

Porównanie aplikacji wspierających zastosowanie metodyk zwinnych w wytwarzaniu oprogramowania

Tomasz Bławucki*, Siarhei Ramanovich, Maria Skublewska-Paszkowska
Politechnika Lubelska, Katedra Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł przedstawia porównanie pod względem wymagań sprzętowych aplikacji wspierających wprowadzanie metodyk zwinnych do procesu wytwarzania oprogramowania. Przedmiotem badań były popularne aplikacje mobilne i internetowe wspomagające procesy Agile w przedsiębiorstwach. W celu określenia znaczenia poszczególnych wymagań technicznych dla użytkowników, przeprowadzono serię eksperymentów badawczych opartych na scenariuszach typowego i brzegowego użytkownika badanych systemów. Na potrzeby przeprowadzonej analizy została dodatkowo opracowana aplikacja wspierająca proces zwinnego wytwarzania oprogramowania. Wyniki pomiarów były rejestrowane za pomocą specjalistycznych narzędzi monitorujących pracę systemu i profilujących działanie przeglądarki internetowej. Rezultaty prac badawczych przedstawiono w formie tabel.

Słowa kluczowe: agile; aplikacje mobilne; aplikacje internetowe; wymagania sprzętowe

*Autor do korespondencji.

Adres e-mail: tomasz.blawucki@pollub.edu.pl

Applications supporting utilization of agile methods in software development process

Tomasz Bławucki*, Siarhei Ramanovich, Maria Skublewska-Paszkowska
Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents a comparison in terms of hardware requirements of applications that supports the agile software development processes. For research purposes, popular mobile and internet applications supporting agile software development were chosen. In order to determine the significance of individual technical requirements for end-users, a series of research experiments, based on scenarios of typical and boundary use was conducted. In addition to research, the application supporting agile software development process was implemented. The results of research were recorded by specialized monitoring and profiling tools. The results of performed work are presented in tabular form.

Keywords: agile; mobile applications; web applications; hardware requirements

*Corresponding author.

E-mail address: tomasz.blawucki@pollub.edu.pl

1. Wstęp

Na przestrzeni lat charakterystyka procesu wytwarzania oprogramowania podlegała ewolucji, czy to ze względu na postęp technologiczny, uzależnienie globalnego rynku od narzędzi informatycznych, czy też zmian zachodzących w samych zespołach projektowych. Niewątpliwie na dzień dzisiejszy ciężko jest sobie wyobrazić szeroko pojęty biznes bez systemów i produktów informatycznych. Tempo globalnej cyfryzacji rynków bezpośrednio wymusza na przedsiębiorstwach informatycznych odpowiednie przyspieszenie wytwarzania oprogramowania, przy jednoczesnym zachowaniu wymaganej przez inwestorów jakości.

Ze względu na często wysoki stopień skomplikowania komercyjnych rozwiązań IT, wymaganego przez klientów, konieczne jest zatrudnienie odpowiednio dużego zespołu projektowego dla ich realizacji. Liczne kadry specjalistów z różnych dziedzin IT realizujących wspólny projekt za sobą obciążenie w postaci utrudnionej komunikacji w grupie oraz konieczności odpowiedniej organizacji pracy, pozwalającej na efektywne wykorzystanie potencjału produkcyjnego. W celu ułatwienia procesu organizacji pracy grup projektowych, przedsiębiorstwa informatyczne często korzystają z narzędzi

w formie specjalistycznego oprogramowania przeznaczonego do wspomagania zwinnego zarządzania procesami wytwarzania oprogramowania.

Wybór systemu wspierającego proces zwinnego wytwarzania oprogramowania, odpowiedniego dla kontekstu projektu i specyfiki pracy zespołu, nie jest trywialny. Decydent powinien wziąć pod uwagę wiele czynników związanych z różnorodnymi aspektami pracy firmy i klienta. Zadanie jest dodatkowo utrudnione przez brak jasno sprecyzowanych zasad i wytycznych doboru takiego oprogramowania. Jedynie kilka publikacji bezpośrednio opisuje proces doboru narzędzi mających wspierać procesy *agile*. Ogólnodostępne blogi eksperckie, czy strony o tematyce *agile* w definicji procedury postępowania przy wyborze dedykowanego oprogramowania skupiają się głównie na dostarczanych funkcjonalnościach [14, 15, 16, 18]. Warto również dodać, że niektóre badania w zakresie dopasowania narzędzi do wprowadzania metodyk zwinnych zostały przeprowadzone na zlecenie firm lub korporacji, co potwierdza istnienie potrzeby eksploracji tego obszaru [1, 2].

Przegląd literatury dotyczącej wymagań technicznych systemów wspierających organizację pracy zespołów projektowych, jako kryterium doboru narzędzi *agile*, wykazał

brak publikacji o podobnej tematyce. Niniejsze opracowanie próbuje wypełnić tę niszę.

