

Krzysztof KUBIAK\*, Magdalena Krystyna WYRWICKA\*\*

## ZARZĄDZANIE PROGRAMEM UTRZYMANIA APLIKACJI MULTIMEDIALNEJ – STUDIUM PRZYPADKU

DOI: 10.21008/j.0239-9415.2020.081.08

Celem artykułu jest diagnoza czynników krytycznych wpływających na zarządzanie programem utrzymania aplikacji multimedialnej w przedsiębiorstwie średniej wielkości. Badanie przeprowadzono z wykorzystaniem metody studium przypadku, obserwacji uczestniczącej, analizy dokumentacji i wywiadów bezpośrednich. Zaprezentowane rezultaty pozwalają lepiej zrozumieć procesy zachodzące w przedsiębiorstwie podczas realizacji programu utrzymania aplikacji multimedialnej. W artykule przedstawiono zidentyfikowane czynniki krytyczne oraz zaproponowano zmiany pozwalające je zneutralizować.

**Słowa kluczowe:** zarządzanie programem, zarządzanie projektami, utrzymanie aplikacji, zespoły programistyczne, zarządzanie technologią

### 1. WPROWADZENIE

Ciągły rozwój technologii, dynamicznie zmieniające się trendy rynkowe oraz potrzeba rozwoju aplikacji wymagają stosowania szczegółowych programów dotyczących ich utrzymania. W skład takich programów wchodzi konkretne projekty, ściśle związane z utrzymaniem aplikacji, których realizacja jest niezbędna z ekonomicznego i technologicznego punktu widzenia.

W literaturze naukowej można odnaleźć wiele publikacji związanych z zagadnieniami zarządzania projektami (zadaniami jednokrotnymi, ryzykownymi, bardzo złożonymi) i zarządzania programami projektów. Jednak problem utrzymania i ade-

---

\* Doktorant Politechniki Poznańskiej, Wydział Inżynierii Zarządzania, ORCID: 0000-0003-2732-1587.

\*\* Politechnika Poznańska, Wydział Inżynierii Zarządzania, ORCID: 0000-0002-9513-7354, ResearcherID: N-6037-2014.

kwatnego aktualizowania oprogramowania – pomimo wagi problemu zarządzania programami utrzymania aplikacji – nie jest rozważany w sposób kompleksowy i nie powstały całościowe publikacje, w których porusza się tę tematykę.

Artykuł ma charakter teoretyczno-empiryczny. Na podstawie badań literatury przedmiotu odniesiono się do pojęcia projektu i zarządzania projektami, a następnie przedstawiono stan wiedzy związany z zarządzaniem programami. W dalszej części zaprezentowano specyfikę zarządzania projektami informatycznymi, zwracając uwagę na szczególne cechy projektów i programów projektów informatycznych. Scharakteryzowano również problem utrzymania aplikacji multimedialnej będący przedmiotem dalszej części pracy.

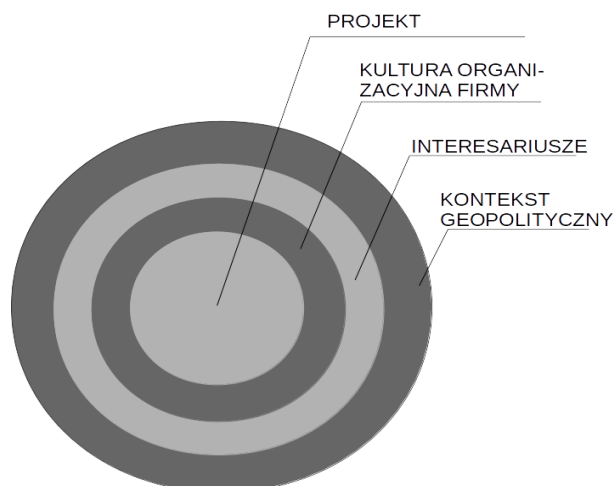
W części empirycznej dokonano opisu metodyki badawczej – przygotowań oraz realizacji studium przypadku z wykorzystaniem obserwacji uczestniczącej i wywiadów bezpośrednich. Ponieważ opracowanie jest oparte na studium przypadku, przedstawiono uzasadnienie wyboru przedsiębiorstwa, w którym przeprowadzono badanie. W punkcie 4 przeanalizowano wyniki badań: obserwacji uczestniczącej, przeglądu dokumentacji projektowej oraz wywiadów bezpośrednich. Artykuł kończy się prezentacją wniosków oraz propozycji zmian w zarządzaniu programem utrzymania aplikacji badanego przedsiębiorstwa.

Celem badania jest poznanie krytycznych aspektów zarządzania programem utrzymania aplikacji multimedialnej w przedsiębiorstwie średniej wielkości. Celem uzupełniającym o charakterze poznawczym jest przedstawienie praktycznego przypadku realizacji programu utrzymania aplikacji multimedialnej w badanej firmie.

## **2. BADANIE LITERATUROWE**

### **2.1. Projekty i zarządzanie projektami**

Każdego roku zmniejsza się liczba powtarzalnych zadań wykonywanych przez aktywnie działające na rynku firmy (Prusak, Wyrwicka, 1997). Globalizacja, zmienne warunki otoczenia czy niepewność towarzysząca prowadzonym przedsięwzięciom determinują coraz powszechniejsze wykorzystanie metod zarządzania projektami w codziennej działalności gospodarczej (Głodziński, 2017). Rosną również wymagania stawiane przed nowo powstającymi projektami, związane z coraz większymi zmianami zachodzącymi w otoczeniu projektu (Wirkus, 2013, 9). Do otoczenia projektu należy zaliczyć m.in.: kulturę organizacyjną przedsiębiorstwa, interesariuszy i kontekst polityczny (rys. 1) (Bodych, Spałek, 2012, 20–21). Aby zakładane cele mogły być osiągnięte, niezbędne jest stosowanie odpowiednich praktyk i metod zarządzania projektami – tradycyjnych (hierarchicznych), czy też zwinnych (typu *agile*) (Kisielnicki, 2016b).



Rys. 1. Otoczenie projektu wpływające na jego realizację.  
Opracowanie własne na podstawie: Bodych, Spalek, 2012

W literaturze przedmiotu można odnaleźć wiele definicji projektu. Według Wysockiego i MacGary'ego (2005, 47) projektem jest sekwencja niepowtarzalnych, złożonych i związanych ze sobą zadań, wsparta dodatkowo określonym celem, ograniczonym budżetem i określonymi wymaganiami. Podobną definicję projektu można znaleźć w publikacji Turnera i Müllera (2003, 7), którzy definiują projekt jako tymczasową organizację, w której zasoby są wykorzystywane do realizacji unikatowych, niepowtarzalnych przedsięwzięć, wykonywanych w warunkach niepewności.

