

dr inż. Przemysław Wójcik*Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie*

ORCID: 0000-0002-4261-847X

e-mail: wojcikp@gmail.com

dr Jacek Wolak*Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie*

ORCID: 0000-0001-5009-2888

e-mail: jwolak@agh.edu.pl

Korzyści i bariery zarządzania wiedzą w projektach wytwarzania oprogramowania

Benefits and barriers of knowledge management in software development projects

Celem opracowania jest wskazanie podstawowych barier i korzyści płynących z zarządzania wiedzą w podgrupie projektów wytwarzania oprogramowania. W pierwszej części artykułu omówiono specyfikę projektów wytwarzania oprogramowania. Następnie został przedstawiony proces badań empirycznych oraz prezentacja uzyskanych wyników. Badania mają charakter pilotażowy i stanowią podstawy do kontynuacji badań nad podjętą problematyką badawczą.

Słowa kluczowe

zarządzanie wiedzą, projekty wytwarzania oprogramowania

The purpose of this article is to identify the basics barriers and benefits of knowledge management in this subgroup of IT projects. In the first part of the article the specificity of software development projects will be discussed. Next, the empirical research process as well as obtained results are demonstrated. The results presented in the paper are regarded as pilot research and constitute the foundations for further research on these issues.

Keywords

knowledge management, software development projects

JEL: M15, D83

Wprowadzenie

Transformację wynikającą z przejścia z gospodarki bazującej na zasobach tradycyjnych w kierunku gospodarki opartej na wiedzy można obserwować, zwłaszcza od połowy ubiegłego wieku (Mbahlati, 2012). Zbiega się to z dynamicznym rozwojem technologicznym, spowodowanym m.in. wyścigiem zbrojeń w czasie II wojny światowej. Impuls rozwojowy w tym obszarze wpłynął i diametralnie zdynamizował obszar badań dotyczących komputeryzacji i informatyki (Miller, 1998, s. 288). Sama postępująca informatyzacja, szczególnie od lat osiemdziesiątych XX wieku, wywarła ogromny wpływ na życie społeczne oraz na uwarunkowania prowadzenia działalności gospodarczej. Od samego początku „ery komputerów” przedsiębiorstwa działają w niesamowicie dynamicznym, turbulentnym oraz konkurencyjnym środowisku. Obecnie w Polsce sektor infor-

matyczny zatrudnia ponad 400 tys. osób, stanowiąc 8% PKB (PARP, 2017). Szczególnymi przypadkami, jeszcze bardziej narażonymi i wrażliwymi na dynamiczne zmiany w sektorze informatyki, są przedsiębiorstwa realizujące projekty wytwarzania oprogramowania.

Projekty wytwarzania oprogramowania to bardzo złożone przedsięwzięcia. Z jednej strony są uwarunkowane potrzebami klienta, który często nie jest w stanie zdefiniować i określić całościowo swoich potrzeb, a z drugiej strony ograniczany jest przez czynniki egzogeniczne, np. zmiany technologiczne, aktywność konkurencji na rynku, zmiany regulacji prawnych itp. (Sener i Karsak, 2012, s. 19–21). Przedsiębiorstwa realizujące projekty wytwarzania oprogramowania ciągle stoją w obliczu zmieniających się wymagań ze strony klienta, muszą eliminować napotkane problemy sprzętowe, techniczne, kłaść nacisk na bezpieczeństwo i integralność zbieranych danych czy



też dostosowywać się do nowych warunków technologicznych (Mockus i Herbsleb, 2001; UMWZ, 2011). Dlatego intensywnie poszukują rozwiązań oraz metod, które pozwolą poprawić dotychczasowe działania, a tym samym utrzymać swoją pozycję na rynku (Gopalkrishna i in., 2012, s. 608). Wielu badaczy w swoich opracowaniach wskazuje, że to właśnie odpowiednie wdrażanie procesów zarządzania wiedzą w organizacji, wspomagających przepływ, dystrybucję oraz materializację wiedzy jest jednym z takich kluczowych czynników (Spalek, 2013; Fazlagić, 2014, s. 127; Kraszewska i Pujer, 2017, s. 7–10).