W przeciągu ostatnich trzech dekad można zauważyć płynne przejście w kontekście metodyk wytwarzania oprogramowania z metody kaskadowej (*ang. waterfall*) do metod zwinnych [4, 5], przy czym podejścia te są diametralnie różne. W przeszłości znakomita większość systemów IT powstawała w podejściu kaskadowym [6], dzisiaj uznawanym jako tradycyjne. Do końca lat 90-tych kaskadowe wytwarzanie oprogramowania było wystarczająco wydajne [13]. Metoda ta jednak około 20 lat temu przestała spełniać rosnące wymagania środowiska IT [12]. Obecnie, według ankiety przeprowadzonej przez J. Jeremiaha [3], metody zwinne są podstawowym podejściem przy procesie wytwórczym systemów informatycznych. Przełom nastąpił w latach 2008-2010, w których wiele znaczących przedsiębiorstw zauważyło zalety *agile* i wdrożyło nowatorskie metodyki we własnych zespołach projektowych, pomimo wad i trudności takich jak: ograniczone planowanie projektów, ograniczenia finansowe, czy wymuszone częste spotkania z docelowym odbiorcą produktu [4].

Zazwyczaj metodyki z rodziny metodyk *agile* nie stanowią kompletnej alternatywy dla tradycyjnych metod zarządzania procesem wytwarzania oprogramowania. Ich umiejętne wykorzystanie pozwala jednak na zastąpienie klasycznego podejścia w tym kontekście [7, 8].

U. Kelter i inni [9] już w 2003 roku przedstawili kluczowe różnice pomiędzy metodykami z rodziny *agile*, a tradycyjnymi podejściami do zarządzania procesami wytwarzania oprogramowania. W badaniach wykorzystano komercyjne doświadczenia praktyków oraz rezultaty z innych, podobnych opracowań. Istotne ze względu na aplikacje wspierające wprowadzanie metodyk zwinnych było zestawienie najważniejszych cech i funkcjonalności wynikających ze specyfikacji poszczególnych metodyk. Obszary, w których zastosowanie podejścia *agile* może przynieść korzyść według autorów to [9]:

- komunikacja, koordynacja i kooperacja współpracowników w obrębie zespołów projektowych;
- bieżące planowanie prac nad realizowanym projektem informatycznym;
- umożliwienie oceny jakości produktu oraz rezultatów pośrednich przez zewnętrznych referentów;
- wytwarzanie i dokumentacja - skupienie się na utrzymaniu jakości produktu, dokumentacja jako integralnej części oprogramowania;
- zarządzanie konfiguracjami projektu - umożliwienie szybszego reagowania na zachodzące zmiany wymagań;
- automatyzacja prostych procesów - generowanie raportów oraz sprawdzanie integralności modułów aplikacji w dużych systemach informatycznych (ciągła integracja).

Wprowadzenie dedykowanych narzędzi *agile*, według autorów, ma niebagatelne znaczenie dla poprawy wydajności zespołów projektowych oraz wzrostu prawdopodobieństwa pomyślnego zakończenia projektów informatycznych. Jako przeciwwagę dla zalet wykorzystania rozwiązań do zwinnego zarządzania projektami IT, przytacza się trudność

konfiguracji oprogramowania do specyfiki projektów oraz adaptacji samych metodyk do organizacji procesu wytwórczego oprogramowania [9].

W 2012 roku G. Azizyan i inni [2] przeprowadzili szczegółową analizę różnorodnych aspektów pracy i struktury przedsiębiorstwa, w celu doboru odpowiedniego narzędzia pozwalającego na skuteczne zwinne zarządzanie wymaganiami oraz optymalizację procesu realizacji projektów. Studium podzielono na dwie główne części: przegląd najważniejszych funkcjonalności i cech oprogramowania takich jak elastyczność i możliwość rozbudowy oraz obserwacja procesów wewnętrznych firmy w celu specyfikacji faktycznych potrzeb odnośnie docelowego narzędzia. Według autorów podział procesu wyboru narzędzia był konieczny do podjęcia właściwej decyzji. Etap pierwszy miał na celu usystematyzowanie faktycznych potrzeb wynikających ze specyfiki procesów oraz środowisk, w których zostały zaadaptowane metodyki zwinne. Jako rezultat analizy otrzymano listę 21 cech systemu, kluczowych z punktu widzenia charakterystyki przedsiębiorstwa, dalej wykorzystanych przy ocenie rozwiązań. Kryteriami wyboru rozwiązań podlegających dalszemu, szczegółowemu badaniu były [2]:

- zainteresowanie przedstawicieli przedsiębiorstwa danym rozwiązaniem;
- popularność rozwiązania;
- wsparcie dla metodyk zwinnych wykorzystywanych w przedsiębiorstwie;
- możliwości integracji z innymi rozwiązaniami;
- licencja oprogramowania;
- wspierane platformy systemowe.