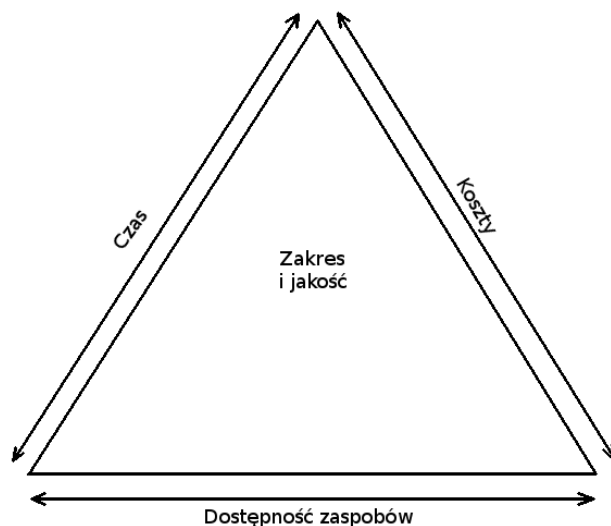
Inne definicje projektu nie odbiegają znacząco od zaprezentowanych powyżej. Zgodnie z metodyką PRINCE2 projekt jest tymczasową organizacją umożliwiającą wytworzenie jednego lub większej liczby produktów końcowych zgodnie z określonym uzasadnieniem biznesowym (Axelos, 2012). Podobną definicję projektu proponuje Skalik (2009, 14), definiując go jako „złożone, niepowtarzalne w zakresie koncepcji i wykonania przedsięwzięcie, realizowane jednorazowo i tymczasowo”.

Bardziej rozbudowaną definicję publikują Jasińska i Szapiro (2014), którzy proponują, aby definiować projekt jako „niepowtarzalne, tymczasowe, interdyscyplinarne przedsięwzięcie, względnie odseparowane od podstawowej działalności oraz mające określony cel, koszt i czas realizacji, które przebiega w sposób sekwencyjny, zespołowy oraz z uwzględnieniem oczekiwanej jakości i wymagań klienta” (Jasińska, Szapiro, 2014, 27). Z kolei według definicji projektu opublikowanej przez Project Management Institute (PMI) traktuje się projekt jako organizację, zauważając, że w każdym przypadku są określone początek i koniec projektu (PMI, 2008, 5). Podobnie jak w pozostałych definicjach, podkreśla się w niej unikalność, cel i rezultaty oraz tymczasowość. PMI zwraca ponadto uwagę, że tymczasowość nie

oznacza, iż okres trwania projektu musi być krótki, a jedynie, że powinien być wyraźnie określony.

Zarządzanie projektami jest więc dynamicznym procesem, który charakteryzuje się wysokim stopniem złożoności. Efektem tego procesu powinno być doprowadzenie projektu do ukończenia go zgodnie z zakładanymi kryteriami dotyczącymi jakości, ceny i terminu (Prusak, Wyrwicka, 1997, 18–20). Na kryteria ceny i terminu zwracają uwagę również Harrison i Lock (2016), wspominając w swojej publikacji o granicach czasowych i kosztowych.

Zarządzanie projektami można ponadto zdefiniować jako wykorzystanie odpowiednich metod i technik w celu doprowadzenia projektu do końca zgodnie z poczynionymi założeniami (Skalik, 2009, 19). Podobną definicję zarządzania projektami można znaleźć w publikacji PMI (2008, 6–7); zgodnie z nią zarządzanie projektami jest odpowiednim stosowaniem umiejętności oraz wykorzystaniem wiedzy i wsparcia narzędziowego w celu uzyskania unikalnego produktu, usługi lub rezultatu. Osiągnięcie zakładanych celów lub rezygnację z realizacji ze względu niemożliwość ich osiągnięcia należy traktować jako zakończenie projektu. Według Wysockiego i McGary'ego (2005, 54, 73) zarządzanie projektami to definiowanie istoty projektu, planowanie, organizowanie, kontrolowanie oraz zamykanie projektu w celu sformułowania wniosków na przyszłość z jednoczesnym utrzymaniem założeń w odniesieniu do kosztów, czasu, zasobów i wykonania zadania na oczekiwanym poziomie jakościowym w pełnym jego zakresie. Swoją koncepcję zarządzania projektem autorzy ilustrują w postaci trójkątnego grafu (rys. 2).



Rys. 2. Zależność czas–koszty–zasoby w kontekście jakości.

Źródło: Wysocki, McGary, 2005, 50–51

Przytoczone powyżej definicje projektu i zarządzania nim mają charakter instrumentalny; podczas analizy literatury przedmiotu należy zauważyć, że autorzy definicji w swoich pracach naukowych doprecyzowują pojęcie zarządzania projektami, wskazując również na procesowy charakter zarządzania projektami. Podejścia procesowego można się doszukać np. w publikacji Kerznera (2017), który definiuje zarządzanie projektami jako stosunkowo nową dyscyplinę, charakteryzując je jako sztukę dopasowania odpowiednich technik w celu uzyskania lepszej kontroli i lepszego wykorzystania zasobów. Podobne aspekty zauważają Harrison i Lock (2016). Zwracają oni uwagę, że projekty wymagają często wykorzystania w procedurze zarządzania niesprawdzonych wcześniej wzorców i procesów, co powoduje wzrost ryzyka w realizacji przedsięwzięcia.

## 2.2. Programy i zarządzanie programami

Realizacja wielu projektów oddzielnie przez niezależne komórki organizacyjne może być nieefektywna. Podczas gdy wiele organizacji koncentruje się na zarządzaniu projektami, tworzenie programów stanowi wyższy poziom zarządzania projektami (Dinsmore, Cabanis-Brewin, 2014). Z tego powodu, dążąc do optymalizacji procesów zarządzania, projekty mające wspólne cele długoterminowe grupuje się w tak zwanych programach (PMI, 2008, 9–10). Turner i Müller (2003, 7) definiują program jako tymczasową organizację, w której łączne zarządzanie grupą projektów (zatwierdzonych uprzednio do realizacji) prowadzi do uzyskania produktów wyższej jakości w wyniku występowania procesów synergii.

Podobną definicję programów przytaczają Wysocki i McGary (2005, 50–51), zauważając, że programy są grupami powiązanych ze sobą projektów, charakteryzującymi się głównie określoną kolejnością ich realizacji i skoordynowaniem zarządzania poszczególnymi projektami. Autorzy dostrzegają ponadto, że programy cechują się znacznie szerszym zakresem prac i zawsze znaczną liczbą celów.

