

# Implementation of solutions for distributed team management in IT sector companies

## Wdrożenie rozwiązań do zarządzania zespołem rozproszonym w firmach sektora IT

Mykhailo Kuzyk\*, Elżbieta Miłoś

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

### Abstract

The article presents the results of a comparative analysis of selected IT tools supporting the management of a distributed IT team in companies from the IT sector. The research was carried out using two methods: scientific experiment (implementation of a test project in the tested tools) and comparative analysis (assessment of tools from the side of management and contractors of a distributed team according to the proposed criteria). The research results were aimed at choosing the best solutions that could be implemented to manage a distributed team in IT sector companies.

*Keywords:* IT project management; distributed teams; remote work; IT tools

### Streszczenie

Artykuł przedstawia wyniki analizy porównawczej wybranych narzędzi informatycznych wspomagających zarządzanie rozproszonym zespołem informatyków w firmach z sektora IT. Badania przeprowadzono z wykorzystaniem dwóch metod: eksperymentu naukowego (wykonanie testowego projektu w badanych narzędziach) i analizy porównawczej (ocena narzędzi od strony kierownictwa i wykonawców rozproszonego zespołu wg zaproponowanych kryteriów). Wyniki badań miały na celu wybór najlepszych rozwiązań, jakie mogłyby zostać wdrożone do zarządzania zespołem rozproszonym w firmach sektora IT.

*Słowa kluczowe:* zarządzanie projektem informatycznym; zespoły rozproszone; praca zdalna; narzędzia informatyczne

\*Corresponding author

Email address: [mykhailo.kuzyk@pollub.edu.pl](mailto:mykhailo.kuzyk@pollub.edu.pl) (M. Kuzyk)

©Published under Creative Common License (CC BY-SA v4.0)

## 1. Wstęp

Intensywny rozwój branży IT oraz zapotrzebowanie na systemy informatyczne w biznesie stawia nowe wyzwania wielu firmom sektora IT, tworzącym rozwiązania informatyczne. Konkurencja na rynku pracy, brak specjalistów z branży IT, rywalizacja w firmach informatycznych sprawia, że wiele z nich podejmuje decyzje o tworzeniu międzynarodowych zespołów rozproszonych, których członkowie znajdują się fizycznie w różnych lokalizacjach i w formie zdalnej pracują nad jednym projektem.

W ostatnich dziesięcioleciach zdalna praca silnie umocniła swoją pozycję na rynku, światowe korporacje informatyczne zatrudniają wykwalifikowanych specjalistów głównie w formie pracy zdalnej. Takie korporacje jak Procter & Gamble, IBM, Accenture i AT & T, częściowo lub całkowicie zrezygnowały z tradycyjnego biura pracy. Dotyczy to głównie obszarów wykorzystania informatycznych technologii, wytwarzania oprogramowania oraz księgowości, co pozwala na znaczne zwiększenie efektywności pracy [1]. Wzrost ilości pracy wykonywanej zdalnie jest związany z globalizacją rynku pracy, zwiększeniem zapotrzebowania na specjalistów wąskich specjalizacji i wymagań klientów.

Zespoły rozproszone i zdalna praca stawiają nowe wyzwania w obszarze komunikacji i zarządzania. Obecnie dla zarządu korporacji i pracowników – członków

zespołów rozproszonych istnieje wiele skutecznych narzędzi do zarządzania projektem dla elastycznego prowadzenia pracy zdalnej i organizacji pracy. Wdrożenie odpowiednich rozwiązań oznacza dla firm umocnienie przewagi ekonomicznej wśród innych firm na rynku [2]. W konsekwencji zatrudniani w tej formie specjaliści utrzymują swoje stanowisko przez długi czas i są bardziej zmotywowani i lojalni w stosunku do pracodawców. Zmniejszenie liczby pełnoetatowych pracowników w biurze, pozwala korporacjom na odzyskanie wysokiego współczynnika skuteczności i znacznie zaoszczędzić na wynajmie miejsc przeznaczonych dla zdalnych pracowników. International Business Machines Corp. oszczędza około 100 mln. \$ rocznie ze zmianą formy pracy na zdalną. W tej firmie około 42% pracowników pracuje zdalnie [3].

Wdrożenie rozwiązań do zarządzania zespołem rozproszonym w firmach sektora IT powinno być poprzedzone analizą dostępnych narzędzi informatycznych i wyborem najlepszych z nich. Przeprowadzona analiza porównawcza może dać rekomendacje w tym obszarze dla firm wytwarzających oprogramowanie.

