

Prof. dr hab. inż. Tomasz Szmuc  
Katedra Informatyki Stosowanej  
Akademii Górniczo-Hutniczej  
Al. Mickiewicza 30, 30-059 Kraków  
tel.: (+12) 617 28 32  
e-mail: [tsz@agh.edu.pl](mailto:tsz@agh.edu.pl)



Kraków, 02-11-2021

## RECENZJA OSIĄGNIĘCIA I OCENA DOROBKU NAUKOWEGO

dr. inż. Mirosława Ochodka

Recenzja niniejsza została przygotowana na zlecenie (umowa z dnia 8. września 2021) prof. dr hab. inż. Andrzeja Jaszkiwicza, Przewodniczącego Rady Dyscypliny Informatyka Techniczna i Telekomunikacja w Politechnice Poznańskiej, w związku z powołaniem (przez Radę Doskonałości Naukowej) komisji habilitacyjnej dla przeprowadzenia postępowania o nadanie stopnia naukowego doktora habilitowanego Nauk technicznych w dyscyplinie Informatyka techniczna i telekomunikacja. Recenzja dotyczy osiągnięć naukowych dr. inż. Mirosława Ochodka.

### 1. Recenzja osiągnięcia naukowego

Recenzja osiągnięcia naukowego została wykonana na podstawie dokumentacji, w tym załączonego cyklu dziewięciu publikacji podstawowych (opublikowanych w latach 2015-2020). Poniżej podano listę publikacji tworzących oceniane osiągnięcie naukowe: *Metody wspierające proces szacowania pracochłonności w projektach informatycznych*.

1. **M. Ochodek**, S. Kopczyńska, M. Staron, *Deep learning model for end-to-end approximation of COSMIC functional size based on use-case names*, Information and Software Technology, vol. 123, 106310, DOI: 10.1016/j.infsof.2020.106310, 2020, ISSN: 0950-5849 **MNiSW/MEiN (2019+) – 140 pkt.**, IF=2,726
2. **M. Ochodek**, M. Staron, W. Meding, J. Bosch, *LegacyPro: A DNA-inspired method for identifying process legacies in software development organizations*, IEEE Software, vol. 37, no. 6, pp. 76-85, DOI: 10.1109/MS.2020.2971894, 2020, ISSN: 0740-7459 **MNiSW/MEiN (2019+) – 100 pkt.**, IF=2,589
3. **M. Ochodek**, R. Hebig, W. Meding, G. Frost, M. Staron, *Recognizing lines of code violating company-specific coding guidelines using machine learning*, Empirical Software Engineering, vol. 25, pp. 220-265, DOI: 10.1007/s10664-019-09769-8, 2020, ISSN: 1382-3256 **MNiSW/MEiN (2019+) – 140 pkt.**, IF=3,156
4. P. Pickerill, H. J. Jungen, **M. Ochodek**, M. Maćkowiak, M. Staron, *PHANTOM: Curating GitHub for engineered software projects using time-series clustering*, Empirical Software Engineering, vol. 25, pp. 2897-2929, DOI: 10.1007/s10664-020-09825-8, 2020, ISSN: 1382-3256 **MNiSW/MEiN (2019+) – 140 pkt.**, IF=3,156
5. **M. Ochodek**, M. Staron, W. Meding, *Simsax: A measure of project similarity based on symbolic approximation method and software defect inflow*, Information and Software Technology, vol. 115, pp. 131-147, DOI: 10.1016/j.infsof.2019.06.003, 2019, ISSN: 0950-5849 **MNiSW/MEiN: (2019+) – 140 pkt.**, IF=2,726
6. S. Kopczyńska, J. Nawrocki, **M. Ochodek**, *An empirical study on catalog of nonfunctional requirement templates: Usefulness and maintenance issues*, Information and Software Technology, vol. 103, pp. 75-91, DOI: 10.1016/j.infsof.2018.06.009, 2018, ISSN: 0950-5849 **MNiSW (2013-2016) – 35 pkt.**, **MNiSW/MEiN (2019+) – 140 pkt.**, IF=2,921
7. **M. Ochodek**, S. Kopczyńska, *Perceived importance of agile requirements engineering practices - A survey*, Journal of Systems and Software, vol. 143, pp. 29-43, DOI: 10.1016/j.jss.2018.05.012, 2018, ISSN: 0164-1212 **MNiSW (2013-2016) – 35 pkt.**, **MNiSW/MEiN (2019+) – 100 pkt.**, IF=2,559