Celem opracowania jest wskazanie podstawowych barier i korzyści płynących z zarządzania wiedzą w podgrupie projektów wytwarzania oprogramowania dla organizacji i zespołu projektowego. W pierwszej części artykułu zostanie zaprezentowana i podkreślona specyfika projektów dotyczących wytwarzania oprogramowania. W kolejnych częściach syntetycznie omówione badania empiryczne nad barierami i korzyściami, tzn. przyjęty przez autorów model badawczy, przebieg i metody badań, a następnie podsumowanie uzyskanych wyników. W ostatniej części pracy, na podstawie analizy wyników i ograniczeń prezentowanych badań pilotażowych, zostaną zaproponowane możliwości i kierunki kontynuacji podjętej problematyki badawczej.

Zarządzanie wiedzą w środowisku projektów wytwarzania oprogramowania

Złożenie dwóch perspektyw zarządzania projektami oraz zarządzania wiedzą generuje specyficzny obszar problemowy, który na nowo otwiera i pozwala badaczom eksplorować te przenikające się warstwy. Z jednej strony, ze względu na swoją definicję tymczasowe, unikalne, niepowtarzalne przedsięwzięcia projektowe (PMI, 2008, s. 4), często o wysokim stopniu złożoności wymagają nieszablonowego podejścia, kreatywności, wysokiej specjalizacji członków zespołu projektowego, z drugiej zaś strony organizacja realizująca projekty w danym sektorze szuka rozwiązań pozwalających taką unikalną wiedzę transferować i kodyfikować na swój użytek oraz potrzeby następnych projektów (Kerzner, 2004; Ismail, Nor i Marjani, 2009; Terzieva, 2014). Zarządzanie wiedzą projektową jest to zarządzanie wiedzą w środowiskach projektowych oraz pomost między procesami zarządzania wiedzą oraz zarządzania projektem (Hanisch i in., 2009, s. 148–160).

Wiedzę projektową można różnorodnie systematyzować w zależności od strategii zarządzania oraz samych potrzeb organizacji. W opracowaniach dot. zarządzania wiedzą projektową badacze wskazują na mnogość ujęć oraz perspektyw. Wybrane klasyfikacje zostały zebrane i przedstawione w tablicy 1.

Analizując zebrane w tablicy klasyfikacje, można dojść do wniosku, że wiedza domenowa, branżowa, wiedza o kliencie czy produkcji może być traktowana jako samodzielny i istotny element. Wielu badaczy (Johansson, Löfström i Ohlsson 2007; Pollack, 2007; Engwall, 2013; Świętoniowska 2015) w swoich pracach wskazuje, że projekty i ich składowe mogą być skutecznie zarządzane tylko w momencie zrozumienia kontekstu, otoczenia środowiska i bliskiego poznania klienta. Dotyczy to projektów każdego typu, lecz czynniki oraz warunki są specyficzne dla danej domeny. Stąd autorzy na potrzeby niniejszego artykułu, ze względu na swoje zainteresowania, wybrali do przedmiotowych badań i analiz projekty wytwarzania oprogramowania. Istotne dla branży wytwarzania oprogramowania było powstanie komitetów lub grup ekspertów, które tworzą specjalne standardy czy wytyczne (np. IEEE czy ISO) tylko dla tego konkretnego obszaru. Przykładem takiego przewodniego dokumentu jest Software Engineering Body of Knowledge.