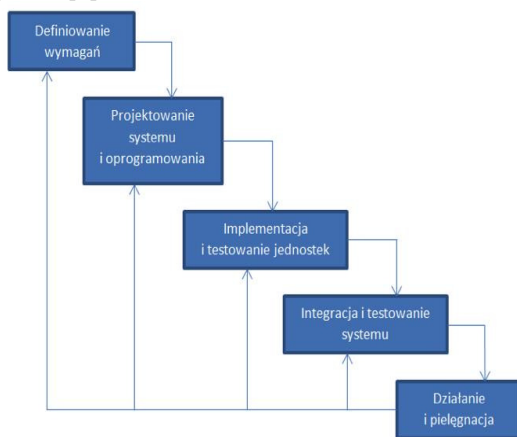
## 2. Podstawowe modele rozwoju oprogramowania

### 2.2. Model kaskadowy

Model kaskadowy – to model procesu wytwarzania oprogramowania, w którym proces rozwoju jest przed-

stawiony jako strumień (Rys.1). Dany strumień sekwencyjnie przechodzi przez fazę analizy wymagań, projektowania, wdrażania, testowania, integracji i wsparcia.

Po przeanalizowaniu wymagań generowana jest lista wymagań w stosunku do oprogramowania. Po jej sformułowaniu, opracowywany jest projekt, podczas którego tworzona jest dokumentacja z opisem wymagań dla programistów i planem realizacji. Następnie projekt jest implementowany w wybranym środowisku programistycznym, kolejne jednostki programistyczne poddawane są testowaniu. Na kolejnym etapie odbywa się integracja poszczególnych komponentów utworzonych wcześniej przez programistów. Po zakończeniu implementacji i integracji przeprowadzane są testy i debugowanie produktu, gdzie są eliminowane wszystkie wady poprzednich etapów wytwarzania oprogramowania. Potem odbywa się etap wdrożenia produktu i jego wsparcie [4].



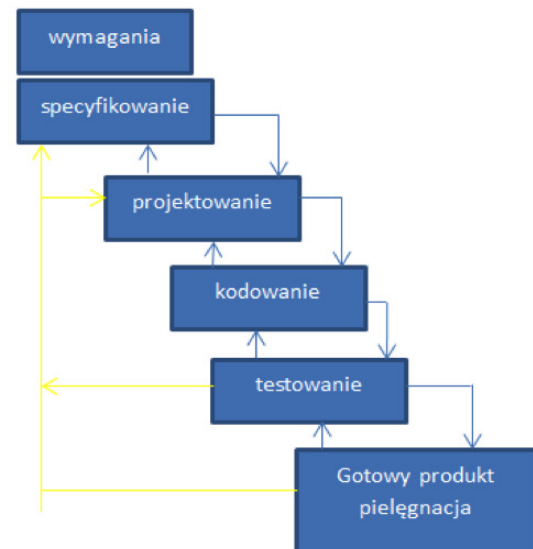
Rysunek 1: Model kaskadowy [8]

## 2.2. Model iteracyjny

Iteracyjny model rozwoju oprogramowania to proces tworzenia oprogramowania, poprzez małe etapy, z równoległą analizą uzyskanych pośrednich wyników, tworzone są nowe wymagania i korygowane są poprzednie etapy pracy (Rys.2). Cykl życia projektu podczas iteracyjnego oprogramowania jest podzielony na kolejne iteracje (etapy), każda z nich tworzy projekt w miniaturze, w tym wszystkie procesy tworzenia oprogramowania, ale w ramach jednego etapu opracowywana jest, tworzona tylko jedna wersja lub oddzielna część [5].

Celem iteracji jest uzyskanie wersji oprogramowania, wraz z nowymi lub zmienionymi funkcjami zaimplementowanymi w bieżącej iteracji, jak i we wszystkich poprzednich. Końcowa iteracja implementuje wszystkie wymagane funkcje produktu.

Czasowe ograniczenia i budżet projektu nie są ustalane, i formują się w trakcie realizacji, jako że wynik niemożliwe jest dokładne określenie zakresu prac dla realizacji ostatecznej wersji.