8. **M. Ochodek**, *Functional size approximation based on use-case names*, Information and Software Technology, vol. 80, pp. 73-88, DOI: 10.1016/j.infsof.2016.08.007, 2016, ISSN: 0950-5849 **MNiSW (2013-2016) – 35 pkt.**, MNiSW/MEiN (2019+) – 140 pkt., IF=2,694
9. J. Jurkiewicz, J. Nawrocki, **M. Ochodek**, T. Głowacki, *HAZOP-based identification of events in use cases. An empirical study*, Empirical Software Engineering, vol. 20, pp. 82-109, DOI: 10.1007/s10664-013-9277-5, 2015, ISSN: 1382-3256 **MNiSW (2013-2016) – 45 pkt.**, MNiSW/MEiN (2019+) – 140 pkt., IF=1,393.

Prace te są powiązane tematycznie (zgodnie z art. 16 ust. 2 Ustawy o stopniach naukowych i tytule naukowym oraz stopniach i tytule w zakresie sztuki, z dnia 14 marca 2003 z późniejszymi zmianami).

### 1.1. Opis osiągnięcia naukowego

Oceniane osiągnięcie naukowe zatytułowane: *Metody wspierające proces szacowania pracochłonności w projektach informatycznych* dotyczy bardzo ważnego działu inżynierii oprogramowania mającego istotny wpływ na decyzje dotyczące warunków realizacji lub przebiegu przedsięwzięć informatycznych.

Habilitant zaproponował zestaw metod wspomagających odpowiednio główne czynności początkowej fazy inżynierii wymagań: (1) ekstrakcja wymagań funkcjonalnych i pozafunkcjonalnych (niefunkcjonalnych), (2) pomiar rozmiaru (pracochłonności) oprogramowania oraz (3) pozyskiwanie i „czyszczenie” danych o projektach historycznych. Badania koncentrują się w obszarze zwinnej inżynierii wymagań (Agile RE). Poniżej opisano podstawowe wyniki w trzech wymienionych obszarach.

#### **Wspomaganie pozyskiwania wymagań funkcjonalnych i niefunkcjonalnych ([7], [9], [6])**

Praca [7] jest logicznym początkiem badań związanych ze wspomaganie procesu pozyskiwania wymagań. Opracowano w niej (na podstawie badań ankietowych wśród profesjonalistów) ranking istniejących praktyk zwinnej inżynierii wymagań. Wyniki pozwoliły na potwierdzenie intuicyjnego wniosku: *w zwinnej inżynierii wymagań główny nacisk jest położony na opracowanie jasnej wizji produktu i współpracę z interesariuszami w ramach krótkich iteracji, w których wymagania stanowią sprzężenie zwrotne sterujące rozwojem produktu*. Kompletność określenia wymagań dla kolejnych iteracji ma zatem istotny wpływ na proces wytwarzania oprogramowania i jakość tworzonego produktu. Opracowane przez Habilitanta metody wspierają pozyskiwanie wymagań funkcjonalnych i niefunkcjonalnych. Wymagania funkcjonalne są tradycyjnie specyfikowane przez przypadki użycia.

Zaproponowana [9] metoda *HAZOP for use cases* (H4U) umożliwia wspomaganie procesu identyfikacji zdarzeń w scenariuszach przypadków użycia. Metoda H4U bazuje na znanej technice HAZOP (*Hazard and Operability Study*) stosowanej do identyfikacji przyczyn i możliwej propagacji hazardu w planowanych/projektowanych przedsięwzięciach. Zaproponowane dostosowanie polega na odpowiednim przypisaniu pierwotnych i wtórnych słów kluczowych umożliwiających ekstrakcję zdarzeń ze scenariuszy. Metoda H4U została zweryfikowana w dwóch grupach testowych: 18 studentów oraz 82 osób zajmujących się profesjonalnie wytwarzaniem oprogramowania. Zastosowanie H4U powiększyło liczbę zidentyfikowanych zdarzeń o 50% i 27% odpowiednio dla grupy pierwszej i drugiej.