Autorzy pracy *A Guide to the Project Management Body of Knowledge* (PMI, 2008) koncentrują się w swojej definicji na tych samych aspektach, podkreślając przede wszystkim dwie cechy programów (PMI, 2013, 9):

- są grupą powiązanych ze sobą projektów,
- skoordynowane zarządzanie tymi projektami przynosi dodatkowe korzyści.

Ponadto autorzy przywołanej powyżej pracy zauważają, że projekt nie musi być elementem żadnego programu, jednak każdy program musi zawierać w sobie projekty. Podkreślają także, że program to grupa powiązanych projektów, podprogramów i działań programowych, których celem jest osiągnięcie dodatkowych korzyści (PMI, 2013).

W innych pozycjach literatury przedmiotu definiuje się program w podobny sposób. Kendrick (2016, 4) zauważa, że na program składa się powiązany wysiłek liderów projektów oraz zespołów pracujących nad tymi projektami. Dinsmore

i Cabanis-Drewin (2014, 4–5) potwierdzają w swojej publikacji powyższe definicje PMI oraz zauważają, że założenia programu mogą być kilkukrotnie modyfikowane podczas jego realizacji, tak aby osiągnąć zakładane cele i zapewnić planowane korzyści. Zwracają także uwagę, że zakres programu może być poszerzany lub zawężany podczas jego realizacji.

Autorzy definicji zarządzania programami zwracają uwagę na aspekty zarządzania projektami, rozszerzając je o możliwość osiągnięcia dodatkowych korzyści. PMI definiuje zarządzanie programem jako zastosowanie w odniesieniu do niego wiedzy, umiejętności, narzędzi i technik w celu spełnienia jego wymagań oraz uzyskania korzyści i kontroli niedostępnej na drodze indywidualnego zarządzania projektami (PMI, 2008, 6–7). Według PMI zarządzanie programami polega na takim dopasowaniu wielu elementów, aby osiągnąć cele programu. Autorzy podkreślają również dodatkowe korzyści, jakimi są zoptymalizowanie lub zintegrowanie kosztów, harmonogramu i wysiłku.

Kendrick (2016, 6–7) zauważa ponadto, że zarządzanie programami wymaga głębokiego zrozumienia procesów synergii i strategii stanowiących podstawy zakładanych celów. Należy zauważyć, że zarządzanie programami jest bardzo podatne na działanie otoczenia i wymaga podejmowania szybkich działań (decyzji), często bez dostępu do pełnej informacji lub porad eksperckich (Dinsmore, Cabanis-Brewin, 2014, 329–330). Autorzy publikacji z zakresu zarządzania programami zwracają również uwagę na nierównomierny stopień złożoności procesu zarządzania programem, który wzrasta, w miarę jak zwiększa się niezależność projektów wchodzących w ich skład (Kendrick, 2016, 8).

### 2.3. Zarządzanie projektami informatycznymi

Procesy tworzenia nowych systemów informatycznych lub procesy modyfikacji systemów istniejących przyjmują bardzo często formę projektów. Projekty informatyczne są jednak specyficzne i odróżniają się od innych projektów przede wszystkim w następujący sposób (Biniek, 2009, 9):

- dotyczą tworzenia aplikacji informatycznych,
- zdecydowana część współpracowników to specjaliści informatycy,
- kierownik projektu jest specjalistą informatykiem.

W przypadku zarządzania projektami informatycznymi należy nastawić się przede wszystkim na działanie w stale zmieniającym się otoczeniu, gdzie każda decyzja wiąże się z dużym ryzykiem (Kisielnicki, 2016a, 109–110). Wystąpienie dodatkowych rodzajów ryzyka, związanych ściśle z projektami informatycznymi, jest mocno akcentowane w literaturze przedmiotu. Pawłowicz (2014, 174) zwraca uwagę na wysoki poziom ryzyka projektów informatycznych. Kisielnicki (2016a, 109–110) zauważa, że w zarządzaniu projektami informatycznymi niektóre zasady

zarządzania projektami występują z większą częstotliwością niż w projektach z zakresu innych dziedzin i zalicza do nich w szczególności:

- zmienność sytuacji,
- konieczność stałych modyfikacji założeń i zachowań,
- pracę w określonych przedziałach czasowych.

Między innymi z opisanych powyżej powodów w pracach nad projektami informatycznymi bardzo często odchodzi się od klasycznych metod zarządzania projektami, kierując się w stronę metod zwinnych (*agile*); należy tu zwrócić uwagę np. na podejście *Scrum*, na programowanie ekstremalne (*extreme programming*<sup>1</sup>), a także na *Future Driven Development*, *Test Driven Development*, *Lean Software Development*, *Dynamic System Development Method* czy *kanban* (Chrapko, 2013; Alahyari, Gorschek, Berntsson Svensson, 2019; Hammarberg, Hubisz, Sundén, 2015). Metody te powstały m.in. jako efekt manifestu *agile*: „Poprzez wytwarzanie oprogramowania oraz pomaganie innym w tym zakresie odkrywamy lepsze sposoby realizowania tej pracy. W wyniku tych doświadczeń zaczęliśmy przedkładać: ludzi i ich wzajemne interakcje (współdziałanie) ponad procedury i narzędzia, działające oprogramowanie nad wyczerpującą dokumentację, współpracę z klientem nad negocjację umów, reagowanie na zmiany nad realizację planów” (Krzemiński, 2014, 21). Należy zauważyć, że twórcy manifestu nie zalecają rezygnacji z wprowadzania procedur, narzędzi czy dokumentacji, jednak wskazują nowe priorytety (Krzemiński, 2014, 21).

Ponieważ w przedsiębiorstwie, którego dotyczy studium przypadku, stosowane są elementy metod programowania ekstremalnego oraz *Scrum*, metody te są poniżej krótko omówione.

Metoda programowania ekstremalnego została zaproponowana przez Becka w latach 90. XX w. (Dajda, Dobrowolski, 2007, 72). Podkreślał on, że przyczyną jej powstania jest podwyższenie jakości wykonywanych projektów oraz poprawa relacji między członkami zespołu (Beck, Anders, 2004). Autorzy kierują się zasadą, że dobre relacje w zespole prowadzą do pozytywnego zakończenia projektów. Programowanie ekstremalne skupia się na aspektach związanych stricte z projektem informatycznym, w szczególności na programowaniu parami, wspólnej własności kodu<sup>2</sup> oraz częstych spotkaniach (Wood, Michaelidos, 2013, 662). W programowaniu ekstremalnym zwraca się również uwagę na przeprowadzanie jednostkowych testów kodu, testów klienckich i testów pierwszego projektu oraz na refaktoryzację (Wood, Michaelidos, 2013, 662).