Interesujące pod względem możliwości oceny również współcześnie dostępnych aplikacji mogą okazać się cechy wyszczególnione przez J. Cabot'a i innych [10]. Najistotniejsze z nich to:

- uniwersalność ze względu na dobór metodyki zarządzania;
- docelowa grupa odbiorców;
- rozmiar zespołów;
- rodzaj licencji oprogramowania oraz jej koszt;
- lokalizacja oprogramowania - czy serwer aplikacji znajduje się na osprzęcie producenta, czy w środowisku klienckim;
- integracja oprogramowania z zewnętrznymi narzędziami;
- możliwość adaptacji aplikacji do zachodzących zmian w obszarze wytwarzania oprogramowania.

M. Taheri i S. MosoudSadjadi [11] w swoim opracowaniu określili następujące cechy oprogramowania do wspierania procesów zwinnych, mające kluczowe znaczenie dla przedsiębiorstw IT. Są to [11]:

- elastyczność systemów pod kątem wymagań przedsiębiorstw;
- łatwość użycia narzędzi przez członków zespołów projektowych;
- rodzaj/rozmiar docelowego przedsiębiorstwa;
- cena oraz zasady licencjonowania oprogramowania;
- wsparcie ze strony producenta;

- funkcjonalność systemów.

Dodatkowo istotnym aspektem są zasady licencjonowania rozwiązań. Autorzy dokonali podziału rozwiązań na oprogramowanie płatne i otwarte źródłowe.

Pomimo dogłębnego przeglądu literatury, nie znaleziono artykułów naukowych, w których jako kryterium wyboru aplikacji do zwinnego zarządzania zespołami projektowymi, odniesiono się do ich wymagań sprzętowych.

2. Aplikacja do zwinnego zarządzania informatycznym zespołem projektowym

Celem dopełnienia niniejszego opracowania autorzy zdecydowali się na projekt i implementację dedykowanego narzędzia *agile*. Zrealizowana aplikacja docelowo ma usprawniać pracę zespołów projektowych w branży IT w kontekście zwinnego wytwarzania oprogramowania, poprzez ułatwienie wprowadzenia metodyk zwinnych do projektu oraz kontroli poprawności ich wykorzystywania. Dodatkowo system ma stanowić wsparcie kierownika zespołu programistów między innymi poprzez możliwość zdalnego przydzielania zadań poszczególnym pracownikom oraz dokumentację postępu prac.

Rozwiązanie dodatkowo ma zespolicz wybrane funkcjonalności obecne w niezależnych, istniejących już na rynku rozwiązaniach. Pozwoli to na ograniczenie liczby narzędzi koniecznych do skutecznego zarządzania pracą zespołów. Rozwiązanie można podzielić na następujące moduły funkcjonalne:

- moduł ewidencji i autoryzacji użytkowników;
- moduł wiadomości i notyfikacji;
- moduł ewidencji zadań;
- moduł ewidencji notatek użytkownika;
- moduł ewidencji i zarządzania grupami projektowymi;
- moduł ewidencji i zarządzania zdarzeniami grupowymi;
- moduł administracyjny.

System informatyczny przyjął formę aplikacji mobilnej działającej na urządzeniach wyposażonych w system operacyjny Android w wersji powyżej 6.0 (Marshmallow) włącznie. Architektura systemu została oparta na modelu klient-serwer, kanał komunikacji między interfejsem klienckim a serwerem został zrealizowany w postaci wymiany dokumentów JSON poprzez REST API. Technologie wybrane do implementacji części serwerowej stanowią: język Python w wersji 3.7 przy wykorzystaniu frameworku web Django w wersji 2.1.7 oraz modułu Django REST Framework. Rolę serwera WSGI pełni instancja Gunicorn w wersji 19.9.0 przeznaczona dla środowisk Unix. Aplikacja kliencka została zrealizowana głównie jako rozwiązanie mobilne, jednak ze względu na pracę administratorów systemu wprowadzono również panel administratora, dostępny poprzez przeglądarkę www.

Na system do zarządzania relacyjnymi bazami danych wybrano PostgreSQL. Operacje na bazie danych (BD) są realizowane poprzez warstwę ORM dostarczoną wraz z frameworkiem Django. Serwer doskonale nadaje się do konteneryzacji i skalowania, odpowiednio do obciążenia zapytaniami.

System w założeniu stanowi bazę do dalszej rozbudowy. Ze względu na ograniczone zasoby, rozwiązanie nie stanowi konkurencji dla obecnych na rynku podobnych narzędzi, a jedynie przedstawia jedną z możliwych ścieżek realizacji takiego systemu.