Użyteczność katalogu szablonów wymagań niefunkcjonalnych badano w pracy [6]. Zastosowanie katalogu szablonów jest praktyką znaną w inżynierii wymagań. Katalog gromadzi doświadczenia z poprzednich projektów i pozwala na ich powtórne użycie w postaci odpowiednich szablonów. W cytowanej pracy opisano kontrolowany eksperyment, w którym dwie grupy użytkowników miały określić wymagania niefunkcjonalne systemu typu *e-commerce*. Użycie katalogu szablonów znacznie poprawiło kompletność i jakość wymagań. Badania symulacyjne w kolejnym kroku pozwoliły określić tzw. dojrzałość katalogu, czyli moment gdy zawartość katalogu stabilizuje się dla kolejnych projektów. Dojrzałość określa zatem granicę stabilizacji katalogu, który nie musi być modyfikowany do czasu zmiany typów projektów lub metodyki.

#### **Wspomaganie pomiaru rozmiaru oprogramowania ([8], [1], [3])**

Szacowanie rozmiaru oprogramowania na podstawie wymagań jest utrudnione ze względu na wysoki poziom ogólności wiedzy o tworzonej produkcie. W pracy [8] zaproponowano dwie dwuetapowe metody pozwalające określać przybliżony rozmiar funkcjonalny na podstawie tytułów przypadków użycia.

W pierwszym etapie jest określany typ celu przypadku użycia (spośród 13 zdefiniowanych), natomiast w drugim przeprowadzane jest szacowanie na podstawie przypisanego celu oraz danych historycznych z wcześniej wykonanych projektów. Zaproponowano również metodę automatycznego znajdowania celu na podstawie tytułu przypadku użycia.

Następnie, w pracy [1], opracowano nową wersję znanej metody COSMIC, pozwalającą na szacowanie rozmiaru na podstawie nazw przypadków użycia. Jest to metoda jednoetapowa, bazująca na modelu regresji realizowanym przez konwolucyjną sieć neuronową. Przeprowadzone eksperymenty potwierdziły lepszą jakość szacowania zarówno względem istniejących metod jak również wcześniej zaproponowanych przez Habilitanta. Pomiar liczby linii kodu źródłowego (LOC) jest znanym alternatywnym podejściem do szacowania pracochłonności utworzenia oprogramowania. Klasyczne podejście wymaga dostępu do kodu źródłowego, najlepiej zapisanego w jednym języku programowania. W początkowej fazie dysponujemy jedynie ogólnym opisem projektu, stąd metoda ta jest praktycznie nieprzydatna. W ramach prac badawczych prowadzonych przez Habilitanta opracowano oprogramowanie *Flexible Code Counter/Classifier (CCFLex)* umożliwiające tworzenie instrumentów pomiarowych LOC na podstawie przykładów. Do rozwiązania wykorzystano algorytmu uczenia maszynowego.

W pracy [3] opisano poprawioną wersję CCFLex oraz przedstawiono zastosowanie tego narzędzia do identyfikacji linii kodu źródłowego niezgodnych z firmowymi standardami. Badano zgodność projektów typu *open source* ze standardami firm Sun i Google oraz kodu wytworzonego przez firmy zajmujące się przemysłowym wytwarzaniem oprogramowania. Uzyskano wysoką dokładność klasyfikacji dla przeprowadzonych eksperymentów, szczególnie dla wytycznych określonych w szerszym kontekście.

#### **Wspieranie pozyskiwania zbiorów danych historycznych ([5], [2], [4])**

Większość zaproponowanych metod korzysta z danych historycznych o wykonanych projektach. Jakość tych danych ma więc istotny wpływ na uzyskiwane wyniki. Wspomaganie pozyskiwania i „czyszczenia” danych jest więc naturalnym uzupełnieniem opisywanych wyżej dwóch podobszarów badawczych. Istotna jest przy tym właściwa klasyfikacja projektów, czyli określenie cech określających podobieństwo projektów. Umożliwia to tworzenie homogenicznych grup (klas) projektów opisywanych odpowiednim zestawem cech. W badaniach skupiono się na ciągłym procesie rozwijania oprogramowania (metodyki zwinne) i tym samym uwzględniano zmiany w trakcie rozwijania projektu. Zaproponowana w [5] nowa miara (zwana SimSAX) oceny podobieństwa projektów informatycznych polega na analizie profilu napływu zgłoszeń o defektach. Profil ma postać szeregu czasowego liczby defektów wykrytych w poszczególnych tygodniach. Profile są przekształcane w ciągi symboli, które są porównywane w celu wykrycia podobnych podsekwencji (motywów). Metodę (po wstępnej kalibracji) przetestowano na sześciu projektach *open source* i pięciu przemysłowych. Uzyskane wyniki potwierdzają przydatność SimSAX, szczególnie w przypadku dłuższych motywów (ponad 32 tygodnie) i alfabetu składającego się z przynajmniej pięciu symboli.