*Scrum* charakteryzuje się większą koncentracją na procesach zarządzania projektami (Chrapko, 2013; Pouya, Seyyed, 2012, 24), przy czym zakłada się dostarczanie oprogramowania w kolejnych, udoskonalanych wersjach, co umożliwia

---

<sup>1</sup> W literaturze można odnaleźć również zapis: *eXtreme Programming* oraz pochodzący od tego zapisu skrót XP.

<sup>2</sup> Możliwość edycji kodu ma wielu członków zespołu, którzy współtworzą go i wspólnie kontrolują, biorąc na siebie odpowiedzialność za jego jakość.

włączenie się w proces tworzenia oprogramowania przyszłym użytkownikom oraz daje zespołowi możliwość samodzielnej organizacji pracy (Górski, 2010, 20–21). Odpowiedzialna za projekt jest jedna osoba, nazywana właścicielem produktu, która staje się również członkiem zespołu realizującego projekt (Pichler, 2014). Podczas realizacji projektu prace przygotowawcze są ograniczane do minimum, a sam produkt powstaje stopniowo i jest dynamicznie rozwijany również po wydaniu rynkowym. Zwraca się przy tym również uwagę na klientów, którzy otrzymują w pierwszym wydaniu produkt o minimalnych funkcjach, a następnie jego aktualizacje, na które mają wpływ również ich opinie (Pichler, 2014; Chrapko, 2013).

## 2.4. Utrzymanie aplikacji

Zarządzanie programem utrzymania aplikacji<sup>3</sup> multimedialnej ma na celu rozwój produktu i przedłużenie w ten sposób cyklu jego życia (Michalski, 2008). Utrzymanie infrastruktury i aplikacji jest procesem długofalowym, trwającym od chwili przygotowania pierwszego oprogramowania komputerowego oraz zaistnienia jego cyklu życia (Sharif, Mokhtar, Buckley, 2011, 1060). Istnieje wiele definicji utrzymania aplikacji, różniących się m.in. stawianymi przed zespołem celami. Do podstawowych definicji należy zaliczyć m.in. poniższe (The Institute of Electrical and Electronics Engineers, 1998):

- modyfikacja oprogramowania wykonana po jego dostarczeniu w celu utrzymania użyteczności programu komputerowego w zmienionym lub zmieniającym się środowisku – cel: utrzymanie adaptacyjne,
- modyfikacja oprogramowania po wykryciu nieprawidłowości (usterek) – cel: utrzymanie korygujące,
- nieplanowane prace konserwacyjno-naprawcze wykonane w celu utrzymania działającego systemu – cel: utrzymanie awaryjne,
- modyfikacja oprogramowania po jego dostarczeniu w celu poprawienia błędów, poprawy wydajności lub innych elementów, jak również modyfikacja mająca na celu dostarczenie produktu do zmodyfikowanego środowiska – cel: konserwacja oprogramowania.

W literaturze przedmiotu można znaleźć również inne definicje, tworzone często na bazie powyższych. W jednej z nich zakłada się, że utrzymanie oprogramowania rozumiane jako wszystkie czynności wsparcia i inżynierii oprogramowania ma na celu zaspokojenie potrzeb użytkownika (Naciri, Janati Idrissi, 2014). Według jej autorów zmiany te mogą powodować ewolucję istniejących funkcji oprogramowania, jak również mogą to być tylko korekty jego anomalii.

---

<sup>3</sup> W publikacji słowo „aplikacja” jest używane w ściśle określonym znaczeniu: program (oprogramowanie) przygotowany na urządzenia końcowe: komputery osobiste, smartfony oraz dowolne urządzenia z przeglądarką internetową, zapewniające połączenie z siecią Internet oraz spełniające ściśle określone wymagania techniczne.



Utrzymanie oprogramowania zostało zdefiniowane również w normie ISO/IEC/IEEE 12207 jako proces zachodzący podczas modyfikacji kodu i związanych z nim zmian w dokumentacji z powodu błędów, niedostatków, problemów, potrzeb poprawy lub adaptacji (IEEE Computer Society, 2017). W normie tej podkreślono również, że celem procesów utrzymania oprogramowania jest modyfikacja istniejącego systemu z zachowaniem jego integralności.

W literaturze podkreśla się również długotrwałość procesu utrzymania aplikacji (Tarwani, Chug, 2016, 1379). Zauważa się, że największa część kosztów sumarycznych całego projektu informatycznego jest przeznaczana na finansowanie utrzymania aplikacji, a nie na jej zaprojektowanie i wykonanie (Tarwani, Chug, 2016, 1379).

### 3. METODY I PROJEKTOWANIE BADAŃ

#### 3.1. Metodyka badań

Jak zaznaczono we wprowadzeniu, celem prowadzonego badania jest zidentyfikowanie krytycznych aspektów zarządzania programem utrzymania aplikacji na podstawie obserwacji pracy zespołu programistów w okresie trzech miesięcy oraz na podstawie informacji uzyskanych od członków tego zespołu. Dodatkowym celem poznawczym jest analiza oraz ocena efektów pracy zespołu nad realizacją programu utrzymania aplikacji mobilnej.

Ponieważ uzyskanie wystarczającej liczby danych do przeprowadzenia wartościowego badania ilościowego nie jest możliwe, autorzy zdecydowali się skorzystać z metody studium przypadku (Kostka-Bochenek, Lisiecka, 2009). W ramach badań zastosowano ponadto poniższe metody (Apanowicz, 2002, 81–88):

- analizę dokumentacji projektowej,
- obserwację uczestniczącą, zaliczaną do grupy badań monograficznych,
- wywiady bezpośrednie.

Firmę wybrano na potrzeby badania z zastosowaniem doboru celowego (Czajka, 2006). Było to przedsiębiorstwo średniej wielkości (na podstawie Rozporządzenia Komisji (WE) 800/2008 z dnia 6 sierpnia 2008 r.) z Wielkopolski, od blisko 30 lat zajmujące się tworzeniem programów komputerowych wymagających długotrwałego wsparcia, rozwijanych lub zastępowanych nowszymi wersjami. Obecnie przedsiębiorstwo wspiera równolegle oprogramowanie dla systemów Windows, Android i iOS oraz serwis *online*. Od początku istnienia przedsiębiorstwo wykonuje oprogramowanie wewnętrznie, w marginalnym zakresie zlecając zadania podmiotom zewnętrznym. Zespół pracuje stacjonarnie w siedzibie firmy, dzięki czemu obserwacja prac jest pełna.

### 3.2. Postępowanie badawcze

Badanie wykonano w czwartym kwartale 2017 r. W jego ramach przeanalizowano dokumentację programów utrzymania aplikacji na czwarty kwartał 2017 r. oraz zapoznano się dokładnie ze stawianymi przed zespołem celami.

