardzo często. Jest to spowodowane postępowaniem w technologiach informatycznych oraz ciągle rosnącymi wymaganiami biznesowymi. Usługi sieciowe mogą być rozwijane w oparciu o dwa style SOAP i REST [1]. Ten artykuł skupia się na zestawieniu szablonów aplikacji wspomagających tworzenie usług sieciowych wykorzystujących protokół SOAP i styl architektury oprogramowania REST. Ponadto zawiera zestawienie cech, wad i zalet porównywanych szablonów. Lista szablonów dostępnych w internecie, wspierających tworzenie usług sieciowych nie jest zbyt obszerna, najważniejszymi pozycjami są: Apache Axis2, Apache CXF, Java Web Services Development Pack / GlassFish, Jersey, Spring WS, Web Services Interoperability Technology. Analizie poddano trzy szablony Apache CXF, Spring WS i Jersey. Na wybór szablonów miało wpływ kilka istotnych kwestii. Apache CXF umożliwia proste i intuicyjne tworzenie usług, wspiera większość standardów specyficznych dla usług sieciowych, coraz częściej zastępuje poprzedniego lidera Apache Axis2. Spring bardzo często jest używany we wszelkiego rodzaju aplikacjach, ze względu na wiele modułów dostarczających rozwiązania dla różnych zagadnień technicznych. Szablon Jersey został wybrany, ze względu na swoją popularność,

którą zyskał między innymi dzięki swojej lekkości i prostocie. W procesie badawczym użyto metryk statycznych oraz dynamicznych, dlatego badania oraz analiza została podzielona na dwa etapy. W pierwszej kolejności analiza bazuje na metrykach statycznych, porównuje przykładowe aplikacje dostarczające usługi. Badania wydajnościowe zostały przeprowadzone z wykorzystaniem aplikacji klienckiej oraz programu SoapUI, umożliwiającego wykonywanie różnego rodzaju testów. Dodatkowym celem badań jest potwierdzenie lub obalenie prawdziwości tezy: SOAP jest wolniejszą, ale bardziej ustandaryzowaną technologią w odróżnieniu od REST, JSON jest dużo lepszym rozwiązaniem we współczesnych aplikacjach od języka znaczników XML. Wybór stylu to ważna decyzja dla projektantów i programistów podczas wdrażania systemów [1]. Artykuł pomaga określić którą z technologii oraz który z użytych szablonów sprawuje się najlepiej w określonych warunkach.

1.1. Style REST i SOAP

SOAP jest protokołem komunikacyjnym opartym o język znaczników, którego celem jest zastąpienie innych protokołów. SOAP określany jest następcą XML-RPC [2]. Początkowo protokół był wykorzystywany do wywołań RPC w sieci, jednak protokół był ciągle rozwijany i od wersji SOAP 1.1 zyskał własną strukturę komunikacyjną. „Interfejsy

usług SOAP są zwykle definiowane przez plik Web Service Description Language (WSDL), opisujący metody, parametry wejściowe i wyjściowe oraz schematy XML dla tych parametrów” [2]. SOAP wykorzystuje paradygmat koperty opartej na standardzie XML do przesyłania informacji i zdalnego wywoływania metod [3]. Sam protokół definiuje reguły przetwarzania przy pomocy węzłów oraz format wiadomości. Wiadomość SOAP – koperta może składać się z elementów: *header*, *body*, *fault*. ”Specyfikacja SOAP definiuje węzeł SOAP, jako jednostkę, która implementuje reguły przetwarzania komunikatów SOAP” [4].

W przeciwieństwie do SOAP, REST nie jest protokołem komunikacyjnym. REST jest najczęściej definiowany jako styl architektoniczny budowy oprogramowania. Usługi sieciowe tworzone zgodnie z tym stylem są często nazywane RESTful Web Services. W ostatnich czasie gwałtowny rozwój aplikacji mobilnych przyczynił się do wzrostu popularności technologii REST. Technologia umożliwia tworzenie lekkich i prostych interfejsów API, ułatwiających integracje zarówno dużych systemów jak i małych aplikacji. Systemy REST do komunikacji pomiędzy serwerem a klientem najczęściej wykorzystują protokół HTTP. „Usługi REST zapewniają dostęp do zasobów poprzez czasowniki protokołu HTTP, wykorzystywane metody to: PUT,DELETE, POST, GET, HEAD” [5]. Klient który chce uzyskać dostęp do zasobu musi wysłać odpowiednie żądanie z identyfikatorem URI. Identyfikatory pełnią kluczową rolę w architekturze REST, są to unikatowe adresy wskazujące na zasób, za ich pomocą odbywa się komunikacja pomiędzy dostawcą a odbiorcą usługi [4]. Wyżej wymienione operacje odpowiadają zaimplementowanym metodom na serwerze, które pozwalają na odczytanie, usunięcie oraz modyfikacje zbioru danych. Usługi REST mogą korzystać z różnych formatów wymiany danych ale najczęściej spotykanymi są pliki XML i JSON. W przeciwieństwie do SOAP przesyłana wiadomość zawiera mniej metadanych między innymi dlatego usługi RESTful są określane jako prostsze i lżejsze. Mimo to usługi te pozwalają na operacje na obiektach zawierających dane nawet do 5 terabajtów [6]. W przypadku REST do dostarczenia opisu usługi można skorzystać z bazującym na XML-u standardzie WADL(Web Application Description Language). Nie zawsze w praktyce jest on stosowany, jednak bardzo ułatwia integracje aplikacji oraz podobnie jak WSDL pozwala na automatyczne tworzenie usług.

1.2. Apache CXF, Jersey, Spring WS

Apache CXF jest szablonem wspomagającym tworzenie usług sieciowych dostępnym na licencji open-source. Usługi sieci Web z użyciem Apache CXF mogą być implementowane z wykorzystaniem różnych protokołów, takich jak SOAP, XML, JSON, RESTful HTTP i obsługiwać różne protokoły komunikacyjne, takie jak HTTP lub JMS (Java Message Service) [7]. Apache CXF pozwala na integracje ze Springiem, w dużej mierze ułatwia to tworzenie usług wykorzystując pliki konfiguracyjne Springa.

Jersey to lekki i prosty szablon służącym do tworzenia usług typu RESTful. Podobnie jak w przypadku Apache CXF, Jersey może być zintegrowany ze Springiem, wspiera JAX-

RS, pozwala na łatwe i szybkie tworzenie usług RESTful przy użyciu adnotacji. Ponadto szablon pozwala na negocjacje treści, dzięki temu komunikacja z usługą może odbywać się w dowolnie wybranym formacie.

Biblioteka Springa jest bardzo obszerna, ponadto jest rozszerzana przez poboczne projekty, obejmując: bezpieczeństwo, przepływy WWW, usługi sieciowe SOAP (usługi sieciowe REST są częścią głównej biblioteki), integracja w przedsiębiorstwie, przetwarzanie wsadowe, integracja z portalami społecznościowymi itd. Spring WS pozwala na stworzenie *contract-first* SOAP Web-Services, które w pierwszej kolejności wymagają stworzenia pliku WSDL lub XSD [8].

2. Metryki i metody

2.1. Metryki

Porównanie wybranych technologii zostało przeprowadzone w oparciu o metryki statyczne oraz metryki dynamiczne. Metryka oznacza dowolną wartość liczbową charakteryzującą oprogramowanie [9]. Według standardu IEE 1061-1998 metryka to funkcja odwzorowująca jednostkę oprogramowania w wartość liczbową, ta wyliczona wartość jest interpretowalna jako stopień spełnienia pewnej własności jakości jednostki oprogramowania [10]. Metryka oprogramowania jest ważnym czynnikiem pozwalającym odróżnić podobne produkty oferowane przez różnych dostawców [11]. Podstawowy podział metryk wyróżnia metryki statyczne i dynamiczne. W metrykach statycznych wyróżnia się metryki rozmiaru, obiektowe, złożoności, pakietów. Metryki statyczne odnoszą się do analizy kodu, są szczególnie przydatne w trakcie implementacji aplikacji. Do metryk dynamicznych zalicza się metryki pozwalające określić zachowanie już działającego produktu, pozwalają określić jak działa napisana usługa, czy spełnia wymagania wydajnościowe i oczekiwania klienta. Wszystkie metryki aby spełniały swoją rolę powinny charakteryzować się prostotą, niezależnością od języka, powinny odzwierciedlać przydatną i konkretną informację oraz powinny dać się obliczyć przez komputer. Poniżej znajdują się definicje poszczególnych metryk.

2.1.1. Metryki statyczne

Liczba linii kodu- Najprostszą miarą oprogramowania, należąca do metryk rozmiaru, jest metryka rozpoznawana pod skrótem LOC(*lines of code*) lub SLOC(*sources lines of code*). Metryka NCLOC jest liczbą linii kodu wykluczając puste linie i linie komentarzy. Na podstawie liczby linii kodu można wyliczyć inne metryki pochodne takie jak jakość, ilość dokumentacji. W części badawczej została użyta do porównania tej samej usługi napisanej przy użyciu różnych technologii, liczba linii kodu liczona będzie zarówno dla całych klas, interfejsów, metod oraz plików konfiguracyjnych.

Liczba klas i interfejsów - Metryka statyczna pozwalająca na określenie liczby klas i interfejsów w aplikacji. Metrykę

wykorzystano do zbadania liczby klas i interfejsów trzech aplikacji spełniających tą samą funkcjonalność.

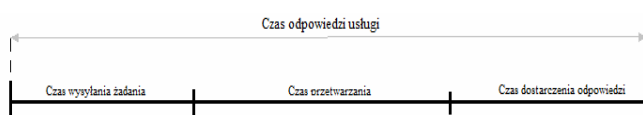
Liczba metod - Prosta metryka pozwalająca określić ilość metod w klasie, interfejsie czy też całej aplikacji. Metryka liczby metod podobnie jak metryka liczby klas i interfejsów została wykorzystana do porównania aplikacji spełniających tą samą funkcjonalność.

Liczba niezbędnych bibliotek - Metryka porównuje liczbę niezbędnych bibliotek potrzebnych do wystawienia usługi w porównywanych szablonach aplikacji. Liczba wymaganych bibliotek w każdym szablonie może być różna, oddziałuje to na inne metryki.

Rozmiar aplikacji - Porównanie aplikacji pod względem rozmiaru na dysku, ilości plików i folderów. W części badawczej zestawiono porównania kodu źródłowego oraz otrzymanego artefaktu w postaci pliku war. Na rozmiar aplikacji może wpływać ilość wykorzystywanych bibliotek, ilość napisanego kodu, ilość folderów i plików. Ta miara została wyrażona w KB.

2.1.2. Metryki dynamiczne

Czas odpowiedzi usługi - „*Service Response Time (SRT)* jest to czas odpowiedzi usługi jaki upłynął pomiędzy wysłaniem żądania a dostarczeniem usługi” [12]. Wyniki badań zostały zaprezentowane w milisekundach(ms). W związku z tym w celu dokładniejszego zbadania problemu czas odpowiedzi podzielono na trzy etapy. Badanie czasu odpowiedzi serwera przeprowadzono przy użyciu aplikacji klienckiej, testów funkcjonalnych, testów obciążeniowych oraz skryptów w narzędziu SoapUI. Rysunek 1 poniżej obrazuje badany czas odpowiedzi z podziałem na czas wysyłania żądania, czas przetwarzania oraz czas dostarczenia odpowiedzi.



Rys 1. Czas odpowiedzi usługi

Obciążenie serwera - metryka pozwalająca określić obciążenie komputera np. zajętość procesora, zajętość pamięci operacyjnej. W tym celu wykorzystano zaawansowane funkcje testów obciążeniowych w programie SoapUI oraz księgarnie internetową.

2.2. Metody pomiaru

Badania wydajnościowe zostały przeprowadzone w lokalnej sieci WIFI na dwóch komputerach (dostawca i konsument usług). W celu otrzymania jak najdokładniejszych wyników, badania przeprowadzono na komputerach przy jak najmniejszej aktywności pamięci i procesora, wszystkie zbędne procesy i zadania systemu

zostały wyłączone. Ponadto sieć została udostępniona jedynie dla dwóch komputerów, a czas obydwu komputerów został zsynchronizowany. Aplikacje dostarczające usługi uruchamiano kolejno na „czystym” kontenerze aplikacji webowych Tomcat 7.0.68. Do badania metryk dynamicznych z wykorzystaniem aplikacji klienckiej użyto metod *nanotime*, *currentTimeMillis* z pakietu JDK oraz biblioteki Apache log4j do logowania czasu w plikach tekstowych. Metoda *currentTimeMillis()* zwraca aktualny czas w milisekundach. Dodatkowo testy funkcjonalne i wydajnościowe zostały przeprowadzone przy użyciu narzędzia SoapUI 5.2.1.

3. Badania

Badania zostały przeprowadzone na trzech aplikacjach działających po stronie serwera, opartych na szablonach Apache CXF, Jersey, Spring WS. Aplikacje dostarczają usługi SOAP i REST. Początkowo aplikacje dostarczały jedynie usługi REST, natomiast po zaimplementowaniu usług SOAP aplikacje dostarczały zarówno usługi REST jak i SOAP. Klientami a zarazem narzędziami pomiarowymi są: aplikacja księgarni oraz program SoapUI. W pierwszej kolejności porównaniu poddane zostały aplikacje dostarczające usługi oferujące tą samą funkcjonalność w aspekcie metryk statycznych. Tabela 1 przedstawia rezultat porównania trzech aplikacji pod względem wykorzystania zasobów komputera. W przypadku pierwszych metryk odnoszących się do kodu źródłowego (liczba linii kodu, liczba metod itd.), najlepiej wypadł szablon Jersey, natomiast najbardziej rzuca się w oczy słaby wynik aplikacji Spring WS. Początkowo wydawać by się mogło, że to Apache CXF ze względu na swoją złożoną strukturę, wielkość bibliotek oraz przeznaczenie głównie dla usług sieciowych w tych metrykach będzie sprawował się najslabiej. Mimo że, różnica pomiędzy projektami Apache CXF i Spring WS jest dość spora, zaimplementowanie dodatkowej usługi SOAP w każdym z projektów wymaga dodania podobnej ilości plików i dodanych linii kodu. Warty zważenia jest fakt, że aplikacja zaimplementowana z użyciem Spring WS potrzebuje najmniej bibliotek oraz po zbudowaniu i umieszczeniu w serwerze Tomcat zajmuje najmniej miejsca. Spowodowane jest to tym, że aplikacje oparte o Apache CXF i Jersey poza własnymi bibliotekami, korzystają dodatkowo z modułów szablonu Spring takich jak Spring Web, Spring Context, natomiast aplikacja oparta o Spring WS korzysta głównie z modułów Spring. Ponadto moduły w Springu są podzielone na wiele komponentów oraz zostały zaimplementowane w ten sposób aby móc działać samodzielnie. Natomiast Apache CXF posiada mniej modułów od Spring WS, są one specyficzne dla usług sieciowych, obszerniejsze, bardziej złożone oraz wspierają więcej standardów usług sieciowych.

Tabela 1. Porównanie stworzonych aplikacji z wykorzystaniem wybranych szablonów.

	Apache CXF		Spring WS		Jersey
<i>Rodzaj dostarczanych usług:</i>	REST	REST/SOAP	REST	REST/SOAP	REST
Liczba linii kodu	893	968	1282	1367	865
Liczba klas	15	17	22	24	15
Liczba interfejsów	5	6	5	6	5
Liczba metod	96	99	112	115	96
Rozmiar kodu źródłowego w kB	76	84	108	116	72
Ilość plików	23	25	33	36	25
Ilość plików konfiguracyjnych(xml, properties)	6	6	7	8	7
Ilość folderów	20	20	19	20	18
Liczba niezbędnych bibliotek(po skompilowaniu)	63	63	42	45	72
Rozmiar niezbędnych bibliotek(po skompilowaniu) w kB	26214	26214	20992	21504	25804
Rozmiar spakowanego programu(war) w kB	22732	23450	18739	19148	22835
Rozmiar aplikacji w tomiecie(folder w webapps) w kB	25498	26316	18739	19149	22835

Po porównaniu kodu aplikacji przeprowadzono testy wydajnościowe, wyniki testów znajdują się w tabelach 2-5. Badania przeprowadzono z użyciem narzędzia Soap UI oraz aplikacją kliencką. Narzędzia nie zawsze odwzorowują prawdziwego klienta dlatego też dla celów badań wydajnościowych zaimplementowano księgarnię internetową, konsumenta usług sieciowych. Jak można zauważyć w wynikach badań czasy odpowiedzi dla Soap UI różnią się od czasów odpowiedzi klienta-księgarni internetowej. Na podstawie tej obserwacji można dojść do wniosku, że nie warto polegać tylko i wyłącznie na różnego rodzaju narzędziach.

Tabela 2. Czas odpowiedzi usługi *getAllBooks* zmierzony aplikacją kliencką.

	Apache CXF	
<i>Rodzaj dostarczanych usług:</i>	REST	SOAP
Czas wysyłania żądania w ms	37	242
Czas przetwarzania w ms	36	34
Czas dostarczenia odpowiedzi w ms	3548	362
Kompletny czas odpowiedzi usługi w ms	3621	3901
Kompletny czas odpowiedzi usługi w ms – wywołanie usługi 20 razy pod rząd	47153	73087

Tabela 3. Czas odpowiedzi usługi *getAllBooks* zmierzony narzędziem SoapUI.

	Apache CXF		Spring WS		Jersey
<i>Rodzaj dostarczanych usług:</i>	REST	SOAP	REST	SOAP	REST
Czas wysyłania żądania w ms	15	75	15	28	16
Czas przetwarzania w ms	48	29	59	38	38
Czas dostarczenia odpowiedzi w ms	1176	2358	1177	2830	1144
Kompletny czas odpowiedzi usługi w ms	1253	2470	1253	2899	1201
Kompletny czas odpowiedzi usługi w ms – wywołanie usługi 20 razy pod rząd	26866	48482	25506	56819	25468

Tabele 2 i 3 świadczą o tym że REST w każdym z badań przewyższa usługi SOAP. W przypadku jednokrotnego wywołania usługi REST najlepiej sprawuje się aplikacja oparta o szablon Jersey, niewiele mniejszy czas odpowiedzi mają pozostałe aplikacje. Ciekawa sytuacja występuje podczas wywołania usługi dwudziestokrotnie, aplikacja oparta o szablon Apache CXF spowalnia i znajdują się na

ostatnim miejscu. Analizując dane z tabeli 3 można stwierdzić, że aplikacja oparta o szablon Apache CXF doskonale radzi sobie z usługami SOAP, jednak jeśli chodzi o usługi REST im większy rozmiar danych, większe obciążenie czas odpowiedzi usługi wzrasta. Warto również wspomnieć o aplikacji wykorzystującej Spring WS, która nie ustępuje znacząco aplikacji wykorzystującej Apache CXF, w przypadku usług SOAP oraz aplikacji wykorzystującej szablon Jersey w przypadku usług REST. Na podstawie danych liczbowych zawartych w tabelach 2 i 3 można zauważyć, że bez względu na rodzaj usługi najwięcej czasu pochłania zwrócenie danych do klienta. Tabela 4 zawiera daną świadczącą o obciążeniu serwera w trakcie korzystania z usług dostarczanych przez trzy aplikacje. Obciążenie serwera mierzono z wykorzystaniem programu Soap UI. W wyniku przeprowadzonych badań okazało się, że aplikacja oparta o szablon Apache CXF wykorzystuje w największym stopniu procesor i pamięć operacyjną zarówno w przypadku

Tabela 5. Wpływ formatu danych na przykładzie usługi *getAllBooks*.

	Apache CXF		Spring WS		Jersey	
<i>Format danych:</i>	XML	JSON	XML	JSON	XML	JSON
Minimalny czas odpowiedzi w ms	1020	827	960	931	972	835
Maksymalny czas odpowiedzi w ms	1196	864	1334	797	1099	858
Średni czas odpowiedzi w ms	1080	847	1101	855	1035	843

Tabela 4. Obciążenie serwera w trakcie wywoływania usługi *getAllBooks*.

	Apache CXF		Spring WS		Jersey
<i>Rodzaj dostarczanych usług:</i>	REST	SOAP	REST	SOAP	REST
Wykorzystanie procesora w procentach podczas wywoływania pojedynczej usługi	6	6	4	4	4
Wykorzystanie pamięci operacyjnej w procentach podczas wywoływania pojedynczej usługi	34	34	33	33	32
Wykorzystanie procesora w procentach podczas wywoływania 20 razy pod rząd tej samej usługi	7	8	5	6	5
Wykorzystanie pamięci w procentach podczas wywoływania 20 razy pod rząd tej samej usługi	35	36	34	34	33
Wykorzystanie procesora w procentach podczas testów obciążeniowych SoapUI w procentach	11	11	8	9	8
Wykorzystanie pamięci w procentach podczas testów obciążeniowych SoapUI	36	37	36	36	34

jednokrotnego wywołania usługi jak i przy pozostałych testach. Pozostałe aplikacje sprawują się lepiej, ale jest to mała różnica. Na obciążenie serwera nie wpływa również to czy wywołano usługę REST czy SOAP.

Analizując dane z tabeli 5 można dojść do wniosku, że warto dobrze dobrać format danych w używanych usługach. Na podstawie danych można wnioskować, że format JSON sprawują się dużo lepiej od formatu XML. Natomiast nie widać znaczącej różnicy w czasach odpowiedzi w zależności od użytego szablonu.

4. Wnioski

W analizie opisano czego może spodziewać się każdy programista wybierając konkretną technologię. Metryki proste wskazują programiście czego może się spodziewać podczas implementacji, natomiast metryki dynamiczne wskazują programiście jak dana aplikacja zachowa się w trakcie działania. W zależności od przeznaczenia tworzonej aplikacji wskazany może być wybór konkretnej technologii. Dla złożonych aplikacji biznesowych, wymagających dużego stopnia bezpieczeństwa, mogących w przyszłości rozrosnąć się o nowe funkcjonalności, zaleca używać się standardu SOAP. REST charakteryzuje się prostotą i sprawuje się dużo szybciej więc użycie go zaleca się w sytuacjach gdy priorytet ma szybkość działania.

Szablonem najlepiej sprawującym się w przypadku usług REST okazał się Jersey, jest lekkim, szybkim i prostym w użyciu szablonem. Z pewnością najlepiej sprawdził by się w prostych aplikacjach internetowych. W przypadku usług SOAP największe wsparcie dla innych technologii ma

szablon Apache CXF, ponadto okazał się najlepszy pod względem czasu odpowiedzi, jednak skutkowało to większym obciążeniem serwera. Dla programisty duże znaczenie może mieć technika tworzenia usług, szablon Spring WS wspiera jedynie technikę *contract-first*. Pomiar metryk dynamiczny wskazał że format JSON jest dużo wydajniejszy od formatu XML.

Badania metryk kodu wykazały, że aplikacja oparta Spring WS jest największa pod względem liczby metod i linii kodu. Jednak po zbudowaniu aplikacji liczba jej bibliotek i rozmiar jest najmniejszy spośród porównywanych aplikacji.

Nie jest zabronione stosowanie protokołu SOAP dla prostych aplikacji, a usług REST dla złożonych aplikacji, zależy to od przeznaczenia aplikacji, zespołu programistycznego itp. Wyników badań i wniosków nie należy traktować jako wytycznych podczas wyboru technologii, mogą one jedynie służyć jako wskazówki. Stworzone aplikacje oferowały jedynie proste funkcjonalności, dalszy rozwój aplikacji można stosować w celu rozbudowy o nowe funkcjonalności, moduły oraz przeprowadzenia dodatkowych badań.

Literatura

- [1] K. Smita, S. Kumar Rath. Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration. Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on. IEEE, 2015.
- [2] M. Lanthaler, C. Gütl. Towards a RESTful service ecosystem. Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on. IEEE, 2010.
- [3] G. Mulligan, D. Gračanin. A comparison of SOAP and REST implementations of a service based interaction independence middleware framework. Simulation Conference (WSC), Proceedings of the 2009 Winter. IEEE, 2009.
- [4] Z. Fryźlewicz, A. Salamon, Podstawy architektury i technologii usług XML sieci WEB, Warszawa : Wydawnictwo Naukowe PWN : Mikom, 2008
- [5] A. Rodriguez. Restful web services: The basics. IBM developerWorks(2008).
- [6] B. P. Upadhyaya, Rest client pattern. Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on. IEEE, 2014.
- [7] N. Balani, R. Hathi. Apache CXF Web Service Development: Develop And Deploy SOAP And Restful Web Services. Birmingham: Packt Publishing, 2009. eBook Collection (EBSCOhost). Web. 6 May 2016.
- [8] H. Sattari, S. Kunjumohamed. Spring Web Services 2 Cookbook : Over 60 Recipes Providing Comprehensive Coverage Of Parctical Real-Life Implementations Of Spring-WS. Birmingham: Packt Publishing, 2012. eBook Academic Collection (EBSCOhost). Web. 29 July 2016.
- [9] L. Westfall, 12 Steps to Useful Software Metrics. Proceedings of the Seventeenth Annual Pacific Northwest Software Quality Conference. Vol. 57. 2005
- [10] Software & Systems Engineering Standards Committee. IEEE Std 1061-1998—IEEE standard for a software quality metrics methodology. IEEE Computer Society, Tech. Rep (1998).
- [11] N. Thio, S. Karunasekera, Automatic measurement of a qos metric for web service recommendation. Software Engineering Conference, 2005. Proceedings. 2005 Australian. IEEE, 2005.
- [12] J. Sun Her ; S. Won Choi ; S. Hun Oh ; S. Dong Kim, A framework for measuring performance in service-oriented architecture. Next Generation Web Services Practices, 2007. NWeSP 2007. Third International Conference on. IEEE, 2007.

Metoda rejestracji i wizualizacji ruchu urządzenia mobilnego

Maciej Kulig*, Jakub Smółka

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

Streszczenie: Artykuł przedstawia możliwość wykorzystania urządzeń mobilnych do rejestracji i wizualizacji ruchu. Przeprowadzone badania dotyczą urządzeń z systemem Android. Na potrzeby badań została napisana aplikacja, która umożliwia odczyt danych z sensora przyspieszenia oraz magnetometru w trzech osiach X, Y i Z. Badania zostały dokonane na dwóch urządzeniach o podobnych parametrach. Wybrane wyniki zostały przedstawione w postaci tabel.

Słowa kluczowe: urządzenia mobilne; rejestracja ruchu; akcelerometr

*Autor do korespondencji

Adresy E-mail: kulig.maciej@pollub.edu.pl, jakub.smolka@pollub.pl

A method for logging and visualization of mobile device motion

Maciej Kulig*, Jakub Smółka

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

Abstract. The article presents the possibility to use mobile devices for recording and visualizing motion. Research focuses on Android devices. The study was written application for mobile devices that allows reading data from the magnetic sensor and acceleration sensor in three axes X, Y and Z. The tests were made on two devices with similar characteristics. Selected results are presented in the tables.

Keywords: mobile devices; acceleration sensor; logging motion device

*Corresponding author:

E-mail addresses: kulig.maciej@pollub.edu.pl, jakub.smolka@pollub.pl

1. Wstęp

W ciągu ostatnich lat popularność urządzeń mobilnych nieustannie rośnie [1,2]. Urządzenia te w życiu człowieka odgrywają coraz większą rolę. Dzięki bardzo szybkiemu rozwojowi technologii i nieustannej miniaturyzacji dzisiejsze urządzenia mobilne są wyposażone w wiele czujników tj.: czujnik zbliżeniowy, akcelerometr, magnetometr, żyroskop, czujnik światła i wiele innych. Z każdym rokiem rośnie liczba aplikacji dostępnych na urządzenia które wykorzystują czujniki w urządzeniach mobilnych [3,4,5]. Czujnik przyspieszenia jest wykorzystywany np.: do sterowania postacią w grach czy sterowania autem. Istnieją też ciekawsze zastosowania akcelerometru np.: wykrywanie zdarzenia upadku [6]. Magnetometr jest wykorzystywany w różnego rodzaju wirtualnych kompasach. Zaimplementowana aplikacja pozwala na rejestrację przyspieszenia jakie działało na urządzenie w trzech osiach x, y i z. Aby móc dokonać pomiarów potrzebny jest dostęp do sensorów wbudowanych w urządzenie.

Dostęp do sensorów na platformie Android jest zapewniony poprzez implementację odpowiedniego interfejsu znajdującego się w systemie. Dane które system przekazuje do zaimplementowanego interfejsu opakowane są w obiekt `SensorEvent`. Zgromadzone dane zostały zaprezentowane na wykresie liniowym z bezpłatnej biblioteki `GraphView`. Biblioteka ta jest napisana w języku Java oraz jest rozpowszechniana na zasadzie licencji `GPLv2` dzięki czemu

można swobodnie ją modyfikować i wykorzystywać w aplikacjach [7].

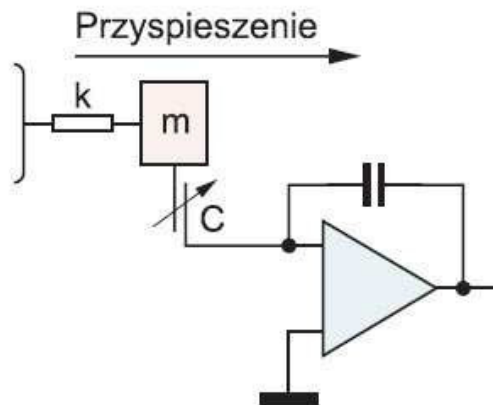
W artykule przedstawiono możliwość rejestracji i wizualizacji ruchu w urządzeniach mobilnych z systemem operacyjnym Android. Aplikacja została zainstalowana na dwóch urządzeniach mobilnych - telefonie i tablecie. Oba urządzenia posiadały podobne parametry techniczne oraz sensory tych samych producentów. Przedstawione zostały wyniki zgromadzone przy pomocy opracowanej aplikacji.