3. Metodyka badawcza

Hipotezą badawczą postawioną w niniejszym artykule było określenie czy wymagania sprzętowe części klienckich systemów informatycznych wspierających zwinne wytwarzanie oprogramowania mogą stanowić jedno z kryteriów wyboru takiego rozwiązania przez przedsiębiorstwa IT. Hipotezę rozpatrzono w dwóch dziedzinach: wydajności aplikacji internetowych oraz dedykowanych aplikacji mobilnych.

Badania pozwoliły na określenie narzutu sprzętowego generowanego przez poszczególne rozwiązania. Miały one formę serii eksperymentów polegających na wielokrotnym powtórzeniu scenariuszy badawczych, przy wykorzystaniu kolejnych aplikacji do zwinnego zarządzania procesem wytwarzania oprogramowania i równoczesne monitorowanie zużycia zasobów urządzenia gospodarza. Każdy scenariusz powtórzono co najmniej pięciokrotnie, celem minimalizacji obciążenia wyników przez wartości odstające i uzyskania bardziej reprezentatywnych wyników. Wskaźnikami technicznymi pracy aplikacji były:

- 1) dla aplikacji mobilnych:
 - zużycie pamięci RAM;
 - transfer sieciowy;
 - czas uruchomienia i reakcji;
 - wykorzystanie procesora;
 - zajętość przestrzeni trwałej urządzenia.
- 2) dla aplikacji internetowych:
 - zużycie pamięci RAM;
 - zużycie pamięci CACHE;
 - czas ładowania strony głównej (bez użycia pamięci CACHE);
 - czas ładowania strony głównej (z użyciem pamięci CACHE);
 - czas budowania widoku strony (bez użycia pamięci CACHE);
 - czas budowania widoku strony (z użyciem pamięci CACHE);
 - objętość załadowanej strony (bez użycia pamięci CACHE);
 - objętość załadowanej strony (z użyciem pamięci CACHE);
 - zużycie procesora (%);
 - liczba pobranych plików (bez użycia pamięci CACHE);
 - liczba pobranych plików (z użyciem pamięci CACHE).

Do badania poszczególnych wskaźników wymagań sprzętowych aplikacji posłużyły scenariusze 1 – 4 opisane w tabeli nr 1.

Tabela. 1. Scenariusze badawcze wykorzystane do badania poszczególnych wskaźników wymagań sprzętowych aplikacji

Scenariusz nr 1		
Wymagania	Przebieg eksperymentu	Cele wykonania scenariusza
Użytkownik uzyskał dostęp do systemu, posiada aktywne konto, jest członkiem odpowiedniej grupy projektowej, obecne uprawnienia pozwalają na aktywną pracę w obrębie wybranego projektu.	Po zalogowaniu użytkownik wybiera projekt do którego został przypisany; dodaje do projektu nowe zadania, przegląda tablicę Scrum/Kanban, edytuje jej wybrane elementy, zmienia status zadań, dokonuje innych działań dla pracownika o podstawowych uprawnieniach.	Dzięki wykonaniu scenariusza, narzędzia profilujące i monitorujące parametry pracy systemu dostarczają informacji na temat średniego wykorzystania zasobów urządzenia przez aplikację takich jak: zużycie procesora, zajętość pamięci RAM czy wykorzystanie sieci.
Scenariusz nr 2		
Wymagania	Przebieg eksperymentu	Cele wykonania scenariusza
Użytkownik posiada aktywne konto w systemie oraz uprawnienia kierownika zespołu lub projektu, może zarządzać danymi istniejących projektów i tworzyć nowe projekty.	Użytkownik po zalogowaniu do aplikacji tworzy nowy projekt oraz wprowadza podstawowe, informacje o nim (nazwa, opis, członkowie i terminy realizacji). Następnie zgodnie z dokumentacją aplikacji tworzy nową tablicę Scrum/Kanban. Użytkownik dodaje nowe zadania do tablicy i przypisuje do wybranych podwładnych.	Dzięki wykonaniu scenariusza, rozszerzono zbiór pomiarów odnoszących się do wymagań technicznych pracy aplikacji takich jak: zużycie procesora, zajętość pamięci RAM czy wykorzystanie sieci. Przetestowano także stabilność pracy aplikacji oraz określono ogólne odczucia związane z użytkowaniem aplikacji.
Scenariusz nr 3		
Wymagania	Przebieg eksperymentu	Cele wykonania scenariusza
Użytkownik posiada dostęp do aplikacji, nie musi posiadać aktywnego konta.	Użytkownik wielokrotnie uruchamia i wyłącza aplikację.	Przy każdorazowym powtórzeniu sekwencji włączenie->wyłączenie aplikacji dokonywany jest pomiar czasu potrzebnego na załadowanie aplikacji, pozwalający na zdefiniowanie komfortu korzystania z niej. Dodatkowo eksperyment

