



Warszawa 2.11.2021

Dr hab. inż. Ilona Bluemke  
Instytut Informatyki, PW  
Warszawa, Nowowiejska 15/19

## **Recenzja rozprawy habilitacyjnej oraz dorobku naukowego dr inż. Mirosława Ochodka**

Recenzja niniejsza została przygotowana na zlecenie Dziekana Wydziału Informatyki i Telekomunikacji Politechniki Poznańskiej, Pana prof. dr hab. inż. Andrzeja Jaszkiwicza (pismo z dnia 2 września 2021), w związku z powołaniem (decyzja Rady Doskonałości Naukowej oraz uchwały Rady Dyscypliny Informatyka Techniczna i Telekomunikacja) na członka komisji i recenzenta, w postępowaniu w sprawie nadania stopnia doktora habilitowanego dr inż. Mirosławowi Ochodkowi, na podstawie osiągnięcia naukowego pt. **Metody wspierające proces szacowania pracochołności w projektach informatycznych.**

### **Ocena osiągnięcia naukowego w postaci cyklu publikacji**

Recenzja osiągnięcia naukowego została wykonana na podstawie dokumentacji, w tym załączonego cyklu 9 publikacji powiązanych tematycznie (zgodnie z art. 219 ust. 1. pkt 2b Ustawy) oznaczonych w materiałach A1-A9 i pokazanych poniżej:

A1. 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

A2. 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

A3. 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

A4. 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

A5. 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

A6. 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

A7. 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

A8. 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

A9. 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

Zgłoszone publikacje stanowią efekt badań Habilitanta prowadzonych w latach 2015-2020. Badania te dotyczą ważnego obszaru informatyki, jakim jest inżynieria oprogramowania ze szczególnym uwzględnieniem problematyki szacowania pracochłonności. Szacowanie pracochłonności jest niezwykle ważne w przedsiębiorstwach informatycznych. Przeszacowanie może spowodować odrzucenie projektu a niedoszacowanie stanowi ryzyko dla realizacji projektu. Mimo że prace dotyczące szacowania pracochłonności są prowadzone od bardzo wielu lat, to ciągle brakuje efektywnych rozwiązań i wybór takiego obszaru badań jest uzasadniony, a uzyskanie nowych rozwiązań może mieć istotne zastosowanie w praktyce. Problemem metod szacowania pracochłonności są braki informacji np. co do wymagań,



zakresu oraz informacji historycznych o podobnych przedsięwzięciach. Habilitant zajął się w swoich pracach właśnie tymi problemami.

Prace stanowiące cykl publikacji zostały przez Habilitanta podzielone na trzy grupy:

- 1- dotyczące procesu pozyskiwania wymagań funkcjonalnych i poza funkcjonalnych w celu obniżenia poziomu niepewności co do zakresu projektu [A6, A7, A9],
- 2- dotyczące pomiaru rozmiaru oprogramowania [A1, A3, A8] oraz
- 3- dotyczące procesu pozyskiwania i „czyszczenia” zbiorów danych historycznych o projektach informatycznych [A2, A4, A5].

Krótką charakterystyką i oceną poszczególnych prac:

W pracy [A7], na podstawie przeglądu literatury zidentyfikowano 31 praktyk projektowych zwinnej inżynierii wymagań. Wykonano badanie ankietowe, z udziałem 138 respondentów zajmujących się wytwarzaniem oprogramowania, które pozwoliło na ocenę ważności tych praktyk. Autorzy opracowali także algorytm rankingowy. Analiza wyników pozwoliła na wyciągnięcie wniosku, że największy nacisk zwinnej inżynierii wymagań kładziony jest na opracowanie „jasnej wizji produktu oraz na współpracę z interesariuszami w ramach krótkich iteracji, tak aby często otrzymywane sprzężenie zwrotne odnośnie wymagań mogło napędzać i sterować rozwojem produktu”. Ranking praktyk projektowych zwinnej inżynierii wymagań jest osiągnięciem Autorów, gdyż nie pojawia się w innych dostępnych opracowaniach. Najważniejsze praktyki zwinnej inżynierii wymagań zidentyfikowane przez Autorów to: dostępność klienta, określenie wspólnej wizji systemu, codzienne spotkania zespołu, organizacja spotkań „demo”, łatwy dostęp do wymagań oraz testowalność wymagań.