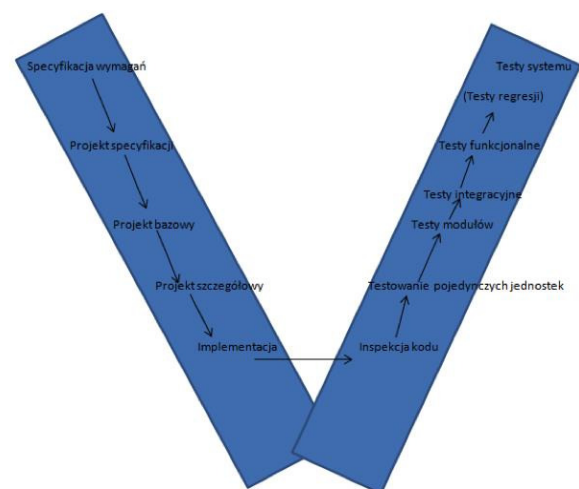
Rysunek 2: Model iteracyjny [8]

## 2.2. Model V

Przykładowe Model V – to ulepszona wersja modelu kaskadowego (Rys.3). Na każdym realizowanym etapie przeprowadzana jest kontrola bieżącego procesu, aby potwierdzić możliwość przejścia do następnego etapu. Testowanie rozpoczyna się od etapu wymagań i dla każdego etapu ustalany jest zestaw testów, które muszą zostać zaliczone, aby produkt mógł wejść w kolejny etap rozwoju [6].

Rysunek 3 prezentuje przebieg pracy w modelu V. Lewe część schematu przedstawia etapy projektowania i programowania, natomiast prawa - etap testowania. Przejście między jednym a drugim etapem przedstawione jest strzałką łączącą oba ramiona.

Dla każdego poziomu testowego tworzony jest dedykowany plan testowy, podczas testowania bieżącego poziomu równolegle tworzy się następny test. Wyniki testów i określenie kryteriów wejścia i wyjścia każdego etapu są określane podczas tworzenia planów testowych. Testowanie w ten sposób odgrywa szczególną rolę w tym modelu projektowania [6].



Rysunek 3: Model V [8]

## 2.2. Model spiralny

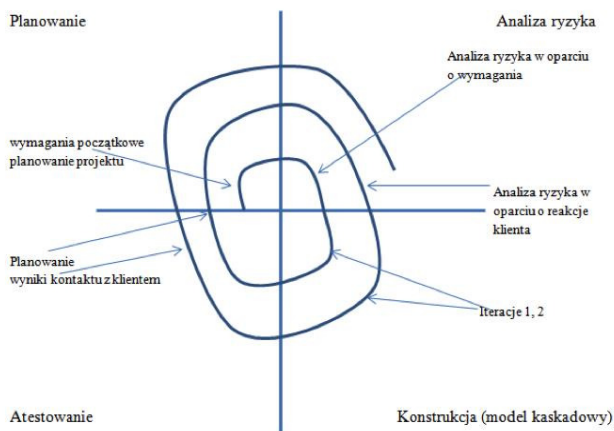
Model spiralny zawiera podstawowe zalety modelu kaskadowego (Rys.4). W tym przypadku w modelu włączono etapy: analizy i zarządzanie ryzykiem, a także wsparcie i zarządzanie. Tworzenie oprogramowania realizowane jest metodą szybkiego wytwarzania aplikacji przy użyciu języków programowania i innych narzędzi programistycznych [7].

Model polega na tym, że każda pętla zawiera zestaw operacji, który ma taką samą liczbę etapów jak modele kaskadowe. Każda część produktu i każdy poziom złożoności, zaczynając od sformułowania wymagań i kończąc kodowaniem programu, ma specjalną uwagę w tym modelu.

Początek procesu wytwarzania rozpoczyna się z określania celów projektu i ich alternatywnych wariantów celów, badania ryzyka, mapowanie planów ich realizacji i przygotowanie do następnej iteracji [7].

Kwadranty nie mają liczby cykli, są wybierane z konieczności, a iteracje są dostosowywane do projektu indywidualnie. Należy zwrócić uwagę, że kodowanie odbywa się później niż w innych modelach w celu minimalizacji ryzyka poprzez wyjaśnienie wymagań dotyczących produktu.

Najważniejszą różnicą od innych jest uwzględnienie ryzyka, na przykład przekroczenie terminów a przekroczenie wydatków. Ten model jest rzadko używany, ale przyczynił się do badania natury iteracyjnego rozwoju i znaczenia uwzględniania ryzyka.