Analiza literatury oraz kluczowych standardów pozwoliła zagregować i krótko scharakteryzować specyficzne i unikalne cechy projektów wytwarzania oprogramowania:

- niematerialny charakter produktu końcowego: produktem projektu wytwarzania oprogramowania jest oprogramowanie lub jego fragment, czyli: specyficzny i zorganizowany zbiór instrukcji (inaczej kod¹), który przetworzony przez maszynę, wykonuje w zdefiniowany sposób określone operacje (Jaszkiewicz, 1997) lub inaczej: zbiór programów komputerowych, procedur oraz potencjalnie powiązanych dokumentów i danych (Standard ISO/IEC 12207:2008);
- specyficzny proces powstawania — najczęściej iteracyjny i przyrostowy, angażujący materialne zasoby teleinformatyczne organizacji oraz niematerialne, indywidualne zasoby wiedzy pracowników (Mishra i Dubey, 2013);
- rozproszony i multidyscyplinarny zespół — zadania projektowe realizowane są za pomocą rozwiązań do zdalnej pracy grupowej, z wykorzystaniem specyficznych narzędzi (np. Jira, Mantis, Confluence), przez członków zespołu o różnych, często bardzo odmiennych umiejętnościach i kompetencjach; ma to gwarantować wielowymiarowe spojrzenie na złożone zadania (Burnell, Priest i Durrett, 2002),
- specyficzne kryteria jakościowe — kryteria dotyczyć mogą aspektów funkcjonalnych i niefunkcjonalnych tworzonego rozwiązania (np. czas odpowiedzi w określonych warunkach, skalowalność, przenoszalność etc.) (Kan, 2002);
- często zmieniająca się specyfikacja i wymagania — określenie kompletnych wymagań dla oprogramowania jest niemożliwe, a szczególnie podczas doprecyzowywania szczegółów następuje

Tablica 1. Klasyfikacje wiedzy projektowej

Autorzy (data publikacji)	Typy wiedzy
Conroy, G. i Soltan, H. (1998)	<ul style="list-style-type: none"> • wiedza techniczna (technika, wiedza technologiczna) • wiedza dot. zarządzania projektami (metodyki, podejścia) • wiedza dot. projektu (wiedza o kliencie)
Damm, D. i Schindler, S. (2002)	<ul style="list-style-type: none"> • wiedza o projektach (wiedza metodyczna) • wiedza w projektach (wiedza członków zespołu projektowego) • wiedza z projektów (zasoby wiedzy i doświadczenia powstałe w wyniku realizacji projektu)
Reich, B. H. (2007)	<ul style="list-style-type: none"> • wiedza behawioralna: kulturowa (o wartościach i normach zachowań interesariuszy projektu) instytucjonalna (znajomość struktury organizacji) • wiedza techniczna: dziedzinowa (o branży, organizacji) procesowa (metodyczna)
Trocki, M. (2011)	<ul style="list-style-type: none"> • wiedza terminologiczna (słownik pojęć, definicji do dalszego wykorzystywania w opisach) • wiedza teoretyczna • wiedza metodyczna (pragmatyczna) • wiedza normatywna (potrzeby, oczekiwania, oceny, kryteria jakościowe)
Zhao, D. i Zuo, M. (2011)	<ul style="list-style-type: none"> • wiedza biznesowa (branżowa) • wiedza o produkcji • wiedza inżynierska (techniczna) • wiedza o zarządzaniu organizacją (struktura, procesy, zasoby) • wiedza dot. zarządzania projektami (metodyki, metody)

Źródło: opracowanie własne.

konieczność zrewidowania początkowych założeń (Despa, 2014);

- specjalistyczne produkty i procesy wspomagające — dokumentacja rozwiązania (np. spis wymagań, diagram przepływu itd.), procesy testowania weryfikujące poprawność i jakość działania względem określonych metryk (Ronkainen i Abrahamsson, 2003).