W pracy [A9] została zaproponowana metoda HAZOP for use cases (H4U), wspomagająca identyfikację zdarzeń w scenariuszach przypadków użycia. Zostały wykonane dwa eksperymenty (ang. controlled experiment). Pierwszy z eksperymentów został przeprowadzony z udziałem 18 studentów a drugi z udziałem 82 osób, które zawodowo zajmują się wytwarzaniem oprogramowania. Celem badania było sprawdzenie czy i w jakim stopniu metoda H4U może wspomagać pracę osób nieposiadających lub posiadających znikomą doświadczenie w pozyskiwaniu i dokumentowaniu wymagań. Efektywność użycia metody H4U we wspomaganie identyfikacji zdarzeń była porównywana z efektywnością podejścia ad hoc (tzn. dowolnego podejścia zastosowanego przez uczestnika eksperymentu). Wyniki eksperymentów pokazały, że podejście do identyfikacji zdarzeń bazujące na metodzie HAZOP pozwala zidentyfikować większą liczbę zdarzeń niż podejście ad hoc natomiast wydłuża to czas przeglądu scenariuszy w poszukiwaniu zdarzeń. Eksperymenty pokazały także, że osiągnięcie wysokiego poziomu kompletności zdarzeń w przypadkach użycia jest w ogólności zadaniem trudnym. Średnia kompletność zdarzeniowa (definiowana jako stosunek liczby zidentyfikowanych zdarzeń do liczby zdarzeń w tzw. złotym rozwiązaniu wzorcowym) mieściła się w przedziale od 0,15 do 0,26 co wskazuje na wysokie ryzyko niekompletności przypadków użycia i może prowadzić do niedoszacowania pracochłonności.

Wymagania niefunkcjonalne są trudne do pozyskania i do poprawnego wyspecyfikowania. Rekomenduje się używanie szablonów specyfikacji wymagań niefunkcjonalnych, aby



zwiększyć kompletność wymagań i zapewnić wysoką jakość ich specyfikacji. Szablon wymagania określa domyślną postać wymagania, uwzględnia przy tym fragmenty opcjonalne i wymagające modyfikacji, tak aby można go było dopasować do konkretnego kontekstu użycia. Zbiór szablonów tworzy katalog szablonów wymagań. W pracy [A6] dokonano empirycznej oceny zastosowania katalogu wymagań niefunkcjonalnych z perspektywy jego użyteczności oraz kosztów utrzymania. Przeprowadzono kontrolowany eksperyment mający na celu zbadanie użyteczności katalogu szablonów do ich pozyskiwania i specyfikowania. W eksperymencie wzięło udział 107 uczestników (studentów), których zadaniem było wyspecyfikowanie wymagań niefunkcjonalnych dla przedstawionego im systemu typu e-commerce. Wyniki eksperymentu pokazały, że użycie katalogu szablonu wymagań pozwala zwiększyć zarówno jakość jak i kompletność specyfikowanych wymagań. W następnym kroku przeprowadzono badanie symulacyjne na zbiorze 2231 wymagań niefunkcjonalnych z 41 projektów informatycznych. Celem badania było zbadanie jak charakterystyki katalogu szablonów wymagań (np. koszt utrzymania) mogą zmieniać się wraz z prowadzeniem kolejnych projektów informatycznych w organizacji. Zaproponowano pojęcie dojrzałości katalogu, który określa moment, w którym zawartość katalogu stabilizuje się pomiędzy kolejno realizowanymi projektami.

W pracach [A6] i [A9] udział dr Ochodka był moim zdaniem (na podstawie oświadczenia współautorów) znacznie mniejszy niż w pracy [A7] i sprowadzał się głównie do analizy danych z eksperymentów.