Rysunek 4: Model spiralny [8]

## 3. Założenia badań

Celem badań jest określenie najlepszego rozwiązania do zarządzania rozproszonym zespołem za pomocą analizy porównawczej wybranych narzędzi informatycznych. Do badań zostało wybranych 6 najpopularniejszych narzędzi wspomagających zarządzanie projektami:

1. GanttPro.
2. Freecamp.
3. SmartSheet.
4. Kanbanize.
5. Clarizen.
6. ActiveCollab

Badania realizowane są za pomocą dwóch metod badawczych: analizy porównawczej i eksperymentu naukowego badającego przydatność tych narzędzi do zarządzania zespołem rozproszonym.

W badaniu postawiono następujące hipotezy badawcze:

H1. Program Kanbanize najlepiej realizuje wszystkie priorytetowe wymagania - z punktu widzenia wykonawców.

H2. Program Smartsheet najlepiej spełnia wszystkie priorytetowe wymagania, które są niezbędne dla skutecznego systemu zarządzania firmą IT - z punktu widzenia kierownictwa.

Na potrzeby badań zdefiniowano kryteria porównawcze, na podstawie których będą oceniane wyżej wymienionych narzędzia - w celu weryfikacji postawionych hipotezy badawcze.

Analizując potrzeby uczestników rozproszonych zespołów informatycznych dla weryfikacji hipotezy 1 zdefiniowano następujące kryteria:

K1: Jakość komunikacji

- Wygoda i możliwość współredagowania
- Efektywne zarządzanie plikami
- Integracja z e-mailem

K2: Elementy wizualne i doświadczenie użytkownika

- Intuicyjność
- Ocena graficznej części interfejsu
- Dostosowanie do wymagań użytkownika
- Ocena procesu tworzenia konta i zapraszanie innych uczestników

K3: Integracja z różnymi platformami.

- Urządzenia mobilne (Android, iOS)
- Prawidłowa współpraca z przeglądarkami

K4: Dodatkowe funkcje

- Wielojęzyczność
- Bezpieczne i szybkie logowanie za pomocą metodą SA,2FA,SSL
- Dostępność i poziom elastyczności konfiguracji
- Wsparcie dla dodatkowych wtyczek

Analizując potrzeby kadry zarządzającej rozproszonym zespołem informatycznym dla weryfikacji hipotezy 2 zdefiniowano następujące kryteria:

K1: Definiowanie funkcjonalności programu

- Możliwość tworzenia wielu projektów
- Wsparcie dla schematu Gantta
- Wsparcie dla metody rozwoju Kanban
- Zarządzanie wieloma użytkownikami
- Szablony projektów
- Integracja z systemami plików
- Import \ Eksport do XML, CVS i inne
- Kontrola wersji
- Możliwość archiwizacji projektu

K2: Zarządzanie zasobami

K3: Raportowanie i analiza

K4: Możliwości produktu

- Okres próbny do zapoznania się
- Funkcjonalność dostępna w wersji podstawowej
- Cena.

Każde badane narzędzie informatyczne zostało ocenione w trakcie tworzenia projektu testowego według

ww. kryteriów, których skale ocen zostały przedstawione w Tabeli 1 i Tabeli 2.

Tabela 1. Ocena kryteriów dla H1

Dla uczestników zespołów:	Ocena
Jakość komunikacji	0 - 6
Wizualizacja interfejsu	0 - 12
Integracja z różnymi platformami.	0 - 4
Dodatkowe funkcje	0 - 12

Tabela 2. Ocena kryteriów dla H2

Dla kierowników:	Ocena
Definicja funkcjonalności programu:	0 - 19
Zarządzanie zasobami	0 - 10
Raportowanie i analiza	0 - 10
Funkcje darmowej wersji	0 - 7

#### 4. Wyniki analizy

W eksperymencie oceny narzędzi wykorzystano przykładowy testowy projekt - stworzenie systemu informatycznego do kompleksowej automatyzacji logistyki transportu. Przy ocenie kryteriów zastosowano system naliczania punktów, na przykład, jeśli funkcja określona jako kryterium oceny jest dostępna dla podstawowej wersji produktu to przyznajemy 1 pkt, w przypadku braku - 0 pkt. Jeśli funkcjonalność ma elastyczne zastosowanie lub dostatecznie rozwinięty potencjał, to wynik może być wyższy. Testowy projekt został zastosowany do wszystkich wybranych programów. Suma punktów dla każdego z kryteriów została pogrubiona, podobnie jak suma wszystkich punktów każdego programu dla każdej grupy potencjalnych uczestników (Tabela 3).