2. Budowa i wykorzystanie akcelerometru i magnetometru w urządzeniach mobilnych

Zdecydowana większość urządzeń mobilnych jest wyposażona w akcelerometr i magnetometr. Akcelerometr jest to elektromechaniczne urządzenie, które pozwala mierzyć siły przyspieszenia w trzech osiach. Na rynku występuje wiele rodzajów akcelerometrów, większość z nich to akcelerometry piezoelektryczne, które charakteryzują się dużymi rozmiarami i nie nadają się do coraz to mniejszych urządzeń mobilnych. W roku 1979 na uniwersytecie Stanforda stworzono pierwszy akcelerometr w technologii MEMS (od ang. microelectromechanical system) [8]. Akcelerometry tego typu są wykorzystywane w wielu dziedzinach np.:

- a) medycyna – pompy infuzyjne, cewniki mierzące ciśnienie krwi,
- b) motoryzacja – czujniki w poduszkach powietrznych, ESP (elektroniczny program stabilizacji), czujniki ciśnienia w kołach,

- c) fotografia – optyczna i elektroniczna stabilizacja obrazu,
- d) sport i rozrywka – krokomierze, aplikacje mierzące aktywność.



Rys. 1. Schemat ideowy akcelerometru z masą bezwładną[8]

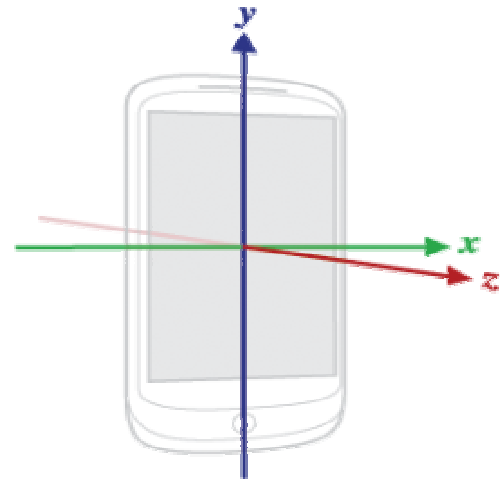
Na rysunku 1 przedstawiono schemat ideowy z masą bezwładną. Najważniejszym elementem tak zaprojektowanego akcelerometru jest masa bezwładna, która jest przymocowana na sprężystych belkach, sprawuje ona rolę elektrody w układzie kondensatora pomiarowego [8]. W następstwie przemieszczenia się wyżej wymienionej masy zmienia się pojemność a co za tym idzie zmienia się napięcie wyjściowe, które w następnym kroku jest przetwarzane przez przetwornik A/C.

Magnetometry czyli inaczej sensory pola magnetycznego są projektowane w oparciu o kilka zjawisk fizycznych, których wybór zależy od charakterystyki mierzonego pola magnetycznego. Można przyjąć że najbardziej popularnym magnetometrem jest czujnik indukcyjny, ponieważ jest to w istocie cewka w której indukowane napięcie zależy od indukcji zmiennego pola magnetycznego. Najczęściej jest wykorzystywany do centrowania map tak aby mapa znajdowała się w pozycji pokrywającej się z kierunkami świata.

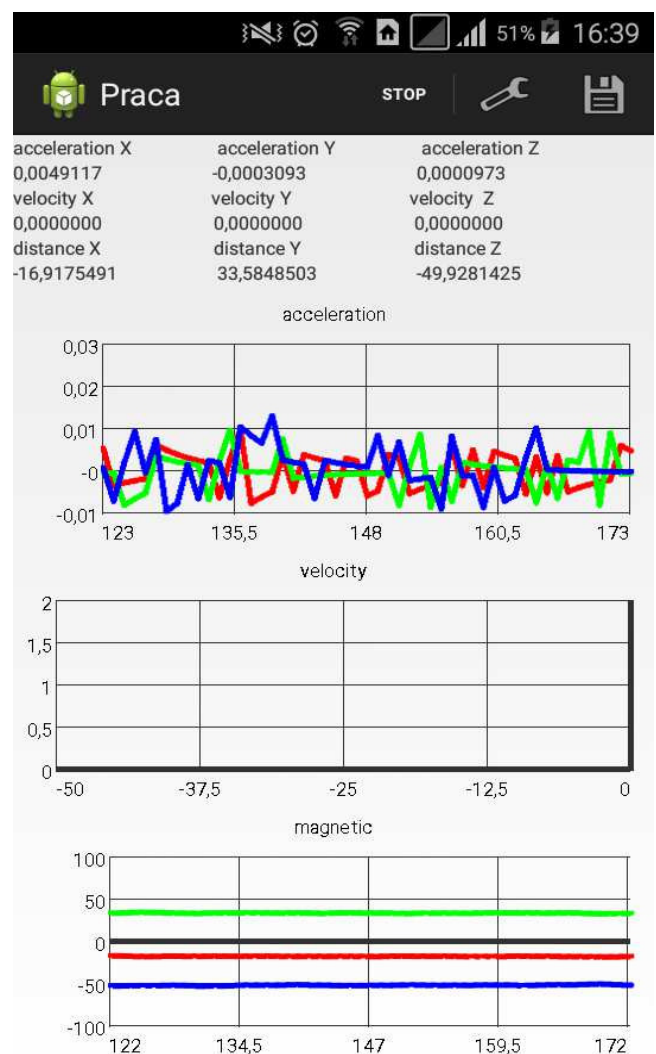
Warto tutaj wspomnieć że pomiary dokonywane na urządzeniach z systemem Android dokonywane są względem układu współrzędnych urządzenia. Układ współrzędnych telefonu przedstawiony został na rysunku 2.

Ważnym aspektem jest też dokładność i czułość tych sensorów. Urządzenie w stanie spoczynku leżące na biurku wskazywało wartości przyspieszenia rzędu $\pm 0.03 \frac{m}{s^2}$ dla tabletu i $\pm 0.02 \frac{m}{s^2}$ dla telefonu.

Następnym ważnym elementem jest czułość oraz podatność na zakłócenia pochodzące z zewnętrznych źródeł. Oba te sensory są bardzo czułe i każdy najmniejszy ruch powoduje zmianę wartości mierzonego sygnału. Na rysunku 3 przedstawiono odczyty z sensorów na urządzeniu, które znajdowało się w spoczynku tzn. leżało nieruchomo na biurku.



Rys. 2. Układ współrzędnych urządzenia z którego korzysta API sensorów[9].



Rys. 3. Wskazania sensorów podczas spoczynku

3. Aplikacja

Na potrzeby prowadzonych badań została napisana aplikacja na platformę Android, która umożliwia pomiar przyspieszenia. Zaimplementowano w niej dwie metody całkowania numerycznego - metodę Heuna i metodę Verleta

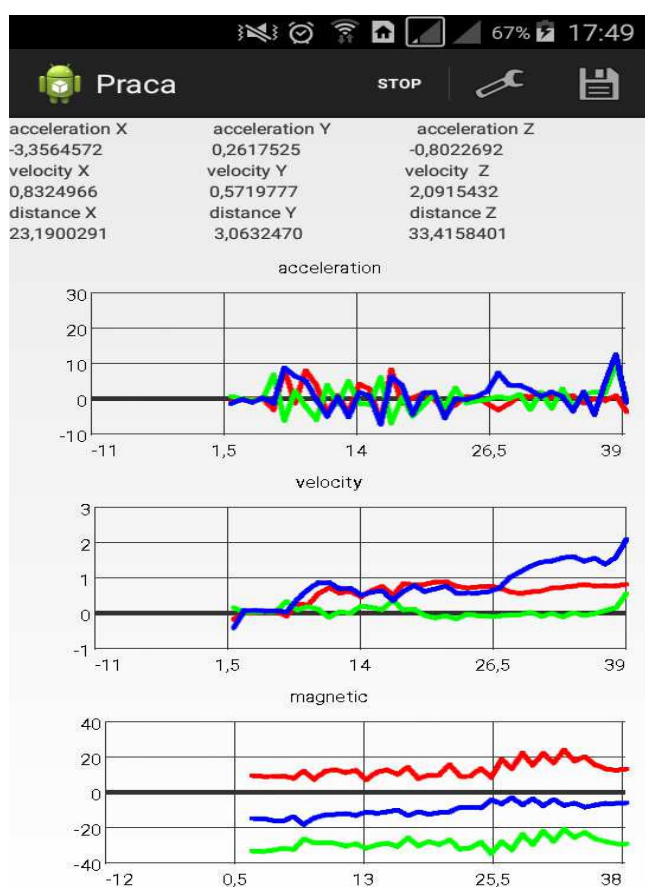
[10,11]. Obie metody posłużyły do przeliczenia drogi przebytej przez urządzenie. Aplikacja pomiarowa wykorzystuje dwa rodzaje sensorów akcelerometr i magnetometr. Dane z magnetometru są niezbędne do przeliczenia wartości sygnału przyspieszenia z układu współrzędnych urządzenia na układ współrzędnych świata.

Stworzona w ramach badania praca umożliwia użytkownikowi wybór wcześniej wspomnianych metod całkowania oraz częstotliwości próbkowania.

Standardowo w systemie Android są dostępne cztery częstotliwości próbkowania [12]:

- a) NORMAL – 5Hz
- b) UI – 16,66 Hz
- c) GAME – 50 Hz
- d) FASTEST – 100Hz

Android umożliwia też ustawienie własnego odstępu pomiędzy rejestracją kolejnych zdarzeń [13]. Aplikacja umożliwia użytkownikowi zapisanie zgromadzonych danych do pliku CSV. Dane odczytane z sensorów są w czasie rzeczywistym prezentowane na wykresach oraz polach tekstowych, które znajdują się nad wykresami. Na rysunku 4 przedstawiono przykładowy ekran główny aplikacji.



Rys. 4. Ekran główny aplikacji

Na rysunku 2.1 przedstawiono przykładowy zrzut ekranu podczas pomiaru. Jak widać główny ekran można podzielić na dwie części górną i dolną. Górna część składa się z pól

tekstowych, w których w czasie rzeczywistym są prezentowane dane. Natomiast dolna część składa się z trzech wykresów, na których w sposób graficzny są przedstawione dane odczytane z sensorów z których można wydzielić trzy wykresy liniowe zaczynając od góry:

- a) wykres przyspieszenia
- b) wykres prędkości
- c) wykres pola magnetycznego

4. Przykładowe odczyty

Jedną z hipotez rozpatrywanych przez autorów jest możliwość wykorzystania stworzonej aplikacji do rejestracji i wizualizacji ruchu urządzeń mobilnych. Badania zostały przeprowadzone na dwóch urządzeniach mobilnych z systemem Android. Producentem telefonu jest firma Samsung wiodący gracz na rynku smartfonów. Telefon jest urządzeniem ze średniej półki co może sugerować lepszej jakości podzespoły i dokładniejsze odczyty sensorów. Natomiast tablet pochodzi od firmy LG, model to G PAD 8.0, urządzenie jest niższej jakości niż telefon. Oba urządzenia posiadają podobne parametry i sensory tych samych producentów. Parametry sensorów różniły się maksymalnym zakresem, rozdzielczością [14]. Dane na temat sensorów odczytane z urządzeń zostały przedstawione w tabeli 3.1.

Pierwsze pomiary były dokonane względem współrzędnych świata a następnie te same badania zostały powtórzone w układzie współrzędnym urządzenia (tzn. bez przeliczania na współrzędne świata). Każdy z pomiarów był dokonany dla obydwu zaimplementowanych metod całkowania numerycznego.

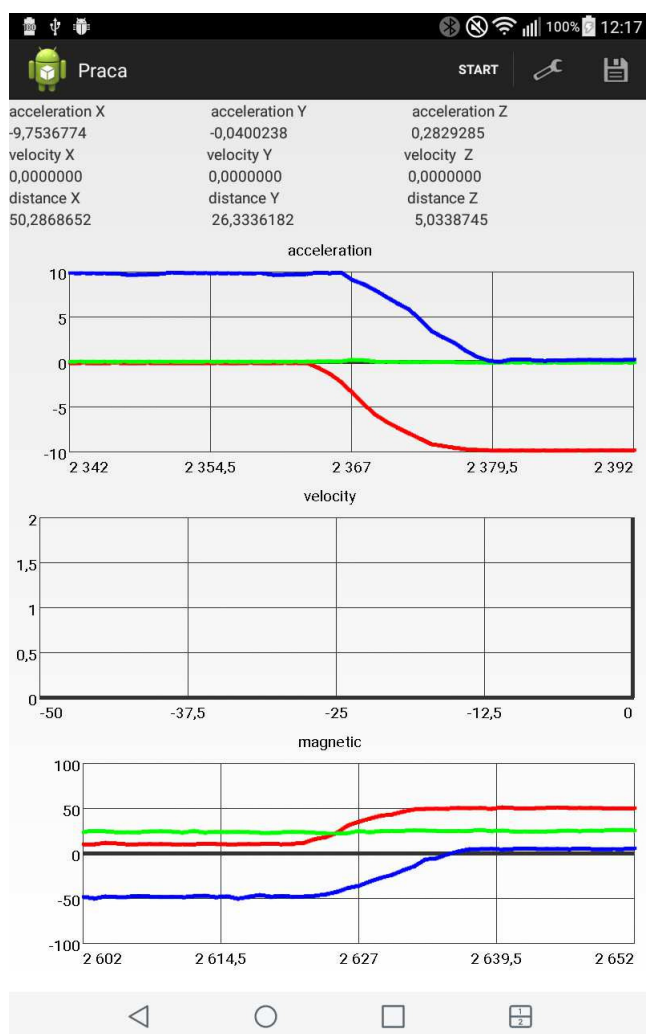
Pomiary były dokonywane na gładkiej powierzchni na dystansie 50 i 100 cm. Każde badanie było powtórzone kilkukrotnie a w tabelach wynikowych zostały przedstawione wyniki średnie z pięciu prób.

Tabela 1. Parametry sensorów obu urządzeń[14,15]

Samsung Galaxy A3	LG G Pad 8
akcelerometr	
producent: Bosch Sensortec	producent: Bosch
wersja: 1	wersja: 1
maksymalny zakres: 19,6133003235 $\frac{m}{s^2}$	maksymalny zakres: 39,2400054932 $\frac{m}{s^2}$
rozdzielczość: 0,0095768068 $\frac{m}{s^2}$	rozdzielczość: 0,0766143799 $\frac{m}{s^2}$
natężenie prądu: 0.13 mA	natężenie prądu: 0.13 mA
magnetometr	
producent: Bosch Sensortec	producent: Bosch
wersja: 1	wersja: 1

maksymalny zakres: 2000,00 μT	maksymalny zakres: brak danych
rozdzielczość: 0,0599999987 μT	rozdzielczość: brak danych
natężenie prądu: 6.0 mA	natężenie prądu: 0.5 mA

Pomiary były dokonywane na gładkiej powierzchni na dystansie 50 i 100 cm. Każde badanie było powtórzone kilkakrotnie a w tabelach wynikowych zostały przedstawione wyniki średnie z pięciu prób.



Rys. 5. Wykres podczas obracania telefonu

Na rysunku 5 przedstawiono wykres, na którym widać jak na początku pomiaru urządzenie znajduje się w spoczynku i pozycji „naturalnej” (wyświetlaczem skierowanym ku górze) a następnie jest obracane na prawy bok. Widać jak na wykresie siła przyspieszenia ziemskiego, która działa wzdłuż osi Z (kolor niebieski na wykresie) zmienia wartość z ok $10 \frac{m}{s^2}$ do wartości 0 a wartość wzdłuż osi X (na wykresie kolor czerwony) zmienia swoją wartość od 0 do $-10 \frac{m}{s^2}$.

Jak widać na trzecim wykresie pole magnetyczne też ulega zmianie rośnie ono w dwóch osiach X i Z natomiast w osi Y

pozostaje ono bez zmian tak jak wartość prędkości która przez cały wykres jest równa 0.

W tabeli 2 przedstawiono wyniki pomiarów przemieszczenia wzdłuż osi X na dystansie 50 cm. Każdy z pomiarów trwał ok dwóch sekund urządzenia były przesuwane wzdłuż blatu biurka powoli żeby wykluczyć wszystkie błędy pochodzące z gwałtownych ruchów.

Tabela 2. Pomiar wzdłuż osi X na dystansie 50cm

Urządzenie	Rzeczywisty dystans [m]	Dystans obliczony [m]	Metoda całkowania	Częstotliwość [Hz]
LG G Pad	0,5	0,4538314	Heun	5
		0,21345827		16,66
		0,08707613		50
		0,013238691		100
		0,6939494	Verlat	5
		0,16029789		16,66
		0,07883833		50
		0,08735664		100
Samsung A3	0,5	0,46926934	Heun	5
		0,2477175		16,66
		0,043609627		50
		0,026183326		100
		0,5060962	Verlat	5
		0,14744999		16,66
		0,04510415		50
		0,009412151		100

Jak widać w wynikach przedstawionych w tabeli 2, wartości najbliższe rzeczywistemu przemieszczeniu były pomiary obliczone na podstawie pomiarów wykonanych z częstotliwością 5Hz. Różnica pomiędzy rzeczywistym przemieszczeniem a przemieszczeniem obliczonym wyniosła ok 5cm co stanowi bardzo dobry wynik. Reszta wyników znacząco odbiega od rzeczywistego przemieszczenia. Najgorsze wyniki uzyskano przy próbkowaniu z najwyższą częstotliwością. Obliczone przesunięcia nie przekroczyły nawet 10% rzeczywistego przemieszczenia. Takie różnice mogły być spowodowane zakłóceniami magnetometru który wskazywał błędne dane co powodowało błędy w dalszych obliczeniach.

Podczas pomiarów na dystansie 100 cm wszystkie pomiary i obliczenia odbiegały od rzeczywistych. Dla najmniejszej wartości próbkowania 5Hz odległość obliczona na podstawie wskazań akcelerometru i magnetometru wyniosła ponad 200 cm więc jak widać dystans przebyty przez urządzenie był dwukrotnie mniejszy od obliczonego. Obliczenia dla pozostałych częstotliwości były bardzo podobne do wartości obliczonych przy krótszym dystansie.

5. Wnioski

W niniejszym artykule przedstawiona została aplikacja mobilna zaimplementowana na platformę Android. Głównym celem aplikacji było zarejestrowanie i wizualizacja wskazań z sensora przyspieszenia w trzech osiach X, Y i Z. Zgromadzone w ten sposób dane pomogły potwierdzić przydatność czujników wbudowanych w urządzenia mobilne przy rejestracji i wizualizacji ruchu. Po wykonaniu badań i analizie wyników można stwierdzić że pomiar odległości ruchu urządzeń mobilnych na podstawie wskazań z czujnika przyspieszenia (akcelerometru) oraz magnetometru jest niemożliwy. Warto tutaj wspomnieć że czujniki wbudowane w urządzenia mobilne są bardzo czułe i wychwycą nawet najmniejszy ruch, jednak są też bardzo podatne na zakłócenia pochodzące z urządzeń zewnętrznych.

Wskazania akcelerometru urządzenia które znajdowało się w spoczynku wahało się w przedziale $\pm 0,03 \frac{m}{s^2}$. Podczas badań zauważono że najbardziej podatny na zakłócenia jest magnetometr, dlatego też powinno się posilkować odczytami z sensora obrotu oczywiście o ile dane urządzenie taki posiada. Dużym zaskoczeniem podczas badań okazały się pomiary przy najwyższej częstotliwości, wydawać by się mogło że większa częstotliwość przełoży się na większą precyzję, jak się później okazało pomiary z najwyższą częstotliwością próbkowania najbardziej odbiegały od rzeczywistego przemieszczenia.

O ile dane zgromadzone podczas pomiaru nie nadają się do wizualizacji ruchu to można je wykorzystać w inny sposób np.: do sterowania postacią w grze, sterowania samochodem, detekcji obrotu urządzenia itp. Jakość i cena urządzenia nie znalazły odzwierciedlenia w wynikach. Obie zaimplementowane metody całkowania numerycznego dawały podobne rezultaty w związku z czym nie da się jednoznacznie odpowiedzieć na pytanie która z tych metod była lepsza.

Literatura

- [1] Historia systemu Android, <http://androidnow.pl/historia-systemu-android/> [07.03.2016]
- [2] Android, the world's most popular mobile platform, <http://developer.android.com/guide/basics/what-is-android.html> [22.05.2016]
- [3] Collins Ch., Galpin M., Kaeppler M.: Android w praktyce. Helion. Gliwice 2012.
- [4] Conder S., Darcey L.: Programowanie aplikacji mobilnych na urządzenia przenośne. Helion, Wydanie drugie, Gliwice 2011.
- [5] Gartner Says Smartphone Sales Surpassed One Billion Units in 2014, <http://www.gartner.com/newsroom/id/2996817> [03.04.2016]
- [6] Paulo Sergio De Brito Andre: Accelerometers: Principles, Structure and Applications. Electrical Engineering Developments Physics Research and Technology, 2013. Computer Animation algorithms and techniques
- [7] Google ujawnił specyfikację platformy Android, http://www.computerworld.pl/news/131045_2/Google.ujawnil.specyfikacje.platformy.Android.html [22.06.2016]
- [8] Żyroskopy i akcelerometry MEMS w elektronice użytkowej, <http://elektronikab2b.pl/technika/12098-zyroskopy-i-akcelerometry-mems-w-elektronice-uzytkowej#.V7FfeJiLRQ> [23.05.2016]
- [9] Sensors Overview, https://developer.android.com/guide/topics/sensors/sensors_overview.html [01.03.2016]
- [10] JT – Automatyka i programowanie, <http://www.jtjt.pl/calowanie-rownan-rozniczkowych> [30.03.2016]
- [11] Metody komputerowe (2003/2004), http://www.se.put.poznan.pl/almamater/wyklady/metody_komputerowe_03-04/ [08.06.2016]
- [12] Android Developers, <https://developer.android.com> [23.06.2016]
- [13] AndroidLand, <http://androidland.pl/sdk/struktura-systemu-android-2/> [11.02.2016]
- [14] Licenses | Android Open Source Project, <https://source.android.com/source/licenses.html> [04.05.2016]
- [15] LG G Pad 8.0 V490 Dane techniczne telefonu :: mGSM.pl, <http://www.mgsm.pl/pl/katalog/lg/gpad8.0v490/LG-G-Pad-8.0-V490.html> [4.06.2016]
- [16] Samsung Galaxy A3 SM-A3009 Galaxy A3 LTE, SM-A300YŻ, SM-A300FU, SM-A300FZ, SM-A300M, SM-A300Y Dane techniczne telefonu :: mGSM.pl, <http://www.mgsm.pl/pl/katalog/samsung/galaxy3sma3009/Samsung-Galaxy-A3-SM-A3009.html> [04.06.2016]

Wykorzystanie User Experience w procesie poprawy GUI aplikacji

Bartłomiej Kultys*, Marek Miłosz

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

Streszczenie. W artykule przedstawiono praktyczne wykorzystanie User Experience w używanej aplikacji w firmie produkcyjnej. Różnice przed i po modyfikacji GUI z wykorzystaniem doświadczenia użytkownika, jak również najlepsze metody projektowania i testowania GUI, zostały przedstawione w artykule. Wyniki badań pozwalają jednoznacznie stwierdzić, że wykorzystanie doświadczenia użytkownika w projektowaniu graficznym interfejsu użytkownika jest dobrą praktyką.

Słowa kluczowe: doświadczenie użytkownika; graficzny interfejs użytkownika; efektywność interfejsu

* Autor do korespondencji.

Adresy E-mail: bartlomiej.kultys@gmail.com, m.milosz@pollub.pl

Use of User Experience in the Process of the Application GUI Improving

Bartłomiej Kultys*, Marek Miłosz

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

Abstract. The article presents practical usage of User Experience in the application used in the production company. The differences before and after modifications of GUI with the use of user experience, as well as the best methods for GUI designing and testing are presented in this paper. The results of the research allow to state clearly, that the use of the user experience in the design of the graphic user interface is a good practice.

Keywords: User Experience; Graphic User Interface; interface efficiency

*Corresponding author.

E-mail address: bartlomiej.kultys@gmail.com, m.milosz@pollub.pl

1. Wstęp

Graficzny interfejs użytkownika (ang. *Graphical User Interface*, GUI) w obecnych czasach jest nieodzownym elementem życia większości osób. Spotykany jest zarówno w telefonach komórkowych, ekranach dotykowych samochodów, a także lodówkach czy pralkach. Tworzenie układu elementów interfejsu podczas tworzenia części funkcjonalnej aplikacji już dawno przestało się sprawdzać. Obecnie tworzenie części graficznej jest oddzielnym działem składającym się na proces tworzenia całej aplikacji. Prowadzone są badania, których zadaniem jest znalezienie efektywnego interfejsu, dopasowanego do jak największej liczby użytkowników. Badania GUI z wykorzystaniem użytkowników pozwoliły na stworzenie nowego terminu: User Experience [1, 2].

Termin User Experience (pol. doświadczenia użytkownika), w skrócie: UX, dotyczy wrażeń użytkownika podczas pracy z urządzeniami interaktywnymi i wykorzystaniem ich w procesie projektowania ich interfejsów. Sześć najbardziej istotnych aspektów UX, które wpływają na wartościowość produktu programowego to [2]:

- 1) Użyteczność (ang. *useful*) – prostota użycia produktu.
- 2) Funkcjonalność (ang. *usable*) – spełnienie potrzeb użytkowników.
- 3) Znajdowalność (ang. *findable*) – łatwość znalezienia produktu.
- 4) Wiarygodność (ang. *credible*) – wzbudzanie zaufania użytkownika.

- 5) Dostępność (ang. *accessible*) – dostępność dla różnych grup użytkowników, w tym również niepełnosprawnych.
- 6) Pożądalność (ang. *desirable*) – wartość wizerunkowa produktu, zachęcająca do użycia.

Aspekty UX tworzą specjalny diagram, zwany *Honeycomb*, czyli plaster miodu – rys. 1.



Rys. 1. Diagram UX Honeycomb [3].

Doświadczenie użytkownika, jak sama nazwa wskazuje, to subiektywne odczucie użytkownika podczas interakcji z interfejsem. Nie jest możliwe stworzenie oprogramowania posiadającego interfejs idealny, który spełniłby wymagania wszystkich osób. W przypadku, gdy jeden użytkownik woli

korzystać z interfejsu za pomocą klawiatury, inny może preferować używanie myszy do osiągnięcia tych samych celów. Dzięki takim właśnie problemom, stawianym projektantom powstała dziedzina projektowania uwzględniająca doświadczenia użytkownika. Najbardziej istotnym czynnikiem jest czynnik ludzki, czyli subiektywne odczucie użytkownika podczas korzystania z interfejsu. Umiejętność stworzenia interfejsu dającego pozytywne odczucia użytkownikowi wymaga zarówno wiedzy ze strony projektantów, rzemiosła, jak i sztuki odkrywania subiektywnych odczuć odbiorcy.

Najważniejszą cechą GUI, oraz głównym powodem, dla którego stosowane jest UX jest jego efektywność. Według Encyklopedii Zarządzania efektywność to [4]:

„Efektywność to rezultat podjętych działań, opisany relacją uzyskanych efektów do poniesionych nakładów. Oznacza najlepsze efekty produkcji, dystrybucji, sprzedaży, promocji.”

Efektywność GUI może być badana na wiele różnych sposobów. Często stosowaną metodą jest pomiar liczby błędów popełnianych przez użytkowników podczas wykonywania zaplanowanego scenariusza. Inną metodą jest okulografia, popularnie zwaną eyetracking [5]. Pozwala na śledzenie ruchu gałek ocznych podczas pracy z GUI. Tańszym oraz łatwiejszym do przeprowadzenia badaniem jest clicktracking, który śledzi ruchy i działania myszy użytkowników, pozwalające stwierdzić łatwość obsługi oraz intuicyjność rozkładu elementów interfejsu.

Jedną z prostszych metod pomiaru efektywności GUI, nie wymagających przy tym udziału użytkownika, jest metoda pomiaru liczby kliknięć. Pozwala ona na pomiar liczby kliknięć koniecznych do zrealizowania zaplanowanego scenariusza [1]. Metoda ta bardzo dobrze sprawdza się w przypadku porównania starej i nowej wersji interfejsu, gdzie zauważalna różnica pozwala jednoznacznie stwierdzić skuteczność wprowadzonych zmian. Najlepsze zastosowanie ma w przypadku aplikacji, w których często powtarzane są te same działania. Metoda ta została zastosowana w niniejszej pracy.

2. Cel, teza i hipotezy badawcze

Aplikacja poddana badaniu w tym artykule jest aplikacją stworzoną do celów rejestracji danych w przedsiębiorstwie przemysłowym. Stworzona została przez lokalny zespół informatyków w roku 2005 i od tamtej pory nie była modyfikowana pod kątem usprawnienia interfejsu graficznego.

Celem pracy jest sprawdzenie, czy modyfikacja GUI aplikacji z zastosowaniem UX przyniesie pozytywny efekt, oraz czy korzystanie z aplikacji po zmianach będzie bardziej efektywne, niż przed ich wprowadzeniem. Z celu pracy wynika następująca teza badawcza:

Wykorzystanie User Experience umożliwia stworzenie interfejsu zwiększające efektywność pracy użytkownika z aplikacją.