Prace [A1, A3, A8] dotyczą wspierania procesu pomiaru rozmiaru oprogramowania. Interesująca jest praca [A8] w której dr Ochodek (jedyny autor tej pracy) zaproponował dwie metody pozwalające szacować rozmiar funkcjonalny tylko na podstawie tytułów przypadków użycia tj. Average Use-Case Goal-aware Approximation (AUCG) oraz Bayesian Network Use-Case Goal Approximation (BN-UCGAIN). W pracy tej zaproponowano taksonomię 13 typów celów przypadków użycia. W pierwszym etapie metod określany jest, na podstawie tytułu przypadku użycia, typ celu zgodnie z zaproponowaną taksonomią. W drugim etapie następuje szacowanie rozmiaru funkcjonalnego na podstawie typu celu przypadku użycia oraz danych historycznych z wcześniej realizowanych projektów. W pracy [A8] zaproponowano także automatyczne podejście do określania celu przypadku użycia na podstawie jego tytułu. Podejście to bazuje na użyciu narzędzi przetwarzania języka naturalnego i algorytmach uczenia maszynowego. Dokładność szacowania rozmiaru funkcjonalnego proponowanych metod została oceniona na zbiorze 26 projektów informatycznych i porównana z dokładnością szacowania na podstawie średniego rozmiaru przypadku użycia (AUC) oraz z dokładnością metody zaproponowanej przez Hussaina, Kosseim i Ormandjiewę (HKO)<sup>1</sup>. Przeprowadzone badania wykazały, że metody AUCG i BN-UCGAIN pozwoliły szacować rozmiar funkcjonalny z większą dokładnością niż metody konkurencyjne (o około 11% do 24%).

W pracy [A1] przedstawiono kolejną metodę szacowania rozmiaru funkcjonalnego COSMIC na podstawie nazw/tytułów przypadków użycia o nazwie DEEP-COSMIC-UC. Jest to metoda jednoetapowa. Na wejściu do modelu DEEP-COSMIC-UC podawany jest tekst tytułu

---

<sup>1</sup> I. Hussain, L. Kosseim, O. Ormandjiewa, Approximation of COSMIC functional size to support early effort estimation in Agile, Data & Knowledge Eng. 85, 2–14, 2013.



przypadku użycia a wyjście stanowi jego szacowany rozmiar funkcjonalny COSMIC. W modelu używane są także wstępnie trenowane modele wektorowej reprezentacji słów (ang. word embeddings), pozwalające zamienić tekst tytułu przypadku użycia na postać wektora liczb. Dokładność szacowania nowego modelu została zbadana na zbiorze 438 przypadków użycia z 27 projektów informatycznych. Proponowany model był w stanie szacować rozmiar COSMIC pojedynczych przypadków użycia z wyższą dokładnością niż model AUC (o około 20%) oraz dwa zaproponowane w pracy [A8] modele AUCG oraz BN-UCGAIN (o około 5-7%). Dodatkowo, statystycznie poprawa dokładności szacowania była zauważalna już po szacowaniu dziesięciu przypadków użycia. Istotną zaletą metody DEEP-COSMIC-UC w stosunku do metod AUCG oraz BN-UCGAIN jest to, że nie wymaga ona zbierania danych historycznych innych niż tytuły przypadków użycia wraz z ich rozmiarem funkcjonalnym. W konsekwencji metoda ta może zostać użyta nawet jeśli w bazie danych historycznych nie gromadzimy informacji o typach celów przypadków użycia.

W pracy [A3], opisano znaczne usprawnienia narzędzia CCFlex<sup>2</sup> (Flexible Code Counter/Classifier) które pozwala na tworzenie instrumentów pomiarów dla specjalizowanych miar rozmiaru oprogramowania tylko na bazie dostarczonych przykładów pomiarów. Zbadano możliwość jego zastosowania do identyfikacji (zliczania) linii kodu źródłowego, które naruszają wewnętrzne wytyczne stosowane w dwóch dużych skandynawskich firmach wytwarzających oprogramowanie. Linie naruszające takie zasady mają istotne znaczenie z perspektywy szacowania pracochłonności i kosztów utrzymania aplikacji (ang. maintenance effort). Przeprowadzono szerokie badania kodu przemysłowego i open source. Wyniki tych badań pokazują, że proponowane podejście do pomiaru rozmiaru fizycznego oprogramowania może być stosowane w warunkach przemysłowych do tworzenia specjalizowanych definicji miary LOC, które w dalszym etapie mogą zostać użyte do budowy modeli szacowania pracochłonności, dedykowanych do konkretnych typów zadań w projektach informatycznych.