Wytwarzanie oprogramowania ze względu na swoją skomplikowaną i złożoną naturę wymaga podejścia dekompozycyjnego, legitymizującego stosowanie podejścia projektowego oraz procesów zarządzania wiedzą. Sam proces wytwarzania oprogramowania (tworzenia kodu) jest unikalną, niepowtarzalną i kreatywną czynnością. Branża oprogramowania ciągle poszukuje nowych metod poprawy własnej produktywności oraz jakości dostarczanego oprogramowania (Gopalkrishna i in., 2012). Stąd ukierunkowane badania empiryczne w problematyce zarządzania wiedzą w specyficznych projektach wytwarzania oprogramowania wydają się naturalnym krokiem.

Model i przebieg badań empirycznych

Przeprowadzone badania empiryczne skierowane były na eksplorację badanej problematyki, a ich celem była identyfikacja barier i korzyści płynących

z zarządzania wiedzą w projektach wytwarzania oprogramowania oraz określenie następnych kierunków do badań pogłębionych.

Badania przebiegały w kilku zasadniczych etapach.

1. Zidentyfikowanie czynników specyficznych dla projektów wytwarzania oprogramowania na podstawie przeglądu literatury oraz wywiadów nieskategoryzowanych, przeprowadzonych w zaprzyjaźnionych firmach informatycznych. Przygotowanie narzędzia badawczego. Badania przeprowadzono metodą computer-aided personal interview (CAPI), tworząc ankietę zawierającą zestaw pytań dotyczących zarządzania wiedzą, m.in. korzyści i barier dot. stosowania procesów zarządzania wiedzą.
2. Objęcie badaniem przedstawicieli strony biznesowej, definiującej produkt, członków zespołów specjalistycznych IT, kadry zarządzającej projektem oraz kierownictwa wyższego szczebla. Badana próba, ze względu na pożądaną charakterystykę próby oraz ograniczony czas, została wybrana metodą kwotową. Przyjęto kwotę w wysokości 200 odpowiedzi. Rozesłano prośbę do 1034 osób, w odpowiedzi uzyskano 194 wypełnione ankiety. Po analizie zawartości danych z zebranych ankiet pod kątem spójności do badania przyjęto 182.
3. Analizowanie oraz weryfikacja zebranych danych z badań ilościowych z zastosowaniem swobodnych wywiadów oraz rozszerzenie tej analizy pod kątem oceny poszczególnych produktów zarządzania wiedzą w projektach wytwarzania oprogramowania.

4. Analizowanie uzyskanego materiału, sformułowanie wniosków z badań.

Założono, że przyjęty model badań oraz sposób ich przebiegu może stać się — po modyfikacji — podstawą kolejnych etapów prac badawczych, bazujących na większych próbach.