Dodatkowo przeanalizowano listę zadań wykorzystywaną do planowania prac, prowadzoną w formie interaktywnej aplikacji Redmine, wspomagającej zarządzanie projektami. Dostęp do listy zadań mają wszyscy pracownicy badanego przedsiębiorstwa. Podczas analizy zwrócono uwagę przede wszystkim na specyfikację oraz na szczegółowość i jasność stawianych przed pracownikami wymagań.

Wszystkie zadania zostały opisane precyzyjnie; wskazano w nich kryteria akceptacji zadania oraz wymagane projekty graficzne.

Na podstawie analizy dokumentacji wyodrębniono projekty wchodzące w skład prowadzonego programu utrzymania aplikacji multimedialnej w badanym okresie:

- „Nowy katalog produktów” (NKP), w którym założono cztery etapy:
    - I – przygotowanie koncepcji i makiety technologicznej (7 zadań składowych zdefiniowanych w systemie),
    - II – przygotowanie prototypu działającego w środowisku testowym (9 zadań składowych zdefiniowanych w systemie),
    - III – wykonanie zadania w środowisku testowym (11 zadań składowych zdefiniowanych w systemie),
    - IV – wdrożenie produkcyjne (4 zadania składowe zdefiniowane w systemie);
  - „Synteza mowy w kursach użytkowników” (SMU), która została podzielona na trzy etapy:
    - I – przygotowanie koncepcyjne i przygotowanie działającego prototypu (18 zadań składowych zdefiniowanych w systemie),
    - II – wdrożenie dla platformy Android (8 zadań składowych zdefiniowanych w systemie),
    - III – wdrożenie dla pozostałych wybranych platform (12 zadań składowych zdefiniowanych w systemie);
  - „Przebudowa mechanizmu tłumaczeń i umożliwienie pełnej obsługi na urządzeniach mobilnych” (PMT) (5 zadań składowych zdefiniowanych w systemie).
- W przypadku każdego z tych projektów precyzyjnie określono w czasie punkty rozpoczęcia prac nad kolejnymi etapami oraz zadaniami przypisanymi do danego etapu. Na tej podstawie określono oczekiwaną realizację każdego z projektów w czwartym kwartale 2017 r.:
- etapy I i II projektu „Nowy katalog produktów”,
  - etap I i 50% etapu II projektu „Synteza mowy w kursach użytkowników”,
  - projekt „Przebudowa mechanizmu tłumaczeń i umożliwienie pełnej obsługi na urządzeniach mobilnych”.

Dokładniejsze informacje na temat realizacji etapów prac projektowych przedstawiono w tab. 1.

Tabela 1. Założenia dotyczące wykonalności projektu w wybranym okresie

Projekt	Oczekiwany stopień wykonania	Oczekiwany stopień wykonania w kolejnych etapach (oczekiwana liczba zadań wykonanych/liczba zadań planowanych)
NKP	50%	etap I – 100% (5/5)
		etap II – 100% (10/10)
		etap III – 0% (0/11)
		etap IV – 0% (0/4)
SMU	50%	etap I – 100% (18/18)
		etap II – 50% (4/4)
		etap III – 0% (0/12)
PMT	100%	100% (5/5)

Opracowanie własne na podstawie badań.

Należy zauważyć, że poza pracami w ramach wymienionych projektów wchodzących w skład badanego programu te same osoby (członkowie działu IT) brały również udział w:

- 1) innych projektach rozwojowych prowadzonych przez firmę, niezwiązanych bezpośrednio z programem utrzymania aplikacji;
- 2) bieżącej działalności przedsiębiorstwa, która w przypadku analizowanego zespołu polega w dużej mierze na:
  - rozwiązywaniu problemów technicznych i technologicznych na potrzeby wewnętrzne,
  - rozwiązywaniu problemów technicznych i technologicznych na potrzeby wybranych zewnętrznych partnerów biznesowych,
  - wsparciu merytorycznym w zakresie technologii dla przygotowywanych planów rozwojowych,
  - pomocy technicznej w nagłych sytuacjach i podczas awarii,
  - zarządzaniu infrastrukturą technologiczną przedsiębiorstwa,
  - prowadzeniu ostatniej linii wsparcia klienta,
  - poprawie błędów wykrytych we wspieranym oprogramowaniu.

Badane przedsiębiorstwo nie wdrożyło w pełni żadnej metody zwinnego zarządzania. W przeszłości próbowano wykorzystywać programowanie ekstremalne, jednak następnie tego zaniechano. W kolejnym okresie firma próbowała wdrożyć metodę *Scrum*, jednak ze względu na konieczność zaangażowania dodatkowych członków zespołu z przyczyn biznesowych nie wdrożono jej w całości. W działaniach zespołu można jednak zaobserwować występowanie elementów obydwu podejść, takich jak m.in.:

- wspólna własność kodu,
- częste spotkania,
- testy jednostkowe,
- systematyczna refaktoryzacja kodu,
- przyrostowe tworzenie oprogramowania w kolejnych wydaniach,
- samoorganizacja prac zespołowych,
- ciągła kontrola informacji zwrotnej od klienta.

W ostatnim etapie badania przeprowadzono wywiady bezpośrednie z pracownikami odpowiedzialnymi za wykonanie kolejnych prac. Wywiad był ukierunkowany na przyczyny ewentualnych niepowodzeń w celu poznania możliwie największej liczby krytycznych aspektów zarządzania programem utrzymania aplikacji.

## **4. WYNIKI BADANIA I DYSKUSJA**

### **4.1. Wyniki obserwacji uczestniczącej**

Podczas badania autorzy codziennie zapisywali obserwacje dotyczące prac zespołu. Na tej podstawie zwrócono uwagę na opisane poniżej sytuacje zachodzące w zespole.