W kolejnej pracy [2] zaproponowano metodę LegacyPro umożliwiającą monitorowanie (zmian) sposobu pracy zespołów projektowych. Metoda ta polega na wykrywaniu dwóch typów motywów: tzw. motywów wewnętrznych (stabilny, powtarzalny styl pracy zespołu) oraz tzw. motywów współdzielonych – zmiany (o ile zachodzą) na terenie organizacji. Ocenę metody przeprowadzono na danych z pięciu projektów *open source* oraz czterech projektów przemysłowych.

Dynamiczny rozwój platform współdzielenia kodu (GitHub, SourceForge) stwarza zapotrzebowanie na rozwój narzędzi wspierających wyszukiwanie właściwego projektu w milionowych bazach. Zaproponowana w pracy [4] metoda PHANTOM (*Project History Analysis of Time-Series Method*) bazuje na częstotliwości zmian w rozwoju projektów na podstawie danych z logów do bazy. Klasyfikacja jest realizowana algorytmami uczenia maszynowego bez nadzoru. Walidację przeprowadzono porównując wyniki klasyfikacji repozytorium GitHub z uzyskanymi przez z zastosowaniem znanego z literatury narzędzia. Zaletą zaproponowanej metody jest jej efektywność potwierdzona radykalnie mniejszymi wymaganiami względem zasobów obliczeniowych. Dodatkowo przeprowadzono porównanie tych narzędzi przy klasyfikacji 100 repozytoriów przemysłowych. W tym przypadku PHANTOM wykazał wyższą o dwa rzędy wydajność, przy tej samej dokładności klasyfikacji.

## 1.2 Ocena osiągnięcia naukowego

Oceniane osiągnięcie naukowe dotyczy wspomagania oceny pracochłonności oprogramowania. W tradycyjnym podejściu (np. metodyki bazujące na modelowaniu) szacowanie jest wykonywane na początku projektu informatycznego, z kolei w metodykach zwinnych przed rozpoczęciem kolejnego cyklu. W pierwszym przypadku dysponujemy wynikającą z analizy ogólną i niekiedy niekompletną wiedzą, która jest użyta do oceny pracochłonności całego przedsięwzięcia. Pracochłonność cyklu ma mniejszy zakres, charakter krótkoterminowy i wykorzystuje informację z poprzednich cykli. W obu metodykach możemy korzystać z doświadczeń zebranych z wcześniej prowadzonych projektów. Próby szacowania pracochłonności były podejmowane praktycznie od początku inżynierii oprogramowania, jednak mimo opracowania różnorodnych metod nie znaleziono zadawalającego rozwiązania. Wynika to z opisanego wcześniej ogólnego i często niekompletnego/niepewnego zestawu danych wejściowych na podstawie których należałoby w miarę precyzyjnie oszacować pracochłonność. Uzyskane tym samym szacowania mogą być obciążone istotnym poziomem niepewności, co przekłada się na zwiększenie ryzyka w podejmowaniu decyzji względem warunków podjęcia lub kontynuacji projektu.

Zaproponowane metody wspierają szacowanie pracochłonności oprogramowania w trzech, opisanych wyżej obszarach: ekstrakcja wymagań funkcjonalnych i niefunkcjonalnych, pomiar rozmiaru (pracochłonności) oprogramowania oraz pozyskiwanie i „czyszczenie” danych o projektach historycznych. Należy podkreślić, że Habilitant nie ograniczył się do wąskiej grupy metod związanych z wybranym aspektem, lecz uwzględnił możliwości wsparcia w wymienionych trzech płaszczyznach. W wyniku opracowano spójny zestaw komplementarnych metod. Istotne jest również dostosowanie opracowanych metod do zwinnych metodyk wytwarzania oprogramowania.