#### 5. Wnioski

Wyniki analizy porównawczej zamieszczone w Tabeli 3 pokazują, że program Smartsheet uzyskał najwyższy wynik spośród wszystkich produktów. Ponadto produkt Kanbanize ma wystarczająco dużo punktów, aby pokazać szeroką funkcjonalność do pracy w projektach ze strony wykonawcy projektu - uczestnika zespołu rozproszonego, co potwierdza pierwszą hipotezę. Z jego mocnych stron warto zwrócić uwagę na prawidłowe wsparcie dla różnych przeglądarek i dostęp do mobilnych wersji tego produktu. Inne narzędzia wykazują niższe wartości sumarycznego kryterium. Podczas analizy wielokrotnie wystąpiła sytuacja, w której w. nie mógł skorzystać z istniejącej, ale niedziałającej funkcjonalności. Przykłady takich sytuacji to: jak import projektu do programu Freecamp, nieprawidłowe działanie Clarizen w przeglądarce Chrome, wyświetlanie Smartsheet w przeglądarce Explorer.

Tabela 3. Wyniki badań

Kryterium	Program					
	GranttPro	Freecamp	Smartsheet	Kanbanize	Clarizen	lab
<b>K1: Definicja funkcjonalności programu</b>	<b>16</b>	<b>11</b>	<b>15</b>	<b>12</b>	<b>10</b>	<b>8</b>
Możliwość tworzenia wielu projektów	1	2	2	2	2	1
Wsparcie dla schematu Gantta	2	1	1	1	1	1
Obsługa metody rozwoju Kanban	1	1	2	2	1	1
Zarządzanie wieloma użytkownikami	2	2	2	2	1	2
Szablony projektów	1	1	2	1	1	1
Integracja z systemami plików	2	2	2	2	0	1
Importuj \ Eksportuj pliki	3	1	3	1	1	1
Kontrola wersji	2	0	1	0	2	0
Możliwość archiwizacji projektu	2	1	0	1	1	0
<b>K2: Zarządzanie zasobami</b>	<b>6</b>	<b>8</b>	<b>9</b>	<b>4</b>	<b>2</b>	<b>7</b>
<b>K3: Raportowanie i analiza</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>8</b>	<b>2</b>	<b>9</b>
<b>K4: Możliwości produktu</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>6</b>	<b>4</b>	<b>7</b>
Okres próbny do zapoznania się	1	1	2	2	2	2
Dostępna funkcjonalność wersji podstawowej	1	1	2	2	2	2
Koszt	3	3	1	2	0	3
<b>K1: Jakość komunikacji</b>	<b>4</b>	<b>4</b>	<b>6</b>	<b>4</b>	<b>3</b>	<b>4</b>
Wygoda i możliwość współredagowania	2	2	2	2	1	2
Efektywne zarządzanie plikami	2	1	2	1	0	1
Praca z pocztą e-mail	0	1	2	1	2	1
<b>K2: Wizualizacja interfejsu</b>	<b>11</b>	<b>8</b>	<b>10</b>	<b>10</b>	<b>6</b>	<b>11</b>
Intuicyjne zrozumienie	3	1	3	3	3	3
Ocena części graficznej interfejsu	3	3	3	3	2	3
Dostosowywanie do wymagań użytkownika	2	1	2	1	0	2
Ocena procesu rejestracji i zapraszanie innych uczestników	3	3	2	3	1	3
<b>K3: Integracja z różnymi platformami</b>	<b>2</b>	<b>4</b>	<b>3</b>	<b>4</b>	<b>2</b>	<b>2</b>
Rodzaje urządzeń mobilnych	0	2	2	2	2	0
Prawidłowe działanie różnych przeglądarek	2	2	1	2	0	2
<b>K4: dodatkowe funkcje</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>8</b>	<b>2</b>	<b>8</b>
Wielojęzyczność	1	2	1	1	0	1
Bezpieczne i szybkie logowanie za pomocą technologii	2	1	2	2	1	1
Dostępność i poziom elastycznej konfiguracji	3	3	3	3	0	2
Obsługa dodatkowych wtyczek	2	3	4	2	1	4
<b>Dla kierowców:</b>	<b>33/46</b>	<b>31/46</b>	<b>37/46</b>	<b>30/46</b>	<b>18/46</b>	<b>31/46</b>
<b>Dla wykonawców:</b>	<b>25/34</b>	<b>25/34</b>	<b>29/34</b>	<b>26/34</b>	<b>13/34</b>	<b>25/34</b>