		pozwala na sprawdzenie stabilności pracy aplikacji oraz pomiar zajętości pamięci RAM oraz skrajnego wykorzystania czasu procesora.
Scenariusz nr 4		
Wymagania	Przebieg eksperymentu	Cele wykonania scenariusza
Użytkownik posiada dostęp do aplikacji oraz podstawowe uprawnienia pozwalające na swobodną nawigację w systemie i wykonywanie rutynowych działań w obrębie projektu/grupy projektowej.	Użytkownik loguje się do systemu i wybiera dowolny projekt, w którym uczestniczy. W obrębie projektu swobodnie nawiguje, przegląda dostępne informacje, zmienia status przypisanych do siebie zadań oraz wykonuje inne, dostępne akcje. Działania mają symulować rutynową pracę z aplikacją w ciągu około pięciu minut na pojedyncze powtórzenie.	W danym badaniu istotne jest zachowanie określonego okresu czasowego korzystania z aplikacji pozwalającego ocenić ją pod względem niezawodności i stabilności pracy. Dodatkowo dokonano pomiaru średniego wykorzystania czasu procesora, pamięci RAM oraz transferu sieciowego w czasie.

W tabeli 2 zaprezentowano specyfikację techniczną urządzenia mobilnego wykorzystanego do badań parametrów pracy aplikacji mobilnych. W tabeli 2 przedstawiono specyfikację techniczną jednostki komputerowej wykorzystanej do badań aplikacji internetowych.

Tabela. 2. Specyfikacja techniczna urządzenia mobilnego wykorzystanego do przeprowadzenia badań

Urządzenie	LG Nexus 5
CPU	QualcommSnapdragon 800@ 2.26GHz
RAM	1.5 GB LPDDR3
Dysk	16GB FLASH
System operacyjny	Android 6.0.1 Marshmallow
Sieć	Download 25 Mb/s, Upload 22 Mb/s

Tabela 3. Specyfikacja techniczna jednostki komputerowej wykorzystanej do przeprowadzenia badań

Urządzenie	Laptop ASUS N55JM
CPU	Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz
RAM	8GB DDR3 1600MHz
Karta graficzna	NVIDIA GeForce GTX 860M
Dysk	SSD 120GB
System operacyjny	Manjaro 18.0.4 IllyriaArchlinux - 64-bitowy
Sieć	Download 25 Mb/s, Upload 22 Mb/s

4. Rezultaty badań

Wyniki badań jasno wskazywały na różnice między badanymi aplikacjami. Zauważalne różnice mogły mieć źródło w stopniu skomplikowania rozwiązań - aplikacje bogatsze funkcjonalnie miały większe wymagania techniczne - lub w stopniu optymalizacji narzędzi oraz technologiach wykorzystanych przez producentów. W tabeli 3 przedstawiono zbiorcze porównanie badanych aplikacji mobilnych, w tabeli 4 przedstawiono rezultaty badań dla aplikacji internetowych. Kolorem zielonym i jasnozielonym oznaczono najlepsze rezultaty, natomiast kolor czerwony oznacza najgorszy wynik wskaźnika w zestawieniu. W tabeli 4 oprócz aplikacji komercyjnych dodano do porównania aplikację autorską *Agile_app*. Aplikacje pod względem narzutu sprzętowego zauważalnie się od siebie różnią, co oznacza że istnieje możliwość dyskryminacji niewłaściwych dla wymagań użytkownika narzędzi na podstawie narzutu sprzętowego.

Obecnie kryteria wyboru aplikacji wspierających proces zwinnego wytwarzania oprogramowania głównie skupiają się na funkcjonalnościach dostarczanych przez dane rozwiązanie [2, 11, 14]. Przeprowadzone badania wykazały, że niektóre wymagania sprzętowe aplikacji klienckich mogą również mieć znaczenie dla przedsiębiorstw IT oraz użytkowników końcowych. Średnie wykorzystanie pamięci RAM znacznie różniące się dla poszczególnych aplikacji internetowych i mobilnych może mieć znaczenie dla zachowania płynności pracy. Wymagania odnośnie szybkości procesora lub optymalizacja wykorzystania pamięci CACHE widocznie wpływają na komfort użytkownika aplikacji mobilnych i internetowych.