W celu udowodnienia lub nie powyższej tezy sformułowane zostały następujące hipotezy:

Hipoteza 1: UX umożliwia pozyskanie informacji o kierunkach poprawy jakości interfejsu.

Hipoteza 2: Istnieje możliwość prostego i jednoznacznego pomiaru efektywności pracy użytkownika z aplikacją.

Hipoteza 3: Poprawiony interfejs z wykorzystaniem UX jest bardziej efektywny.

3. Metodyka badań

Przedmiotem badania jest wspomniana aplikacja Rejestr Kart Pracy Maszyn (RKPM). Głównym zadaniem aplikacji jest rejestrowanie kart pracy maszyn w przedsiębiorstwie produkcyjnym przemysłu tytoniowego. Dodatkowo spełnia ona szereg innych, istotnych dla obszaru produkcji funkcji, które znacznie ułatwiają i przyspieszają analizę danych. Spośród wszystkich funkcjonalności aplikacji, badaniem objęte zostały trzy najczęściej wykorzystywane scenariusze, które stosowane są przez około 90% całego czasu pracy z RKPM. Scenariuszami tymi są:

UC1 – rejestracja kart pracy maszyn. Dziennie dodawanych jest około 120 kart. Karty dodawane są dla trzech typów maszyn: 22 maszyn pakujących, 15 maszyn papierosowych oraz 7 maszyn filtrowych pracujących na trzech zmianach przez siedem dni w tygodniu.

UC2 – rejestracja nowych gatunków papierosów. Nowy wyrób dodawany jest zawsze wtedy, gdy nastąpiła jakkolwiek modyfikacja wielkości, składu, wyglądu produktu, lub został stworzony całkowicie nowy wyrób. Ze względu na wiele różnych rynków zbytu, nowe wyroby dodawane są nawet kilkukrotnie w ciągu dnia.

UC3 – rejestracja szkartu, czyli wadliwego produktu nadającego się do ponownego użycia. Niewłaściwie wytworzony produkt nie może trafić do sprzedaży, przez co musi zostać oznaczony jako szkart, co pozwoli na jego ponowne wykorzystanie.

Głównymi użytkownikami aplikacji są administratorzy zmiany, którzy codziennie wprowadzają dane o produkcji z każdej maszyny dla każdej zmiany. Wprowadzone dane muszą być wspierane słownikami, które również regularnie aktualizowane są przez pracowników produkcji, a następnie wprowadzone dane są przetwarzane przez program i wyświetlane w formie odpowiednio dostosowanych raportów, przeglądanych przez koordynatorów produkcji, kierownictwo oraz zarząd firmy.

Badania zostały podzielone na kilka głównych etapów.

Pierwszym etapem był pomiar istniejącej aplikacji wybraną metodą pomiaru efektywności interfejsu graficznego. Ze względu na specyficzne zastosowanie aplikacji, a także wąską grupę osób jej używających, wybrano metodę zliczania

maksymalnej oraz minimalnej liczby kliknięć, które są konieczne do wykonania zaplanowanego scenariusza.

Drugim etapem, będącym częścią metody spiralnej, było określenie zbioru wymagań, jakie powinna spełniać nowa wersja interfejsu. Etap ten w znacznym stopniu wykorzystuje doświadczenia użytkowników, korzystających z aplikacji najczęściej.

Trzecim etapem badań było stworzenie projektu prototypu interfejsu na bazie zebranych w poprzednim etapie wymagań. Każde wymaganie było analizowane osobno i wspólnie z użytkownikiem oraz była szukana najlepsza metoda jego spełnienia.

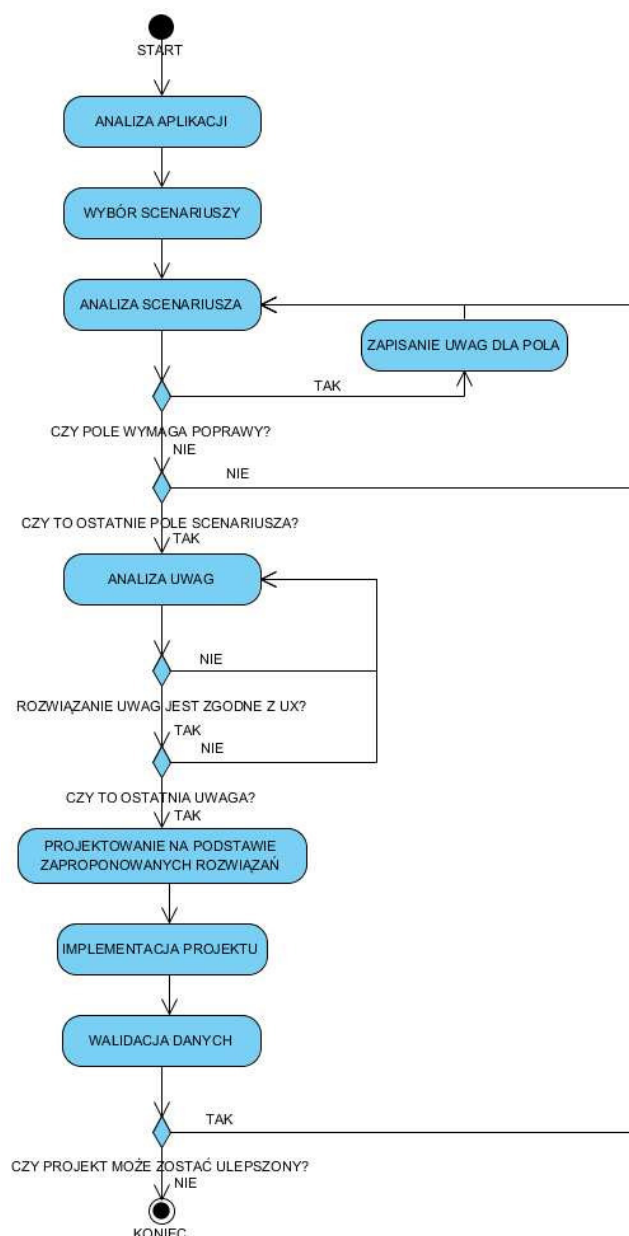
Stworzenie prototypu wysokiej jakości, które było czwartym etapem, bazowało na stworzonym projekcie i pozwoliło na wizualizację pracy wykonanej wcześniej.

Ostatnim – piątym – etapem metody spiralnej było testowanie prototypu przez użytkowników.

Metoda spiralna, składająca się z czterech głównych etapów, począwszy od zbioru wymagań dotyczących zmian, wykonana została dwukrotnie. Druga pętla metody spiralnej pozwoliła na dopracowanie wykonanych w pierwszej pętli zmian oraz ulepszenie interfejsu o wymagania nie zauważone wcześniej.

Końcowym etapem badania był pomiar prototypu tą samą metodą, co na początku oraz porównanie wyników ze sobą. Poniżej przedstawiony został schemat blokowy głównej części badania (rys. 2.).

Każdy z czterech etapów będących częścią metody spiralnej w mniejszym bądź większym stopniu wykorzystuje doświadczenia użytkownika do ich wykonania. Największy wkład w ulepszaniu interfejsu użytkownicy mają w początkowej oraz w końcowej części. Zbieranie wymagań prawie w całości zależne jest od użytkowników i ich doświadczeń w pracy z aplikacją. Najmniejsze zaangażowanie użytkowników wymagane jest podczas tworzenia prototypu aplikacji, gdzie większą rolę przejmują projektanci. Testowanie, podobnie jak w pierwszym etapie odbywa się głównie przy pomocy użytkowników.



Rys. 2. Schemat blokowy zastosowanej metody spiralnej.

4. Rezultaty badań

W wyniku pracy powstały prototypy nowego GUI aplikacji. Przykład takiego interfejsu przedstawiono na rys. 3.

Porównanie efektywności pracy, mierzonej liczbą kliknięć w interfejsie, ze starym i zmodyfikowanym interfejsem aplikacji przedstawiono w tab. 1-3.

Tabela 1. Porównanie pomiaru kliknięć scenariusza UC1

Wersja	Minimum [kliknięcia]	Maksimum [kliknięcia]	Średnia [kliknięcia]
Pierwotna	123	263	193
Prototyp	74	219	146,5

Rys. 3. Przykład prototypu zmodyfikowanego GUI.

Tabela 2. Porównanie pomiaru kliknięć scenariusza UC2

Wersja	Minimum [kliknięcia]	Maksimum [kliknięcia]	Średnia [kliknięcia]
Pierwotna	30	62	46
Prototyp	21	52	36,5

Tabela 3. Porównanie pomiaru kliknięć scenariusza UC3

Wersja	Minimum [kliknięcia]	Maksimum [kliknięcia]	Średnia [kliknięcia]
Pierwotna	41	49	45
Prototyp	21	34	27,5

Zmniejszenie liczby kliknięć w interfejsie można przedstawić w postaci względnej, wykorzystując następujący indeks przyrostu efektywności R:

$$R = 100\% - \frac{K_{\text{prot}}}{K_{\text{pier}}} 100\% \quad (1)$$

gdzie: K_{prot} – liczba kliknięć w nowej, prototypowej wersji GUI, K_{pier} – liczba kliknięć w pierwotnej wersji GUI.

W tab. 4 przedstawione zostały wartości indeksu przyrostu efektywności dla poszczególnych scenariuszy.

Tabela 4. Wartości indeksu przyrostu efektywności dla różnych scenariuszy i wariantów pracy z GUI aplikacji

Wartość	Scenariusz		
	UC1	UC2	UC3
Minimum	39,9%	30,0%	48,8%
Maksimum	16,8%	16,2%	30,7%
Średnia	24,1%	20,7%	38,9%

5. Dyskusja

Przedstawione wyniki wyraźnie pokazują pozytywny efekt zastosowania User Experience w projektowaniu graficznego interfejsu użytkownika. Dodatkowo wszystkie wartości tab. 4 wskazują, że prototyp w każdym scenariuszu przyniósł pozytywny skutek. Hipoteza 1 została więc potwierdzona.

Możliwość ilościowego i, w konsekwencji, jednoznacznego porównania liczby kliknięć wersji pierwotnej oraz prototypowej wersji interfejsu potwierdza hipotezę 2. Metoda ta jest jednocześnie najprostszą i najtańszą metodą oceny efektywności pracy z GUI.

Analiza rezultatów tab. 1-4 częściowo potwierdza hipotezę 3. Pełne potwierdzenie tej hipotezy wymagało by dokładniejszych badań na reprezentatywnej grupie, różnorodnych użytkowników. Badania te są zwykle dość kosztowne.

6. Wnioski

Zarówno wyniki, jak i dyskusja na temat badania, oraz potwierdzenie (w tym częściowe) wszystkich postawionych hipotez badawczych pozwalają jednoznacznie stwierdzić, że teza postawiona na początku badań również może być uznana za potwierdzoną. Jednoznaczne wyniki przemawiające za wysoką efektywnością wykorzystania UX podczas ulepszania interfejsu są niezbitym dowodem na skuteczność tej metody.

Przedstawione dane pozwalają zaobserwować jedynie ilościową wartość kliknięć. Oczywiście testowanie interfejsu jest znacznie szerszym zagadnieniem niż pomiar kliknięć. Procentowa różnica ilości kliknięć nie przekłada się jednak na taką samą różnicę czasową, która zostaje zaoszczędzona podczas wypełniania pól. Czas wypełniania formularza jest wartością subiektywną dla każdego użytkownika i w dużej mierze zależy od znajomości samego interfejsu jak i umiejętności obsługi klawiatury oraz myszy przez obsługującego go użytkownika. W przypadku list rozwijanych z dużą ilością pozycji, czas może zostać dodatkowo wydłużony na szukanie odpowiedniej wartości. Warte przeanalizowania są również czas konieczny do wykonania scenariusza i ilość błędów popełnianych podczas wypełniania formularza. Mimo braku analizy błędów popełnianych przez użytkowników, zastosowanie list rozwijanych zamiast pól tekstowych oraz automatyczne wypełnianie niektórych pól najczęściej podawanymi wartościami daje pewność, że zarówno liczba błędów, jak i czas wykonania scenariusza, uległy zmniejszeniu w stosunku do wersji pierwotnej.

Literatura

- [1] M. Miłosz, Ergonomia systemów informatycznych, Politechnika Lubelska, Lublin (2014).
- [2] C. Kraft, User Experience, Innovation a press, New York (2012).
- [3] <http://windowanddoor.com/article/junejuly-2016/user-experience> [28.06.2016]
- [4] <https://mfiles.pl/pl/index.php/Efektywno%C5%9B%C4%87> [21.03.2016]
- [5] <https://pl.wikipedia.org/wiki/Okulografia> [29.07.2016]
- [6] M. Plechawska-Wójcik, S. Mora, Ł. Wójcik, Assessment of User Experience with Responsive Web Applications using Expert Method and Cognitive Walkthrough, ICEIS 2013 - 15th International Conference on Enterprise Information Systems, 2013, s. 60-67.
- [7] J. Allen, J. Chudley, Projektowanie witryn internetowych User eXperience. Smashing Magazine, tłum. Walczak T., Helion, Gliwice (2013).
- [8] S. Krug, Don't make me think! A Common Sense Approach to Web Usability, New Riers, Wydanie drugie, Berkeley (2006).
- [9] T. Roden, Building the Realtime User Experience, O'Reilly, Sebastopol (2010).
- [10] K. Kołodziejczyk, M. Plechawska-Wójcik, Wykorzystanie metodyki projektowania zorientowanego na użytkownika na przykładzie interfejsu banku on-line, Problemy Współczesnej Inżynierii – Technologie programistyczne, Politechnika Lubelska, Lublin, 2014.
- [11] G. Lasa, D. Justel, A. Retegi, Eyeface: A new multimethod tool to evaluate the perception of conceptual user experiences, Mondragon University, Arrasate Mondragon, 2015.
- [12] W. Royce, Managing the development of large software systems, Międzynarodowa Konferencja Inżynierii Oprogramowania, Monterey, 1987.
- [13] <http://semanticstudios.com/about/> [27.06.2016]
- [14] http://semanticstudios.com/user_experience_design/ [20.06.2016]

Analiza możliwości wykorzystania architektury SOA do integracji systemów informatycznych

Paweł Waręciak*, Piotr Kopniak

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

Streszczenie. Celem niniejszej pracy jest pokazanie różnych metod integracyjnych systemów informatycznych oraz koncepcji SOA. Porównanie ich wymagań w stosunku do SOA oraz przedstawienie czy SOA spełnia te wymagania i jak je realizuje.

Słowa kluczowe: SOA; systemy informatyczne; SOAP; REST; Webservice; integracja systemów

* Autor do korespondencji.

Adres E-mail: pawel.wareciak@gmail.com

Analysis of the possibilities of using SOA to integrate IT systems

Paweł Waręciak*, Piotr Kopniak

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

Abstract. The aim of this work is to show different integration methods of IT systems with SOA conception. Compare their requirements to SOA and present if SOA meets these requirements and how implement them.

Keywords: SOA; IT systems; SOAP; REST; Webservice; system integration

*Corresponding author.

E-mail address: pawel.wareciak@gmail.com

1. Wstęp

Współczesne aplikacje generują duże ilości danych, które następnie trzeba przetworzyć. Niekiedy takie systemy stają się skomplikowane w utrzymaniu i niepodatne na nowe zmiany. Wraz z rozwojem takich aplikacji zachodzi potrzeba integracji pomiędzy nimi. Istnieją różne możliwości integracji aplikacji, między innymi Architektura Zorientowana na Serwisy - SOA (ang. Service Oriented Architecture).

Integracja systemów jest to proces, który pozwala na wykorzystanie danych jakie rejestruje i generuje system informatyczny, funkcji tego systemu oraz procesów biznesowych innym systemom informatycznym. Główne dziedziny integracyjne to [1]:

- integracja aplikacji,
- integracja danych,
- integracja procesów.

Integracje aplikacji można określić jako integracje oprogramowania w celu automatycznej wymiany informacji pomiędzy aplikacjami często stworzonymi przez różnych producentów i działającymi w jednym przedsiębiorstwie lub w przedsiębiorstwach partnerskich.

Drugim typem integracji jest integracja danych, która polega na eliminacji zbędnych powtórzeń z baz danych, czyli redundancji danych. Dzięki temu różne systemy informatyczne operują na takich samych danych, a aktualizacje danych propagowane są do wszystkich zainteresowanych aplikacji. Zapewnia to jednolitość danych we wszystkich integrowanych systemach.

Ostatnim typem integracji jest integracja procesów, polegająca na współdzieleniu procesów biznesowych pomiędzy kilkoma aplikacjami, w taki sposób, że elementy

procesów mogą być wielokrotnie wykorzystywane przez różne aplikacje i możliwa jest budowa różnych procesów na ich podstawie. Procesy takie wykorzystują zwykle wiele systemów informatycznych i wykraczają poza granice jednego przedsiębiorstwa.

Rozwiązania stosowane przy integracji systemów możemy podzielić na kilka typów [2]:

- portale informacyjne,
- rozproszone procesy biznesowe,
- współdzielona logika biznesowa,
- architektura zorientowana na usługi (SOA),
- transfer i replikacja danych,
- integracja aplikacji (EAI),
- obiekty rozproszone - RPC i CORBA,
- systemy kolejkowe.

Zastosowanie integracyjne w portalach informacyjnych cechuje udostępnienie jednego interfejsu np. strony internetowej. Użytkownik może zobaczyć wszystkie interesujące go dane z wielu systemów informatycznych na jednej stronie internetowej (portalu). Informacje są agregowane z wielu systemów i pokazane np. w formie modułów - sekcji na stronie, np. portletów, które reprezentują fragmenty różnych systemów.

Rozproszone procesy biznesowe i współdzielona logika biznesowa opierają się na integracji procesów biznesowych wewnątrz jednego systemu. Zdarza się że procesy biznesowe pochodzą z innych systemów, wtedy mówimy że jest to integracja B2B.

Transfer lub replikacja danych występuje kiedy zachodzi potrzeba synchronizacji danych przy integracji z innym systemem. Zastosowanie takiego rozwiązania można spotkać np. w systemach CRM (Client Relationship

Management). Przy takim podejściu przechowujemy dane z kilku aplikacji w jednym miejscu lub synchronizujemy dane w przechowywane w wielu miejscach. Poza zaletami takiego podejścia, tzn. spójnymi danymi we wszystkich aplikacjach, mogą pojawić się problemy wpływające na działanie poszczególnych z nich. Może to generować problemy przy niekompatybilnych modelach danych w nowszych wersjach aplikacji. Do replikacji na bazie danych wykorzystuje się wbudowane rozwiązania. Transfer danych to wymiana danych pomiędzy systemami za pomocą plików lub połączeń sieciowych wykorzystujących Gniazda TCP.

EAI służy do integracji aplikacji oraz procesów wewnątrz organizacji. Głównym celem jest przekształcenie aplikacji w spójną strukturę.

Gniazda zapewniają dwukierunkowy kanał do komunikacji za pomocą jednego gniazda TCP.

RPC i CORBA są to mechanizmy zdalnego wywoływania procedur. Umożliwiają współdzielenie funkcjonalności pomiędzy systemami. Pierwszy z nich wymaga takiego samego języka programowania do stworzenia komunikujących się, rozproszonych w sieci obiektów. CORBA dzięki zastosowaniu komponentów pośrednich ORB nie ma takiego wymagania i jest rozwiązaniem bardziej uniwersalnym.

Kolejki służą do przesyłania komunikatów lub plików pomiędzy integrowanymi aplikacjami w sposób zapewniający dotarcie do odbiorcy w określonej kolejności nawet w przypadku awarii sieci i czasowego braku komunikacji. Dodają warstwę pośrednią dla innych komponentów. Przykładem takiego rozwiązania jest Java Message Services (JMS).

SOA jest architekturą do tworzenia złożonych systemów informatycznych, która wg specyfikacji posiada cechy wszystkich opisywanych metod integracji systemów. Niniejszy artykuł analizuje czy SOA rzeczywiście nadaje się do integracji systemów i w jaki sposób można to zrobić.

2. Możliwości integracyjne systemów informatycznych

2.1. JMS

Java Message Services (JMS) można zdefiniować jako standard przesyłania wiadomości. Jest to popularny sposób integrowania aplikacji, poprzez wykorzystanie luźnych powiązań oraz przesyłania komunikatów asynchronicznie [3]. Architektura JMS zawiera następujące elementy [4]:

- **dostawce JMS** - dostarcza narzędzia do administrowania oraz implementuje interfejsy
- **klienta JMS** - klienci operują na komunikatach, tworzą, wysyłają oraz je odbierają
- **narzędzia do zarządzania** - umożliwiają tworzenie oraz zarządzanie połączeniami

Wiadomości są przekazywane poprzez jedną z dwóch możliwości:

- **kolejkę (Queue)** – wiadomość kierowana jest do jednego adresata,
- **temat (Topic)** - wiadomości są subskrybowane przez wielu odbiorców.

Wykorzystanie kolejek zapewnia luźną integrację systemów informatycznych, co oznacza że komunikacja nie wymaga jednoczesnego połączenia klientów i nie musi być

przewodzona w czasie rzeczywistym. Takie podejście jest głównie wykorzystywane do budowania warstw pośrednich w celu wykonywania usług dla innych komponentów.

JMS umożliwia integrację aplikacji oraz procesów. Poprzez zastosowanie asynchronicznego generowania zdarzeń można bezpośrednio przekazywać zdarzenia do punktu zbiorczego. Takie podejście pozwala na wyzwalamie kolejnych zdarzeń w procesie.

2.2. RPC i CORBA

Technologia zdalnego wywoływania procedur (RPC - ang. Remote Procedure Call) została opracowana i zaczęła być powszechnie stosowana w roku 1980 [5]. Podczas wywołania RPC parametry procedury są zestawione w żądaniu i przesłane na serwer. Zazwyczaj żądania są przesyłane za pomocą protokołu UDP albo TCP oraz zawierają informacje o nazwie procedury z listą jej parametrów [5]. Po dotarciu żądania do serwera jest ono konwertowane oraz procedura jest weryfikowana przez serwer. W odpowiedzi serwer aplikacji odsyła wartości zwrócone przez procedurę do klienta który oczekuje na żądanie. Komunikat RPC zawiera zarówno zbiór wartości zwróconych przez procedurę lub informacje o błędzie. Wywoływanie synchroniczne jest typowym trybem dla operacji na zdalnych procedurach, również w przypadku Common Object Request Broker Architecture (CORBA) [5]. CORBA jest rozszerzeniem RPC, która opisuje podstawową infrastrukturę do tworzenia zdalnych wywołań procedury poprzez obiekty. Systemy rozproszone mogą pracować na różnych systemach oraz wykorzystywać różne języki programowania.

RPC i CORBA są odpowiednim standardem dla rozproszonych obiektów tzw. fragmentów aplikacji działających na innych środowiskach, które współpracują ze sobą. Zaprojektowanie rozwiązania w taki sposób umożliwia integrację aplikacji.

2.3. Gniazda

Gniazda są protokołem sieciowym, który umożliwia zestawienie dwukierunkowego kanału komunikacji za pośrednictwem pojedynczego połączenia TCP [6]. Protokoły TCP oraz UDP które mogą zostać wykorzystane poprzez gniazda, cechuje zapewnienie połączenia zawsze pomiędzy dwoma procesami [7]. W przypadku WebSocket połączenie jest nawiązywane za pomocą protokołu HTTP. Aplikacja która inicjuje połączenie wysyła specjalne żądanie HTTP z adresem URL punktu końcowego z którym chce nawiązać połączenie. Jeśli serwer zaakceptuje połączenie, tworzona jest odpowiedź która zostaje przesłana do klienta. W takim wypadku połączenie TCP może zostać zestawione i będzie aktywne dopóki któraś ze stron nie zerwie połączenia albo nie zostanie przekroczony limit czasu z powodu bezczynności.

Wykorzystanie gniazd umożliwia integrację aplikacji, gdzie zachodzi potrzeba aktualizacji danych w czasie rzeczywistym np. w przypadku danych giełdowych.

2.4. Architektura SOA

Większość aplikacji jest stworzona w architekturze trójwarstwowej. W takim podejściu dużą wagę przywiązuje się do interfejsu użytkownika. Bardzo często spotykamy potrzebę rozszerzenia takiej aplikacji o możliwość wymiany

danych z innym systemem. Niestety nie zawsze taka możliwość została przewidziana podczas projektowania aplikacji. Dlatego przy tradycyjnym podejściu, kiedy zachodzi potrzeba integracji z innym systemem, rozwiązanie jest przygotowane na szybko bez przemyślanej koncepcji. Takie rozwiązania nierzadko były dostosowane do jednego systemu, co uniemożliwiało zastosowanie istniejącego rozwiązania do integracji z innymi systemami. Największym problemem takiego podejścia jest budowana nieelastyczność na zmiany aplikacji oraz duża pracochłonność i koszt nowych modyfikacji. Architektura zorientowana na usługi wprowadza koncepcję integracji różnych systemów przy zastosowaniu wydzielonych usług oraz podziału funkcjonalności aplikacji na usługi [8].

Architektura zorientowana na usługi określa koncepcję tworzenia usług. Głównymi cechami charakteryzującymi usługi są [9]:

- Kontrakt - interfejs usługi, który zawiera informacje o zasadach komunikacji, danych wejściowych, wyjściowych oraz metodach, które udostępnia dana usługa sieciowa.
- Luźne powiązania - usługi udostępniają pewien kontrakt danych. Ukrywając jak jest realizowana dana funkcjonalność.
- Abstrakcja - zawiera tylko podstawowe informacje na temat usługi, poza udostępnionym kontraktem. W całości ukrywa jak realizowana jest dana funkcjonalność.
- Ponowne wykorzystanie - usługi zawierają podzieloną logikę pomiędzy sobą, przy podziale logiki w łatwy sposób można wykorzystywać ponownie stworzoną funkcjonalność.
- Autonomiczność - każda usługa jest autonomiczna. Odpowiedzialność takiego serwisu ogranicza się do realizowanej funkcjonalności.
- Bezstanowość - minimalizuje użycie zasobów, nie przechowując informacji o kliencie. Brakuje informacji o wykonanych wcześniej operacjach.
- Wykrywalność - stworzone z myślą o wykorzystaniu w rozproszonych systemach. Usługi są uzupełniane metadanymi za pomocą, których mogą być skutecznie wykryte i zinterpretowane.
- Kompozycja - mogą składać się z kilku funkcjonalności realizowanych poprzez inne usługi, tzn. tworzyć złożone usługi, które rozdzielają zadanie pomiędzy nimi.

Poniżej przedstawione zostaną standardy, na których opiera się architektura SOA, tzn. usługa WebService, UDDI, SOAP i ESB. Przedstawiony zostanie także inny typ usługi - REST.

2.5. Usługą WebService

Termin WS (Web Services), czyli usługę sieciową możemy zdefiniować jako oprogramowanie stworzone do wymiany danych pomiędzy maszynami poprzez sieć, wykorzystując do tego zbiór technologii (WSDL, XML, SOAP, UDDI)[10]. Usługi sieciowe są jednym z rozwiązań stosowanych przy implementacji architektury zorientowanej na usługi. Bazując na standardach komunikacji poprzez protokół wymiany wiadomości kończąc na kontrakcie danych. Są reużywalnymi elementami budowy procesów biznesowych w systemach informatycznych.

2.6. UDDI

UDDI (Universal Description, Discovery and Integration)[11] - technologia pozwalająca na publikację i wyszukiwanie informacji o usługach WebService. Jest to otwarty standard zarządzany przez organizację OASIS [12]. Opiera się na mechanizmie publikacji - wyszukania - powiązania. Skorzystanie z usługi musi zostać poprzedzone opublikowaniem jej w rejestrze. Konsument wyszukuje usługi dostępne w rejestrze, po odnalezieniu konkretnej usługi następuje powiązanie jej z konsumentem, który otrzymuje możliwość korzystania z operacji danej usługi. Informacje wykorzystywane w UDDI są opisane za pomocą rejestru biznesowego, który można podzielić na trzy grupy:

- białe strony (white pages) - dane kontaktowe organizacji dostarczającej usługę,
- żółte strony (yellow pages) - klasyfikacja biznesu i usług,
- zielone strony (green pages) - informacje o udostępnionej usłudze np. z odnośnikiem do specyfikacji.

2.7. REST

Innym typem usługi sieciowej rzadko wykorzystywanym w rozwiązaniach SOA, ale opisanej dla uzupełnienia wiedzy o usługach sieciowych jest REST. Jest to koncepcja tworzenia usługi/protokołu, nieposiadająca opisu zestawu operacji danej usługi. Transmisja danych jest wykonywana w oparciu o protokół HTTP. Usługa REST bazuje na następujących założeniach [13]:

- **Adresowalne zasoby** – wszystkie dostępne zasoby danej usługi powinny być adresowalne za pomocą unikalnego adresu URI
- **Ujednolicony interfejs** – usługa posiada prosty dobrze określony zestaw metod, za pomocą, których uzyskuje się dostęp do jej zasobów.
- **Wielopostaciowa reprezentacja** – część zasobów w zależności od potrzeb może przybierać różną postać tzn. jeden zasób dostępny pod tym samym adresem URL, może mieć format XML-owy lub JSON-owy lub też zupełnie w zależności od potrzeb i jego zastosowania.
- **Bezstanowość** – usługa nie przechowuje stanu pomiędzy kolejnymi jej wywołaniami, wszystkie żądania są zupełnie od siebie niezależne.

REST wykorzystując protokół HTTP udostępnia metody (POST, GET, PUT, DELETE).