Trzecia grupa prac [A2, A4, A5] dotyczy procesu pozyskiwania i „czyszczenia” zbiorów danych historycznych o projektach informatycznych. Jest to ważne zagadnienie w inżynierii oprogramowania gdyż dane te są wykorzystywane między innymi do kalibracji modeli szacowania pracochłonności.

W pracy [A5] została zaproponowana nowa miara oceny podobieństwa projektów informatycznych o nazwie SimSAX. Ocena podobieństwa dokonywana jest na podstawie analizy tzw. profilu zgłoszeń o defektach w oprogramowaniu (ang. defect-inflow profile). Profile defektów transformowane są do sekwencji symboli oraz porównywane, by wykryć powtarzające się, podobne podsekwencje tzw. motywy. Została opracowana procedura kalibracji parametrów pomiaru podobieństwa z użyciem miary SimSAX. Autorzy [A5] przeprowadzili kalibrację metody pomiarowej na podstawie danych z dwóch dużych projektów przemysłowych. Moje wątpliwości budzi jakość kalibracji robionej tylko na dwóch projektach. Wykorzystując informacje pochodzące z wywiadów z uczestnikami tych

---

<sup>2</sup> M. Ochodek, M. Staron, D. Bargowski, W. Meding, R. Hebig. Using machine learning to design a flexible LOC counter, In 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), pp. 14-20. IEEE, DOI: 10.1109/MALTESQUE.2017.7882011, 2017.



projektów byli w stanie określić rzeczywiste podobieństwa pomiędzy projektami w trakcie ich realizacji oraz skonfrontować je z podobieństwami wskazywanymi przez SimSAX. Przeprowadzono także analizę podobieństw pięciu projektów przemysłowych i sześciu projektów open source. Przeprowadzone analizy wykazały, że miara SimSAX może zostać użyta do znajdowania podobieństw między projektami, zwłaszcza gdy wyszukiwane są długie motywy. Zaproponowana metoda kalibracji pozwala dostosować miarę do konkretnego kontekstu organizacji czy też projektu.

W pracy [A2] zaproponowana została metoda o nazwie LegacyPro, która wykorzystuje miarę SimSAX do znajdowania „zaszłości” w sposobie pracy zespołów programistycznych po przeprowadzeniu transformacji w organizacji (np. po wprowadzeniu innych metod zarządzania projektami). Metoda LegacyPro może także mieć zastosowanie do ciągłego monitorowania sposobu pracy zespołów projektowych. W przypadku wykrycia znaczących zmian w sposobie pracy należałoby powtórzyć analizę danych historycznych pod kątem ich adekwatności do nowej sytuacji i ew. dokonać re-kalibracji używanej metody szacowania pracochłonności. LegacyPro bazuje na wykrywaniu dwóch typów motywów w profilach defektów – motywów wewnętrznych w ramach danego projektu (ang. internal motifs) oraz motywów współdzielonych, czyli występujących w następujących po sobie projektach (ang. shared motifs). Motywy wewnętrzne pozwalają określić na ile powtarzalny jest sposób pracy danego zespołu, natomiast motywy współdzielone pozwalają wykrywać zmiany (lub ich brak) na poziomie całej organizacji. Proponowana metoda była sprawdzana na danych z czterech projektów przemysłowych i pięciu projektów open source, charakteryzujących się bardzo długim czasem rozwoju. Metoda pozwoliła wykryć podobieństwa oraz różnice w sposobie rozwoju tych projektów. Badania symulacyjne pokazały, że istnieje możliwość zamiany standardowego wyjścia metody LegacyPro do postaci klasyfikatora binarnego, który można użyć do automatycznego wykrywania zmian w sposobie realizacji przedsięwzięć informatycznych. W kontekście dostosowywania metod szacowania pracochłonności użycie metody może pozwolić na utrzymanie jednorodności i aktualności zbiorów danych historycznych gromadzonych w ramach danej organizacji.

W pracy [A4] zaproponowano metodę pozwalającą filtrować „rzeczywiste” projekty w zbiorze repozytoriów kodu oraz narzędzie PHANTOM (Project History Analysis of Time-Series Method) do tego celu. Metoda wykorzystuje algorytmy uczenia maszynowego bez nadzoru i analizę częstotliwości zmian w historii rozwoju projektów na podstawie danych z logów Git. Porównanie metody i narzędzia PHANTOM z metodą i narzędziem zaproponowanym przez Munaiah i innych<sup>3</sup> pokazało że PHANTOM jest zdecydowanie bardziej wydajne i dokładne. Porównania wykonano na dużych repozytoriach projektów.