Inżynieria oprogramowania wymaga zazwyczaj eksperymentalnej weryfikacji nowych metod lub algorytmów. Proponowane koncepcje/metody winny być starannie weryfikowane w dziedzinie ich zastosowań. Weryfikacja winna obejmować możliwie szeroką grupę projektów, z uwzględnieniem wytwarzanych przemysłowo. Jest to często dodatkowe utrudnienie dla środowisk uczelnianych, szczególnie mających słabsze kontakty z przemysłem wytwarzania oprogramowania. Ponadto firmy zazwyczaj ograniczają dostęp do swoich repozytoriów. Opisane w przedstawionych publikacjach rozwiązania charakteryzują się systematycznym wywodem, a ich użyteczność jest starannie weryfikowana. Weryfikacja zaproponowanych metod jest realizowana nie tylko na projektach typu *open source*, lecz również prowadzonych w warunkach przemysłowych. Nie bez znaczenia jest tu kilkuletnia współpraca z Chalmers University of Technology w Goeteborgu, co zapewne ułatwiło pozyskanie danych o projektach z dwóch dużych firm szwedzkich. Dodatkowym rezultatem jest opisane wyżej oprogramowanie (CCFLex, SimSAX, LegacyPro) implementujące wybrane metody.

W podsumowaniu oceny merytorycznej należy stwierdzić, że osiągnięte wyniki są oryginalne, a ich praktyczna użyteczność została eksperymentalnie wykazana. Zaproponowane metody efektywnie wspomagają proces szacowania pracochłonności projektów informatycznych, co stanowi wartościowy wkład do inżynierii wymagań, z uwzględnieniem metodyk zwinnych.

Przedstawiony cykl publikacji jest tematycznie spójny i dobrze opisuje oceniane osiągnięcie naukowe. Wszystkie spośród przedstawionych dziewięciu artykułów są publikowane w czasopiśmie z listy Journal Citation Report (JCR) i tym samym znajdują się na liście A czasopism opublikowanej przez Ministerstwo Nauki i Szkolnictwa Wyższego (Ministerstwo Edukacji i Nauki). Są to zatem czasopisma wysokiej rangi o cyrkulacji międzynarodowej. Oprócz autorskiej pozycji [8] wszystkie pozostałe publikacje są zespołowe, przy czym wyróżnić można rdzeń zespołu współpracujący z Habilitantem. W pozycjach [1-5] współautorami są również osoby z Chalmers University of Technology, ponadto w artykułach [2, 3] współuczestniczyli pracownicy firm szwedzkich. Dostarczone z dokumentacją oświadczenia pozwalają stwierdzić, że w opisywanych badaniach Habilitant pełnił decydującą rolę. Powyższe świadczy dodatkowo o umiejętnościach współpracy w zespołach, co jest aktualnie pożądaną cechą badacza.

## 2. Ocena dorobku naukowego

Działalność naukowa habilitanta zawiera się w obszarze inżynierii oprogramowania. Wprawdzie badania koncentrują się w obszarze inżynierii wymagań, lecz można znaleźć pojedyncze prace dotyczące testowania. Istotna liczba prac, szczególnie po uzyskaniu stopnia doktora, jest publikowana w wysoko

punktowanych czasopismach. Poniżej podano punktację z rozbiem rodzaj: publikacje w czasopismach i rozdziałach/materiałach konferencyjnych. W podkategoriach wyróżniono okresy przed i po uzyskaniu stopnia doktora. Punkty podano wg skali obowiązującej w roku ukazania się publikacji, natomiast w nawiasach liczbę obliczoną wg skali Rozporządzenia Ministerstwa z dnia 18 grudnia 2019 (2019+) roku.

1. Artykuły w czasopismach – łącznie 16, suma punktów 949 (1 660)
  - a. po uzyskaniu stopnia doktora – 11 artykułów, suma punktów 854 (1320)
  - b. przed doktoratem – 5 artykułów, suma punktów 95 (340)
2. Artykuły w rozdziałach/materiałach konferencyjnych – łącznie 25, suma punktów wg listy 2019+ wynosi (885), w tym:
  - a. po uzyskaniu stopnia doktora – 18 artykułów, suma punktów (595)
  - b. przed doktoratem – 7 artykułów, suma punktów (290).