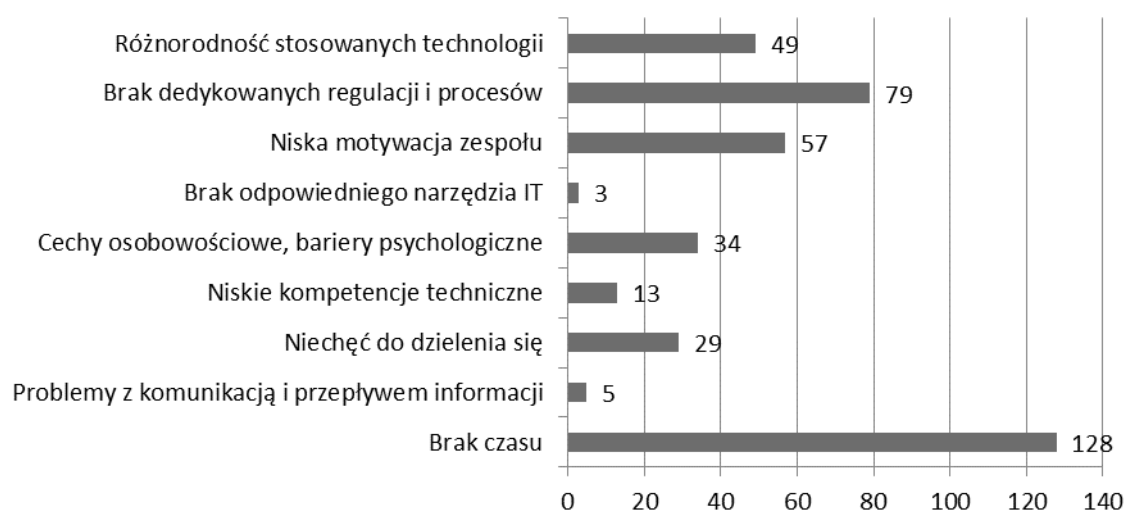
Wyniki badań empirycznych

Przyjęty model badawczy opierał się na poprzedzeniu badań analizą literatury przedmiotu celem identyfikacji już znanych problemów i barier z zakresu zarządzania wiedzą w projektach oraz korzyści płynących z ich wykorzystywania. Badanie pozwoliło zebrać i skompilować w sposób syntetyczny (rysunek 1) najczęściej wskazywane bariery występujące w procesie zarządzania wiedzą w projektach wytwarzania oprogramowania. W pozycjach literaturowych szczególnie podkreśla się istotę zarządzania wiedzą w projektach, definiuje się nawet dedykowane role i obowiązki (Schnidler i Eppler, 2003) do wspierania tych aktywności. Zaskakujący u respondentów jest fakt, że jako największą barierę dla zarządzania wiedzą w projektach wytwarzania oprogramowania przedstawiają brak czasu (70%). Jest to zapewne związane z natłokiem zadań i przedkładaniem wartości, jaką jest dostarczenie produktu dla klienta nad elementy pośrednio związane z jego wytwarzaniem. Może się to też częściowo wpisywać w najmodniejsze w ostatnich latach w branży wytwarzania oprogramowania ramy postępowania zgodne ze Scrum (Rubin, 2012) (metodyka zwinna zgodna z manifestem agile)².

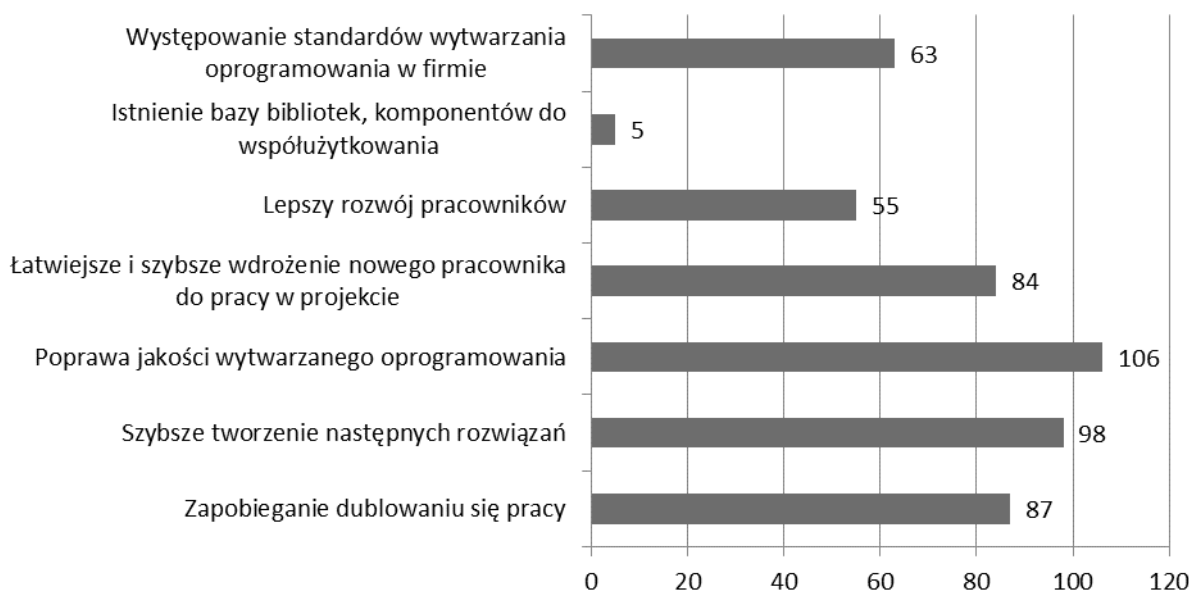
Pochodną tego problemu, który jest również bardzo istotną barierą, a na co wskazują respondenci (ponad 40%), jest brak regulacji i polityk obligujących pracowników do zarządzania wiedzą. Najpewniej jest to wynik braku stosowania modeli dojrzałości dot. zarządzania wiedzą oraz projektami w organizacji. Dwie inne bariery, o których warto tutaj wspomnieć, to niska motywacja zespołu (ponad 30%) oraz różnorodność technologiczna (26%). O ile pierwsza, ze wspomnianych powyżej przyczyn może występować z powodu wielu czynników endo- i egzogenicznych w stosunku do członków zespołu (np. płaca, charakter umowy, konflikty, bariery kulturowe itp.), to ta druga bariera jest naturalnym i nieuniknionym następstwem rozwoju technologii. W bardzo krótkim okresie w dzisiejszych czasach można zaobserwować diametralne zmiany technologiczne, infrastrukturalne, architektoniczne oraz koncepcyjne w sektorze informatycznym. Kontrola, a także spójność technologiczna przedsięwzięć zazwyczaj jest definiowana na poziomie strategicznym organizacji.