- **GET** – stosuje się w celu pobrania danego zasobu.
- **PUT** – z jego pomocą tworzy się nowy zasób pod wskazanym adresem URI oraz aktualizuje istniejące.
- **POST** – może być stosowany zamiennie z PUT, ale przyjęło się, że z pomocą metody POST zapisuje się w usłudze dane mające charakter hierarchiczny.
- **DELETE** - służy do usunięcia zasobu z usługi.

2.8. SOAP

W celu wymiany wiadomości usługi sieciowe typu WebService wykorzystują protokół SOAP. Protokół ten umożliwia przesłanie danych w formacie tekstowym i binarnym. Opisuje sposób kodowania oraz wymiany wiadomości w formacie XML. Dokumenty SOAP najczęściej są przysyłane za pomocą protokołu HTTP/HTTPS, ale możliwe jest też przesłanie wiadomości za pomocą protokołu

SMTP, FTP, RMI. SOAP opakowuje wiadomości i składa się z kilku elementów [14]:

- **Envelope** - wszystkie wiadomości SOAP zawierają ten element razem z przestrzenią nazw <http://schemas.xmlsoap.org/soap/envelope/>.
- **Nagłówek** - opcjonalny parametr, który może zostać przesłany. Zazwyczaj wykorzystuje się go do przesłania dodatkowych informacji odnośnie uwierzytelnienia, podpisu cyfrowego itp. Jeśli nagłówek jest niezrozumiały dla usługi, zostanie on pominięty.
- **Treść** - jest to treść wiadomości w formacie XML
- **Załączniki** - przechowują dane binarne

2.9. ESB

Enterprise Service Bus (ESB) jest to baza strukturalna i komunikacyjna dla usług w architekturze SOA, zakłada tworzenie systemu opartego o usługi oraz sterowanego przy pomocy zdarzeń. ESB nie są konieczne, ale są często stosowane przy architekturze SOA. ESB ułatwia dynamiczne łączenie, zarządzanie oraz interakcje, udostępniając wspólną infrastrukturę. Wymiana danych odbywa się przez szynę, która ma za zadanie rozdzielać wiadomości w zależności od konfiguracji. Architektura ESB zakłada szynę jako centralny model danych w połączeniu pomiędzy serwisami. Szyna zawiera wszystkie serwisy, które są związane z systemem oraz wiadomości które są przekazywane z oraz do serwisów za jej pośrednictwem. Zaletą takiego rozwiązania jest mała ilość połączeń pomiędzy usługami. ESB zakłada jeden model danych, skutkuje to tym że teoretycznie każdy serwis integrujący się z szyną powinien posiadać ten sam model danych. Takie wymaganie uniemożliwiło by wręcz integrację z innymi systemami. Rozwiązaniem tego problemu są adaptery, które umieszcza się na styku pomiędzy serwisem, a szyną. Służą one to konwersji danych do poszczególnych modeli [15].

2.10. Ocena możliwości zastosowania SOA przy integracji systemów informatycznych

Zastosowanie SOA umożliwia wykorzystanie 3 typów integracji. Architektura zorientowana na usługi pozwala na integrację aplikacji (EAI) poprzez wykorzystanie wydzielonych usług do wymiany informacji z innymi aplikacjami. W SOA mamy WebService który przesyła asynchroniczne wiadomości, ale dzięki SOAP ma jeszcze kontrolę błędów i niepowodzeń i generalnie służy do integracji nie tylko danych ale też współdzielenia funkcji biznesowych. SOA umożliwia integrację procesów biznesowych, przy pomocy szyny ESB. W tym wypadku ESB występuje w roli pośrednika pomiędzy wykonaniem procesów biznesowych (BPEL) a usługą, zarządzając również przysyłanymi komunikatami.

Kolejną możliwością jest integracja danych, gdzie SOA ignoruje przetwarzanie danych masowych, operując na poziomie usług. Automatycznie zakłada się zatem, że zdalna procedura (usługa) zaprojektowana została w taki sposób, by obsłużyć wszystkie możliwe sytuacje biznesowe związane ze zdalnym dostępem do zasobów.

3. Podsumowanie

Zastosowanie architektury zorientowanej na usługi do integracji systemów informatycznych jest dobrym pomysłem,

ponieważ wprowadza koncepcje wydzielenia funkcjonalności do niezależnych usług. Przy takim podejściu systemy stworzone w koncepcji SOA, mogą wymieniać dane oraz tworzyć procesy biznesowe złożone z kilku aplikacji.

Porównując podejście wykorzystywane w architekturze SOA, do innych możliwości integracyjnych, koncepcja architektury zorientowanej na usługi umożliwia integrowanie usług dla wielu klientów. Taką możliwość mają jeszcze kolejki. RPC oraz gniazda cechują się integracją typu punkt-punkt. Główną wadą RPC jest blokowanie klienta, który oczekuje na odpowiedź od serwera. Koncepcja SOA wymusza wytwarzanie oprogramowania w oparciu o luźno powiązane usługi, co przy innych typach integracji nie zawsze jest możliwe. Przy takim podejściu koszty późniejszych zmian są bardzo duże. Poprzez narzut złożoności testowanie SOA jest znacznie trudniejsze, niż w innych typach integracji. Stworzone aplikacje przedstawiają podział systemu na usługi, co zapewnia większą elastyczność w odróżnieniu do integracji punkt-punkt. Zastosowanie takiego podejścia umożliwia ponowne wykorzystanie usług przez inne systemy oraz współdzielenie danych. Kolejną cechą jest możliwość budowy procesów biznesowych oraz wykorzystanie danych z jednego systemu w kolejnych. SOA poprzez swoją koncepcję umożliwia integrowanie aplikacji, danych oraz procesów. Wprowadza to bardzo elastyczne podejście ponownego wykorzystania usług, stawiając SOA na lepszej pozycji podczas wyboru możliwości integracji systemów informatycznych.

Literatura

- [1] Dr. Lui M., Gray M., Chan A., Long J., Pro Spring Integration, Apress, Nowy Jork, 2011
- [2] Kiermasz W., Integracja systemów informatycznych w architekturze zorientowanej na usługi, Instytut Informatyki PW, 2009.
- [3] Schaefer C., Ho C., Harrop R., Pro Spring, Fourth Edition, Apress, Nowy Jork, 2014
- [4] JMS, <http://docs.oracle.com/javase/5/tutorial/doc/bncdx.html> [16.04.2016]
- [5] Bolton F., Pure CORBA, Sams, 2001
- [6] Coward D., Java WebSocket Programming, Oracle Press, 2013
- [7] Lombardi A., WebSocket, O'Reilly Media, Inc., Sebastopol, 2015
- [8] Rotem-Gal-Oz A., SOA Patterns, Manning, Nowy Jork, 2012
- [9] Plunkett T., Chou D., Balasubramanian R., Thomas P., Roy S., Tost A., Erl T., SOA with Java: Realizing Service-Oriented with Java Technologies, New Jersey Prentice Hall, 2014
- [10] Dirksen J., SOA Governance in Action: REST and Web Service Architectures, Manning Publications, 2012
- [11] UDDI, <http://uddi.xml.org> [16.04.2016]
- [12] Organization for the Advancement of Structured, Information Standards. Strona domowa organizacji OASIS. <http://www.oasis-open.org/home/index.php> [16.04.2016]
- [13] Shin S. SUN TECH DAYS, <http://www.oracle.com> [16.04.2016]
- [14] W3C Organization. SOAP, <https://www.w3.org/TR/soap12-part1> [16.04.2016]
- [15] DiMaggio L., Conner K., Kumar M. B., Cunningham T.: JBoss ESB, Packt Publishing, Birmingham 2012

Zbadanie wydajności aplikacji internetowych utworzonych z wykorzystaniem Spring MVC oraz JavaServer Faces

Nazar Kovalchuk*, Edyta Łukasik

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

Streszczenie. Celem artykułu było porównanie wybranych szkieletów programistycznych w technologii Java. Do analizy zostały wybrane Spring MVC oraz JavaServer Faces. Została przeprowadzona analiza teoretyczna związana z funkcjonowaniem badanych frameworków. Dla porównania wybranych szkieletów programistycznych były utworzone dwie aplikacje, posiadające podobną funkcjonalność. Wyniki przeprowadzonych badań dotyczących wydajności oraz pomiaru metryk kodu utworzonych aplikacji zostały pokazane w postaci wykresów.

Słowa kluczowe: Spring MVC; JavaServer Faces; Java; aplikacja internetowa

* Autor do korespondencji.

Adres E-mail: koval014051993@gmail.com

Efficiency test of web applications created using Spring MVC and JavaServer Faces frameworks

Nazar Kovalchuk*, Edyta Łukasik

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

Abstract. The purpose of the article was to compare selected frameworks for web applications Java technology. Spring MVC and JavaServer Faces were selected for the analysis. Theoretical analysis related with the functioning of the examined frameworks was conducted. For comparison of selected frameworks there were created two applications that have similar functionality. Performance testing and measuring code metrics of created applications were conducted.

Keywords: Spring MVC; JavaServer Faces; Java; web application

*Corresponding author.

E-mail address: koval014051993@gmail.com

1. Wstęp

Celem artykułu jest porównanie wybranych szkieletów programistycznych w technologii Java. Do analizy zostały wybrane frameworki Spring MVC oraz JavaServer Faces. Analiza obejmuje teoretyczne aspekty ich funkcjonowania oraz wykorzystanie w praktyce.

W celu utworzenia aplikacji internetowych wykorzystano następujące narzędzia:

- szkielety programistyczne Spring MVC oraz JavaServer Faces;
- zintegrowane środowisko programistyczne NetBeans [1] oraz Eclipse [2];
- server MySQL [3];
- Maven – narzędzie do budowy projektu w technologii Java: kompilacji, tworzenia jar, generowania dokumentacji[4].

Dla porównania Spring MVC oraz JSF została przeprowadzona analiza teoretyczna związana z funkcjonowaniem badanych szkieletów programistycznych, która wskazała podobieństwa oraz różnice w działaniu obu ram projektowych. Omawiając równocześnie mocne i słabe strony każdego ze szkieletów w sytuacjach typowych dla

działania aplikacji internetowych, w analizie zwrócono uwagę na najważniejsze aspekty związane z działaniem badanych frameworków, czyli:

- architektura;
- model programistyczny;
- walidacja danych;
- internacjonalizacja.

Jedną z najlepszych metod oceny wydajności jest wykonanie testów z 1 aktywnym użytkownikiem, 2 użytkownikami, 4 użytkownikami, 8 użytkownikami itp[5].

Najważniejszymi parametrami, które należy rejestrować w każdym teście wydajności, są przepustowość (liczba żądań na sekundę) oraz czas odpowiedzi.

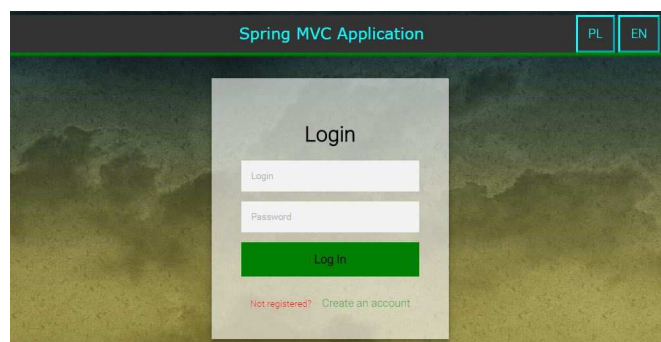
2. Opis aplikacji

Utworzone zostały dwie aplikacje oparte o: Spring MVC oraz JavaServer Faces. Każda z nich posiada następujące funkcjonalności:

- logowanie użytkownika;
- rejestracja użytkownika;
- użytkownicy aplikacji posiadają dwie role (admin oraz prosty użytkownik);

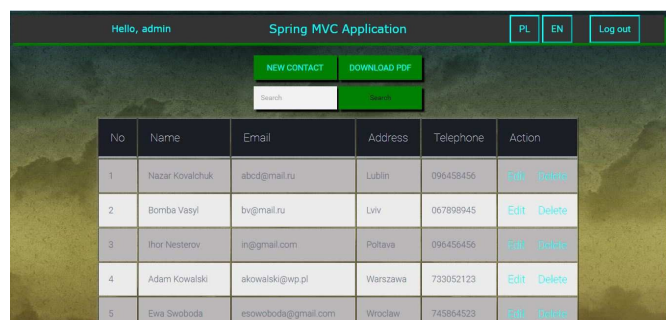
- dodawanie nowego kontaktu;
- usuwanie oraz edycja wybranego kontaktu;
- wyszukiwanie kontaktów;
- pobranie listy kontaktów w formacie PDF oraz Excel;
- walidacja wprowadzonych danych;
- zmiana języka aplikacji (angielski oraz polski).

Wygląd przykładowych ekranów działających aplikacji został pokazany na rysunkach 1, 2 i 3. Po uruchomieniu aplikacji pojawia się okno logowania użytkownika przedstawione na rys. 1.



Rys. 1. Logowanie użytkownika w aplikacji Spring MVC

Przy zalogowaniu użytkownika jako administratora pojawia się główne okno aplikacji jak na rys.2.



Rys. 2. Główne okno aplikacji Spring MVC

Aplikacja posiada wybór języka: polski lub angielski. Na rysunku 3 przedstawiono wygląd aplikacji JSF w języku polskim.



Rys. 3. Zmiana języka aplikacji JavaSever Faces

3. Badania wydajności

Do zbadania wydajności utworzonych aplikacji zostało wykorzystane narzędzie JMeter.

W celu przeprowadzenia testów wydajności były utworzone dwa scenariusze testów. Pierwszy scenariusz obejmował operacji odczytu danych, czyli akcji, które nie mogły mieć wpływ na stan aplikacji.

Drugi scenariusz obejmował operacji, które wpływały na stan aplikacji. Do operacji takiego typu można odnieść dodawanie nowych użytkowników, modyfikacje kontaktów oraz ich usuwanie.

Testy dotyczące wydajności zostały przeprowadzone na obu badanych aplikacjach opartych o Spring MVC oraz JavaServerFaces.

Dane testy były przeprowadzone dla różnej ilości użytkowników od 1 do 512.

Wyniki pierwszego testu aplikacji opartej o Spring MVC są przedstawione w tabeli 1.

Tabela 1. Wydajność aplikacji opartej o Spring MVC przy pierwszym teście

Liczba użytkowników	Liczba żądań/sek	Czas odpowiedzi [ms]
1	60	13
2	111	15
4	137	24
8	164	40
16	174	72
32	189	155
64	148	413
128	146	832
256	136	1509
512	126	2471

Wyniki drugiego testu aplikacji opartej o Spring MVC są przedstawione w tabeli 2.

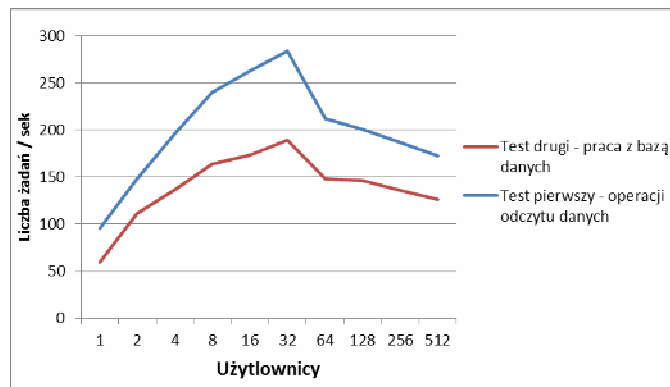
Tabela 2. Wydajność aplikacji opartej o Spring MVC przy drugim teście

Liczba użytkowników	Liczba żądań/sek	Czas odpowiedzi [ms]
1	96	2
2	148	2
4	196	3
8	240	6
16	263	9
32	284	10
64	212	12
128	201	13
256	187	303
512	173	1530

Na podstawie otrzymanych wyników został utworzony wykres przedstawiony na rys. 4 dla porównania wydajności aplikacji Spring MVC przy pierwszym i drugim teście.

Jak widać na podstawie wykresu przedstawionego na rys.4 przy drugim teście, który dotyczył pracy z bazą danych,

wydajność aplikacji spadła, co jest oczekiwanym wynikiem, ponieważ na takie operacje aplikacja potrzebuje więcej zasobów.



Rys. 4. Wydajność aplikacji opartej o Spring MVC przy pierwszym i drugim teście.

Wyniki pierwszego testu aplikacji opartej o JavaServer Faces są przedstawione w tabeli 3.

Tabela 3. Wydajność aplikacji opartej o JavaServer Faces przy pierwszym teście

Liczba użytkowników	Liczba żądań/sek	Czas odpowiedzi [ms]
1	19	25
2	35	29
4	57	34
8	94	41
16	109	72
32	112	140
64	103	306
128	96	651
256	94	1435
512	86	2755

Wyniki drugiego testu aplikacji opartej o JavaServer Faces są przedstawione w tabeli 4.

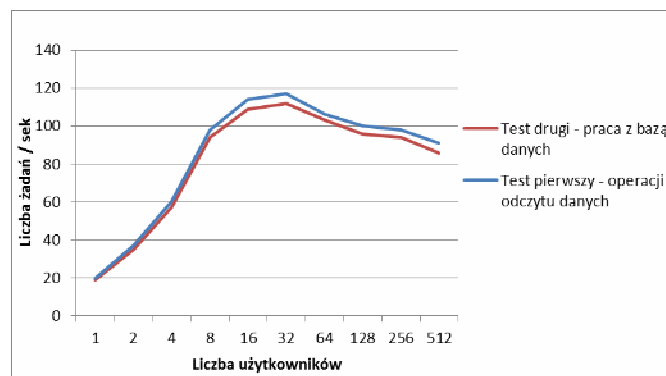
Tabela 4. Wydajność aplikacji opartej o JavaServer Faces przy drugim teście

Liczba użytkowników	Liczba żądań/sek	Czas odpowiedzi [ms]
1	20	27
2	37	27
4	60	33
8	98	41
16	114	70
32	117	140
64	106	304
128	100	651
256	98	1289
512	91	2821

Na podstawie otrzymanych wyników został utworzony wykres przedstawiony na rys. 5 dla porównania wydajności aplikacji JavaServer Faces przy pierwszym i drugim teście.

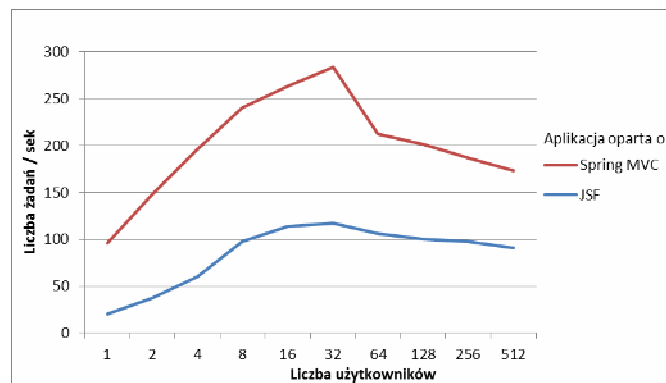
Jak widać na wykresach pokazanych na rys. 5 wyniki obu testów aplikacji opartej o JavaServer Faces są bardzo podobne. Wynika to z wielkości przesyłanego ciasteczka (około 5 KB),

które przy zwiększonej liczbie żądań zaczyna odgrywać znaczną rolę.

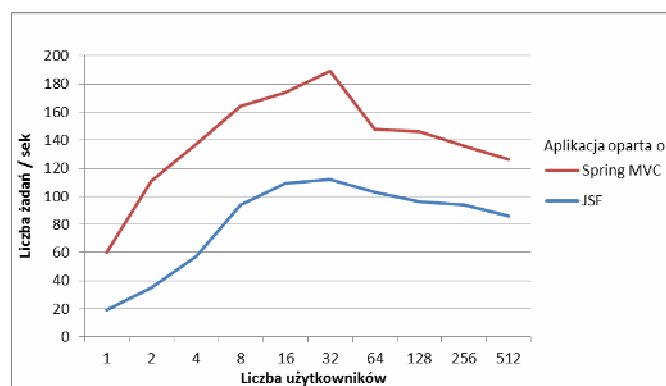


Rys. 5. Wydajność aplikacji opartej o JavaServer Faces przy pierwszym i drugim teście

Dla porównania wydajności aplikacji Spring MVC oraz JavaServer Faces były zbudowane wykresy na podstawie otrzymanych wyników, które pokazano na rys. 6 i 7.



Rys. 6. Porównanie wydajności aplikacji Spring MVC oraz JavaServer Faces przy pierwszym teście



Rys. 7. Porównanie wydajności aplikacji Spring MVC oraz JavaServer Faces przy drugim teście

Z otrzymanych wyników przedstawionych na rys. 6 i 7 można wnioskować, że aplikacja oparta o Spring MVC jest bardziej wydajna, niż aplikacja oparta o JavaServer Faces. Wydajność aplikacji liniowo narasta w przypadku zmiany liczby klientów od 1 do 32. Jednak w przedziale od 32 do 512 klientów, wzrost wydajności zaczyna powoli spadać. Punkt ten nazywa się punktem nasycenia. To jest kluczowe znaczenie,

które trzeba znaleźć podczas testów wydajności. Punkt nasycenia mówi o tym, że osiągnięto najwyższą wydajność dla aplikacji. Wzrost liczby użytkowników powyżej tej wartości tylko wydłużył czas reakcji, nie zwiększając ogólnej wydajności aplikacji.

4. Metryki kodu

Celem pomiaru metryk kodu było porównanie wielkości zaimplementowanych aplikacji opartych o Spring MVC oraz JavaServer Faces.

Pomiar metryk kodu został przeprowadzony za pomocą wtyczki środowiska Eclipse o nazwie Metrics oraz wtyczki środowiska NetBeans o nazwie SourceCodeMetrics. Wyniki pokazano w tabeli 5.

Tabela 5. Metryki kodu aplikacji opartych o Spring MVC oraz JavaServer Faces

Metryka	Spring MVC	JavaServer Faces
Liczba linii kodu	568	508
Liczba klas	7	5
Liczba pól	14	23
Liczba metod	69	45
Liczba linii kodu metod	257	236
Liczba pakietów	5	5

Zmierzona została liczba linii kodu, artefaktów oraz elementów klas. W analizie uwzględniono klasy DAO odpowiedzialne za dostęp do danych oraz logikę biznesową aplikacji z wykorzystaniem szkieletów Spring MVC oraz JavaServer Faces. W przypadku linii kodu zostały zmierzone dwie metryki. Pierwsza z nich jest związana z całkowitą liczbą linii kodu całej aplikacji, druga dotyczy wyłącznie kodu umieszczonego w metodach. Aplikacja oparta o JavaServer Faces zawiera mniej linii kodu niż aplikacja oparta o Spring MVC.

Tabela 6 przedstawia porównanie liczby linii kodu podzielonego według funkcjonalności. Wprowadzono podział na kod kontrolerów, kod walidacji, kod odpowiedzialny za internacjonalizację oraz kod dostępu do bazy danych (DAO).

Tabela 6. Metryki kodu aplikacji opartych o Spring MVC oraz JavaServer Faces

Liczba linii kodu	Spring MVC	JavaServer Faces
Kontrolery	146	52
Walidacja	20	-
Internacjonalizacja	22	24
DAO	140	210
Liczba linii kodu metod	257	236
Liczba pakietów	5	5

Aplikacja oparta o JavaServer Faces nie posiada w kodzie logiki związanej z walidacją wprowadzanych danych. Wszystkie aspekty walidacji wprowadzanych danych udało się zaimplementować na stronach internetowych.

W przypadku kontrolerów aplikacja wykorzystująca Spring MVC posiada więcej kodu, ponieważ kontrolery zawierają metody walidacji wprowadzonych danych.

Ilość kodu odpowiedzialnego za internacjonalizację jest podobna w obu aplikacjach opartych o Spring MVC oraz o JavaServer Faces.

W przypadku klas odpowiedzialnych za dostęp do bazy danych aplikacja wykorzystująca Spring MVC jest łatwiejsza. Spring MVC posiada gotową klasę JdbcTemplate, która pozwala zmniejszyć ilość kodu.

5. Wnioski

Celem badań było porównanie dwóch wybranych szkieletów programistycznych w technologii Java. Analiza została przeprowadzona dla Spring MVC oraz JavaServer Faces. Przedstawione w artykule porównanie miało za zadanie ułatwienie programiście dokonanie wyboru między dwoma alternatywnymi rozwiązaniami.

Przeprowadzona analiza teoretyczna związana z funkcjonowaniem badanych szkieletów programistycznych wskazała podobieństwa oraz różnice w ich działaniu.

Spring MVC realizuje koncepcję aplikacji internetowej opartej o żądania i odpowiedzi [6]. Używa do tego kontrolerów, działających na modelu. JavaServer Faces umożliwia tworzenie aplikacji internetowych z naciskiem na interfejs użytkownika zbudowany w sposób komponentowy [7].

Przeprowadzone zostały badania wydajności oraz pomiar metryk kodu utworzonych aplikacji. Badania wydajności zostały wykorzystane do oceny zachowania aplikacji w sytuacji dużego obciążenia, co jest ważną kwestią w przypadku aplikacji dostępnych dla szerokiego grona osób. Aplikacja oparta o Spring MVC jest bardziej wydajnościowa, niż aplikacja oparta o JavaServer Faces.

Pomiar metryk kodu wskazał, że aplikacja wykorzystująca Spring MVC jest większa ze względu na liczbę linii kodu oraz artefaktów programistycznych.

Literatura

- [1] https://netbeans.org/features/index_ru.html [14.06.2016].
- [2] <http://hightech.in.ua/content/art-eclipse-platform> [08.06.2016].
- [3] <http://danneo.ru/coding/read-mysql-a-control-system-of-databases.html> [17.06.2016].
- [4] <https://maven.apache.org/what-is-maven.html> [16.06.2016].
- [5] http://www.ibm.com/developerworks/ru/library/wes-1208_hare/ [21.07.2016].
- [6] Amuthan G.: Spring MVC: przewodnik dla początkujących. Gliwice: Wydawnictwo Helion, cop. 2015.
- [7] [David Geary, Cay S. Horstmann: JavaServer Faces. Gliwice : Wydawnictwo Helion, cop. 2011.

Analiza porównawcza narzędzi do budowy prototypów interfejsów

Stanisław Lipski*, dr inż. Marek Miłoś

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

Streszczenie. Analizę porównawczą narzędzi do prototypowania interfejsów użytkownika przeprowadzono przy pomocy metody mieszanej polegającej na realizacji aktywnego eksperymentu z wykorzystaniem narzędzi na grupie badawczej, przeprowadzeniu badań ankietowych oraz analizie eksperckiej parametrów technicznych. Rezultaty badań pozwoliły wyznaczyć wartości zmiennych diagnostycznych, które to użyte w metodzie selekcji wielokryterialnej posłużyły do wyboru najlepszego narzędzia.

Słowa kluczowe: interfejs użytkownika; prototypowanie; narzędzia wspomagające; wybór wielokryterialny

*Autor do korespondencji.

Adresy E-mail: staslipski@interia.pl, m.milosz@pollub.pl

Comparative analysis tools for interfaces prototyping

Stanisław Lipski*, Marek Miłoś

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

Abstract. Comparative analysis tools for user interfaces prototyping was conducted using a mixed method consisting in the implementation of active experiment with the use of tools for the research group, conducting surveys and expert analysis of technical parameters of tools for prototyping. Study results have helped determine the values of diagnostic variables that have been used in the multi-selection method of the best tool selection.

Keywords: user interface; prototyping; supporting tools; multi-criteria selection

*Corresponding author.

E-mail addresses: staslipski@interia.pl, m.milosz@pollub.pl

1. Wstęp

Istotnym elementem w cyklu wytwarzania oprogramowania, wpływającym na końcowy rezultat odgrywa prototypowanie. Jest ono stosowane w początkowej fazie wytwarzania oprogramowania, które stanowi dopełnienie dokumentacji ukazującej zakres wymagań projektu. Zastosowanie prototypów w fazie projektowania ma na celu umożliwienie projektantom testowanie nowych koncepcji dotyczących konstrukcji systemu. Oceniając prototyp, projektanci mogą identyfikować wymagania funkcjonalne, problemy z użytecznością i wydajnością, które mogą być rozpatrywane przed fazą realizacji. Prototypowanie szczególnie rekomendowane jest do projektowania interfejsów użytkownika. Sposób przekazania wymagań do GUI (ang. *Graphical User Interface*) w postaci tekstowej nie jest wystarczający, co może doprowadzić do błędnej interpretacji przez zespół projektowy. Połączenie prototypowania ze współpracą z użytkownikiem końcowym może być efektywnym sposobem projektowania graficznych interfejsów użytkownika [2].

Dokonując podziału prototypów, można uwzględnić kryteria obejmujące zakres projektowanego systemu.

Prototypem poziomym (ang. *Horizontal Prototype*) nazywa się prototyp, który ukazuje szeroki zakres działania systemu z uwzględnieniem tylko istotnych elementów. Budowa tego rodzaju prototypu ma na celu zobrazowanie wymagania przyszłego systemu i ułatwić zatwierdzenie

specyfikacji projektu. Do głównych zalet tego prototypu zaliczyć można lepsze ukazanie potrzeb użytkownika oraz rzeczową ilustrację specyfikacji wymagań [1, 13].

Prototypowanie pionowe (ang. *Vertical Prototype*) jest to rodzaj szczegółowego prototypu, który uwzględnia jedynie pewne aspekty całego projektu. Stworzenie prototypu pionowego wybranego fragmentu systemu, może posłużyć kontroli wybranej architektury przy założeniu pewnych scenariuszy testowych. Prototypowanie pionowe wykorzystane może być w początkowej fazie projektowania jak również na etapie kreowania różnych wariantów aplikacji [3].