- Pracownicy punktualnie i sumiennie pojawiają się w miejscu pracy; nie ma trudności z utrzymaniem dyscypliny zespołu projektowego. W przypadku ewentualnych niedyspozycji zadania są sprawnie przekazywane innym członkom zespołu. Wszyscy pracownicy dobrze rozumieją swoje role oraz zakresy kompetencji. Bardzo rzadko wchodzi w zakres kompetencji innej osoby.
- Atmosfera podczas spotkań zespołu jest swobodna i oceniana przez autora jako przyjacielska. Pracownicy jednak często zwracają uwagę na zbyt dużą liczbę spotkań, które dezorganizują im plan dnia pracy. Zauważają również, że poranne spotkanie dotyczące organizacji dnia pracy często znacznie przekracza zaplanowany czas 20 minut. W trakcie badania zdarzały się spotkania trwające 40 minut, a dwa trwały blisko 70 minut.
- W trakcie pracy pracownicy często i chętnie korzystają z wiedzy innych członków zespołu, w szczególności z wiedzy specjalistycznej, związanej z historią rozwoju danego produktu. Bardziej doświadczeni pracownicy sprawiają wrażenie osób z radością dzielących się wiedzą i wskazują historyczne konotacje prowadzące do wybranych konkretnych rozwiązań.
- W zespole często pojawiają się narzekania na braki w dokumentacji lub na jej nierzetelność. Duża rotacja pracowników powoduje, że niektóre braki w dokumentacji wymagają poświęcenia dodatkowego czasu na ocenę badanej sytuacji.
- Można zauważyć, że pracownicy innych działów bez skrupowania przerywają pracę osobom z działu IT, gdy pojawiają się jakiegokolwiek problemy. Poruszają

- różne tematy i oczekują od nich szybkiej odpowiedzi lub wykonania dodatkowych zadań. Często powoduje to zmianę planu dnia pracy, co bywa dla pracowników frustrujące.
- Wykonawcy projektu zwracają również uwagę na komplikację w wykonywaniu niektórych zadań, związaną z daną architekturą aplikacji, która wymaga wykonywania zadań w określonej kolejności. Ze względu na rozdrobnienie technologiczne nieobecność pojedynczych pracowników wpływa bezpośrednio na łańcuch prac w ramach projektu. Na problem ten zwracali uwagę wszyscy zaangażowani w projekt pracownicy podczas porannych spotkań.
  - Pracownicy akcentują zbyt szeroki zakres realizowanego projektu. W ocenie większości z nich projekt jest zbyt duży w stosunku do liczby zaangażowanych osób, co czyni ich pracę trudniejszą oraz mniej dokładną.
  - Pracownicy zwracali również uwagę na trudności z dostępem do specjalistów w konkretnych dziedzinach.

#### **4.2. Analiza stopnia realizacji projektów wchodzących w skład badanego programu**

W trakcie prac badani pracownicy precyzyjnie określali wykonawców każdego zadania oraz sumiennie oznaczali statusy wykonywanych zadań, co umożliwiło zebranie danych i wykonanie analizy stopnia realizacji projektu.

Procentowe wyniki prezentują ilościową analizę zadań wchodzących w skład konkretnego etapu. Nie uwzględniają one estymowanego czasu pracy oraz realnego czasu pracy nad realizacją zadań wchodzących w skład konkretnego etapu. Wszystkie dane dostępne były w systemie zarządzania projektami.

W analizach celowo nie uwzględniono estymowanego czasu pracy oraz rzeczywistego czasu pracy ze względu na niepełne odnotowywanie czasu pracy przez pracowników w systemie. Firma nie narzuciła pracownikom obowiązku notowania czasu wykonywania zadań, w związku z czym robią to tylko nieliczni.

W badaniu nie analizowano też liczby zadań związanych z innymi projektami rozwojowymi i zadań dotyczących bieżącej działalności przedsiębiorstwa oraz ich estymacji czasowej. Zadania te zostały celowo pominięte, ponieważ przedsiębiorstwo nie prowadzi ścisłej kontroli tych procesów, co czyniłoby dane niepełnymi. Kadra zarządzająca wie, że liczba prac związanych z bieżącą działalnością oraz ich zakres są trudne do oszacowania, a błędy szacunkowe w przeszłości były bardzo duże. Z tego powodu wprowadzono wykonywanie zadań bieżących w formule wolnego wyboru: pracownicy w wolnym czasie pobierają zadania z kolejki według własnego uznania z uwzględnieniem przypisanych do zadań priorytetów. Wykorzystywane w przedsiębiorstwie podejście wymaga dokładnego planowania projektów oraz całego programu utrzymania aplikacji. Program ten jest tworzony z wykorzystaniem danych zgromadzonych w poprzednich okresach. Dzięki takiemu podejściu jest możliwe oszacowanie realnych terminów realizacji zadań.

Po zakończeniu badania przeprowadzono analizę efektów prac na dzień 31 grudnia 2017 r. na podstawie informacji zawartych w systemie informatycznym. Uzyskane wyniki przedstawiono w tabeli 2.

Tabela 2. Stan projektów po okresie trzech miesięcy

Projekt	Osiągnięty stopień wykonania	Osiągnięty stopień wykonania w kolejnych etapach (liczba zadań wykonanych /liczba zadań planowanych)
NKP	40%	etap I – 100% (5/5)
		etap II – 60% (6/10)
		etap III – 0% (0/11)
		etap IV – 0% (0/4)
SMU	42,5%	etap I – 100%
		etap II – 25%
		etap III – 0%
PMT	100%	100%

Opracowanie własne na podstawie badań.

Korzystając z informacji zgromadzonych w systemie zarządzania projektami, udało się zidentyfikować powody opóźnień prac w ramach projektów NKP i SMU.

Projekt „Nowy katalog produktów” został zrealizowany w 40% (zakładany stopień wykonania: 50%). Zespół nie mógł wykonać prac w zakładanym zakresie z kilku przyczyn. W projekcie wykorzystano zupełnie nową technologię, z którą pracownicy nie mieli wcześniej do czynienia. Z tego powodu wiele nowych problemów stanowiło dla nich duże wyzwania. Rozwiązania problemów oraz same problemy były odnotowywane w systemie zarządzania projektem. Praca na tym etapie wymagała jednak dodatkowych konsultacji oraz niezaplanowanego przeznaczenia czasu na lekturę dokumentacji technicznej.

Podczas prac nad prototypem zespół kilkakrotnie stanął przed koniecznością podjęcia decyzji dotyczącej sposobu wykonania konkretnego zadania i rozwiązania występującego problemu. Każdą tego typu decyzję poddawano dyskusji i jeśli było to możliwe, symulowano możliwe warunki brzegowe w celu uzyskania pełnego obrazu przewidywanych sytuacji.

W fazie prototypowania zakładano umieszczenie prototypu na serwerze zewnętrznym, zarządzanym przez obsługującą go firmę, jednak częstotliwość zmian i niezbędna gruntowna analiza wydajności wymagały skonfigurowania serwera na maszynie znajdującej się w przedsiębiorstwie. Dodatkowy czas poświęcony na konfigurację nie był uwzględniony w szacowanym czasie realizacji projektu, jed-

nak liczba zadań się nie zmieniła – zadanie to zastąpiło zadanie planowane do zlecenia jednostce zewnętrznej i wykonano je w ramach firmy.