Tabela 4. Wyniki pomiarów parametrów pracy aplikacji mobilnych-wartości średnie, mediany i odchylenia standardowe badanych wskaźników

Parametry	Agile_app	MeisterTask	Yodiz	nTask	yougile	Jira	Trello
Zużycie pamięci RAM % – średnia	2,12	7,16	2,40	2,26	1,94	4,22	2,84
Zużycie pamięci RAM % – mediana	2,10	7,20	2,40	2,30	1,90	4,20	2,80
Zużycie pamięci RAM % – Och. Std. (odchylenie standardowe)	0,26	0,42	0,25	0,40	0,21	0,19	0,27
Zużycie pamięci RAM [MB] – średnia	32,56	109,98	36,86	34,71	29,80	64,82	43,62
Zużycie pamięci RAM [MB] – mediana	32,26	110,59	36,86	35,33	29,18	64,51	43,01
Zużycie pamięci RAM [MB] – Och. Std.	3,98	6,39	3,92	6,20	3,19	2,95	4,15
Transfer sieciowy Upload [MB] – średnia	0,194	0,110	0,056	0,094	0,051	0,669	0,458
Transfer sieciowy Upload [MB] – mediana	0,202	0,111	0,056	0,091	0,051	0,667	0,467
Transfer sieciowy Upload [MB] – Och. Std.	0,063	0,023	0,019	0,047	0,020	0,084	0,066
Transfer sieciowy Download [MB] – średnia	0,527	0,160	0,284	0,320	0,238	0,667	0,614
Transfer sieciowy Download [MB] – mediana	0,532	0,165	0,283	0,272	0,162	0,605	0,533
Transfer sieciowy Download [MB] – Och. Std.	0,128	0,035	0,112	0,127	0,167	0,176	0,180
Czas uruchomienia i reakcji [s] – średnia	1,75	1,74	3,22	6,82	3,30	2,40	3,55
Czas uruchomienia i reakcji [s] – mediana	1,77	1,74	3,22	6,82	3,32	2,26	3,55
Czas uruchomienia i reakcji [s] – Och. Std.	0,11	0,03	0,02	0,03	0,07	0,32	0,04
Wykorzystanie procesora [%] – średnia	53,8	72,6	44,6	78,6	83,4	81,0	79,4
Wykorzystanie procesora [%] – mediana	54,0	72,0	43,0	78,0	85,0	81,0	79,0
Wykorzystanie procesora [%] – Och. Std.	13,26	7,96	5,57	4,62	7,02	6,28	5,18
Zajętość przestrzeni trwałej urządzenia [MB]	26,26	22,75	14,66	24,14	38,24	54,83	31,50

Podczas badań niektóre aplikacje znacznie przekraczały akceptowalny czas oczekiwania użytkownika na reakcję lub uruchomienie 2~3 sekund, negatywnie wpływając na komfort użytkownika [19]. Jedynie aplikacje internetowe *Yodiz* i *nTask* oraz aplikacje mobilne *MeisterTask*, *JiraCloud* oraz opracowana aplikacja *Agile_app* spełniły to kryterium.

Tabela 5. Wyniki pomiarów parametrów pracy aplikacji internetowych-wartości średnie, mediany i odchylenia standardowe badanych wskaźników