Prototypowanie może opierać się na dwóch wariantach związanych z porzuceniem prototypu lub jego dalszą ewaluacją. **Porzucenie prototypu** (ang. *Throwaway Prototype*) polega na stworzeniu modelu, który ostatecznie zostanie porzucony i nie będzie dalej rozwijany. Przygotowanie tego typu prototypu służy głównie do ukazania wymagań lub do podjęcia decyzji wyboru odpowiedniej wersji systemu [2, 4]. Inną metodą jest **prototypowanie ewolucyjne** (ang. *Evolutionary Prototyping*) mające na celu stworzenie dokładnego prototypu, który stopniowo będzie udoskonalany aż do produktu finalnego. Podstawowym czynnikiem jest współpraca z użytkownikami końcowymi, którzy dostarczają kluczowych informacji dotyczących zmian w przyszłym systemie [5, 10, 14].

2. Metoda selekcji wielokryterialnej

Wybierając odpowiednie narzędzie należy przeanalizować wiele czynników, które wpływają na wybór tego odpowiedniego. Przeprowadzając analizę porównawczą badane są pewne cechy produktów pod względem efektywności czy przydatności danych narzędzi w konkretnych projektach. Uzyskane wyniki pomogą zaobserwować różnice pomiędzy poszczególnymi narzędziami i dokonać odpowiedniego wyboru.

Analiza wielokryterialna wykorzystywana jest przy wspomaganiu procesu decyzyjnego. Stosowana do oceny i wyboru najkorzystniejszego rozwiązania wśród analizowanych wariantów. Przed przystąpieniem do analizy należy określić zmienne diagnostyczne, czyli zestaw zmiennych przedstawiających charakterystykę wybranych obiektów. Dobór odpowiednich zmiennych diagnostycznych w znacznym stopniu wpływa na wyniki badania, a nieodpowiednia selekcja może negatywnie wpłynąć na końcowe rezultaty. Wyodrębnienie zmiennych diagnostycznych dla narzędzi do prototypowania odbywa się w tym przypadku przy pomocy kryteriów poza statystycznych, czyli merytorycznych i formalnych. Wybierane są te cechy, które dla badanego przedmiotu odgrywają istotną rolę do przeprowadzenia analizy porównawczej [12].

W analizie porównawczej zmienne (kryteria) można traktować w sposób współmierny lub zastosować wobec nich odpowiednie wagi różnicowe, których wyodrębniona wartość stanowi o ważności danego kryterium. Przy nadawaniu wag muszą być spełnione dwa kryteria:

- wagi zdefiniowane, jako nieujemne,
- suma wartości zmiennych diagnostycznych powinna być równa jeden

Dokonując wielowymiarowej analizy porównawczej na wybranych zmiennych diagnostycznych wiąże ze sobą konieczność przeprowadzenia transformacji w celu ujednolicenia obserwowanych parametrów. Proces polegający na zmianie parametrów nazywany jest transformacją normalizacyjną. Jednym z typów normalizacji jest unitaryzacja, polegająca na uzyskaniu zmiennych o znormalizowanym zakresie przedstawionym przez różnicę pomiędzy wartościami maksymalnymi i minimalnymi. Wynikiem otrzymanej wartości jest zmienna w przedziale [0;1]. Ogólną formułę normalizacji można przedstawić wzorem:

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

gdzie: X_{ij} – zaobserwowana wartość j-tej zmiennej w i-tym obiekcie, $\min\{X_{ij}\}$ – minimalna wartość j-tej zmiennej w i-tym obiekcie, $\max\{X_{ij}\}$ – maksymalna wartość j-tej zmiennej w i - tym obiekcie

Po przeprowadzeniu normalizacji poszczególnych zmiennych można przystąpić do uporządkowania zbioru

obiektów opartej na metodzie bez wzorcowej. Polega na dokonaniu uśrednienia unormowanych wcześniej wartości zmiennych przypisując im odpowiednie wagi [9]. Do wyznaczenia zmiennej syntetycznej uwzględnia się następujący wzór:

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

gdzie: Z_{ij} – znormalizowana wartość zmiennej i-tym obiekcie w j-tej zmiennej, W_j – waga kryterium j-tej zmiennej.

3. Narzędzia informatyczne wspomagające prototypowanie interfejsu

Przy budowaniu prototypu interfejsu można skorzystać z szeregu dostępnych narzędzi dzięki którym w łatwy sposób można stworzyć makietę systemu. Posłużyć można się uniwersalnymi narzędziami takimi jak Microsoft PowerPoint lub Microsoft Visio. Istnieją również narzędzia dedykowane wyłącznie do budowy prototypu interfejsu. Różnią się między sobą liczbą dostępnych komponentów, stopniem odzwierciedlenia czy rodzajem oferowanej licencji [2, 11]. Do przeprowadzenia analizy porównawczej wybrane zostały trzy narzędzia komputerowe wspomagające prototypowanie.

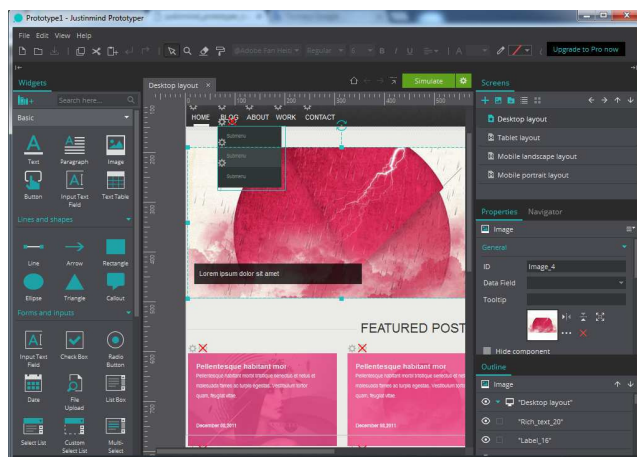
3.1. Narzędzie Justinmind

Justinmind to oprogramowanie do tworzenia interaktywnych prototypów stron internetowych i aplikacji mobilnych. Prezentowane narzędzie udostępnia szeroki zakres gotowych komponentów, dzięki którym możliwe jest zbudowanie najbardziej wymagającego projektu, bez znajomości języków programowania. Konstruowanie interfejsu odbywa się poprzez przeciąganie komponentów i umieszczanie ich na wybranym ekranie. Każdy z gotowych elementów posiada możliwość edycji i dostosowania go do indywidualnych potrzeb jak również zastosowanie interakcji pomiędzy poszczególnymi modułami. Stworzony interfejs można w łatwy sposób przetestować i przekazać innym osobom zaangażowanym w projekt. W przypadku aplikacji mobilnych możliwe jest przetestowanie prototypu na dowolnym urządzeniu przy pomocy dedykowanej aplikacji na systemy operacyjne Android i IOS. Podczas projektowania zespołowego możliwa jest jednoczesna praca wielu osób nad tym samym projektem z funkcją dodania oceny i komentarza dla prezentowanych komponentów. Narzędzie oferuje również wygenerowanie szczegółowej dokumentacji technicznej wykonanego projektu i wyeksportowania go do dowolnego formatu [6]. Wygląd głównego okna programu przedstawiony został na rys. 1.

3.2. Narzędzie Pidoco

Kolejne prezentowane narzędzie do prototypowania interfejsów działa w modelu chmury obliczeniowej SaaS (ang. *Software as a Service*) i nie wymaga pobierania ani instalacji na stacjach roboczych. Do korzystania

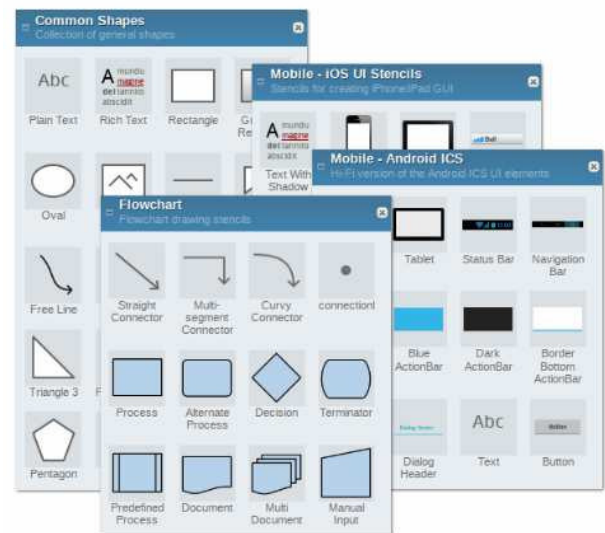
z oprogramowania wymagany jest dostęp do sieci Internet i przeglądarka internetowa. Narzędzie oferuje możliwość planowania, projektowania i testowania stron internetowych, aplikacji mobilnych i aplikacji biznesowych. System umożliwia użytkownikom tworzenie szkieletów makiet i interaktywnych prototypów interfejsów niskiej jakości bez konieczności programowania. Kreowanie interfejsów w danym narzędziu bazuje na przeciąganiu komponentów z dostępnych zasobów i umieszczaniu ich w tworzonej makiecie – rys. 2. Wobec każdego elementu można zastosować predefiniowaną interakcję i umożliwić użytkownikom możliwość poruszania się pomiędzy poszczególnymi ekranami systemu [7].



Rys. 1 Okno projektowe narzędzia Justinmind

3.3. Narzędzie Pencil Project

Pencil Project służy tak jak poprzednie narzędzia do tworzenia makiet stron internetowych, aplikacji mobilnych czy do projektowania rozbudowanych systemów. Narzędzie to jest programem typu desktop i rozpowszechniane jest na licencji typu Freeware. Oprogramowanie można pobrać ze strony producenta i zainstalować na komputerze z systemami operacyjnymi Windows, Os X lub Linux. Pencil Project zapewnia dostęp do różnych kolekcji elementów interfejsu użytkownika, dedykowane są również dla systemów mobilnych Android czy IOS [8]. Na rys. 3 przedstawione są komponenty do wykorzystania w programie Pencil Project.



Rys. 3 Dostępne komponenty programu Pencil Project

4. Problem badawczy i plan badań

Problemem poruszonym w danej pracy jest zagadnienie prototypowania i dedykowane narzędzia do budowania prototypów interfejsów. Każde z narzędzi posiada swój indywidualny zestaw cech, które mogą mieć wpływ na cały proces prototypowania i efekt finalny. Dokonując analizy porównawczej, można zweryfikować przydatność poszczególnych narzędzi i wybrać odpowiednie oprogramowanie, które sprosta wymaganiom projektu.

W pracy postawiona została następująca teza:

Istnieje możliwość wyboru optymalnego narzędzia do budowy prototypu interfejsu.

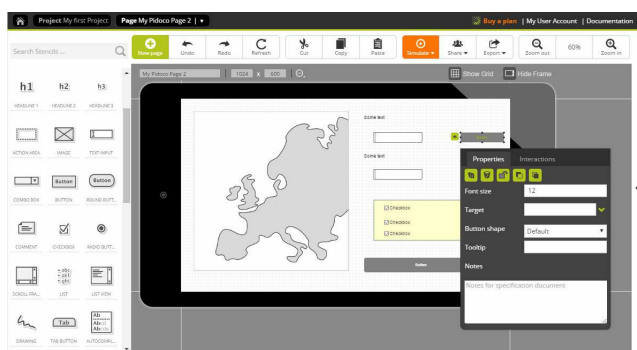
Do przedstawionych problemów zostały również sformułowane następujące hipotezy:

H1. Możliwe jest sformułowanie jednoznacznego kryterium do porównania efektywności narzędzia do budowy prototypów.

H2. Różne narzędzia do budowy prototypów interfejsów mają zróżnicowaną efektywność.

4.1. Przebieg badań

Badanie obejmowało przeprowadzenie pewnego doświadczenia na grupie ludzi, którego celem było zaobserwowanie efektywności wybranych narzędzi, przy budowie zdefiniowanego prototypu systemu. Za jednym z kryteriów efektywności narzędzia uwzględniony został łączny czas zaznajomienia się z danym narzędziem jak również zbudowania prototypu interfejsu. Przed przystąpieniem do badań zdefiniowana została dokumentacja systemu na podstawie, której musiał zostać stworzony model systemu. Dokumentacja została przekazana osobom zaangażowanym w doświadczenie. Każde z narzędzi było testowane przez trzy osoby biorące udział w badaniu. Po zapoznaniu się z danym narzędziem i zbudowaniu prototypu



Rys. 2 Główne okno narzędzia Pidoco

dodatkowo każdy uczestnik badania musiał wypełnić ankietę, składającą się z pewnej liczby pytań, w której miał dokonać oceny danego narzędzia pod względem przyjazności interfejsu, ilości dostępnych komponentów czy stabilności pracy.

Badanie uwzględniało następujące elementy:

- określenie kryterium analizy porównawczej dla narzędzi do budowy prototypów,
- określenie parametrów technicznych,
- wybór parametrów efektywnego prototypowania,
- nadanie wybranym kryteriom odpowiednich wag
- przedstawienie czasów budowy prototypu uzyskanych przez członków grupy badawczej,
- prezentacja wyników ankiet,
- przeprowadzenie normalizacji danych z ankiety przy pomocy wzoru (1),
- przypisanie poszczególnym cechom narzędzi odpowiedniej ilości punktów na podstawie własnych obserwacji i danych uzyskanych z ankiety,
- przedstawienie zsumowanych wyników dla poszczególnych narzędzi przy pomocy wzoru (2),
- ocena uzyskanych wyników analizy.

4.2. Charakterystyka grupy badawczej

Grupa badawcza składała się z dziewięciu osób, która miała za zadanie stworzyć prototyp interfejsu użytkownika w przydzielonym im narzędziu i zmierzyć całkowity czas budowy prototypu systemu. Aby z przeprowadzonych badań można było uzyskać wiarygodne wyniki, a analizę przeprowadzić w sposób rzetelny wybrano grupę badawczą składającą się z absolwentów uczelni wyższych na kierunku informatyka.

Osoby biorące udział w doświadczeniu, nie знаły wcześniej przydzielonego im narzędzia, dzięki czemu była możliwość zbadania całkowitego czasu nauki i tworzenia prototypu.

4.3. Specyfikacja projektu

Do zweryfikowania efektywności wybranych narzędzi sporządzona została dokumentacja projektu z wymaganiami systemu. Projekt obejmował stworzenie prototypu aplikacji mobilnej pod system Android lub iOS, która umożliwi komunikację tekstową pomiędzy osobami, rejestrowanymi i zalogowanymi do systemu.

Wymagania aplikacji mobilnej zostały zdefiniowane w następujący sposób:

- osoba, która chce korzystać z aplikacji musi przejść proces rejestracji użytkownika, podając takie dane jak login, hasło, e-mail, rok urodzenia, i zdjęcie profilowe;
- logowanie do aplikacji mobilnej odbywa się poprzez wprowadzenie przez użytkownika loginu i hasła zadeklarowanych podczas rejestracji;
- użytkownik, który zapomniał swojego loginu lub hasła ma możliwość przywrócenia swoich danych poprzez wprowadzenie podanego podczas rejestracji adresu e-mail, na który wysłany zostanie login z nowym hasłem;
- zalogowany użytkownik ma możliwość nawiązania rozmowy tekstowej z innymi osobami, które dokonały

rejestracji, poprzez zastosowanie wyszukiwania przy wprowadzeniu odpowiedniego kryterium (login, przedział wiekowy) lub skorzystać z dodatkowej funkcji losowego wyszukiwania kontaktów;

- użytkownicy mają możliwość wysyłania pomiędzy sobą wiadomości tekstowych;
- osoba korzystająca z aplikacji mobilnej dysponuje możliwością tworzenia własnej grupy znajomych.

Wymagania prototypu są następujące:

- zbudowany prototyp musi wykazywać pełną interaktywność w taki sposób, aby osoby testujące prototyp aplikacji mogły poruszać się pomiędzy poszczególnymi ekranami systemu;
- wykorzystać dostępne komponenty i zastosować wobec nich odpowiednią interakcję i dynamikę w ramach testowanego narzędzia.

4.4. Dobór zmiennych diagnostycznych

Dobierając parametry porównawcze brane były pod uwagę czynniki, które mogą wpływać na właściwy dobór oprogramowania do budowania prototypów interfejsu użytkownika dla konkretnego projektu jak również wpływ na efektywność procesu prototypowania. Nie wszystkie parametry są tak samo ważne i przy porównywaniu należy zastosować odpowiednie wagi, które ukażą, jakie parametry są istotne, a które w mniejszym stopniu wpłyną na wyniki badania. Parametry zostały podzielone na dwie kategorie. W pierwszej zostały uwzględnione parametry techniczne oprogramowania zaprezentowane w tab. 1.

Tabela 1. Parametry techniczne oprogramowania

Nazwa kryterium (waga)	Wskaźniki	Przypisane wagi
Uruchomienie na systemach operacyjnych (0,3)	wszystkie	1 pkt.
	Windows	0,50 pkt.
	Mac Os X	0,50 pkt.
Budowanie prototypów na platformy (0,2)	wszystkie	1 pkt.
	Android	0,25 pkt.
	iOS	0,25 pkt.
	Web	0,25 pkt.
Odzwierciedlenie interfejsu (0,15)	Duże	0,50 pkt.
	Średnie	0,30 pkt.
	Małe	0,20 pkt.
Eksport prototypu (0,2)	wszystkie	1 pkt.
	HTML	0,33 pkt.
	Aplikacja mobilna	0,34 pkt.
	PDF, .DOC, .PNG	0,33 pkt.
Cena narzędzia (0,15)	Wersja komercyjna	0,12 pkt.
	Wersja bezpłatna	0,70 pkt.
	Wersja próbna	0,18 pkt.

Druga grupa kryteriów odnosi się już do samego procesu prototypowania. Uwzględnione zostały tu czynniki, które realnie wpływają na szybkość wykonanego prototypu. Najważniejszym czynnikiem, który otrzymał najwyższą wagę jest czas wykonania prototypu, który zależy w głównej mierze od łatwości użytkowania i przyswojenia danego narzędzia. Niższy czas budowania prototypu jest lepiej punktowany. Kryteria zostały przedstawione w tab. 2.

Tabela 2. Parametry efektywności

Nazwa kryterium (waga)	Wskaźniki	Przypisane wagi
Czas budowy prototypu (dane uzyskane z eksperymentu) (0,45)	Najniższy czas	0,75 pkt.
	Średni czas	0,20 pkt.
	Najwyższy czas	0,05 pkt.
Wsparcie (poradniki, dokumentacja) (0,05)	Duże	0,75 pkt.
	Średnie	0,15 pkt.
	Małe	0,10 pkt.
	Brak	0 pkt.
Testowanie na urządzeniach mobilnych (0,1)	Tak	1 pkt.
	Nie	0 pkt.
Interaktywność prototypu (0,05)	Duża	0,75 pkt.
	Średnia	0,15 pkt.
	Niska	0,10 pkt.
Stopień trudności budowania prototypu (dane uzyskane z ankiety) (0,1)	Niski (1-2)	0,75 pkt.
	Średni (2-4)	0,15 pkt.
	Wysoki (4-5)	0,10 pkt.
Stabilność pracy narzędzia (dane uzyskane z ankiety) (0,1)	Duża (4-5)	0,75 pkt.
	Średnia (2-4)	0,15 pkt.
	Niska (1-2)	0,10 pkt.
Szybkość działania programu (dane uzyskane z ankiety) (0,1)	Duża (4-5)	0,75 pkt.
	Średnia (2-4)	0,15 pkt.
	Niska (1-2)	0,10 pkt.
Liczba dostępnych komponentów (dane uzyskane z ankiety) (0,05)	Duża (3-4)	0,75 pkt.
	Średnia (2-3)	0,15 pkt.
	Niska (1-2)	0,10 pkt.

5. Rezultaty badań

Z przeprowadzonych badań uzyskano prezentowane w tab. 3 czasy budowy prototypów interfejsów dla poszczególnych narzędzi.

Tabela 3. Czasy budowy prototypów

Czas w godzinach	Narzędzia do prototypowania		
	Pencil Project	Justinmind	Pidoco
t1	1:22	1:55	0:16
t2	1:10	1:45	0:38
t3	1:35	1:20	0:46
Średni czas	1:22	1:4	0:42

Uśrednione wyniki ankiet otrzymane od uczestników badania przedstawione zostały w tab. 4.

Do wybranych kryteriów zostały przydzielone punkty zgodnie z uwzględnionymi kryteriami zawartymi w tab. 1 dla każdego z badanych narzędzi. System przydzielania punktów uwzględniał informacje zawarte na stronach internetowych producentów oprogramowania, jak również została przeprowadzona indywidualna weryfikacja każdego kryterium dla wybranych narzędzi. Wyniki zostały zaprezentowane w tab. 5.

Do ukazania jednoznacznych wyników analizy efektywnego prototypowania dla przedstawionych narzędzi przypisane zostały punkty do wybranych wcześniej parametrów. System naliczania punktów uwzględniał analizę każdego narzędzia pod kontem wcześniej ustalonych kryteriów z uwzględnieniem poszczególnych progów punktowania. Część punktów została uzyskana na podstawie uzyskanych ankiet po przeprowadzeniu normalizacji skali. tab. 6 przedstawia przydzielone punkty i znormalizowaną sumę wyników dla poszczególnego narzędzia.

Tabela 4. Wyniki ankiet

Treść pytania (skala)	Narzędzie do prototypowania		
	Pencil Project	Justinmind	Pidoco
1. Stopień trudności budowania prototypu (1-5)	1,67	2,33	2,67
2. Stabilność pracy narzędzia (1-5)	4	5	3,67
3. Szybkość działania programu (1-5)	4,67	4,67	4,33
4. Liczba dostępnych komponentów (1-4)	2,67	3,67	2,33

Tabela 5. Ocena narzędzi względem parametrów technicznych

Nazwa kryterium	Pencil Project (punkty)	Justinmind (punkty)	Pidoco (punkty)
Uruchomienie na systemach operacyjnych	1	1	1
Budowanie prototypów na wybrane platformy	1	1	1
Odzwierciedlenie interfejsu	0,2	0,5	0,3
Eksport prototypu	0,33	1	1
Cena narzędzia	0,7	0,3	0,3
Suma wyników po normalizacji	0,7	0,82	0,79

Tabela 6. Ocena narzędzi względem parametrów technicznych

Nazwa kryterium	Pencil Project (punkty)	Justinmind (punkty)	Pidoco (punkty)
Czas budowy prototypu	0,2	0,05	0,75
Wsparcie (poradniki, dokumentacja)	0,15	0,75	0,75
Testowanie na urządzeniach mobilnych	0	1	1
Interaktywność prototypu	0,10	0,75	0,75
Stopień trudności budowania prototypu	0,75	0,15	0,15
Stabilność pracy	0,75	0,75	0,15
Szybkość działania programu	0,75	0,75	0,75
Liczba dostępnych komponentów	0,15	0,75	0,15
Suma wyników po normalizacji	0,35	0,51	0,59

6. Analiza porównawcza narzędzi do prototypowania interfejsów

Na podstawie formuły (1) i (2) wyznaczone zostały wyniki dla parametrów technicznych i czynników efektywności. Oba główne parametry objęte analizą potraktowane zostały w sposób równoważny i przypisana została im jednakowa waga. Suma wyników po normalizacji przedstawia końcową liczbę uzyskanych punktów przez każde z narzędzi. Końcowe zestawienie punktów uzyskanych przez poszczególne narzędzie przedstawia tab. 7.

Tabela 7. Końcowe zestawienie wyników analizy porównawczej

Nazwa kryterium	Pencil Project (punkty)	Justinmind (punkty)	Pidoco (punkty)
Parametry techniczne	0,7	0,82	0,79
Parametry efektywności	0,35	0,51	0,59
Suma wyników po normalizacji	0,53	0,67	0,70

Analizując tab. 3 z uzyskanymi czasami można zauważyć, że każde badane narzędzie uzyskało inny czas, co przekłada się na różną efektywność budowania prototypu. Najniższy średni czas uzyskało narzędzie Pidoco. Na podstawie przeprowadzonej oceny poszczególnych narzędzi pod względem efektywności i po zsumowaniu wyników, można jednoznacznie stwierdzić, odczytując z tab. 7, że narzędzie Pidoco uzyskało najwyższą liczbę punktów. Głównym czynnikiem, który zadecydował o wyniku był uzyskany czas budowania prototypu. Drugim narzędziem, które uzyskało mniejszą liczbę punktów jest Justinmind.

Z przedstawionych wyników cząstkowych wynioskować można, że narzędzie to cechuje rozbudowany interfejs, który posiada bardzo dużą liczbę komponentów w przeciwieństwie do pozostałych narzędzi. Narzędzie, które w tej klasyfikacji otrzymało najmniejszą liczbę punktów to oprogramowanie Pencil Projekt. Do najsłabszych cech można zaliczyć liczbę dostępnych komponentów czy interaktywność prototypu. Brak możliwości przetestowania prototypu na urządzeniach mobilnych również wpłynęła na niską ocenę.

7. Wnioski

Na podstawie przeprowadzonych badań i uzyskanych wyników zamieszczonych w ramach niniejszej pracy można stwierdzić, że narzędzia do prototypowania interfejsów cechują się różną efektywnością. Odwołując się do hipotezy, która mówi, że różne narzędzia do budowy prototypów interfejsów mają zróżnicowaną efektywność i dokonując analizy tab. 3 można jednoznacznie zauważyć, że każde prezentowane narzędzie uzyskało inny wynik. Oznacza to, że hipoteza H1 została udowodniona.

Dokonując analizy sformułowanych parametrów zamieszczonych w tab. 6, które realnie wpływają na efektywność prototypowania można odnieść się do drugiej hipotezy, która mówi, że możliwe jest sformułowanie

jednoznacznego kryterium do porównania efektywności narzędzia do budowy prototypów. W tym przypadku również druga hipoteza (H2) została udowodniona.

Przeprowadzając ostateczne porównanie prezentowanych narzędzi zamieszczonych w tab. 7 i odwołując się do postawionej tezy, która mówi, że istnieje możliwość wyboru optymalnego narzędzia do budowy prototypu interfejsu, odczytać można, że najbardziej optymalnym narzędziem dla przedstawionego problemu okazało się oprogramowanie Pidoco. Z uzyskanej analizy i wyników można jednoznacznie stwierdzić, że postawiona teza została potwierdzona.

Biorąc pod uwagę przeprowadzoną analizę, można wysnuć stwierdzenie, że wybór odpowiedniego narzędzia zależy od rodzaju projektu i wymagań stawianych, co do prototypowania. Każdy nowo budowany system wymaga indywidualnego podejścia w stosunku do procesu prototypowania. Stawiając odpowiednie warunki, możliwe jest wybranie odpowiedniego narzędzia, które jest w stanie sprostać oczekiwaniom projektu.

Literatura

- [1] A. Sears, J. Jacko, Human- Computer Interaction: Development Process, Taylor & Francis Group, 2009.
- [2] A. Leonidis, M. Antona, C. Stephanidis, Rapid Prototyping of Adaptable User Interfaces, International Journal of Human-Computer Interaction, 2012, 213-235.
- [3] K. Sacha, Inżynieria oprogramowania. Wydawnictwo naukowe PWN, 2010.
- [4] EvolutionaryPrototyping
http://www.teachict.com/as_a2_ict_new/ocr/A2_G063/331_sys_ems_cycle/prototyping_RAD/miniweb/pg2.htm
(dostęp 05.2016r)
- [5] P. Weichbroth, M. Sikorski, User interface prototyping, techniques, methods and tools, Uniwersytet Ekonomiczny w Katowicach, 2015.
- [6] Justinmind: Prototyping platform for web and mobile apps
<https://www.justinmind.com> (dostęp 07.2016)
- [7] Wireframing Design Tools- Pidoco
<https://pidoco.com/en/tour/features> (dostęp 07.2016)
- [8] Prototype Tools- Pencil Project
<http://pencil.evolus.vn/Features.html> (dostęp 07.2016)
- [9] T. Panek, Statystyczne metody wielowymiarowej analizy porównawczej. Szkoła Główna Handlowa - Oficyna Wydawnicza, 2009.
- [10] A. Sears, J. Jacko, Human- Computer Interaction: Development Process. Taylor & Francis Group, 2009.
- [11] J. Arnowitz, M. Arent, N. Berger, Effective Prototyping For Software Makers. Amsterdam, Morgan Kaufmann, 2007.
- [12] A. Cid-López, M. Hornos, R. Carrasco, E. Herrera-Viedma, Applying a linguistic multi-criteria decision-making model to the analysis of ICT suppliers' offers. Expert Systems with Applications, 57, 2016, 127-138.
- [13] W. Bleek, M. Jeenicke, R. Klischewski, Developing Web-based Applications through e-Prototyping. Hamburg University, Department for Informatics, Software Engineering Group, 2002
- [14] L. Teixeira, V. Saavedra, C. Ferreira, J. Simoes, B. Santos, Requirements Engineering Using Mockups and Prototyping Tools: Developing a Healthcare Web-Application, Heidelberg, 2014, 652-663.

Analiza porównawcza jakości serwisów internetowych polskich banków

Michał Osiński*, Marek Miłoś

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

Streszczenie. W artykule przedstawiono rezultaty analizy porównawczej jakości serwisów internetowych i mobilnych szesnastu największych banków działających na terenie Polski. Analizę przeprowadzono pod kątem wielokryterialnego wyboru najbardziej odpowiedniego serwisu w konkretnych warunkach. Zdefiniowano i wykorzystano w analizie szesnaście różnych kryteriów z trzech obszarów: funkcjonalność i dostępność, innowacyjność oraz bezpieczeństwo. Przy pomocy metody ważonych kryteriów wyznaczono ranking serwisów w każdej z grup kryteriów i dla kryterium końcowego. Analiza wielokryterialna pozwoliła wskazać systemy bankowe o najwyższej jakości.