Projekt „Synteza mowy w kursach użytkowników” został zrealizowany w 42,5% (zakładany stopień wykonania: 50%). W dokumentacji wykonania zadań można było odnaleźć dwie podstawowe przyczyny opóźnienia. Pierwszą było znaczne wydłużenie etapu I ze względu na zmianę wykonawcy zadania. W rezultacie duża część kodu została poddana refaktoryzacji<sup>4</sup>. Drugą przyczyną było duże zaangażowanie czasowe osób odpowiedzialnych za interfejs graficzny w realizację innych projektów. Zbyt duża liczba krytycznych zadań bieżących powodowała opóźnienie i czasowe zablokowanie prac, wpływając bezpośrednio na stopień wykonania zadania.

Projekt przebudowy mechanizmu tłumaczeń został ukończony w określonym trzymiesięcznym terminie, jednak warto zauważyć, że pierwotnie jego realizacja planowana była na koniec października, a finalnie komponent pojawił się w wersji produkcyjnej pod koniec listopada.

Podczas prac nad tym zadaniem pojawiły się problemy związane z definiowaniem wymagań oraz dużym skomplikowaniem istniejącego systemu. Były to główne przyczyny, dla których termin ukończenia zadania w badanym projekcie został znacząco przekroczony.

### 4.3. Analiza wyników wywiadów bezpośrednich

Kończąc badanie w pierwszych dwóch tygodniach stycznia 2018 r., przeprowadzono wywiady bezpośrednie z członkami zespołu. Wywiady miały na celu wskazanie krytycznych aspektów powodujących powstałe opóźnienia.

Pracownicy w większości potwierdzali obserwacje opisane w p. 4.1. Wszyscy zwracali uwagę na destrukcyjny wpływ zadań bieżących na realizację projektów. Podkreślali przy tym, że liczba tych zadań ciągle rośnie. Zaznaczono również, że stosunkowo duża liczba zadań bieżących wymaga natychmiastowej reakcji, przez co trzeba wielokrotnie odrywać się od zadań wykonywanych w ramach projektów. Niektórzy sygnalizowali, że na powrót do przerwane zadania projektowego trzeba poświęcić dodatkowy czas.

Pracownicy zwracali uwagę również na ograniczenia liczebności kadry. Jako przyczyny opóźnień wskazywali nieplanowane urlopy oraz zwolnienia lekarskie. Podkreślali, że liczniejsza kadra pozwoliłaby lepiej planować zadania i unikać sytuacji, w których nieobecność jednego pracownika wpływa na wykonanie zadań. W badanym przypadku jeden z pracowników był nieobecny przez okres trzech

---

<sup>4</sup> Refaktoryzacja (z ang. *refactoring*) – w programowaniu: zmiana w budowie kodu, która nie wpływa bezpośrednio na żadną z jego funkcji, a jedynie na czytelność i organizację (strukturę) kodu.

tygodni, na który składały się: nieplanowany wcześniej tygodniowy urlop w listopadzie oraz dwutygodniowa absencja spowodowana chorobą w grudniu.

W wyniku przeprowadzonych wywiadów udało się wykryć jedną z kluczowych przyczyn opóźnienia w realizacji projektu syntezy mowy, związaną z budową prototypu. Pracownicy wielokrotnie zmieniali podejście i poświęcili dużo czasu na zaprojektowanie rozwiązania. W pierwszych dniach stycznia, mimo uprzedniej gruntownej analizy, wykonawcy musieli jednak ponownie przeprojektować powstałe rozwiązanie. W tym kontekście zwracali szczególną uwagę na brak ekspertów, którzy mogliby wykorzystać swoją wiedzę do zaprojektowania lepszego rozwiązania już za pierwszym razem.

Pracownicy zwracali w wywiadach uwagę na dobrą atmosferę pracy w zespole oraz współpracę. Podkreślali przy tym, że realizowany projekt pozytywnie wpływał na ich rozwój osobisty. Dostrzegali też rozwój produktu, do którego się przyczynili, i widzieli w nim efekt swojej pracy. Mimo wszystko krytycznie spoglądali na powstający produkt, wskazując błędy w realizacji projektów i założeń. Jako możliwą przyczynę opóźnień wskazywali zbyt małe wykorzystanie testów weryfikujących. Zdawali sobie sprawę, że wykonanie testów kodu (automatycznych, jednostkowych) wydłużyłoby prace nad projektem, jednak ich zdaniem wykonanie produktu bez testów nie było finalnie krótsze, ponieważ należało poświęcić dużo czasu na wykrycie błędów oraz wprowadzanie poprawek.

Pracownicy zwracali również uwagę na niesprzyjające warunki pracy, co było związane z liczbą pracowników zgromadzonych w poszczególnych pomieszczeniach. Uważają oni, że hałas utrudniał im skupienie się na zadaniu oraz był przyczyną przerw w pracy.

Niektórzy pracownicy wskazywali również na nieprecyzyjne zdefiniowanie stawianych przed nimi wymagań technicznych. Jako przyczynę takiego stanu rzeczy podawali problemy w komunikacji między osobami z działów nietechnicznych i technicznych.

## 5. PODSUMOWANIE I WNIOSKI

Mimo dużego stopnia złożoności problemów, z jakimi badane przedsiębiorstwo musi sobie radzić podczas prac nad programem utrzymania aplikacji multimedialnej, badane przedsiębiorstwo zrealizowało ponad 80% programu zakładanego w czwartym kwartale 2017 r.

Cel prowadzonego badania został osiągnięty i udało się wskazać krytyczne aspekty zarządzania programem utrzymania aplikacji multimedialnej w przedsiębiorstwie średniej wielkości. Należą do nich:

- zła sytuacja kadrowa przedsiębiorstwa oraz brak odpowiedniej zastępowalności poszczególnych pracowników w przypadku wydarzeń o charakterze losowym,



- brak odpowiedniego planowania zadań z uwzględnieniem czasu poświęconego na wykonanie innych zadań o charakterze bieżącym w przypadku zespołu łączącego prace nad programem z bieżącym wsparciem działalności,
- niezapewnienie warunków pozwalających na wykonywanie pracy w możliwie największym komforcie oraz ciszy,
- niestaranne utrzymywanie dokumentacji technicznej,
- brak wdrożonego systemu transferu wiedzy w przypadku zmian kadrowych związanych z rotacją pracowników.

Opierając się na wynikach przeprowadzonego badania, przedsiębiorstwo powinno w przyszłości zwracać większą uwagę na dokładność analizy pracochłonności zadań oraz starać się zredukować liczbę zadań bieżących. Firma powinna usprawnić komunikację z innymi działami, mając na celu minimalizację zadań wymagających natychmiastowej reakcji (zadania określane przez pracowników w wywiadach jako silnie zakłócające plany pracy). Oczekiwanym przez pracowników posunięciem powinno być też zatrudnienie dodatkowych pracowników oraz zapewnienie wiedzy eksperckiej w tworzonym zespole.