Parametry	MeisterTask	Yodiz	nTask	yougile	Jira	Trello
Zużycie pamięci RAM [MB] – średnia	245,40	190,60	216,60	124,00	328,60	141,00
Zużycie pamięci RAM [MB] – mediana	262,00	188,00	217,00	122,00	348,00	157,00
Zużycie pamięci RAM [MB] – Och. Std. (odchylenie standardowe)	50,98	19,82	22,23	13,21	51,74	30,56
Zużycie pamięci CACHE [MB] – średnia	2,09	2,20	2,99	2,06	5,36	1,69
Zużycie pamięci CACHE [MB] – mediana	2,09	2,20	2,99	2,06	5,36	1,69
Zużycie pamięci CACHE [MB] – Och. Std.	0,00	0,00	0,00	0,00	0,00	0,00
Czas ładowania strony głównej (bez użycia pamięci CACHE) [s] – średnia	3,21	3,69	2,71	1,24	8,05	3,85
Czas ładowania strony głównej (bez użycia pamięci CACHE) [s] – mediana	3,17	3,66	2,83	1,26	8,13	3,90
Czas ładowania strony głównej (bez użycia pamięci CACHE) [s] – Och. Std.	0,11	0,62	0,34	0,07	0,27	0,22
Czas ładowania strony głównej (z użyciem pamięci CACHE) [s] – średnia	2,35	2,82	1,77	0,86	5,09	3,46
Czas ładowania strony głównej (z użyciem pamięci CACHE) [s] – mediana	2,31	2,80	1,82	0,87	5,12	3,44
Czas ładowania strony głównej (z użyciem pamięci CACHE) [s] – Och. Std.	0,04	0,20	0,19	0,10	0,22	0,09
Czas budowania widoku strony (bez użycia pamięci CACHE) [s] – średnia	4,18	4,60	2,79	2,27	8,23	3,37
Czas budowania widoku strony (bez użycia pamięci CACHE) [s] – mediana	4,17	4,72	2,82	2,29	8,29	3,39
Czas budowania widoku strony (bez użycia pamięci CACHE) [s] – Och. Std.	0,09	0,62	0,20	0,07	0,29	0,20
Czas budowania widoku strony (z użyciem pamięci CACHE) [s] – średnia	3,28	3,78	1,81	1,66	5,10	1,97
Czas budowania widoku strony (z użyciem pamięci CACHE) [s] – mediana	3,22	3,81	1,85	1,57	5,06	1,99
Czas budowania widoku strony (z użyciem pamięci CACHE) [s] – Och. Std.	0,083	0,076	0,189	0,322	0,236	0,041
Objętość załadowanej strony (bez użycia pamięci CACHE) [MB] – średnia	2,10	2,30	3,46	2,18	5,60	1,82
Objętość załadowanej strony (bez użycia pamięci CACHE) [MB] – mediana	2,10	2,30	3,50	2,10	5,60	1,80
Objętość załadowanej strony (bez użycia pamięci CACHE) [MB] – Och. Std.	0,00	0,00	0,09	0,11	0,00	0,04
Objętość załadowanej strony (z użyciem pamięci CACHE) [MB] – średnia	0,01	0,10	0,47	0,12	0,24	0,13
Objętość załadowanej strony (z użyciem pamięci CACHE) [MB] – mediana	0,01	0,10	0,47	0,12	0,24	0,13
Objętość załadowanej strony (z użyciem pamięci CACHE) [KB] – Och. Std.	0,00	0,00	0,00	0,00	0,00	0,00
Zużycie procesora [%] – średnia	11,00	13,00	13,80	10,60	16,20	10,20
Zużycie procesora [%] – mediana	11,00	12,00	15,00	10,00	16,00	11,00
Zużycie procesora [%] – Och. Std.	2,16	2,35	2,39	1,82	2,86	1,92
Liczba pobranych plików (bez użycia pamięci CACHE) – średnia	31,00	105,00	119,60	113,20	94,60	81,60
Liczba pobranych plików (bez użycia pamięci CACHE) – mediana	31,00	105,00	121,00	111,00	94,00	82,00
Liczba pobranych plików (bez użycia pamięci CACHE) – Och. Std.	0,00	0,00	2,88	3,96	1,34	1,67
Liczba pobranych plików (z użyciem pamięci CACHE) – średnia	11,00	20,00	38,20	18,60	43,20	39,80
Liczba pobranych plików (z użyciem pamięci CACHE) – mediana	11,00	20,00	40,00	18,00	43,00	40,00
Liczba pobranych plików (z użyciem pamięci CACHE) – Och. Std.	2,16	1,87	4,97	1,52	1,48	0,84

Pomimo braku celowej optymalizacji aplikacji, wyniki badania wymagań sprzętowych w porównaniu z pozostałymi narzędziami nie różniła się znacząco lub, w niektórych przypadkach osiągała lepsze wyniki. Dodatkowo, podczas eksperymentów kilkakrotnie doszło do awarii aplikacji mobilnych *JiraCloud* oraz *nTask*. Ciekawą obserwacją jest fakt ogromnej popularności rozwiązania z oferty firmy *Atlassian*, [1, 10] pomimo najwyższych zmierzonych wymagań sprzętowych. Niektóre aplikacje mobilne znacząco części danych i plików przechowywała lokalnie na urządzeniu, co pozwalało na dostęp do części informacji bez dostępu do sieci Internet lub przy wykorzystaniu łącz o niskiej przepustowości. Te aplikacje odpowiadały wymaganiom dostępności i funkcjonalności [10, 11] stawianym przez niektóre przedsiębiorstwa.

Uzyskane wyniki pozwoliły na stwierdzenie małego znaczenia wymagań sprzętowych części klienckich systemów informatycznych wspierających zwinne wytwarzanie oprogramowania w kontekście wyboru tych rozwiązań przez przedsiębiorstwa IT. Nie wykluczono jednak okoliczności w których może wystąpić konieczność ich uwzględnienia, np. w przypadku skrajnych ograniczeń technicznych przedsiębiorstwa.

5. Podsumowanie i wnioski

Celem badań było sprawdzenie czy aspekty techniczne działania aplikacji tak mobilnych jak i aplikacji internetowych w zakresie zwinnego zarządzania projektami informatycznymi mogą mieć znaczenie w procesie wyboru odpowiedniego narzędzia przez przedsiębiorstwa IT. Aktualnie proces ten jest stosunkowo chaotyczny i obciążony miarą popularności danych rozwiązań.

Przedstawione badania wykazują mierną optymalizację narzędzi, będących aktualnie liderami na rynku oprogramowania *agile*. Istotnie gorsze wyniki wynikają ze stopnia rozbudowania i skomplikowania tych rozwiązań.