Powoduje to wiele błędów w przypadku braku szczegółowej dokumentacji i utrata kontekstu podczas tłumaczenia języka interfejsu. GanttPro doskonale nadaje się do realizacji podstawowych funkcji projektu, takich jak alokacja ról, zasobów wirtualnych, ma dobre narzędzia do importowania i eksportowania programów, co zapewnia wygodę w realizacji małych projektów.

Clarizen, który w badaniach uzyskał najniższą notę, ma niską elastyczność, zarówno dla menedżerów, jak i dla wykonawców projektu, wiele potrzebnych funkcji nie jest dostępnych lub mają istotne ograniczenia.

Wszystkie badane produkty miały zaimplementowane główne funkcje, takie jak tworzenie wielu projektów z podstawowymi informacjami w przestrzeni roboczej, podstawowe udogodnienia w zarządzaniu zespołem i zapewnianie przypisywania ról i ograniczeń w pracy. Obsługiwane były również powiadomienia dla uczestników w różnej lokalizacji dla szybkiej reakcji i kontroli. Przy intuicyjnym rozumieniu interfejsu zostało zauważone indywidualne podejście do każdego produktu, niektóre programy posiadały podobny interfejs do powszechnie znanych programów lub umożliwiały dostosowanie układu interfejsu do wymagań użytkownika i różnych form instrukcji.

Wykonana analiza porównawcza potwierdziła słuszność postawionych hipotez badawczych. Wyniki badań pokazały, że efektywna praca personelu w ramach jednego projektu, szybkie współdziałanie ze sobą, utrzymanie odpowiedniego poziomu motywacji i zainteresowania wprost proporcjonalnie wpływają na sukces projektu w przyszłości. Co więcej dobór odpowiedniego narzędzia zgodnie z wymaganiami kierownictwa, pozwala pomóc kontrolować postęp projektu jako całości, szybko reagować na problemy lub zmiany w trakcie pracy, podsumować wykonaną pracę, co

przynosi maksymalne korzyści zarówno menedżerom, tak i wykonawcom. Elastyczne ustawienia, w tym dostęp do wersji mobilnej, pozwala zmaksymalizować efektywność pracy nad zadaniami, a szybki przegląd wykonanej pracy, pozwala zrealizować wszystkie wymagania wykonawców, co ułatwia pracę łatwiejszą i pozwala na dostęp z dowolnego miejsca na świecie.

## Literatura

- [1] J. Mulki, F. Bardhi, F. Lassk, J. Nanavaty-Dahl, Set up Remote Workers to Thrive, 2009, [https://www.researchgate.net/publication/264844669\\_Set\\_Up\\_Remote\\_Workers\\_to\\_Thrive](https://www.researchgate.net/publication/264844669_Set_Up_Remote_Workers_to_Thrive), [25.05.2020].
- [2] Przyszłość pracy zdalnej, <https://www.apa.org/monitor/2019/10/cover-remote-work>
- [3] Zespół wirtualny pochodzenie, definicja i zakres, <https://www.managementstudyguide.com/virtual-team.htm> [25.05.2020].
- [4] M. Con, Niestandardowe historie: elastyczny rozwój oprogramowania, St. Petersburg, 2019.
- [5] Definicja modelu integracyjnego, <https://www.qalight.com.ua/baza-znaniy/iterativnaya-model-iterative-model>, [25.05.2020].
- [6] V. Pirogov, Systemy informacyjne i bazy danych: organizacja i projektowanie, BHV-Petersburg, 2009.
- [7] Yu. Izbachkov, V. Petrov, A., Vasiliev, I. Telina, Systemy informacyjne: Podręcznik dla szkół średnich, wydanie trzecie, St. Petersburg, 2011.
- [8] Miejsce testowania w modelach cyklu tworzenia oprogramowania, <https://www.testerzy.pl/artykuly/miejsce-testowania-w-modelach-cyklu-tworzenia-oprogramowania>, [25.05.2020].