Respondenci wskazywali w swoich odpowiedziach, że wymiernymi korzyściami płynącymi z zastosowania procesów zarządzania wiedzą (rysunek 2) są: poprawa jakości wytwarzanego oprogramowania, szybsze tworzenie następnych rozwiązań, zapobieganie dublowaniu się pracy oraz szybsze wdrażanie nowego pracownika. Zapobieganie dublowaniu się pracy wydaje się bardzo istotne, szczególnie w organizacjach zatrudniających tysiące osób, gdzie redundancja wysiłków przy realizacji podobnej pracy może być bardzo kosztowna, a trudna do wychwycenia. Tym bar-

Rysunek 1. Bariery występujące w procesie zarządzania wiedzą w projektach wytwarzania oprogramowania



Źródło: opracowanie własne.

Rysunek 2. Korzyści płynące z wykorzystania wiedzy w projektach wytwarzania oprogramowania

Źródło: opracowanie własne.

dziej ciekawe jest postrzeganie zarządzania wiedzą jako czynnikiem diametralnie wpływającym korzystnie oraz bezpośrednio na wytwarzany produkt projektu, przy jednoczesnym braku wdrożonych, dedykowanych regulacji w tym obszarze oraz sygnalizowanie braku wystarczającego czasu w badanych organizacjach. Tutaj rodzi się pytanie: czy organizacje świadomie to pomijają, czy jest to efekt zaniedbań oraz pracy ukierunkowanej tylko na dostarczanie klientowi wartości, bez postrzegania jej w rozumieniu długofalowym dla organizacji.

Podsumowanie

W artykule przedstawiono wybrane wyniki oraz wnioski z badań empirycznych dotyczących barier i korzyści płynących z zarządzania wiedzą w projektach wytwarzania oprogramowania. Przeprowadzone badania miały charakter badań pilotażowych, dlatego też posiadają szereg ograniczeń dotyczących ich zakresu oraz wielkości próby.

Przypisy

¹ W środowisku IT uważa się, że kod sam w sobie jest nośnikiem wiedzy.