Współczynnik *Impact Factor* dla dorobku publikacyjnego kształtuje się następująco: po uzyskaniu stopnia naukowego doktora –10 publikacji, sumaryczny IF = 26.479; przed doktoratem 4 publikacje i odpowiednio IF = 3.212. Baza Web of Science indeksuje 26 publikacji cytowanych 107 razy (bez autocytowań) w 91 artykułach. Indeks Hirscha wg WoS wynosi H=5. Uwzględnienie dodatkowych cytowań w trzech publikacjach nieindeksowanych podnosi sumę cytowań (bez autocytowań) do wartości 123 oraz H-indeks osiąga wartość 6. Baza Scopus indeksuje 39 publikacji cytowanych 263 razy (w tym 23 autocytowania) w 209 artykułach. Indeks Hirscha = 8. Uwzględnienie cytowań w dodatkowych 18 nieindeksowanych w Scopus artykułach nie zmienia wartości H-indeksu.

Przytoczone wyżej dane świadczą o rozpoznawalności i uznaniu osiągnięć na arenie międzynarodowej. Zapewne cytowalność i indeks Hirscha powiększą się, gdyż ważne publikacje ukazały się w ostatnich dwóch latach. Podobnie jak w przypadku prac tworzących osiągnięcie naukowe, w większości są to artykuły współautorskie, często w podobnych konstelacjach współautorów jak w przypadku publikacji stanowiących osiągnięcie naukowe.

Rozpoznawalność Habilitanta w międzynarodowym środowisku naukowym potwierdzają również liczne recenzje, które wykonał dla czasopism: *Information and Software Technology* (15), *Journal of Software and Systems* (10), *IET Software* (6), *e-Informatica Software Engineering Journal* (13), *Software Quality Journal* (5), *Empirical Software Engineering* (2), *e-Informatica Software Engineering Journal* (2), *IEEE Software* (2), *IEEE Access* (2), *Journal of Software Evolution and Process* (2), *Science of Computer Programming* (2), *Software and Systems Modeling* (2), *Computer Science and Information Systems* (1).

Dr M. Ochodek pełnił rolę *principal investigator* w projekcie *Anomaly Detection in Wireless Networks using Machine Learning*. Projekt był realizowany w ramach współpracy Software Center (Chalmers University of Technology) z firmami szwedzkimi Ericsson AB, Axis Communication AB oraz Grundfos. Jest współautorem pracy, która została wyróżniona Best Paper Award na konferencji Software Quality Days (2019). Habilitant był członkiem zespołu, który otrzymał nagrodę prestiżową nagrodę Emerald's Real Impact Award. Nagroda została przyznana w uznaniu za praktyczne znaczenie badań w obszarze pomiaru oprogramowania oraz wdrożone rozwiązania w przemyśle ICT.

### 3. Dorobek dydaktyczny i organizacyjny

Dr Mirosław Ochodek prowadził różnorodne zajęcia dydaktyczne (wykłady, zajęcia laboratoryjne i projektowe) z przedmiotów w obszarze inżynierii oprogramowania. Zajęcia te zostały wysoko ocenione przez studentów w anonimowych badaniach ankietowych. Wypromował 50 dyplomatów (studia I oraz II stopnia). Współpraca ze studentami zaowocowała ośmioma wspólnymi publikacjami. Habilitant pełnił również rolę promotora pomocniczego w trzech zakończonych pomyślnie przewodach doktorskich. Prowadził również, jako starszy wykładowca, zajęcia dydaktyczne (architektura oprogramowania, zwinne metodyki wytwarzania) w trakcie półrocznego pobytu na Chalmers University of Technology. Jest jednym z organizatorów Studia Rozwoju Oprogramowania – nauczania w formie realizacji projektu. Zespoły realizujące projekty są tworzone ze studentów pierwszego i drugiego stopnia. Ważnym

elementem jest w tej metodzie współpraca z firmami. Takie podejście stanowi przybliżenie do warunków przemysłowego wytwarzania oprogramowania. Działalność ta została trzykrotnie nagrodzona przez Rektora Politechniki Poznańskiej. Dr M. Ochodek wdrożył ponadto szereg innowacyjnych metod wspomagających nauczanie Inżynierii Oprogramowania. Należy tu wspomnieć o metodzie oceny architektury oprogramowania przez odgrywanie ról (*technical drama*), czy też połączenie treści wykładów laboratoriów i projektów przez realizację projektu (*capstone project*) wymagającego rozwiązania różnorodnych problemów.