Podsumowując, osiągnięcia Habilitanta w ramach w ramach pierwszej grupy uważam za drugorzędne. Za główne wartościowe osiągnięcia uznaję te związane z grupa drugą oraz trzecią. Oryginalnymi osiągnięciami są tutaj:

---

<sup>3</sup> Munaiah N, Kroh S, Cabrey C, Nagappan M, Curating GitHub for engineered software projects. Empirical Software Engineering, vol. 22, pp. 3219–3253, <https://doi.org/10.1007/s10664-017-9512-6>, 2017



- Metody szacowania rozmiaru funkcjonalnego tylko na podstawie tytułów przypadków użycia tj. Average Use-Case Goal-aware Approximation (AUCG) oraz Bayesian Network Use-Case Goal Approximation (BN-UCGAIN).
- Metoda szacowania rozmiaru funkcjonalnego COSMIC na podstawie nazw/tytułów przypadków użycia o nazwie DEEP-COSMIC-UC
- Identyfikacja / zliczanie linii kodu, które naruszają określone wytyczne.
- Miara oceny podobieństwa projektów informatycznych - SimSAX.
- zaproponowano metodę pozwalającą filtrować „rzeczywiste” projekty w zbiorze repozytoriów kodu oraz narzędzie PHANTOM (Project History Analysis of Time-Series Method)

Należy przy tym podkreślić pragmatyczne, komplementarne podchodzenie do podejmowanych problemów – koncepcjom towarzyszą prace nad skonstruowaniem narzędzi wspomagających proponowane podejścia oraz eksperymentalna weryfikacja stawianych tez, często na rzeczywistych, przemysłowych projektach.

Całość cyklu ma w znacznym stopniu charakter jednotematyczny, spaja je problematyka szacowania pracochłonności w projektach informatycznych. Opiniowany cykl publikacji oceniam pozytywnie i uważam że w wystarczającym stopniu spełnia wymagania stawiane rozprawom habilitacyjnym.

Osiem z przedstawionych w cyklu dziewięciu prac, to prace wielu autorów. We wniosku znajdują się informacje o wkładzie merytorycznym Habilitanta i oświadczenia współautorów. Trochę jednak brakuje mi informacji „ilościowych” - na ile procent współautorzy oceniają wkład dr M. Ochodka. Kilka nazwisk powtarza się wielokrotnie (np. Staron- 5, Kopczyńska - 3, Nawrocki -2), świadczy to o dobrze pracujących zespołach badawczych, jednak właśnie w tych przypadkach interesowały by mnie także dane „ilościowe” .

### **Ocena dorobku naukowego**

W wykazie opublikowanych rozdziałów w monografiach naukowych znajduje się 25 pozycji, są to referaty z konferencji międzynarodowych i polskich.

Artykuły w czasopismach naukowych włącznie z pracami znajdującymi się w cyklu to 17 pozycji. Oglądając dorobek naukowy opisany przez Habilitanta, trudno mi było rozróżnić prace po doktoracie. Moje wątpliwości rozwiązała analiza dorobku naukowego wykonana przez Bibliotekę Politechniki Poznańskiej. Z tej analizy wynika, że po doktoracie dr M. Ochodek opublikował 11 prac w czasopismach naukowych o łącznej liczbie punktów 1320, i współczynniku Impact Factor = 26,5 a także artykuły na konferencjach o łącznej liczbie punktów 854. Są to moim zdaniem dobre wyniki punktowe.

Kilka nazwisk współautorów występuje często w tych publikacjach (np. Nawrocki, Staron, Jurkiewicz, Kopczyńska), szkoda że Habilitant nie zamieścił wyjaśnienia dlaczego te nazwiska tak często występują, mogę domniemywać że były to zespoły badawcze realizujące projekty.

Baza Web of Science indeksuje 26 publikacji dr Ochodka, cytowanych 107 razy (bez autocytowań), indeks Hirscha równy 5. Cytowania zamieszczone są w 91 artykułach cytujących (bez autocytowań).