Słowa kluczowe: bankowość elektroniczna; ocena jakości; serwisy internetowe

*Autor do korespondencji.

Adres E-mail: michal.osinski@pollub.edu.pl

Comparative analysis of the quality of Polish banks internet services

Michał Osiński*, Marek Miłoś

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

Abstract. The article presents the results of a comparative analysis of the quality of websites and mobile websites of sixteen largest banks operating in Poland. The analysis was conducted taking into account multiple criteria of choosing the optimal website under certain circumstances. Sixteen different criteria in three different areas have been defined, those areas being: functionality and accessibility, innovation, safety. Using the method of weighted criteria, a ranking has been made in each of the groups and for the final criterion. Multiple-criteria decision analysis allowed the identification of the highest quality banking systems.

Keywords: electronic banking; analysis of the quality; websites

*Corresponding author.

E-mail address: michal.osinski@pollub.edu.pl

1. Wprowadzenie

Sprzedaż produktów i usług w Internecie od pewnego czasu stała się zjawiskiem powszechnym. Firmy decydują się na nią, znając powszechną opinię, która mówi że „jeśli nie ma Cię w sieci to znaczy, że nie istniejesz”. Polski rynek usług bankowych również przeszedł istotne przemiany odnoszące się do struktury i funkcjonowania podmiotów na rynku. Banki używają nowych technologii do obsługi swoich klientów. Internet stanowi główny kanał dystrybucji oferowanych usług, a tradycyjne placówki bankowe powoli odchodzą do lamusa.

Jednym z początkowych efektów wykorzystania Internetu w bankowości było stworzenie stron informacyjnych, które z upływem czasu zostały wzbogacone o tzw. bankowość elektroniczną, wdrażaną w celu ułatwienia komunikacji, umocnienia pozycji na rynku i obniżenia kosztów świadczenia usług. Bankowość elektroniczna zdobyła akceptację tych klientów, którzy cenią wygodę, szybkość obsługi oraz możliwość samodzielnego wykonywania niektórych czynności bankowych [1].

Liczba osób korzystających z bankowości elektronicznej rośnie w bardzo szybkim tempie. W raporcie „Bankowość internetowa i płatności bezgotówkowe” według stanu na koniec 2015 roku, podano, iż liczba klientów indywidualnych z podpisaną umową dającą dostęp do bankowości internetowej wynosiła 30,3 mln i w porównaniu do roku poprzedniego

wzrosła o prawie 21% [2]. Liczba aktywnych klientów, czyli takich, którzy logują się przynajmniej raz w miesiącu wynosi 14,5 mln [2].

Bankowy serwis internetowy wysokiej jakości ma duże znaczenie dla klientów, co potwierdzają badania „Omnibus 2011” [3], gdzie 38% badanych wskazało funkcjonalną bankowość elektroniczną jako czynnik, który w znaczący sposób przyczynia się do wybrania oferty konkretnego banku. Na maksymalnym skróceniu czasu potrzebnego do załatwienia spraw w banku w szczególności zależy młodym użytkownikom bankowości - studentom i absolwentom [3].

Niniejszy artykuł prezentuje analizę serwisów internetowych i mobilnych szesnastu największych instytucji finansowych w Polsce, a także ich porównanie z rekomendacją wyboru najlepszych systemów.

2. Metody oceny jakości

W celu zbadania jakości serwisów internetowych banków działających na terenie Polski należy wybrać odpowiednią metodę ich porównania. Modeli oceny serwisów internetowych jest wiele i w znacznym stopniu się od siebie różnią. Część z nich polega na wypełnianiu ankiet przez różne grupy użytkowników, a następnie analizowaniu ich, inne na jednoosobowych badaniach. Modele różnią się także wymiarami używanymi do opisu jakości stron. Posiadają

różną ich liczbę, dzieląc wymiary na kategorie lub też nie. Część modeli różnicuje wymiary lub kategorie pod względem ich wagi, inne uznają wszystkie za równorzędne.

Popularną metodą oceny jakości stron internetowych jest WebQual [4]. Metodę inaczej zwaną eQual opracowało dwóch profesorów: Stuart Barnes oraz Richard Vidgen w celu pozyskania opinii o jakości witryny od jej użytkowników. Z upływem czasu metoda ewaluowała i w kolejnych jej wersjach wymiary oceny jakości były dostosowywane do badanej branży [5]. Polega ona na wypełnieniu kwestionariusza, który zawiera pytania na temat takich aspektów jak: jakość informacji na stronie internetowej, jakość obsługi, jakość integracji z użytkownikiem, a także design strony [4]. Odpowiedzi udzielone na pytania w każdej z kategorii są punktowane i na ich podstawie wydawana jest ocena końcowa.

Kolejną metodą stosowaną do oceny jakości portalu internetowego jest e-Servqual [6]. Jest to model przystosowany do badania jakości usług w Internecie, wywodzący się bezpośrednio z modelu Servqual, który przystosowano do oceny jakości usług tradycyjnych. W modelu tym portale oceniane są przy pomocy 7 kryteriów [6]: Niezawodność, Empatia, Namacalność, Łatwość nawigacji, Odpowiednia reprezentacja, Dokładność oraz Bezpieczeństwo. Dla każdego z kryteriów w ankiecie do oceny występuje określona liczba stwierdzeń na temat jakości serwisów. Respondenci przyznają punkty od 1 do 5 oznaczające w jakim stopniu się z nimi zgadzają [6].

Oceną jakości witryn internetowych zajmuje się również Witold Chmielarz z Katedry Systemów Informacyjnych Zarządzania Uniwersytetu Warszawskiego [7]. W swoich publikacjach szeroko opisuje temat analizy jakości. Do badań Witold Chmielarz wykorzystuje metodę punktową polegającą na przyznaniu od 0 do 1 punktu za poszczególne kryteria. Są one podzielone na grupy w zależności od badanego tematu. W każdej grupie poszczególne kryteria zostają poddane ocenie przez ekspertów. Przydzielają oni punkty w zależności od tego w jakim stopniu dane kryterium zostało spełnione. Następnie z tych ocen zostaje wyciągnięta mediana i wystawiana jest ocena w postaci punktowej dla każdego kryterium [8].

Analiza taksonomiczna służy do pogrupowania oraz uporządkowania obiektów o wielu cechach w przestrzeni wielowymiarowej. Istnieje wiele metod taksonomicznych pomocnych w klasyfikacji tychże obiektów [9]. Do najczęściej stosowanych należą: metody centroidalne, Warda, k najbliższych sąsiedztw oraz taksonomii wrocławskiej [10]. Różnią się one między sobą przede wszystkim sposobem obliczania odległości taksonomicznej. Metody taksonomiczne można podzielić na bezwzorcowe, czyli takie, które opierają się na wyliczonych średnich arytmetycznych, oraz na metody wzorcowe [11]. W bezwzorcowych należy skonstruować miernik syntetyczny agregatowy, na podstawie znormalizowanych wartości cech. Natomiast we wzorcowych skonstruować należy sztuczne punkty odniesienia czyli tzw. taksonomiczne mierniki rozwoju. Następnie mierzona jest odległość od tych wzorców i następuje konstrukcja miernika syntetycznego [10].

Metoda taksonomii wrocławskiej jest metodą niehierarchiczną, która pozwala na podział badanych obiektów na grupy o tych samych cechach. Obiekty te można uporządkować liniowo lub nieliniowo, a następnie odwzorować na płaszczyźnie punkty przestrzeni wielowymiarowej, w taki sposób aby odległość między nimi była jak najmniejsza [12].

W celu oceny jakości systemów internetowych polskich banków można również skorzystać z optymalizacji wielokryterialnej. Jest ona możliwa do zastosowania w przypadkach gdzie konieczny jest wybór pomiędzy wieloma celami (kryteriami). Metod optymalizacji wielokryterialnej jest wiele. Do najpopularniejszych należą [13]:

- 1) Metoda ważonych kryteriów - polega na sprowadzeniu poszczególnych funkcji celu w jedną funkcję celu (kryterium zastępcze) poprzez ich zsumowanie [14].
- 2) Metoda ograniczonych kryteriów – polega na zminimalizowaniu wartości funkcji jednego z kryteriów przy ograniczeniach innych funkcji [14]. Metoda ta jest dokładniejsza niż metoda ważonych kryteriów.
- 3) Metoda optymalizacji hierarchicznej – polega na sprowadzeniu optymalizacji wielokryterialnej do jednokryterialnej poprzez wykonywanie ich w zadanej kolejności [15].

3. Obiekt badań

Do analizy porównawczej początkowo wybrano 16 największych pod względem sumy aktywów banków w Polsce według miesięcznika finansowego Bank [16]:

- 1) PKO Bank Polski S.A.,
- 2) Bank Pekao S.A.,
- 3) Bank Zachodni WBK S.A.,
- 4) mBank S.A.,
- 5) ING Bank Śląski S.A.,
- 6) Getin Noble Bank S.A.,
- 7) Bank Millennium S.A.,
- 8) Bank BGŻ BNP Paribas S.A.,
- 9) Raiffeisen Polbank S.A.,
- 10) Bank Handlowy w Warszawie S.A.,
- 11) Bank Gospodarstwa Krajowego,
- 12) Alior Bank S.A.,
- 13) Deutsche Bank S.A.,
- 14) Bank BPH S.A.,
- 15) Credit Agricole Bank Polska S.A.,
- 16) Bank Ochrony Środowiska S.A.

Podczas pracy okazało się, że nie wszystkie banki umożliwiają potencjalnym klientom dostęp demo do bankowości elektronicznej. Z tego względu zaistniała konieczność wyłączenia z analizy czterech podmiotów: Bank Handlowy w Warszawie S.A., Alior Bank S.A., Deutsche Bank S.A., Bank BPH S.A. Ponadto z analizy wyłączyć należało Bank Gospodarstwa Krajowego, który jest powołany do wspierania programów rządowych, samorządowych oraz rozwoju regionalnego i nie oferuje kont bankowych dla klientów indywidualnych.

Ostatecznie więc w przeprowadzonej analizie jakości weźmie udział 11 banków z czołówki sektora finansowego w Polsce.

4. Metoda badawcza

Hipoteza główna o wysokiej jakości serwisów internetowych polskich banków zostanie poparta udowodnieniem hipotez szczegółowych:

- H1. Oferty banków są zróżnicowane i skierowane do różnych grup docelowych.
- H2. Istnieje możliwość efektywnego wykorzystania metody kryteriów ważonych do porównania jakości serwisów internetowych polskich banków.
- H3. Interfejsy systemów bankowych są responsywne.
- H4. Wszystkie usługi oferowane przez banki są dostępne przez serwis internetowy.

Trudno obiektywnie zmierzyć jakość portali internetowych. Nie wszystkie z metod oceny jakości serwisów internetowych można zastosować do oceny bankowości elektronicznej. Analizę porównawczą zdecydowano się przeprowadzić nieco zmodyfikowaną metodą punktową, której zastosowanie proponuje w swoich pracach profesor Witold Chmielarz z Uniwersytetu Warszawskiego [17]. Metoda ta jest przejrzysta i przyjazna zarówno dla osoby przeprowadzającej badanie jak i późniejszych odbiorców wyników tych badań. Główne modyfikacje jakie zdecydowano się wprowadzić polegają na zmianie sposobu przyznawania punktacji. Za spełnienie konkretnych kryteriów przyznane zostaną punkty w skali od 0 do 5 (jedynie wartości całkowite), a nie jak w pierwotnej metodzie (0, 0.25, 0.50, 0.75 lub 1 punkt). Większy rozstęp pomiędzy możliwymi wynikami pozwoli na dokładniejszą i bardziej zróżnicowaną ocenę poszczególnych systemów. Ze względu na konstrukcję kryteriów, umożliwiającą jednoznaczne wyznaczenie ich wartości na podstawie analizy serwisów, badania zostały przeprowadzone jednoosobowo na próbie badawczej szesnastu systemów internetowych.

Do porównania wybrano szesnaście kryteriów, które podzielono na trzy kategorie:

- 1) Funkcjonalność i dostępność.
- 2) Innowacje.
- 3) Bezpieczeństwo.

W celu ustalenia oceny końcowej przyjmuje się, że każda z powyższych kategorii w równym stopniu wpływa na ocenę końcową i suma wartości poszczególnych kategorii wynosi 1.

W każdej z kategorii zawarto określoną liczbę kryteriów, za które przyznano punkty w skali od 0 do 5. Nie przyznawano punktów niepełnych. I tak odpowiednio:

- 5 punktów to najwyższy możliwy do osiągnięcia wynik. Przyznaje się je pod warunkiem osiągnięcia pełnej satysfakcji,
- 4 punkty przyznaje się, gdy osiągnięto satysfakcję, lecz dostrzega się możliwość pewnych ulepszeń w danej dziedzinie,

- 3 punkty to wynik środkowy. Oznacza spełnienie kryterium w stopniu znacznym, lecz pozostawia ono wiele do życzenia,
- 2 punkty to spełnienie kryterium w niewielkim stopniu,
- 1 punkt przyznaje się w przypadku znikomego spełnienia kryterium,
- 0 punktów oznacza całkowite niespełnienie danego kryterium.

Korzystając z optymalizacji wielokryterialnej, a ściślej z metody ważonych kryteriów, ocena końcowa zostaje wyznaczona z następującego wzoru:

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

Gdzie:

Q – liczba kategorii,
 N – liczba kryteriów w danej kategorii,
 w_k – waga k -tej kategorii gdzie $w_1=w_2=\dots=w_n$, $w_1+\dots+w_n=1$,
 y_{ki} – wartość i -tego kryterium w k -tej kategorii,
 m_k – maksymalna liczba punktów za kryterium.

5. Kryteria doboru serwisu internetowego banku

Kryteria oceny jakości serwisów dobrano tak, by ich wartości można było w sposób jednoznaczny (i obiektywny) wyznaczyć na podstawie analizy oferty i serwisu internetowego banku. Natomiast dobór kryteriów oraz ich wag jest subiektywnym wkładem autora.

Funkcjonalność i dostępność

W kategorii funkcjonalność i dostępność zawarto kryteria świadczące o dostępności i intuicyjności produktu oraz takie, które ułatwiają korzystanie z niego. Zaliczają się do nich:

- 1) Kanał otwarcia konta – Jakimi drogami można otworzyć konto, np. w placówce, przez Internet, telefon, aplikację mobilną banku.
- 2) Responsive Web Design – Wykonanie serwisu w technologii RWD – 5 punktów, brak wykonania w tej technice – 0 punktów.
- 3) Kanały dostępu do konta – Możliwość wykonywania operacji zarówno w placówce, poprzez system transakcyjny banku, infolinie, a także aplikację mobilną banku.
- 4) Zmiana limitów – Możliwość zmiany limitów przelewów oraz limitów dotyczących płatności kartami przez serwis internetowy banku.
- 5) Powiadomienia – Dopuszczalność ustawienia powiadomień o ruchach na rachunku za pomocą wiadomości SMS, mail lub powiadomienia push.
- 6) Wydajność serwisu – Mierzona za pomocą narzędzia PageSpeed Insights. 0 punktów w przypadku uzyskania wyniku z przedziału 0-20, 1 punkt dla przedziału 21-36, 2 punkty dla przedziału 37-52, 3 dla przedziału 53-68, 4 dla przedziału 69-82, 5 za wynik 83 lub wyższy.

- 7) Zgodność ze standardem WCAG 2.0 AA – Ocena wystawiona na podstawie badania strony za pomocą narzędzia AChecker. Liczba punktów tym większa im mniejsza liczba wykrytych problemów.
- 8) Usługi dodatkowe – Udostępnianie przez bank możliwość zakupu przez system transakcyjny dodatkowych usług takich jak: ubezpieczenia, fundusze, doładowanie telefonu na kartę, czy wymianę walut.

Innowacje

Innowacje to kategoria, w której punkty przyznawane są za wdrażanie nowych usług i produktów, które nie są jeszcze standardem na polskim rynku usług bankowych. W tej kategorii ocenie podlegały następujące kryteria:

- 1) Oferta dla Ciebie – Czy bank proponuje klientom przez Internet produkty na podstawie dotychczasowej relacji z bankiem.
- 2) Planer wydatków – Czy system kategoryzuje wydatki ułatwiając ich przegląd.
- 3) Innowacyjne możliwości płacenia – Możliwość płatności legitymacją / kartą zbliżeniową / naklejką zbliżeniową / NFC / BLIK. Przyznawany jeden punkt za każdą z powyższych możliwości.
- 4) Aplikacja mobilna – Czy bank posiada aplikację na różne mobilne systemy operacyjne (po jednym punkcie za aplikację na system operacyjny iOS, Android oraz dowolny inny. Maksymalnie 3 pkt). Czy bank umożliwia przysyłanie pieniędzy na numer telefonu (1 pkt), Funkcja Skanuj i płać (1 pkt).
- 5) Ocena aplikacji mobilnej – Ocena aplikacji mobilnej przez użytkowników sklepu Google Play. 0 punktów w przypadku uzyskania wyniku poniżej 2.5, 1 punkt dla przedziału 2.5-2.9, 2 punkty dla przedziału 3.0-3.4, 3 dla przedziału 3.5-3.9, 4 dla przedziału 4.0-4.4, 5 za wynik 4.5 lub wyższy.
- 6) Inne – Dodatkowe osiągnięcia w dziedzinie innowacji. Maksymalnie 5 punktów.

Bezpieczeństwo

Choć w kategorii bezpieczeństwo jest niewiele kryteriów, ma ona duże znaczenie, przy wyborze najlepszego systemu bankowego. Bezpieczeństwo pieniędzy jest niezwykle ważne i należy zwracać uwagę na proces logowania oraz potwierdzania transakcji bankowych dokonywanych za pośrednictwem Internetu. Ocenie podlegały kategorie:

- 1) Proces logowania – Hasło maskowane, obrazek bezpieczeństwa, potwierdzenie logowania hasłem SMS lub kodem jednorazowym, szyfrowanie strony, klawiatura ekranowa. Po jednym punkcie za każdy spełniony powyższy podpunkt.
- 2) Bezpieczeństwo transakcji – Punkty przyznawane za możliwość włączenia powiadomień o dokonanej transakcji oraz za usługę 3-D Secure.

6. Rezultaty badań

Po wykonaniu szczegółowej analizy serwisów internetowych polskich banków nastąpiło wypunktowanie cech w każdej z określonych kategorii.

Kategorię Funkcjonalność i dostępność podzielono na 8 cech, co daje łącznie możliwość zdobycia 40 punktów. Tutaj liderem okazał się Bank Zachodni WBK, który zdobył 31 punktów. Na drugim miejscu z wynikiem 30 punktów uplasował się Bank Millennium, a na kolejnym miejscu z 29 punktami znalazł się ING Bank Śląski. Szczegółowe wyniki przedstawia tabela 1.

Tabela 1. Ranking systemów w kategorii Funkcjonalność i dostępność

Pozycja	Bank	Wynik
1	Bank Zachodni WBK S.A.	31
2	Bank Millennium S.A.	30
3	ING Bank Śląski S.A.	29
4	PKO Bank Polski S.A.	28
5	Bank Pekao S.A.	27
6	mBank S.A.	25
7	Credit Agricole Bank Polska S.A.	24
8	Raiffeisen Polbank S.A.	23
9	Getin Noble Bank S.A.	22
10	Bank Ochrony Środowiska S.A.	21
11	Bank BGŻ BNP Paribas S.A.	20

W rankingu systemów bankowych w kategorii Innowacje największą liczbę punktów zdobyły: Bank Millennium oraz Bank Zachodni WBK. Osiągnęły one po 28 punktów na 30 możliwych. Kolejny w rankingu – mBank zdobył 26 punktów. Szczegółowe wyniki w przedstawia tabela 2.

Tabela 2. Ranking systemów bankowych w kategorii Innowacje

Pozycja	Bank	Wynik
1	Bank Millennium S.A.	28
1	Bank Zachodni WBK S.A.	28
3	mBank S.A.	26
4	Getin Noble Bank S.A.	22
4	ING Bank Śląski S.A.	22
6	PKO Bank Polski S.A.	19
7	Bank Pekao S.A.	18
8	Bank BGŻ BNP Paribas S.A.	12
9	Credit Agricole Bank Polska S.A.	11
10	Raiffeisen Polbank S.A.	9
11	Bank Ochrony Środowiska S.A.	2

W kategorii Bezpieczeństwo wyróżniono cechy dotyczące procesu logowania oraz bezpieczeństwa dokonywanych transakcji. Do zdobycia było maksymalnie 10 punktów. Tę liczbę udało się osiągnąć systemom aż dwóch banków: Banku BGŻ BNP Paribas oraz Banku Zachodni WBK. Za ledwie o jeden punkt mniej zdobył Bank Millennium, natomiast z wynikiem ośmiu punktów uplasowały się: Bank Pekao, ING Bank Śląski oraz Getin Noble Bank. Szczegółowe wyniki przedstawiono w tabeli 3.

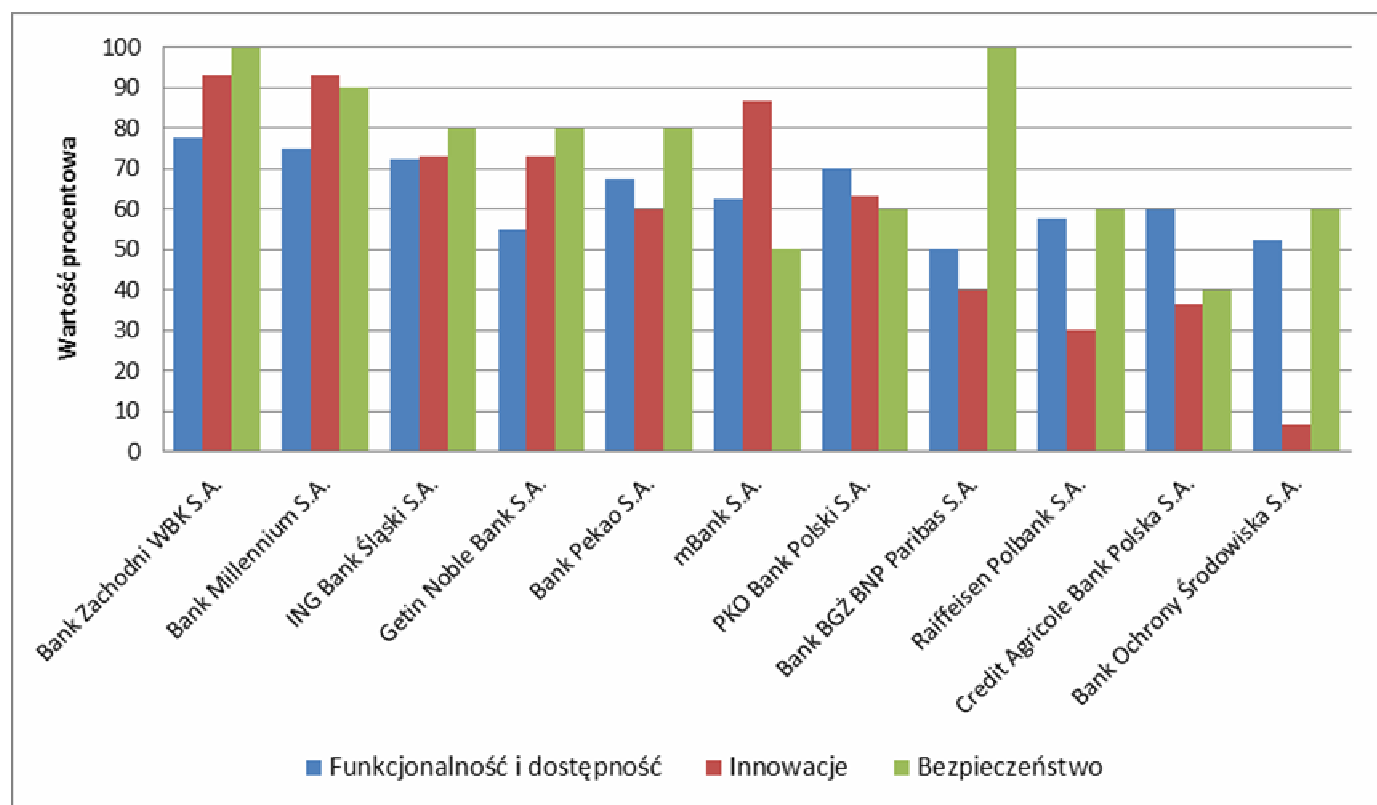
Tabela 3. Ranking systemów bankowych w kategorii Bezpieczeństwo

Pozycja	Bank	Wynik
1	Bank BGŻ BNP Paribas S.A.	10
1	Bank Zachodni WBK S.A.	10
3	Bank Millennium S.A.	9
4	Bank Pekao S.A.	8
4	ING Bank Śląski S.A.	8
4	Getin Noble Bank S.A.	8
7	Raiffeisen Polbank S.A.	6
7	PKO Bank Polski S.A.	6
7	Bank Ochrony Środowiska S.A.	6
10	mBank S.A.	5
11	Credit Agricole Bank Polska S.A.	4

Wynik końcowy dla każdego banku obliczany jest na podstawie punktów uzyskanych w każdej kategorii ze wzoru (1). Wyniki porównania serwisów internetowych polskich banków zostały przedstawione w tabeli 4. Natomiast wykres 1 prezentuje wyniki porównania w poszczególnych kategoriach przedstawione w postaci procentowej.

Tabela 4. Porównanie serwisów internetowych banków

Pozycja	Bank	Wynik
1	Bank Zachodni WBK S.A.	90,28
2	Bank Millennium S.A.	86,11
3	ING Bank Śląski S.A.	75,28
4	Getin Noble Bank S.A.	69,44
5	Bank Pekao S.A.	69,17
6	mBank S.A.	66,39
7	PKO Bank Polski S.A.	64,44
8	Bank BGŻ BNP Paribas S.A.	63,33
9	Raiffeisen Polbank S.A.	49,17
10	Credit Agricole Bank Polska S.A.	45,56
11	Bank Ochrony Środowiska S.A.	39,72



Wykres 1. Porównanie serwisów internetowych banków w trzech kategoriach

7. Wnioski

Hipoteza H2 została potwierdzona. W wyniku przeprowadzonych badań serwisów internetowych polskich banków pierwsze miejsce przypadło dla Banku Zachodniego WBK S.A., który zajął również pierwsze miejsce w każdej z trzech kategorii i zdobył 90,28 punktów na 100 możliwych. Drugie miejsce zdobył Bank Millennium S.A. (86,11 punktów), a na trzecim znalazł się ING Bank Śląski S.A. osiągając 75,28 punktów. W czołówce rankingu zaobserwować można więc dość duże różnice pomiędzy analizowanymi systemami bankowymi. Kolejne dwa miejsca z bardzo zbliżoną liczbą punktów to: Getin Noble Bank S.A. (69,44) oraz Bank Pekao S.A. (69,17). Dalsze miejsca z nieco większą różnicą punktową to systemy: mBank S.A. (66,39), PKO Bank Polski S.A. (64,44), Bank BGŻ BNP Paribas S.A. (63,33), Raiffeisen Polbank S.A. (49,17), Credit Agricole Bank Polska S.A. (45,56) oraz Bank Ochrony Środowiska S.A. (39,72).

Dokonana analiza częściowo potwierdza tezę o wysokiej jakości serwisów internetowych polskich banków. Serwisy banków zostały przeanalizowane pod wieloma kryteriami. Ich oferty są zróżnicowane i skierowane do różnych grup docelowych. Analizowane banki są bankami uniwersalnymi, świadczą one zróżnicowane usługi dla klientów indywidualnych jak i dla przedsiębiorstw. Część z nich ma w swojej ofercie konta dla ludzi młodych o obniżonych progach opłat. Potwierdza to hipotezę H1.

Nie wszystkie usługi oferowane przez banki są dostępne przez serwis internetowy. Część z nich jest bardziej rozbudowana i oferuje wiele usprawnień i udogodnień, inne są proste i można tam wykonać jedynie podstawowe operacje. Hipoteza H4 nie została potwierdzona.

Największym zaskoczeniem in minus jest fakt, iż w chwili obecnej tylko dwa z przeanalizowanych systemów (Bank Millennium S.A. oraz ING Bank Śląski S.A.) zostały wykonane w technologii Responsive Web Design. Technika ta, do projektowania stron stosowana jest już od kilku lat, jednak w bankowości elektronicznej wykorzystywana jest dość rzadko. Zamiast niej większość banków posiada uproszczone wersje mobilne swoich systemów oraz aplikacje mobilne. W związku z powyższym należy uznać, że hipoteza H3 została zweryfikowana negatywnie.

Brak potwierdzenia hipotez H3 i H4 wskazuje, że pomimo dotychczasowych osiągnięć serwisy internetowe polskich banków powinny być ulepszane.