Dr Mirosław Ochodek pełnił również szereg funkcji organizacyjnych. Był kierownikiem zadania w projekcie TECH-INFO odpowiedzialnym za organizację staży studenckich i projektów wykonywanych wspólnie z firmami wytwarzającymi oprogramowanie. W latach 2012-2015 był Sekretarzem Komitetu ds. Współpracy z IFIP, działającym przy Komitecie Informatyki PAN. Ponadto był członkiem komitetów organizacyjnych konferencji CEE-SET 2007, KKIO 2014, IFIP Doctoral Seminar on Software-Intensive Systems 2015. Był również członkiem komitetów programowych kilkunastu konferencji międzynarodowych. Zorganizował dla firm szereg warsztatów i szkoleń w obszarze inżynierii oprogramowania. W ramach współpracy z przemysłem wykonał wiele ekspertyz, opracował poradniki w zakresie szacowania pracochłonności i pomiaru rozmiaru oprogramowania. Uzyskał również wiele ważnych specjalistycznych certyfikatów w obszarze zarządzania projektami i inżynieria oprogramowania.

#### 4. Wniosek końcowy

W ocenianym osiągnięciu naukowym Habilitant zaproponował spójny zestaw komplementarnych metod wspomagających szacowania pracochłonności oprogramowania. Metody te wspierają czynności w trzech obszarach: ekstrakcja wymagań funkcjonalnych i niefunkcjonalnych, pomiar rozmiaru (pracochłonności) oprogramowania oraz pozyskiwanie i „czyszczenie” danych o projektach historycznych. Proponowane rozwiązania zostały pozytywnie zweryfikowane na przykładach różnorodnych projektów typu open source jak również wytwarzanych przemysłowo. Dodatkowo zostało wytworzone oprogramowanie (CCFLex, SimSAX, LegacyPro) implementujące wybrane metody. Uzyskane wyniki są oryginalne i wartościowe, szczególnie ze względu na istotność dokładnej oceny pracochłonności, w warunkach ogólnych i bywa niekompletnych danych w początkowej fazie projektu lub cyklu. Wymagało to użycia różnych podejść (przetwarzania języka naturalnego, uczenie maszynowe), czy też pomysłowej modyfikacji istniejących metod (HAZOP, COSMIC). Wyniki zostały opublikowane w znaczących czasopismach o zasięgu międzynarodowym.

Publikacje niewymienione w zestawie określającym osiągnięcie naukowe stanowią merytoryczne uzupełnienie i w niewielkim stopniu rozszerzają (testowanie, innowacyjne metody nauczania inżynierii oprogramowania) zakres tematyczny. Świadczy to o ugruntowanej tematyce i konsekwentnej realizacji poszczególnych etapów badań. Należy podkreślić, że międzynarodowa pozycja jest potwierdzona przez liczne recenzje wykonane dla ważnych czasopism – łącznie 64 zarejestrowane w bazie Publons.

W dorobku dydaktycznym należy wyróżnić wdrożenie innowacyjnych metod w nauczanie inżynierii oprogramowania. Działalność organizacyjna dr. inż. Mirosława Ochodka charakteryzuje się różnymi formami współpracy z przemysłem wytwarzającym oprogramowanie. Jest to istotny element budowania środowiska współpracy między uczelnią a przemysłem.

Oceny końcowe osiągnięcia naukowego, dorobku po uzyskaniu stopnia doktora oraz działalności okołonaukowej pozwalają na jednoznaczne stwierdzenie, że spełnione są wymagania dotyczące uzyskania stopnia naukowego doktora habilitowanego (Ustawa z dnia 14 marca 2003 o stopniach i tytułach w zakresie sztuki z późniejszymi zmianami), co prowadzi do silnego wniosku o dopuszczenie dr. inż. Mirosława Ochodka do kolejnego etapu postępowania habilitacyjnego w celu nadania stopnia naukowego doktora habilitowanego Nauk inżyniersko-technicznych w dyscyplinie Informatyka techniczna i telekomunikacja.



Tomasz Szmuc