Baza Google Scholar indeksuje 49 publikacji dr Ochodka, cytowanych w latach 2006-2021, z czego 30 publikacji jest cytowanych łącznie 447 razy (w tym autocytowania), a od roku 2016 publikacje cytowane są 283 razy a indeks Hirscha jest równy 11.

Baza Scopus indeksuje 39 publikacji dr Ochodka, cytowanych 263 razy (w tym 23 autocytowania) w latach 2006-2021, indeks Hirscha równy jest 8. Cytowania zamieszczone są w 209 artykułach cytujących (w tym 23 autocytowania).

Dorobek publikacyjny Habilitanta nie jest zbyt liczny ale jest to dorobek dobrej jakości o czym świadczą wysokie wartości punktowe, spora liczba cytowań i dość dobre wartości indeksu Hirscha. Jestem przekonana że uzyskany dorobek publikacyjny, na tle dorobku innych zakończonych przewodów habilitacyjnych, można uznać za wystarczający.

### **Osiągnięcia dydaktyczne**

Dorobek dydaktyczny jest lakonicznie opisany we wniosku (mniej niż jedna strona) toteż trudno go dokładnie ocenić. Habilitant podaje jedynie że prowadzone zajęcia dydaktyczne dotyczą inżynierii oprogramowania i że są wysoko oceniane przez studentów. We wniosku można znaleźć także informację, że wypromował od 2011 roku ponad 50 dyplomantów. Myślę, że tak duża liczba dyplomantów wynika z prowadzenia zespołów projektowych w Studio Rozwoju Oprogramowania (Software Development Studio) i promowania prac wieloautorskich. Prowadzenie zajęć dydaktycznych w obszarze badań naukowych na pewno wpływa na ich dobrą jakość. Jednak we wniosku brakuje dokładniejszych informacji o prowadzonej dydaktyce np. czy Habilitant wprowadził nowe przedmioty itp. Z informacji o zatrudnieniu dr Ochodka wynika, że już w 2008 roku był zatrudniony jako asystent więc do 2021 roku musiał prowadzić wiele zajęć dydaktycznych i szkoda że pewnych informacji o tych zajęciach nie zamieścił we wniosku.

**Ze względu na brak precyzyjnych danych we wniosku, nie jestem w stanie ocenić dorobku dydaktycznego.**

### **Osiągnięcia organizacyjne, współpraca międzynarodowa i inne**

Dorobek organizacyjny też jest skrótowo przedstawiony. Dr Mirosław Ochodek uczestniczył w organizacji kilku konferencji, recenzował artykuły w dobrych czasopismach o tematyce związanej z inżynierią oprogramowania.

Dr Mirosław Ochodek współpracuje z uniwersytetem w Goteborgu a wyniki tej współpracy można znaleźć w publikacjach, także tych należących do cyklu.



Dr Mirosław Ochodek posiada kilka certyfikatów w obszarze inżynierii oprogramowania. Współpracuje z firmami branży IT.

Biorąc pod uwagę powyższe pozytywnie oceniam dorobek organizacyjny, współpracę międzynarodową i działalność popularyzatorską.

### **Inne uwagi**

Niezbyt podoba mi się sposób napisania wniosku. Pomieszczone są informacje o dydaktyce i współpracy międzynarodowej. Opisy dydaktyki na Politechnice Poznańskiej i na University of Gothenburg zajmują podobną ilość miejsca, a czas pracy w obu instytucjach różni się znacząco. Opis dorobku naukowego jest mało czytelny i gdyby nie opracowanie Biblioteki Politechniki Poznańskiej trudno by mi było ten dorobek ocenić.

Mam też zastrzeżenia do opisu cyklu publikacji. Habilitant opisuje zawartość poszczególnych prac w ramach trzech podgrup. Wolałabym by wyraźniej zaznaczył co nowego wnosi dana praca w stosunku do innych rozwiązań.

### **Podsumowanie**

Biorąc pod uwagę osiągnięcia przedstawionego cyklu publikacji, odpowiednika rozprawy habilitacyjnej oraz uzyskany dorobek naukowy, organizacyjny i współpracę międzynarodową uważam, że spełnione są wymagania zawarte *Ustawie* i na tej podstawie wnoszę o dopuszczenie dr inż. Mirosława Ochodka do dalszych etapów postępowania habilitacyjnego.

*Ilona Blumbe*