Literatura

- [1] E. Ślęzak, E. Guzek, *Innowacyjna Bankowość Internetowa: Bank Web 2.0*, Warszawa, Wolters Kluwer Polska, 2012

- [2] NetB@nk, Raport: Bankowość internetowa i płatności bezgotówkowe, IV kwartał 2015, Związek Banków Polskich, http://zbp.pl/public/repozytorium/wydarzenia/images/marzec_2016/konferencja/Netbank_Q4_finalna_na_konferencje_20160322.pdf [01.04.2016]
- [3] D. Bednarska-Olejniczak, *Zachowania młodych konsumentów na rynku usług bankowych*. Zeszyty Naukowe Wyższej Szkoły Bankowej We Wrocławiu, 2012
- [4] S. Barnes, R. Vidgen, Assessing the quality of auction Web sites, *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, p. 10, 2001
- [5] S. Barnes, R. Vidgen, An Evaluation of Cyber-Bookshops: The WebQual Method, *International Journal of Electronic Commerce*, vol. 6, no. 1, p. 11, 2001
- [6] H. W. Webb, L. A. Webb, SiteQual: an integrated measure of Web site, *The Journal of Enterprise Information Management*, Vol. 17, No. 6, 2004
- [7] W. Chmielarz, M. Zborowski, Analiza usług bankowości elektronicznej dla klientów indywidualnych w Polsce metodą punktową i konwersji. In D. Jelonek & T. Turek, eds. *Wiedza i technologie informacyjne: nowe trendy badań i aplikacji: monografia*, Sekcja Wydawnictw Wydziału Zarządzania Politechniki Częstochowskiej, pp. 145–157, Częstochowa, 2013
- [8] W. Chmielarz, Comparative analysis of electronic banking services in selected banks in Poland in 2013, *Research Papers of the Wrocław University of Economics*, no. 316, pp. 16-29, 2013
- [9] B. Prus, M. Szylar, Evaluation of possibility to use typological procedure and Wrocław taxonomy to analyse agriculture development conditions on the example of the former Kraków voivodeship communes, *Geomatics, Landmanagement & Landscape*, no. 4, pp. 83-101, 2015
- [10] A. Binderman, Wielokryterialne metody analizy zróżnicowania polskiego rolnictwa w 2009 roku, *Metody Ilościowe w Badaniach Ekonomicznych / Szkoła Główna Gospodarstwa Wiejskiego*, no. 2, 2011
- [11] N. Prat, I. Comyn-Wattiau, J. Akoka, A Taxonomy of Evaluation Methods for Information Systems Artifacts, *Journal of Management Information Systems*, vol. 32, no. 3, pp. 229-267, 2015
- [12] S. Kauf, Wykorzystanie metody taksonomii wrocławskiej w segmentacji rynku, *Studia Ekonomiczne / Uniwersytet Ekonomiczny w Katowicach*, no. 195, 2014
- [13] M. Ehrgott, *Multicriterial optimization*, 2nd edition, Springer, Berlin, 2005
- [14] Optymalizacja Wielokryterialna, A.M.J. Skulimowski, <http://brasil.cel.agh.edu.pl/~13sustrojny/on/index.html> [01.06.2016]
- [15] T. Lachowicz, Optymalizacja wielokryterialna decyzji w zagadnieniach bezpieczeństwa funkcjonowania podmiotu, *Studia Ekonomiczne / Uniwersytet Ekonomiczny w Katowicach*, no. 235, p. 144, 2015
- [16] G. Elaniuk, Rok silnych wrażeń, *Miesięcznik Finansowy Bank*, nr 6(278), 2016
- [17] W. Chmielarz, Comparative analysis of electronic banking services in selected banks in Poland after the 2009-2010 crisis, *Studies & Proceedings Polish Association for Knowledge Management*, no. 58, pp. 20-32, 2012

Analiza wydajności aplikacji webowej we frameworku AngularJS

Yurii Vavrin*, Edyta Łukasik

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 44A, 20-618 Lublin, Polska

Streszczenie. Celem artykułu było tworzenie i analiza aplikacji webowej na bazie frameworku AngularJS. Dla aplikacji zostało napisane API po stronie serwera z połączeniem do bazy danych. Była przeprowadzona analiza teoretyczna związana z funkcjonowaniem badanego frameworku. Przeanalizowane wady i zalety podejścia frameworku do tworzenia aplikacji webowych. Były przeprowadzone badania wydajności oraz linii kodu utworzonej aplikacji.

Słowa kluczowe: AngularJS; JavaScript; aplikacja webowa; framework.

*Autor do korespondencji.

Adres E-mail: yuravavrin@gmail.com@industi.com

Analysis of web-application with AngularJS framework

Yurii Vavrin*, Edyta Łukasik

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 44A, 20-618 Lublin, Poland

Abstract. The purpose of the paper was to create and analyze web-application with AngularJS framework. Server-side API with access to database was written to support the application. Theoretical research on AngularJS functionalities has been done. Analyzed framework approach for creating applications advantages and disadvantages. Performance testing and measuring code of created application were conducted.

Keywords: AngularJS; JavaScript; web application; framework.

*Corresponding author.

E-mail address: yuravavrin@gmail.com

1. Wstęp

Technologie informatyczne mają coraz większe znaczenie w życiu każdego człowieka. Szybki rozwój techniki potrzebuje szybkiego rozwoju oprogramowania, dlatego metody i języki programowania wciąż są ulepszone lub powstają nowe. Jednym z najpopularniejszych kierunków w programowaniu jest programowanie internetowe. Szeroki dostęp do sieci internetowej spowodował jeszcze szybszy rozwój technologii dedykowanych tym rozwiązaniom. Pojawiły się różne biblioteki, platformy czy frameworki, które nie tylko ułatwiają życie programistom, ale przede wszystkim ulepszają jakość tworzonych stron m.in. szybkość ich działania. Ich znajomość jest niezbędna dla każdego kto, zamierza tworzyć systemy i aplikacje internetowe.

Framework jest to gotowy szkielet służący do budowy aplikacji. Zawiera on definicje mechanizmu działania aplikacji oraz jej strukturę [1]. Dostarcza także zestawu narzędzi tj. komponentów i bibliotek do wykonywania określonych zadań. Korzystając z frameworka programista tworzy aplikację poprzez rozbudowę i dostosowanie poszczególnych komponentów pod wymagania projektu. Dzięki temu możliwe jest oszczędzanie czasu i pieniędzy na pisanie wielu niezbędnych funkcji od zera. Głównym językiem używanym do tworzenia stron WWW i aplikacji internetowych jest skryptowy język programowania JavaScript. W tej technologii powstało kilka frameworków, np. AngularJS, EmberJS, ReactJS, BackboneJS ta inni.

W niniejszym artykule przedstawiono jeden z najpopularniejszych frameworków JavaScript - AngularJS. Wskazano jego zalety i wady. W tej technologii stworzona została prosta aplikacja pozwalająca na wymianę informacji pomiędzy znajomymi. Opisano stworzony interfejs oraz funkcjonalności aplikacji. Przeprowadzona została analiza możliwości i uncjalności, która oferuje programiście AngularJS. Przeprowadzono także badania dotyczące wydajności stworzonej aplikacji i frameworka.

2. Zalety i wady AngularJS

AngularJS to chyba najszybciej rozwijający się framework JavaScript w ciągu ostatnich pięciu lat. Powstał on w 2009 roku w Google, a jego autorami są Misko Hevery oraz Adam Abrons. Obecnie Abrons nie pracuje już w Google, a projektem razem z Hevery zajmują się Igor Minar i Vojt Jin [2]. Framework posiada pierwsze miejsce wśród frameworków od 2014 roku. Na rysunku 1 przedstawiono porównanie popularności AngularJS (niebieski) i BackboneJS (czerwony).

Wsparcie dla programistów używających tego frameworka jest dobre, gdyż społeczność go używająca jest duża a ilość informacji na jego temat wystarczająca. Jak przystało na dojrzały framework, którego patronem jest Google, posiada on też całkiem dobrą i aktualną dokumentację [4]. Jeśli chodzi o rozmiar projektu, to nie jest tak mały, jak ponad połowa innych frameworków, ponieważ całość źródeł ściągniętych na dysk zajmuje 9,6MB (wersja 1.5.8) [5]. W zamian za to

AngularJS zawiera wbudowane wszystkie niezbędne składniki. Teoretycznie można by się nawet obejść bez jQuery, ponieważ AngularJS zawiera jego zmniejszoną wersję jQuery Lite.



Rys. 1. Popularność AngularJS i BackboneJS [3]

AngularJS implementuje wzorzec Model-View-Controller (MVC) po stronie klienta. Zawiera on w pełni wybudowany mechanizm bindowania danych. Używanie tego mechanizmu jest bardzo proste – wystarczy przypisać do obiektu *\$scope* właściwość lub funkcję. Od tego momentu staje się ona (ta właściwość lub funkcja) obiektem obserwowanym. Jeśli jest podpięta pod widok, jest on automatycznie powiadamiany o jej zmianie w kontrolerze i na odwrót – jeśli zmiana nastąpi w kontrolerze, widok natychmiast o tym wie („two-way data binding”) [6].

W AngularJS szablon widoków nie są realizowane za pomocą zewnętrznej biblioteki, tylko są wbudowane we framework. Zastosowano tutaj połączenie szablonów tekstowych i deklaratywnych, tzn. używane są zarówno „placeholder” jak i adnotacje za pomocą atrybutów HTML – wszystko w zależności od potrzeb.

W sposobie opartym na atrybutach HTML, mogą wystąpić problemy wydajnościowe, ponieważ kosztem wygody programisty, konieczne jest stałe obserwowanie wszystkich elementów DOM podpiętych pod zdarzenia [6]. Z drugiej strony, szablony tekstowe są mniej prawidłowe semantycznie ponieważ, aby podpiąć zdarzenie do elementu DOM konieczne jest dodanie placeholdera w miejsce atrybutu.

AngularJS proponuje dwie możliwości składowania danych. Standardowo, framework dostarcza serwis *\$http*, który pozwala na wywoływanie żądań asynchronicznych do serwera (autorzy zakładają, że używając Angulara nie potrzebne jest jQuery).

Drugim sposobem jest podejście REST. Do tego celu służy serwis *\$resource*, który nie znajduje się w głównej części frameworka. Serwis daje możliwość po jednym konfigurowaniu (konfiguruje się specjalnie pod potrzeby) wysłać skomplikowane żądania do serwera.

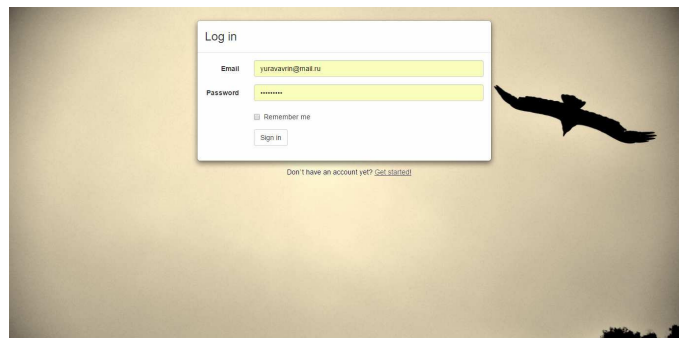
3. Funkcjonalność aplikacji

Utworzona aplikacja oparta o AngularJS posiada następujące funkcjonalności:

- logowanie użytkownika;

- rejestracja użytkownika;
- walidacja wprowadzonych danych;
- zmiana informacji o użytkowniku (takich jak imię, nazwisko, data urodzenia, zmiana zdjęcia itp.);
- dodawania powiadomienia;
- przegląd i lajkowanie swoich powiadomień oraz powiadomień swoich przyjaciół;
- wyszukiwanie i dodawanie przyjaciół;
- usuwanie przyjaciół.

Po wpisaniu adresu strony w przeglądarce, pierwsze co widzi użytkownik jest to strona logowania. Pokazano ją na rys. 2. Logowanie i rejestracja zostały zrealizowane jako animowane okienka modalne.

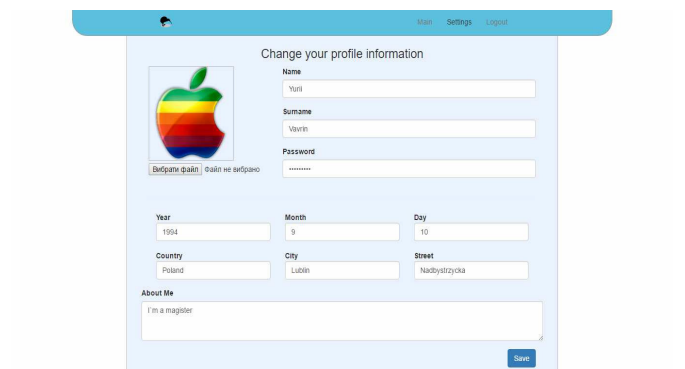


Rys. 2. Logowanie do systemu

W celu przejścia do strony rejestracji wystarczy kliknąć na przycisk *Get Started*, pod modalnym oknem. Po kliknięciu modalne okno zamyka się i pojawia się okno rejestracji. W oknie tym analogicznie, jak w przypadku modalnego okna logowania, po kliknięciu przycisku *Login*, pod modalnym oknem, ponownie otwiera się okno logowania.

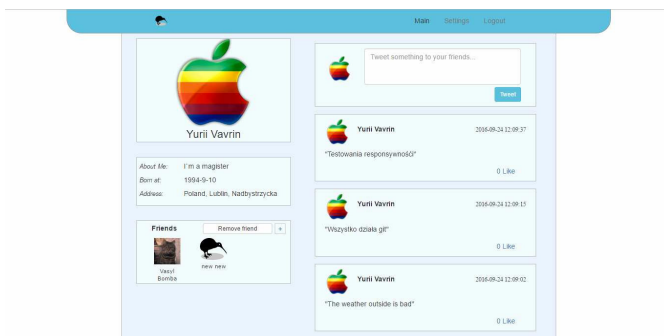
Główna część aplikacji została podzielona na *Główną stronę* i na *Ustawienia użytkownika*. Przełączyć się między nimi można za pomocą górnego menu, które też zawiera funkcję wylogowanie z aplikacji.

W ustawieniach użytkownik ma możliwość zmiany informacji o sobie, takich jak: adres, imię i nazwisko, czy hasło logowania. Strona z ustawieniami użytkownika pokazana została na rys. 3.



Rys. 3. Ustawienia użytkownika w aplikacji

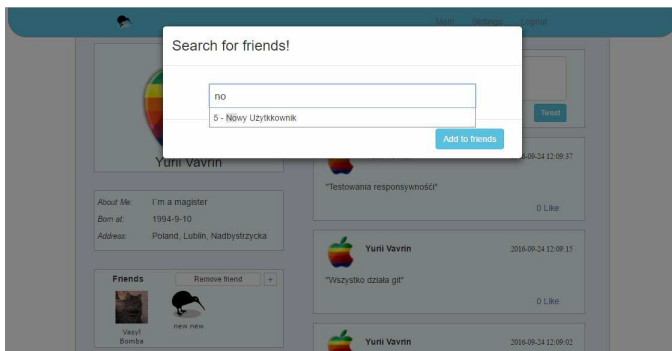
Wygląd głównej strony aplikacji przedstawiono na rys. 4. Na tej stronie w jej lewej części użytkownik widzi swoje zdjęcie, imię i nazwisko, a także swoje dane, natomiast w prawej części widzi informacje od przyjaciół.



Rys. 4. Główna strona aplikacji

Po prawej stronie znajduje się lista powiadomień, działająca na podobnej zasadzie jak w serwisie Twitter. W liście powiadomień są powiadomienia własne oraz powiadomienia przyjaciół. Istnieje możliwość dodania nowego powiadomienia. Powiadomienia są sortowane według daty tworzenia, od najnowszych do najstarszych. Oprócz tego, użytkownik ma możliwość lajkować powiadomienia.

Oprócz tego, na stronie jest lista przyjaciół, z przyciskami „Dodaj nowego” i „Usuń przyjaciela”. Po kliknięciu dodawania nowego przyjaciela, otwiera się okienko modalne z polem dla wyszukiwania. Przedstawiono je na rys. 5.

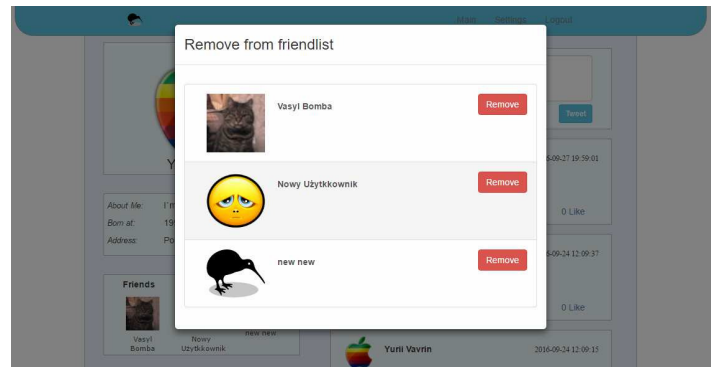


Rys. 5. Okno wyszukiwania przyjaciół

Po wpisywaniu kolejnej litery, aplikacja wysyła żądania do serwera, który szuka w bazie danych użytkowników z podobnym imieniem lub nazwiskiem.

Zgodnie z tym jak powinna działać aplikacja jedno-stronicowa, lista przyjaciół a także lista powiadomień odświeżają się automatycznie bez odświeżenia całej strony.

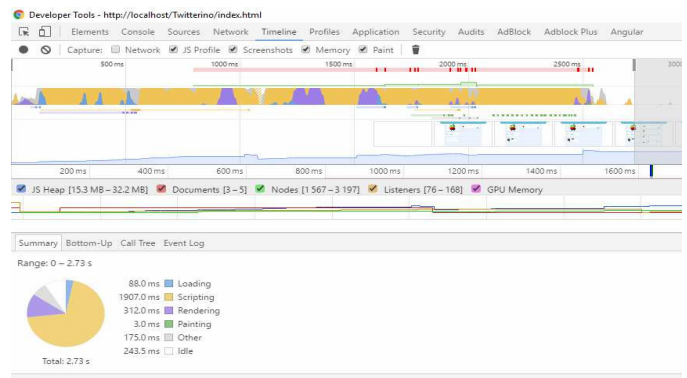
Przycisk usuwania przyjaciół otwiera okno modalne z listą przyjaciół. Przy każdym jest przycisk służący do jego usunięcia. Przykład takiej listy pokazano na rys. 6. W celu usunięcia użytkownika z listy przyjaciół, wystarczy kliknąć „Usuń” obok wskazanego przyjaciela. Spowoduje to odświeżenie listy użytkowników i powiadomień, bez odświeżenia całej strony.



Rys. 6. Okno usuwania przyjaciół

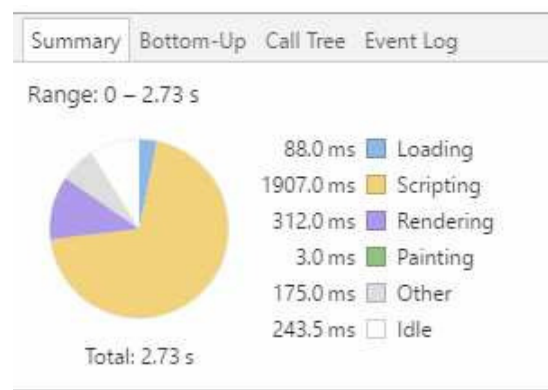
4. Badania wydajności

Chyba najważniejszym parametrem jakiegokolwiek programu komputerowego jest jego wydajność i szybkość działania. Dla badania wydajności aplikacji webowej najlepiej skorzystać z funkcjonalności samej przeglądarki, ponieważ przeglądarka jest kompilatorem kodu aplikacji webowej i ma dostęp bezpośrednio do charakterystyk wydajności. Przeglądarka Google Chrome posiada bardzo szczegółowe standardowe urządzenie służące do oceny wydajności strony – Timeline.



Rys. 7. Wyniki wydajności aplikacji

Na rysunku 7. można zobaczyć czas ładowania strony i wszystko co się działo w trakcie. Najbardziej interesujący jest punkt *Podsumowanie*. Wyniki dla testowanej aplikacji pokazano na rys. 8.



Rys. 8. Szczegóły wydajności aplikacji

Dzięki prostej liście wyników, możliwe jest przeanalizowanie szybkości ładowania aplikacji. Na wykresie przedstawionym na rys. 8 podany jest czas ładowania strony, działania skryptu, renderowania ta innych procesów.

Ładowania plików w tym frameworku jest dość szybkie. Dzieje się tak dlatego, że posiada on wiele wbudowanych funkcjonalności i rozbudowaną strukturę, a to powoduje, iż programista oszczędza czas na pisanie kodu. Ponieważ napisano mało linii kodu, w rezultacie pliki są małe i szybko się ładują.

Najwięcej czasu w ładowaniu aplikacji zajmuje skrypting. Angular jest w tym wolniejszy od większości frameworków przez „two-way binding”, który ustawia „nasłuchiwanie” każdej zmiennej obiektu *\$scope*. Kolejną przyczyną takiego spowolnienia jest obsługa widoków w sposób atrybutowy, co powoduje traktowanie kodu HTML jako część skryptu przez przeglądarkę [5].

Razem, czas ładowania zajmuje 2730 ms. Nie jest to czas krytycznie duży dla małej aplikacji webowej, ale w wypadku dużego projektu czas ładowania aplikacji w AngularJS może znacznie wzrosnąć, dlatego warto zwrócić uwagę na różne metody optymalizacji działania strony.

5. Linie kodu

Linia kodu to podstawowa metryka rozmiaru programu komputerowego. Może ona też określać, ile czasu lub ile osób potrzeba, by ten program napisać.

Często używana jest jednostka pochodna KLOC (kiloes of lines of code) – tysiące linii kodu (zwykle mierzony jest kod źródłowy). Nie ma ustalonego standardu mierzenia linii kodu, dlatego wartość LOC może być liczona z pustymi liniami lub bez nich, z komentarzami lub bez nich. Wartość jest też zależna od użytego języka programowania [7].

Ilość linii kodu z pustymi liniami, policzona dla stworzonej aplikacji została przedstawiona w tabeli 1. Oddzielnie policzono linii kodu w plikach HTML i JavaScript. Liczba tych drugich jest o 25% więcej.

Tabela 1. Liczba linii kodu z pustymi liniami w aplikacji AngularJS

Typ	Resultat
Pliki HTML	305
Pliki JS	420
Razem	725

Ze względu na to, że AngularJS działa też bezpośrednio z kodem HTML, liczba linii kodu w plikach HTML zwykle jest nieco większa w porównaniu z innymi frameworkami. Z innej strony, dzięki lepszej strukturze i większej liczbie wbudowanych funkcjonalności, w plikach *.js Angular przeważnie znacząco wyprzedza wszystkich konkurentów.

W wyniku, dla tworzenia aplikacji w AngularJS potrzeba znacznie mniej pisania kodu niż w większości innych

frameworków, chociaż czasami osiąga się to kosztem wydajności.

6. Wnioski

Zaimplementowano działającą aplikację webową z zastosowaniem frameworku JavaScript - AngularJS oraz API obsługujące aplikacje.

Funkcjonalności aplikacji pozwalają na takie działania jak: logowanie, rejestrację, zmianę danych o użytkowniku takich jak imię, nazwisko, data urodzenia, zdjęcie. Główna strona aplikacji pozwala na wyszukiwanie, dodawanie oraz usunięcie przyjaciół użytkownika, a także na przegląd, dodawanie i lajkowanie powiadomień stworzonych przez użytkownika lub jego przyjaciół.

Aplikacja jest „jedno-stronicowa” (single-page), tzn. że wszystkie zmiany wprowadzone przez użytkownika, aktualizują się bez odświeżenia całej strony. Wszystkie dane są zapisywane do bazy danych MySQL.

W realizacji aplikacji zastosowano języki HTML, CSS, JavaScript i PHP. Użyto technologii AJAX i Bootstrap, natomiast w celu komunikacji z bazą danych MySQL.

Uwypuklone zostały największe zalety i główne wady omawianego frameworka. Do zalet należy zaliczyć: możliwość budowania aplikacji w oparciu o architekturę MVC, wstrzykiwanie zależności ułatwiające współpracę komponentów, obustronne bindowanie oraz wbudowaną komunikację REST.

Przeprowadzone zostały badania przedstawiają statystyki dotyczące wydajności stworzonej aplikacji oraz metryki linii kodu. Po porównywaniu tych wyników z aplikacjami tworzonymi w innych frameworkach należy podjąć decyzję o jego wyborze lub nie do tworzonego systemu.

Literatura

- [1] Hiller C.: Choosing a JavaScript framework. Bleeding Edge Press, 2014.
- [2] AngularJS - <https://en.wikipedia.org/wiki/AngularJS> [15.08.2016].
- [3] <https://www.google.com/trends/> [27.08.2016].
- [4] <https://angularjs.org/> [24.07.2016]
- [5] Smith. I.: AngularJS in 60 minutes. Wahling Consulting, 2013.
- [6] Козловский П.: Разработка веб-приложений с использованием AngularJS. 2014
- [7] https://en.wikipedia.org/wiki/Source_lines_of_code [9.09.2016]

Metodyka tworzenia oprogramowania a jakość kodu – analiza porównawcza dwóch przypadków

Bartłomiej Zalewski*, Marek Miłosz

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

Streszczenie. Jakość kodu zależy od stosowania dobrych praktyk jego wytwarzania. W artykule przedstawiono metryki pomiaru jakości kodu tworzonego zgodnie z paradygmatem obiektowym i narzędzia informatyczne wyznaczające ich wartości. Rozpatrzono dwa przypadki rozwoju oprogramowania przez różne zespoły w różny sposób zarządzane. Przeanalizowano jakość kodu w kolejnych pięciu wersjach programów. Rezultaty badań pozwoliły na sformułowanie wniosku o przewadze metodyki lekkiej, na co wskazują lepsze wartości większości metryk.

Słowa kluczowe: jakość kodu; metryki pomiaru; metodyki wytwarzania oprogramowania

*Autor do korespondencji.

Adres E-mail: b.r.zalewski@gmail.com

Methodology of software development vs. code quality – a comparative analysis of two cases

Bartłomiej Zalewski*, Marek Miłosz

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

Abstract. Code quality is strongly dependent on using best coding practices during its development. This paper presents various code quality metrics in object oriented programming and computer tools to its automatic measurement. Two cases of software development by two different teams were considered. Code quality was analyzed in five following program versions. This study shows better value of almost (but not all) code quality metrics developed using agile methodology. It raises the conclusion about agile methodology advantage.

Keywords: code quality; measurement metrics; methodologies of software development

*Corresponding author.

E-mail address: b.r.zalewski@gmail.com

1. Wstęp

Aby wytwarzać dobre jakościowo oprogramowanie należy nieustannie poddawać je testom. Najczęściej programy testuje się pod względem spełniania funkcjonalności. Często zapomina się lub jawnie lekceważy testowanie jakości kodu, który realizuje poszczególne funkcjonalności.

Statyczna analiza kodu umożliwia sprawdzenie jakości kodu użytego w aplikacji. Dzięki analizie statycznej można na bieżąco sprawdzać, czy tworzony kod jest zgodny z regułami projektowania systemów informatycznych, które przekładają się na cechy oprogramowania takie jak niezawodność, złożoność zarządzania czy utrzymanie. Obliczanie wartości metryk mających za zadanie sprawdzić jakość napisanego kodu nie musi wiązać się z koniecznością poświęcania na to dodatkowego czasu, ponieważ istnieją narzędzia lub dodatki do IDE (ang. *Integrated Development Enviroment*), które robią to automatycznie na wybranym projekcie [1]. Jakość wytworzonego oprogramowania wynikać może również z tego, w jaki sposób realizowany był projekt, którego wynikiem było to oprogramowanie. Obecnie znaleźć można kilka podejść do realizacji projektu informatycznego. Metodami polecanymi, powszechnie uważanymi za skuteczne, są metodyki zwinne (ang. *Agile*), np. SCRUM, Extreme Programming, TDD (ang. *Test-Driven Development*). Są to metodyki lekkie i elastyczne, zorientowane na dostarczenie produktu. Jednocześnie wiele projektów realizowanych jest bez oparcia o sprawdzone

metodyki. Spowodowane jest to często niewiedzą lub chęcią zaoszczędzenia czasu oraz pieniędzy, co niekoniecznie przynosi zamierzony cel [2]. Stosowanie metodyk realizacji projektów często wiąże się z pójściem na kompromis. Dla przykładu metodyka SCRUM wymaga planowania zadań na tydzień lub więcej w przód, co w niektórych zespołach może wydawać się nierealne, gdyż właściciel produktu wymaga od nich podejścia ad-hoc. Panuje opinia, że metodyki zwinnego programowania stosuje się tylko w dużych firmach. W rzeczywistości jednak takie praktyki stosowane są również w małych przedsiębiorstwach [3]. Metodyki zwinne wprowadzają również większą dyscyplinę przy realizowaniu projektu informatycznego [2].

2. Metryki pomiaru jakości kodu

W trakcie rozwoju programowania zorientowanego obiektowo zaczęto tworzyć zasady, dzięki którym wytwarzany kod miał posiadać lepszą jakość. Jakość ta miała przekładać się na łatwiejsze zarządzanie, testowanie kodu, zmniejszenie jego złożoności oraz pozwalać na ponowne wykorzystanie już istniejących modułów bez konieczności ich powielania. Różni badacze dziedziny inżynierii oprogramowania analizując dobre praktyki i wzorce wytwarzania oprogramowania zorientowanego obiektowo przedstawiali własne miary mające ocenić, jak bardzo wytworzony kod spełnia daną zasadę. W tym artykule przedstawione zostaną następujące metryki: złożoność cyklomatyczna McCabe'a, miary Halsteada,

metryki z zestawu MOOD, metryki Roberta C. Martina oraz zestaw metryk CK.

Złożoność cyklomatyczna McCabe'a (ang. *Cyclomatic Complexity* – CC) jest metryką, która została zaprojektowana z myślą o strukturalnych językach programowania, ale aktualnie znajduje swoje zastosowanie w określeniu miejsc trudnych do testowania oraz utrzymania w obiektowych językach programowania [4]. Zasada wyliczania tej miary polega na określeniu liczby ścieżek wykonania danej części oprogramowania [4].