## LITERATURA

- Alahyari, H., Gorschek, T., Berntsson Svensson, R. (2019). An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations. *Information and Software Technology*, 105, 78–94.
- Apanowicz, J. (2002). *Metodologia ogólna*. Gdynia: Wydawnictwo Diecezji Peplińskiej Bernardinum.
- Axelos (2012). *Managing successful projects with Prince2*. London: TSO.
- Beck, K., Anders, C. (2004). *Extreme Programming Explained*. Boston: Addison Wesley.
- Biniek, Z. (2009). *Wybrane elementy zarządzania projektem informatycznym*. Warszawa: Wyższa Szkoła Finansów i Zarządzania w Warszawie, Vizja Press & IT.
- Bodych, M., Spalek, S. (2012). *PMO Praktyka zarządzania projektami i portfelem projektów w organizacji*. Gliwice: Helion.
- Chrapko, M. (2013). *SCRUM o zwinnym zarządzaniu projektami*. Gliwice: Helion.
- Czakon, W. (2006). Łabędzie Poppera – case studies w badaniach nauk o zarządzaniu. *Przegląd Organizacji*, 9 (80), 9–12.
- Dajda, J., Dobrowolski, G. (2007). Współczesne metody systemowego wytwarzania oprogramowania. *Automatyka / Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie*, 11, 1–2, 71–79.
- Dinsmore, P.C., Cabanis-Brewin, J. (2014). *The AMA Handbook of Project Management*. New York–Atlanta–Brussels–Chicago–Mexico City–San Francisco–Shanghai–Tokyo–Toronto–Washington, D.C.: AMACOM Division of American Management Association International.

- Głodziński, E. (2017). *Efektywność w zarządzaniu projektami*. Warszawa: Polskie Wydawnictwo Ekonomiczne.
- Górski, T. (2010). Zwinność i dyscyplina w podnoszeniu efektywności zespołów projektowych. *Biuletyn Instytutu Systemów Informatycznych*, 6, 19–26
- Hammarberg, M., Hubisz, J., Sundén, J. (2015). *Kanban: zobacz, jak skutecznie zarządzać pracą!* Gliwice: Helion.
- Harrison, F., Look, D. (2016). *Advanced Project Management: A Structured Approach*. New York: Routledge.
- IEEE Computer Society (2017). ISO/IEC/IEEE 12207. Systems and software engineering – Software life cycle processes.
- Jasińska, K., Szapiro, T. (2014). *Zarządzanie procesami realizacji projektów w sektorze ICT*. Warszawa: Wydawnictwo Naukowe PWN.
- Kendrick, T. (2016). *How to Manage Complex Programs: High-Impact Techniques for Handling Project Workflow, Deliverables, and Teams*. New York–Atlanta–Brussels–Chicago–Mexico City–San Francisco–Shanghai–Tokyo–Toronto–Washington, D.C.: AMACOM.
- Kerzner, H.R. (2017). *Project management. A system approach to planning, scheduling and controlling*. New Jersey: Wiley.
- Kisielnicki, J. (2016a). Kierownik projektu informatycznego i jego rola w zespole realizującym projekt. *Studia Informatica Pomerania*, 4(42).
- Kisielnicki, J. (2016b). Zarządzanie projektami badawczo-rozwojowymi – system komunikacji. *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu*, 421, 240–242.
- Kostka-Bochenek, A., Lisiecka, K. (2009). Case study research jako metoda badań naukowych. *Przegląd Organizacji*, 10(837), 25–29.
- Krzemiński, M. (2014). *Agile. Szybciej, łatwiej, dokładniej*. Gliwice: Helion.
- Michalski, R. (2008). Jakość użytkowa w procesie wytwarzania oprogramowania. *Operations Research & Decisions*, 4, 71–88.
- Naciri, S., Janati Idrissi, M. (2014). Third-Party Application Maintenance Management. *International Journal of Computer Applications*, 103(6).
- Pawłowicz, M. (2014). Metoda efektywnego zarządzania projektem teleinformatycznym w instytucji publicznej. *Zeszyty Naukowe Wyższej Szkoły Oficerskiej Wojsk Lądowych im. gen. T. Kościuszki*, 2(172).
- Pichler, R. (2014). *Zarządzanie projektami ze scrumem. Twórz produkty, które pokochają klienci*. Gliwice: Helion.
- Pouya, F., Seyyed, H. (2012). An Agile Method for E-Service Composition. *International Journal of Computer Science Issues*, 9(1).
- PMI (2008). *A Guide To The Project Management Body of Knowledge (PMBOK Guide)*. Atlanta: Project Management Institute.
- PMI (2013). *Standard for Program Management*, 3<sup>rd</sup> ed. Atlanta: Project Management Institute.
- Prusak, W., Wyrwicka, M. (1997). *Zarządzanie projektami*. Poznań: Zachodnie Centrum Organizacji sp. z o.o.

- Sharif, K., Mokhtar, M., Buckley, J. (2011). Open source programmers' information seeking during software maintenance. *Journal of Computer Science*, 7(7).
- Skalik, J. (Ed.) (2009). *Zarządzanie projektami*. Wrocław: Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu.
- Tarwani, S., Chug, A. (2016). Agile Methodologies in Software Maintenance: A Systematic Review. *Informatica*, 40(4).
- The Institute of Electrical and Electronics Engineers (1998). *IEEE Standard For Software Maintenance – IEEE Std 1219-1998*. New York: The Institute of Electrical and Electronics Engineers.
- Turner, J., Müller, R. (2003). On the nature of the project as a temporary organization. *International Journal of Project Management*, 21, 1–8.
- Wirkus, M. (2013). *Zarządzanie projektami i procesami. Teoria i przypadki praktyczne*. Warszawa: Difin.
- Wood, S., Michaelidos, G. (2013). Successful extreme programming: Fidelity to the methodology or good teamworking? *Information and Software Technology*, 55(4).
- Wysocki, K.R., McGary, R. (2005). *Efektywne zarządzanie projektami*. Gliwice: Helion.

## MANAGEMENT OF THE MULTIMEDIA APPS MAINTENANCE PROGRAM – A CASE STUDY

### Summary

In the article, the author tries to answer the question: what critical factors influence the management of a multimedia application maintenance program in a medium-sized enterprise? The research was conducted according to case study methodology. The author used the following methods: participant observation, documentation analysis, and face-to-face interviews. The results of this research help us to better understand the processes in the enterprise during the implementation of the application maintenance program. The study identified critical factors and proposed changes.

**Keywords:** program management, apps maintenance, programming teams, technology management