² Manifest agile — deklaracja wspólnych zasad dla zwinnych metod tworzenia oprogramowania, która m.in. stawia działające oprogramowanie, ponad obszerną dokumentację. Oficjalna strona internetowa manifestu zwinności: <https://agilemanifesto.org/>

Organizacje wytwarzające oprogramowanie zmagają się w swojej codziennej działalności z bardzo wymagającym rynkiem oraz starają się optymalizować sposób swojego działania. Odpowiedzialność za zachowanie konkurencyjności, ciągłe udoskonalanie sposobu pracy przy zachowaniu efektywności ekonomicznej leżą w gestii zarządu przedsiębiorstw oraz kadry wysokiego szczebla. Ten nieustanny proces wymaga ustawicznego pozycjonowania własnych działań na tle innych konkurentów, a także wewnętrznych oraz zewnętrznych wymagań (np. technologicznych, prawnych). Wyniki tych badań wstępnych pokazują, że mogą one zaoferować interesujący wgląd w realne problemy zespołów realizujących projekty wytwarzania oprogramowania, wskazać realne bariery oraz korzyści dla organizacji w momencie implementacji procesów zarządzania wiedzą.

Badania wykazują również duży potencjał związany z rozszerzeniem ich zakresu. Wydaje się, że interesujące wnioski można uzyskać, rozszerzając te badania w celu poszukiwania sposobów niwelowania zidentyfikowanych barier lub też poprzez realizację badań w innej podkategorii projektów informatycznych.