Wskaźniki techniczne mimo tego że dziś nie są już tak istotne jak kiedyś, nadal mogą stanowić wartościowe kryterium dla małych lub średnich przedsiębiorstw IT, poszukujących aplikacji wspierających ich procesy *agile*. Pomimo obniżenia kosztów utrzymania sieci dla przedsiębiorstw, wybór łącz o niższej przepustowości, nadal pozwalających na swobodną pracę firmy może doprowadzić do kolejnych oszczędności. Również duże firmy, wybierając aplikacje *agile* o niższych wymaganiach odnośnie przepustowości sieci ograniczają prawdopodobieństwo nadmiernego przeciążenia infrastruktury.

Aplikacje mobilne o niższym narzucie sprzętowym będą płynnie działały na starszych lub tańszych urządzeniach, dzięki czemu przedsiębiorstwa planując zakup urządzeń służbowych dla pracowników mają możliwość kolejnych oszczędności.

Optymalizacja wykorzystania pamięci CACHE przez aplikacje internetowe oraz użycia lokalnej pamięci urządzenia mobilnego do przechowywania danych aplikacji w przypadku aplikacji mobilnych ma duże znaczenie dla komfortu pracy użytkownika. Dzięki tym działaniom znacząco spada czas oczekiwania na załadowanie treści stron i widoków, aplikacje są bardziej responsywne a łącze internetowe nie jest niepotrzebnie obciążane.

Przeprowadzone badania odnosiły się wyłącznie do wymagań sprzętowych wybranych aplikacji przeznaczonych do zwinnego zarządzania projektami informatycznymi. Przegląd literatury wykazał brak podobnych opracowań, co może wskazywać na potencjalny kierunek dalszych badań.

Pomimo braku celowej optymalizacji, autorska aplikacja zaprezentowana w artykule nie odbiegała znacząco od pozostałych analizowanych rozwiązań pod względem wymagań sprzętowych. W niektórych przypadkach osiągała lepsze wyniki, od innych badanych narzędzi.

Zasadność przeprowadzonych eksperymentów aktualnie jest podważalna ze względu na ciągle rosnącą moc obliczeniową, zmniejszający się koszt pamięci RAM oraz

przestrzeni dyskowej czy też powszechny dostęp do szerokopasmowej sieci Internet. Większe zasoby sprzętowe nie oznaczają jednak braku konieczności optymalizacji aplikacji i respektowania ograniczeń docelowych urządzeń klienckich na które obecnie trafiają.

Literatura

- [1] VersionOne.com. The 1-13th Annual "State of Agile" - Survey
- [2] G. Azizyan, M. Magarian, M. Kajko-Mattsson. The Dilemma of Tool Selection for Agile Project Management. ICSEA 2012, The Seventh International Conference on Software Engineering Advances. 2012
- [3] Survey:Is agile the new norm?,<https://techbeacon.com/app-dev-testing/survey-agile-new-norm>, [10.08.2019]
- [4] M. S. Raunak, D. Binkley. Agile and other trends in software engineering, 2017
- [5] A. M. Dima, M. A. Maassen. From Waterfall to Agile software: Development models in the IT sector, 2006 to 2018. Impacts on company management, 2018
- [6] S. Balaji, M. Sundararajan Murugaiyan. Waterfall vs v-model vs agile: A comparative study on SDLC, 2012
- [7] K. Vilkki. When Agile is not enough. 2010
- [8] D. Turk, R. France, and B. Rumpel. Limitations of agile software processes. 2014
- [9] U. Kelter, M. Monecke, M. Schild. Do we need Agile Software Development tools? 2003
- [10] J. Cabot, G. Wilson. Tools for Teams: A Survey of Web-Based Software Project Portals. 2009
- [11] M. Taheri, S. M. Sadjadi. A Feature-Based Tool-Selection Classification for Agile Software Development. 2015
- [12] N. Abbas, A. M. Gravell, G. B. Wills. Historical Roots Of Agile Methods: Where Did "Agile Thinking" Come From?. 2008
- [13] Gilb, T. Evolutionary Delivery versus the "waterfall model". ACM SIGSOFT Software Engineering Notes. 1985
- [14] 7 Criteria for Selecting the Best Project Management Software and Collaboration Tool, <https://www.planview.com/resources/articles/selecting-best-project-management-software/>, [21.09.2019]
- [15] Buyers' Guide for Beginners Selecting PM Software, <https://www.projectmanager.com/blog/buyers-guide-selecting-pm-software>, [21.09.2019]
- [16] How to pick right agile tool, <https://www.smartsheet.com/how-pick-right-agile-tool>, [21.09.2019]
- [17] Agile tool selection, <https://www.slideshare.net/choldorf/agile-tools>, [21.09.2019]
- [18] Choosing Project Management Software: A Buyer's Guide, <https://www.businessnewsdaily.com/9976-project-management-software-buyers-guide.html>, [21.09.2019]
- [19] F. Fui-Hoon Nah, A study on tolerable waiting time: how long are Web users willing to wait?, 2007