Miary Halsteada dotyczą funkcji i argumentów danego modułu oprogramowania [5]. Halstead przedstawił pięć następujących miar [5]: długość programu (ang. *Length of a Program*), słownik programu (ang. *Vocabulary*), objętość (ang. *Volume*), trudność (ang. *Difficulty*) oraz pracochłonność (ang. *Effort*). Na bazie miar przedstawionych przez Halsteada opracowano metryki bardziej złożone, m.in. [6] poziom programu (ang. *Program Level*), czas implementacji (ang. *Time*) a także szacunkowa liczba błędów (ang. *Number of Delivered Bugs*).

Zestaw metryk MOOD (ang. *Metrics for Object-Oriented Design*) to zbiór składający się z sześciu metryk, takich jak: współczynnik polimorficzności (ang. *Polymorphism Factor - PF*), współczynnik ukrycia atrybutów (ang. *Attribute Hiding Factor - AHF*), współczynnik ukrycia metod (ang. *Method Hiding Factor - MHF*), współczynnik dziedziczenia atrybutów (ang. *Attribute Inheritance Factor - AIF*), współczynnik dziedziczenia metod (ang. *Method Inheritance Factor - MIF*) oraz współczynnik powiązań (ang. *Coupling Factor - CF*). Istotnym, z perspektywy zarządzania projektem, jest fakt, że zastosowanie metryk MOOD skutkuje możliwością szybkiej i ogólnej oceny jakości oprogramowania [7]. Metryki MOOD pozwalają na pomiar elementów programu takich jak: hermetyzacja, polimorfizm, dziedziczenie i powiązania pomiędzy klasami. Sześć istniejących metryk odpowiada za pomiar poszczególnych składników paradygmatów programowania zorientowanego obiektowo [8].

Metryki Roberta C. Martina to zbiór składający się z pięciu metryk. Są to metryki: powiązania do wewnątrz (ang. *Affarent Coupling - Ca*), powiązania na zewnątrz (ang. *Efferent Copling - Ce*), abstrakcja (ang. *Abstractness - A*), niestabilność (ang. *Instability - I*) oraz znormalizowana odległość od ciągu głównego (ang. *Normalized Distance from Main Sequence - Dn*). Metryki te dotyczą zależności pomiędzy klasami w projekcie. Dwie najważniejsze metryki opierają się na rozróżnieniu powiązań na powiązania do wewnątrz i powiązania na zewnątrz [9].

Metryki Chidamera i Kemerera nazywane są powszechnie zestawem metryk CK [10]. Jest to obecnie jeden z najpopularniejszych zbiorów metryk obiektowych pozwalający na pomiar złożoności klas [11]. Zawiera on sześć metryk odnoszących się do różnych aspektów obiektowości: dziedziczenia, złożoności klasy, powiązania pomiędzy klasami i spójności. Metryki te pozwalają na niskopoziomowe szacowanie jakości kodu w obiektowych językach

programowania. W skład zbioru metryk CK wchodzi [10]: ważona liczba metod w klasie (ang. *Weighted Methods per Class - WMC*), odpowiedź klasy (ang. *Response for a Class - RFC*), głębokość drzewa dziedziczenia (ang. *Depth of Inheritance Tree - DIT*), liczba pośrednich potomków klasy (ang. *Number of Childre of Class - NOC*), powiązania między klasami (ang. *Coupling Between Objects - CBO*) oraz brak spójności metod (ang. *Lack of Cohesion in Methods - LCOM*).

W tabeli 1 przedstawiono optymalne wartości metryk, które zostały użyte do porównania jakości dwóch projektów rozpatrywanych w tej pracy. W tabeli zawarto tylko te metryki, które posiadają określoną wartość optymalną.

Tabela 1. Wartości optymalne metryk użyte do porównania projektów informatycznych. Źródło: opracowanie własne na podstawie [12].

Nazwa metryki	Optymalna wartość
Ca	0-500
Ce	0-20
I	0,7 - 1 pakiety niestabilne 0 - 0,3 pakiety stabilne
Dn	0,2
WMC	20
RFC	20-100
DIT	0-6
NOC	2-5
CBO	0-7
LCOM2	0-1

3. Narzędzia informatyczne do analizy jakości kodu

W konsekwencji rozwoju badań nad jakością kodu powstały narzędzia umożliwiające zautomatyzowany pomiar wartości z wykorzystaniem poszczególnych metryk jakości kodu. Obecnie istnieje wiele programów umożliwiających pomiar kodu stworzonego w różnych językach programowania. Są to programy niezależne lub zintegrowane ze środowiskami rozwoju oprogramowania. Istnieje kilka istotnych programów umożliwiających badanie jakości kodu napisanego w języku C#. Są to narzędzia analityczne w Visual Studio, SonarQube, NDepend, Source Monitor.

Visual Studio od roku 2007 pozwala na przanalizowanie projektu pod kątem wartości pięciu metryk [13]. Narzędzie to nosi nazwę Visual Studio Code Metrics Powertool for Visual Studio 2015. Umożliwia ono pomiar metryk [14]: *Maintainability Index* (metryka indeksu trudności utrzymania kodu), *Cyclomatic Complexity* (metryka złożoności cyklomatycznej), *Depth of Inheritance* (metryka głębokości drzewa dziedziczenia z zestawu metryk CK), *Class Coupling* (metryką powiązań pomiędzy klasami z zestawu metryk CK) oraz *Lines of Code* (liczba linii kodu).

Oprogramowanie SonarQube, zwane uprzednio Sonar [15], jest platformą open source. SonarQube wylicza wartości podstawowych metryk, np. złożoność cyklomatyczna, liczba linii kodu. Wyniki każdej z sekcji mierzonej przez SonarQube mogą być prezentowane w formie graficznych raportów [15].

Source Monitor jest darmowym programem dostarczającym podstawowych informacji na temat kodu. Source Monitor mierzy: liczbę linii kodu, łączną liczbę

instrukcji, procent komentarzy w kodzie, procent komentarzy będących dokumentacją, liczbę klas, średnią liczbę metod w klasie, średnią liczbę instrukcji w metodzie, maksymalną oraz średnią złożoność klasy, maksymalną oraz średnią wielkość drzewa dziedziczenia. Dodatkowo możliwe jest wyświetlenie grafu kiviata obrazującego, czy wartości metryk mieszczą się w rekomendowanych wartościach [16].

NDepend jest najpopularniejszym narzędziem do badania jakości kodu na platformie .NET [17]. W przeciwieństwie do pozostałych programów jest narzędziem płatnym. Posiada jednak dwutygodniową wersję próbną dającą te same możliwości, co płatna wersja programu. NDepend umożliwia zautomatyzowanie procesu analizy kodu poprzez zintegrowanie go z programami do ciągłej integracji (ang. *Continuous Integration*), np. TeamCity lub z mechanizmem MSBuild. Umożliwia to generowanie szczegółowego raportu przy każdej kompilacji programu bez konieczności ręcznego uruchamiania analizy. NDepend pozwala również na integrację z Visual Studio. Posiada również opcję raportowania otrzymanych wyników do przejrzystych stron html [17].

4. Plan i rezultaty badań

Celem pracy badawczej było sprawdzenie postawionej następującej tezy:

Stosowanie dobrych praktyk wytwarzania oprogramowania zorientowanego obiektowo podnosi wartość metryk sprawdzających jakość kodu.

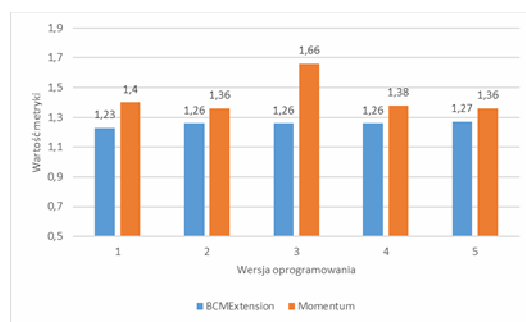
Eksperyment polegał na porównaniu dwóch programów komputerowych pod kątem wartości metryk mierzących jakość napisanego kodu. Oba programy były wytwarzane w tych samych technologiach i przy użyciu jednakowych narzędzi programistycznych. Programy różniły się stosowanymi metodykami wytwarzania oprogramowania, a mianowicie projekt BCMExtension rozwijany jest przez zespół programistów, których praca sterowana jest metodologią SCRUM, a projekt Momentum rozwijany jest bez stosowania żadnej metodyki. Do obliczeń wartości metryk jakości napisanego kodu został wybrany program NDepend. Do obróbki statystycznej otrzymanych wartości poszczególnych metryk jakości oprogramowania każdego z projektów użyty został program Microsoft Excel 2016. Wybranych zostało pięć wersji każdego z programów. Przy wyborze wersji programów polegano na sprintach wykonywanych przy wytwarzaniu oprogramowania BCMExtension. Wybrano wersję programu ze sprintów numer 1, 10, 20, 30 i 40. Odstępy czasowe pomiędzy poszczególnymi wersjami wyniosły od 2,5 do 3 miesięcy co pozwoliło na zauważenie różnic pomiędzy poszczególnymi wersjami programów. Następnie wybrano wersje programu Momentum adekwatną do dat ukończenia poszczególnych sprintów programu BCMExtension. Do określenia jakości kodu badanych programów posłużyła metryka: złożoność cyklomatyczna McCabe'a – do określenia złożoności metod zawartych w programach. By zobrazować CC każdego z programów została wyliczona średnia tej metryki oraz dodatkowo wartość maksymalna. By określić powiązania pomiędzy poszczególnymi klasami w projekcie wyliczone

zostały wartości metryk z zestawu CK. Metryki z zestawu CK odnoszą się do konkretnych klas programu, więc w eksperymencie została obliczona wartość średnia oraz wartość maksymalna każdej z metryk tego zestawu. Dodatkowo obliczona została liczba klas, które przekraczają przyjęte optymalne wartości każdej z metryk. Do określenia poprawności konstrukcji pakietów porównywanych programów zostały wyliczone wartości poszczególnych metryk z zestawu Roberta C. Martina.

Szczegółowe rezultaty badań zostały przedstawione w pracy [18].

5. Analiza porównawcza jakości kodu w obu aplikacjach

Wyniki przeprowadzonego eksperymentu posłużyły do przeprowadzenia analizy porównawczej badanych projektów. Analiza opiera się na porównaniu poszczególnych wartości metryk projektu BCMExtension w odniesieniu do wartości metryk jakości projektu Momentum.

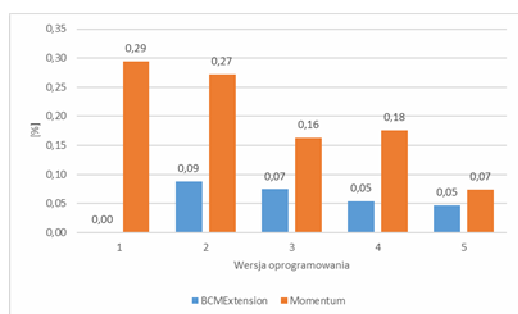


Rys. 1. Porównanie średnich wartości metryk złożoności McCabe'a dla poszczególnych wersji programu BCMExtension i Momentum.

Rys. 1 wskazuje, że średnia wartość metryki McCabe'a dla wszystkich wersji programu BCMExtension jest niższa, niż dla programu Momentum. Ostatnie wersje obu programów różnią się średnią wartością metryki CC o 0,09, na korzyść programu BCMExtension. Metryka CC wskazuje na trudność utrzymania i testowania badanego kodu. Program BCMExtension osiągnął nieznacznie niższe wartości tej metryki w porównaniu z programem Momentum. Oznacza to, że program BCMExtension powinien być łatwiejszy w utrzymaniu i testowaniu od programu Momentum.

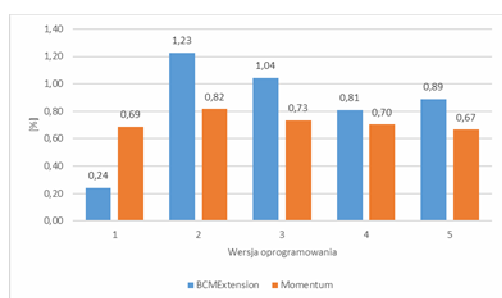
Dla zobrazowania jakości klas zawartych w programach BCMExtension i Momentum uśredniono wartość każdej z metryk zestawu CK. Na rys. 2 przedstawiono procentowy udział klas przekraczających optymalną wartość metryki LCOM2 w programach BCMExtension i Momentum. Liczba klas przekraczających optymalną wartość tej metryki została określona przez zsumowanie klas, dla których wartość metryki LCOM2 przekracza wartość 1. Udział klas przekraczających optymalną wartość metryki LCOM2 jest niewielki zarówno w programie BCMExtension oraz Momentum i nie przekracza 0,3%. Projekt BCMExtension osiąga jednak lepsze wartości metryki LCOM2 w porównaniu do projektu Momentum. Ostateczna różnica procentowa jest spowodowana większą liczbą klas występujących w wersji piątej programu BCMExtension. Niski udział procentowy

klas przekraczających optymalną wartość metryki LCOM2 świadczy o wysokiej jakości obu programów.



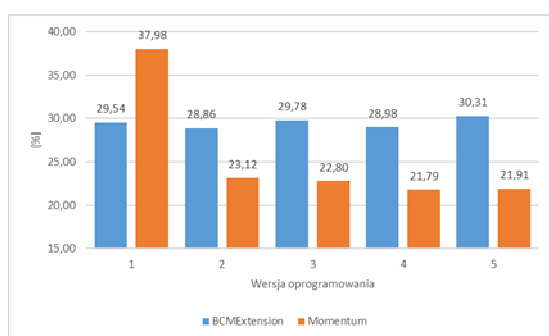
Rys. 2. Porównanie procentowego udziału typów przekraczających optymalną wartość metryki LCOM2 dla poszczególnych wersji programów BCMExtension i Momentum.

Inaczej niż w przypadku poprzedniej metryki, procentowe wartości klas przekraczających odpowiednią wartość metryki NOC są minimalnie gorsze dla projektu BCMExtension. Różnica ta nie przekracza 1%. Rys. 3 potwierdza, że oba projekty reprezentują wysoką jakość mierzoną metryką NOC.

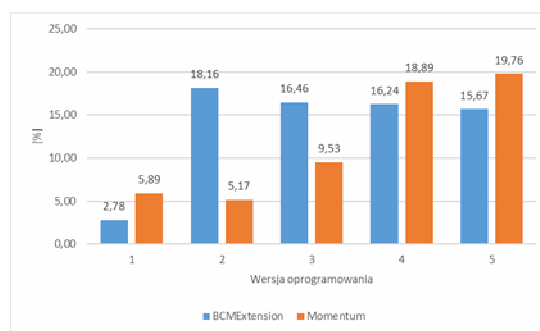


Rys. 3. Porównanie procentowego udziału typów przekraczających optymalną wartość metryki NOC dla poszczególnych wersji programów BCMExtension i Momentum.

Oba projekty reprezentują wysoki poziom klas przekraczających poprawną wartość metryki CBO (rys. 4), co powinno zaalarmować zespoły wytwarzające programy BCMExtension i Momentum i zdopingować do poprawy kodu klas przekraczających optymalne wartości tej metryki.



Rys. 4. Porównanie procentowego udziału typów przekraczających optymalną wartość metryki CBO dla poszczególnych wersji programów BCMExtension i Momentum.

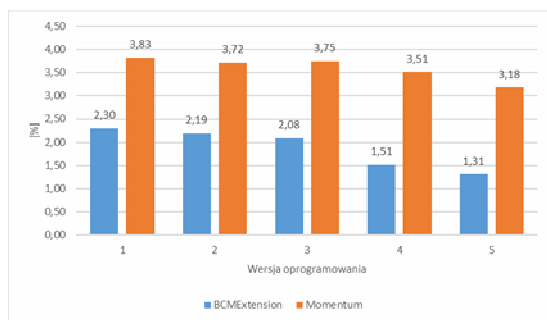


Rys. 5. Porównanie procentowego udziału typów przekraczających optymalną wartość metryki RFC dla poszczególnych wersji programów BCMExtension i Momentum.

Kolejnym analizowanym rezultatem przeprowadzonego badania jest procentowy udział klas przekraczająca poprawną wartość metryki RFC przedstawiony na rys. 5. Pierwsze trzy wersje programu Momentum posiadają poprawny odsetek klas przekraczających optymalną wartość metryki RFC, który nie przekracza 10%. Dużo gorzej wypada wersja czwarta i piąta. W porównaniu do wersji trzeciej procent klas źle zaimplementowanych wzrasta dwukrotnie w wersji czwartej, osiągając blisko 20% wynik w wersji piątej. Program BCMExtension najniższy procent (wynoszący 2,5%) klas przekraczających rekomendowaną wartość metryki RFC osiąga w pierwszej wersji programu. W kolejnej wersji następuje gwałtowny wzrost do 18% niepoprawnie napisanych klas. Kolejne wersje programu BCMExtension notują delikatny spadek odsetka klas przekraczających poprawną wartość metryki RFC, by ostatecznie osiągnąć 15% w wersji piątej.

Program BCMExtension osiągnął dużo gorsze wyniki, niż program Momentum, w wersji drugiej i trzeciej. Od wersji czwartej BCMExtension prezentuje jednak niższy odsetek niepoprawnych klas, by ostatecznie w wersji piątej programu posiadać 5% mniej klas przekraczających optymalną wartość metryki RFC. Metryka RFC wskazuje na złożoność klasy, a wartości tej metryki większe niż 100 wskazują na zbyt dużą złożoność klasy powodującą problemy z testowaniem i utrzymaniem kodu. Oba projekty posiadają wysoki procent klas źle zaimplementowanych względem metryki RFC. Zespoły wytwarzające oba projekty powinny popracować nad poprawą kodu niepoprawnych klas.

Porównanie przedstawione na rys. 6 obrazuje, że program BCMExtension posiada niższy procent klas przekraczających poprawną wartość metryki WMC, we wszystkich pięciu wersjach programu, w porównaniu z programem Momentum. Piąta wersja programu BCMExtension posiada ponad dwukrotnie niższy procent niepoprawnie zaimplementowanych klas w porównaniu do piątej wersji programu Momentum. Oba projekty cechuje niski odsetek źle napisanych klas nieprzekraczający 4%.

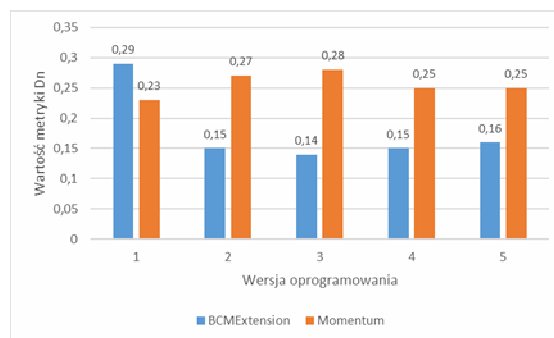


Rys. 6. Porównanie procentowego udziału typów przekraczających optymalną wartość metryki WMC dla poszczególnych wersji programów BCMExtension i Momentum.

Do wyliczenia metryki I świadczącej o niestabilności danej klasy posłużyły wartości dwóch metryk: Ca i Ce. Wartości tej metryki dla pakietów warstwy kontrolera programów BCMExtension i Momentum są równe 1, co jednoznacznie określa te pakiety jako niestabilne. Inaczej rzecz się ma z pakietami warstwy biznesowej obu programów. Ich wartości oscylują pomiędzy 0,5 a 0,64, co uniemożliwia jednoznaczne określenie ich stanu. Są to pakiety, których wartości metryki I wskazują na to, że są to pakiety niestabilne bardziej niż stabilne, wartości I dla tych pakietów nie osiągają wartości 0,7 będącej minimalną optymalną wartością dla pakietów niestabilnych. Bliżej optymalnej wartości dla metryki I są pakiety warstwy biznesowej wersji od 2 do 5 programu BCMExtension w porównaniu z pakietami programu Momentum. Nie osiągnięcie optymalnej wartości metryki I może skutkować dużym oddaleniem od ciągu głównego, czyli wartością metryki Dn.

Kolejną analizowaną metryką z zestawu metryk Roberta C. Martina jest metryka A wskazująca na stosunek klas abstrakcyjnych do wszystkich klas pakietu. Wartości tej metryki wskazują na praktyczną nieobecność klas abstrakcyjnych w pakietach warstwy kontrolera pakietów programu BCMExtension i Momentum. Wartości metryki A dla pakietów warstwy biznesowej obu programów wahają się pomiędzy 0,15 a 0,21, w ostatecznych, piątych wersjach programów osiągają wartości 0,21 dla programu BCMExtension i 0,15 dla programu Momentum.

Analiza danych z rys. 7 doprowadza do wniosku, że każda wersja programu Momentum posiada pakiet warstwy biznesowej przekraczający maksymalną poprawną wartość metryki Dn. Na tym samym wykresie widać, że pakiet warstwy biznesowej, pierwszej wersji programu BCMExtension również przekracza maksymalną poprawną wartość metryki Dn. Pakiety warstwy biznesowej kolejnych wersji programu BCMExtension mieszczą się poniżej granicy maksymalnej dopuszczalnej wartości tej metryki. Warstwa biznesowa programu Momentum powinna zostać poddana analizie oraz przeprojektowaniu, by obniżyć wartość metryki Dn.



Rys. 7. Porównanie metryki Dn warstwy dostępu do danych, dla poszczególnych wersji programów BCMExtension i Momentum.

6. Wnioski

Celem badań było przeanalizowanie jakości kodu dwóch programów tworzonych przez różne zespoły programistów, których pracę determinowała inna metodyka wytwarzania oprogramowania. Przedstawiona teoria z zakresu metryk jakości oprogramowania, a także dobrych praktyk wytwarzania oprogramowania pozwalają stwierdzić, że istnieją metryki kodu umożliwiające jednoznaczne określenie jego jakości. Celem publikacji było również przeprowadzenie analizy jakościowej dwóch programów oraz późniejsza analiza porównawcza wyników.

Rezultaty badań (tab. 2 i 3) pokazują, że choć wartość niektórych metryk jest podobna to istnieją metryki, gdzie wartości dla programów BCMExtension i Momentum różnią się w sposób znaczący na korzyść programu BCMExtension. Widać to w przypadku metryk RFC, WMC z zestawu metryk CK oraz metryki Dn. Pokazuje to, że programy tworzone z zastosowaniem różnych metod i technik mają różną jakość. Pomimo różnic dotyczących metod zarządzania projektami BCMExtension oraz Momentum nie można jednoznacznie stwierdzić, który z projektów jest jakościowo lepszy. Istnieją metryki takie jak metryki RFC i WMC, pochodzące z zestawu metryk CK, oraz metryka Dn, z zestawu metryk Roberta C. Martina dla pakietu biznesowego, w których program BCMExtension osiąga znacznie lepsze bądź trochę lepsze wyniki od programu Momentum. Z drugiej strony, wartość metryki CBO z zestawu metryk CK jest znacznie gorsza dla projektu BCMExtension. Oba projekty osiągają porównywalne wyniki, jeżeli chodzi o metrykę złożoności cyklomatycznej McCabe'a, metryki LCOM2 i NOC z zestawu metryk CK oraz metrykę Dn dla pakietów warstwy kontrolera z zestawu metryk Roberta C. Martina.

Choć program BCMExtension nie zawsze osiąga lepsze wartości metryk sprawdzających jakość kodu od programu Momentum to z przeprowadzonej analizy można wysnuć wniosek, że kod programu BCMExtension jest łatwiejszy w utrzymaniu i testowaniu niż kod programu Momentum. Wyjaśnieniem nie zawsze lepszych wyników jakości kodu BCMExtension może być nie zawsze stosowanie się do zasad dobrego wytwarzania kodu przez programistów, a co za tym idzie niedokładna weryfikacja jakości kodu przeprowadzana po wytworzeniu nowej funkcjonalności. Zły wpływ na jakość kodu BCMExtension mogą mieć również pozostałości kodu wytworzonego przed stosowaniem metodyki SCRUM.

Tabela 2. Wartości metryk użytych w badaniu dla 5. wersji programów BCMExtension i Momentum.

Metryka	BCMExtension	Momentum
LoC	55666	40005
Śr. CC	1,27	1,36
Śr. LCOM2	0,038	0,028
Śr. NOC	1,27	0,73
Śr. DIT	1,54	1,57
Śr. CBO	14,16	13,22
Śr. RFC	117,55	48,04
Śr. WMC	1,43	3,19
Ca (warstwa biz./kontro.)	212 / 0	172 / 0
Ce (warstwa biz./kontro.)	356 / 1738	263 / 1006
A (warstwa biz./kontro.)	0,21 / 0	0,15 / 0,01
I (warstwa biz./kontro.)	0,63 / 1	0,6 / 1
Dn (warstwa biz./kontro.)	0,16 / 0	0,25 / 1

Tabela 3. Procentowy udział elementów przekraczających optymalną wartość metryk zestawu Roberta C. Martina dla 5. wersji programów BCMExtension i Momentum.

Metryka	BCMExtension [%]	Momentum [%]
LCOM2	0,05	0,07
NOC	0,89	0,67
DIT	0	0
CBO	30,31	21,91
RFC	15,67	19,76
WMC	1,31	3,18

W celu potwierdzenia wpływu metodyki SCRUM na jakość kodu potrzebne są dodatkowe badania obu projektów, z uwzględnieniem przedstawionych wyników. Dobrym pomysłem byłoby również zintegrowanie narzędzi do automatycznej weryfikacji jakości kodu wytworzonej funkcjonalności z projektem BCMExtension, Przyczynić się to może do poprawy jakości kodu wytwarzanego przez zespół programistów oraz zwiększyć świadomość stosowania dobrych praktyk. Mogłoby to również zaowocować pisaniem czystego kodu w szybszym czasie. Badania na ten temat prowadzą obecnie Taibi, Janes, Lenarduzzi [19]. Wyniki metryk dla programu BCMExtension, jeżeli nawet okazywały się gorsze niż wyniki metryk dla programu Momentum, to wykazywały tendencję do poprawy. Może to oznaczać, że po przeprowadzeniu głębszej refaktoryzacji kodu, wszystkie metryki jakości kodu programu BCMExtension będą lepsze niż metryki jakości programu.

Na podstawie tej pracy można wysnuć ogólny wniosek, że:

Stosowanie dobrych praktyk wytwarzania oprogramowania zorientowanego obiektowo podnosi wartość metryk sprawdzających jakość kodu.

Literatura

- [1] Čeponis, J., Venčkauskas, A., Čėponienė, L., Zonys, A.: Extending Rule Set for Static Code Analysis in .NET. Platform. Information Technology And Control, 45, 2016, 99-108.

- [2] Iivari J.: The relationship between organizational culture and the deployment of agile methods. Information and Software Technology, 53, 2011, 509-520.
- [3] Holmström H., Alahyari H., Bosh J.: Climbing the "Stairway to Heaven" -- A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on, 2012, 392-399.
- [4] Vinju J.J., Godfrey M.W.: What Does Control Flow Really Look Like? Eyeballing the Cyclomatic Complexity Metric. Source Code Analysis and Manipulation (SCAM), 2012 IEEE 12th International Working Conference on, 2012, 154 - 163.
- [5] Posnett D., Hindle A., Devanbu P.: A simpler model of software readability. MSR '11 Proceedings of the 8th Working Conference on Mining Software Repositories, 2011, 73-82.
- [6] Yousef A.H.: Extracting software static defect models using data mining. Ain Shams Engineering Journal, 6, 2015, 133-144.
- [7] Elish M.O., Al-Yafei A.H., Al-Mulhem M.: Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems: A case study of Eclipse. Advances in Engineering Software, 42, 2011, 852-859.
- [8] Bluemke I.E., Zajac P., Metryki MOOD w systemie Rational Rose w: red. Huzar Z., Mazur Z., Problemy i metody inżynierii oprogramowania, Wydawnictwa Naukowo – Techniczne, Warszawa, 2003.
- [9] Martin R.C., Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki, Helion, 2015.
- [10] Radjenović D., Heričko M., Torkar R., Živković A.: Software fault prediction metrics: A systematic literature review. Information and Software Technology, 55, 2013, 1397-1418.
- [11] http://www.inmost.org.pl/articles/Metryki_obiektowe_jako_ws_kaAniki_jakoAci_kodu_i_projektu [04.06.2016].
- [12] Szyjewski Z., Muszyńska K., Zarządzanie projektami i modelowanie procesów, Polskie Towarzystwo Informatyczne, Warszawa, 2013.
- [13] <https://avandeursen.com/2014/08/29/think-twice-before-using-the-maintainability-index/> [08.06.2016].
- [14] <https://msdn.microsoft.com/en-us/library/bb385914.aspx> [08.06.2016].
- [15] Arapidis C.: Sonar Code Quality Testing Essentials: Achieve Higher Levels Of Software Quality With Sonar. Birmingham, Packt Publishing, 2012.
- [16] Derezińska A., Rudnik M.: Quality Evaluation of Object-Oriented and Standard Mutation Operators Applied to C# Programs. Lecture Notes in Computer Science, 7304, 2012, 42-57.
- [17] Kayarvizhy N., Kanmani S.: An Automated Tool for Computing Object Oriented Metrics Using XML. Advances in Computing and Communications, 191, 2011, 69-79.
- [18] Zalewski B.: Metryki oceny jakości oprogramowania i ich stosowalność. Praca magisterska pod kierunkiem Miłosza M., Politechnika Lubelska, Lublin, 2016, 71.
- [19] Taibi D., Janes A., Lenarduzzi V.: Towards a Lean Approach to Reduce Code Smells Injection: An Empirical Study. Agile Processes, in Software Engineering, and Extreme Programming, 251, 2016, 300-304.