Bibliografia

- Burnell, L. J., Priest, J. W. i Durrett, J. B. (2002). Teaching distributed multidisciplinary software development. *IEEE software*, 19(5), 86–93. <https://doi.org/10.1109/ms.2002.1032859>
- Conroy, G. i Soltan, H. (1998). ConSERV, as a continual audit concept to provide traceability and accountability over the project life cycle. *International Journal of Project Management*, 16(3), 185–197. [https://doi.org/10.1016/s0263-7863\(97\)00038-0](https://doi.org/10.1016/s0263-7863(97)00038-0)
- Damm, D. i Schindler, M. (2002). Security issues of a knowledge medium for distributed project work. *International Journal of Project Management*, 20(1), 37–47. [https://doi.org/10.1016/s0263-7863\(00\)00033-8](https://doi.org/10.1016/s0263-7863(00)00033-8)
- Despa, M. L. (2014). Comparative study on software development methodologies. *Database Systems Journal*, 5(3), 37–56.
- Engwall, M. (2003). No project is an island: Linking projects to history and context. *Research policy*, 32(5), 789–808. [https://doi.org/10.1016/s0048-7333\(02\)00088-4](https://doi.org/10.1016/s0048-7333(02)00088-4)
- Fazlagić, J. (2014). *Innowacyjne zarządzanie wiedzą*. Warszawa: Difin.
- Gopalkrishna, B., Rodrigues, L. L., Poornima, P. K. i Manchanda, S. (2012). Knowledge Management in Software Companies — An Appraisal. *International Journal of Innovation, Management and Technology*, 3(5).
- Hanisch, B., Lindner, F., Mueller, A. i Wald, A. (2009). Knowledge management in project environments. *Journal of Knowledge Management*, 13(4), 148–160. <https://doi.org/10.1108/13673270910971897>
- Ismail, W. K. W., Nor, K. M. i Marjani, T. (2009). The role of knowledge sharing practice in enhancing project success. *Institute of Interdisciplinary Business Research*, 1(7), 34–52.
- ISO (2008). *ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes*. Genewa: ISO. <https://doi.org/10.3403/30236650u>
- Jaszkiewicz, A. (1997). *Inżynieria Oprogramowanie*. Gliwice: Helion.
- Johansson, S., Löfström, M. i Ohlsson, Ö. (2007). Separation or integration? A dilemma when organizing development projects. *International Journal of Project Management*, 25(5), 457–464. <https://doi.org/10.1016/j.jproman.2006.11.006>
- Kan, S. H. (2002). *Metrics and models in software quality engineering*. Boston: Addison-Wesley Longman Publishing.
- Kerzner, H. (2003). *Advanced project management: Best practices on implementation*. Hoboken: John Wiley & Sons.
- Kraszewska, M. i Pujer, K. (2017). *Konkurencyjność przedsiębiorstw. Sposoby budowania przewagi konkurencyjnej*. Wrocław: Exante.
- Mbhalati, O. J. (2012). The Genesis Of The Knowledge-Based View. *Journal of Knowledge Management Practice*, 13(2).
- Miller, C. R. (1998). Learning from history: World War II and the culture of high technology. *Journal of Business and Technical Communication*, 12(3), 288–315. <https://doi.org/10.1177/1050651998012003002>
- Mishra, A. i Dubey, D. (2013). A comparative study of different software development life cycle models in different scenarios. *International Journal*, 1(5), 64–69.
- Mockus, A. i Herbsleb, J. (2001). *Challenges of global software development*. Referat na konferencji: Seventh international software metrics symposium. London. <https://doi.org/10.1109/metric.2001.915526>
- PARP. (2017). *Perspektywy rozwoju polskiej branży ICT do roku 2025*. Warszawa: Ministerstwo Rozwoju.
- PMI. (2008). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Campus Boulevard: Project Management Institute. <https://doi.org/10.1109/ieeestd.2011.6086685>
- Pollack, J. (2007). The changing paradigms of project management. *International Journal of Project Management*, 25(3), 266–274. <https://doi.org/10.1016/j.jproman.2006.08.002>
- Reich, B. H. (2007). Managing knowledge and learning in IT projects: A conceptual framework and guidelines for practice. *Project Management Journal*, 38(2), 5–17. <https://doi.org/10.1177/875697280703800202>
- Ronkainen, J. i Abrahamsson, P. (2003). *Software development under stringent hardware constraints: Do agile methods have a chance?* Referat na konferencji: International Conference on Extreme Programming and Agile Processes in Software Engineering. Berlin. https://doi.org/10.1007/3-540-44870-5_10
- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular Agile process*. Boston: Addison-Wesley Longman Publishing.
- Schindler, M. i Eppler, M. J. (2003). Harvesting project knowledge: a review of project learning methods and success factors. *International Journal of Project Management*, 21(3), 219–228. [https://doi.org/10.1016/s0263-7863\(02\)00096-0](https://doi.org/10.1016/s0263-7863(02)00096-0)
- Sener, Z. i Karsak, E. E. (2012). A decision model for setting target levels in software quality function deployment to respond to rapidly changing customer needs. *Concurrent Engineering*, 20(1), 19–29. <https://doi.org/10.1177/1063293x11435344>
- Spalek, S. (2013). Dzielenie się wiedzą projektową w polskich przedsiębiorstwach — zarys problematyki. *Zarządzanie i Finanse*, 11(1, cz. 2), 305–314.
- Świętoniowska, J. (2015). Podejście kontekstowe w zarządzaniu projektami. *Studia Ekonomiczne*, 216, 118–134.
- Terzieva, M. (2014). Project Knowledge Management: How organizations learn from experience. *Procedia Technology*, 16, 1086–1095. <https://doi.org/10.1016/j.protcy.2014.10.123>
- Trocki, M. (red.). (2011). *Zarządzanie wiedzą w projektach*. Warszawa: SGH.
- UMWZ. (2011). *Analiza sytuacji rynkowej dla działalności gospodarczej na obszarze województwa zachodniopomorskiego dla branży informatycznej*. Gdańsk: Urząd Marszałkowski Województwa Zachodniopomorskiego. <https://doi.org/10.18276/er.2015.24-13>
- Zhao, D. i Zuo, M. (2011). *Knowledge transferred across projects and its characteristics in an IT service enterprise*. Referat na konferencji: 2011 International Conference of Information Technology, Computer Engineering and Management Sciences. Nanjing, Jiangsu, China. <https://doi.org/10.1109/icm.2011.139>