Prof. dr hab. inż. **Kazimierz Subieta**
Instytut Podstaw Informatyki PAN
ul. Ordona 21, 01-237 Warszawa
oraz
Polsko-Japońska Wyższa Szkoła Technik Komputerowych
ul. Koszykowa 86, 02-008 Warszawa
subieta@pjwstk.edu.pl

Warszawa, 9 listopada 2010 r.

# Opinia

dotycząca rozprawy habilitacyjnej dr inż. **Barbary Begier**

**„Doskonalenie jakości oprogramowania poprzez włączenie użytkowników w proces jego wytwarzania"**

**Dane formalne rozprawy habilitacyjnej i jej kontekstu:**

- Przewód habilitacyjny toczy się przed Radą Wydziału Informatyki Zachodniopomorskiego Uniwersytetu Technicznego w Szczecinie.
- Rozprawa ma postać książki wydanej nakładem Wydawnictwa Politechniki Poznańskiej, Poznań 2007, ISSN 0551-6528, ISBN 978-83-7143-363-4. Dostępna w Księgarni Politechnik, Poznań, ul. Piotrowo 3.
- Książka zawiera 6 rozdziałów (razem 211 stron). Bibliografia zajmuje 13 stron (259 pozycji cytowanych). Książka jest napisana w języku polskim i zawiera streszczenie w języku angielskim (1 strona). Tekst rozprawy jest prezentowany na 182 stronach. Rozprawa zawiera także 4 załączniki, razem 7 stron.
- Pozostałe przedstawione mi dokumenty są następujące:
  - Kopia dyplomu doktorskiego;
  - Życiorys naukowy (3 str.);
  - Kwestionariusz osobowy (2 str.);
  - Curriculum Vitae (2 str.);
  - Autoreferat (17 str.);
  - Pełny wykaz publikacji po doktoracie (71 pozycji);
  - Wykaz publikacji punktowanych (podzbiór pełnego wykazu, 31 pozycji);
  - Wykaz publikacji napisanych wspólnie z innymi autorami (podzbiór pełnego wykazu, 17 pozycji);
  - Wykaz ważniejszych opracowań niepublikowanych (18 pozycji);

- Wykaz niektórych prac naukowo badawczych zastosowanych w praktyce (4 pozycje);
- Oświadczenia Autorki oraz innych autorów odnośnie procentowego wkładu w przygotowanie publikacji (41 oświadczeń);
- 14 wybranych publikacji Autorki związanych tematycznie z przedłożoną rozprawą habilitacyjną.

**Omówienie dziedziny przedmiotowej rozprawy habilitacyjnej**

Przedmiotem rozprawy habilitacyjnej dr Barbary Begier jest inżynieria oprogramowania. Dziedzina ta jest w tej chwili kluczowa dla bardzo wielu współczesnych zastosowań informatyki, w tym szczególnie zastosowań związanych z biznesem. Inżynieria oprogramowania zajmuje się wszystkimi aspektami tworzenia dużych aplikacji informatycznych, w tym aspektami technicznymi, jakościowymi, organizacyjnymi, wydajnościowymi, ekonomicznymi, ergonomicznymi, społecznymi, psychologicznymi, etycznymi, itd. Rynek związany z budową dużych systemów informatycznych jest szacowany w skali światowej na setki miliardów dolarów rocznie. Jeszcze wyżej są szacowane efekty ekonomiczne wynikające z ich zastosowań. Fakty te posiadają następujące doniosłe konsekwencje dla środowiska naukowego:

- Pragmatyczna waga badań i odkryć w dziedzinie inżynierii oprogramowania jest nieporównywalnie większa niż w innych dziedzinach naukowych, gdyż odkrycia te mogą być powiązane z bezpośrednimi potrzebami społeczeństwa oraz konkretnymi efektami ekonomicznymi i cywilizacyjnymi.
- Ważne potrzeby ekonomiczne, społeczne i cywilizacyjne stojące za tą dziedziną przyciągnęły do niej bardzo wielu naukowców i specjalistów, czego efektem jest lawina artykułów, książek, dokumentacji i innych materiałów. Bibliografie z zakresu inżynierii oprogramowania obejmują setki tysięcy pozycji, stanowiąc poważną barierę dla nowicjuszy. Proporcje wysiłku naukowca, chcącego być na bieżąco z danym tematem, rozkładają się zupełnie inaczej, niż to ma miejsce w innych dyscyplinach naukowych. Można z grubsza oszacować, że ponad 90% pracy naukowca pracującego w tej dziedzinie musi być poświęcone temu, co na dany temat napisali inni, zaś co najwyżej 10% może być poświęcone własnym pomysłom i badaniom.
- Spontanicznie rozwijane technologie, lawina często przypadkowych i partykularnych metodologii, implementacji, teorii, pojęć, standardów i innych pomysłów spowodowały niemały chaos, w którym trudno zorientować się nawet specjalistom. Nie istnieje jednoznaczna i klarowna terminologia, codziennie powstaje wiele nowych terminów, różne terminy mają zbliżone znaczenie, ten sam termin oznacza co innego w nieco różnym kontekście (w innym systemie), niektóre terminy umierają. Akademicki postulat budowy „podstaw", „zasad" i „teorii" jest w tej dziedzinie nie do zrealizowania, gdyż postęp wynalazczy jest zbyt duży, aby mieć nawet cień gwarancji, że „podstawy" staną się efektywną wiedzą, „zasady" nie są skutkiem braku wyobraźni i myślenia życzeniowego, zaś „teorie" nie są pseudo-naukowymi utopiami.
- W tej sytuacji jedynym skutecznym sposobem naukowego działania w tej dziedzinie jest włączenie się do nurtu wynalazczości. Każdy wynalazek musi być poparty obiektywnym świadectwem skuteczności w praktycznych sytuacjach, a tego nie da się uzyskać w inny

sposób niż przez sprawdzenie pomysłu w rzeczywistej skali problemu. Waga implementacji w tej dziedzinie jest tak duża, że popularne staje się hasło „jeden działający prototyp jest wart 100 publikacji naukowych". Budowa aplikacji informatycznych przy pomocy proponowanych narzędzi i metodologii, próby zastosowania ich w praktyce, porównanie ich z innymi podobnymi propozycjami jest sprawą kluczową w tworzeniu świadomości problemu, pozwala zrozumieć, gdzie jest problem a gdzie go nie ma, pozwala uniknąć czepiania się utopii i pseudo-rozwiązań.

- Powstały setki użytecznych systemów, narzędzi, technologii, języków i metodologii związanych z inżynierią oprogramowania oraz tysiące ich zastosowań, których budowa, zrozumienie aspektów projektowych i użytkowych, jest często niezbędne do tego, aby można było uważać daną osobę za kompetentnego specjalistę w tym zakresie.
- Nasycenie rynku systemami komputerowymi oraz społeczno-ekonomiczna waga ich zastosowań powoduje konieczność istnienia w kraju dostatecznie mocnej bazy naukowej i dydaktyczno-weryfikacyjnej w zakresie szkolenia specjalistów z tego zakresu, ich weryfikacji, oraz weryfikacji związanych z tym projektów lub produktów.

Polskie środowisko naukowe posiada stanowczo za mało samodzielnych pracowników naukowych specjalizujących się w dziedzinie inżynierii oprogramowania i aktywnie zaznaczających swoją obecność w społeczności naukowej. Powodów tej sytuacji jest kilka, m.in. zbyt małe nakłady na naukę, ucieczka polskich naukowców na lepiej opłacane posady w biznesie lub w innych krajach, praktycznie brak projektów zmierzających do stworzenia własnych (nawet prototypowych) systemów lub narzędzi. Istotnym czynnikiem jest także ucieczka polskich naukowców w inną tematykę, pozornie bardziej „naukową" (taką jak metody formalne), a w istocie sprzyjającą twórczości o niskiej randze z punktu widzenia potrzeb społeczeństwa. Powodem tych ucieczek jest brak polityki naukowej zmierzającej do preferowania dziedzin rzeczywiście potrzebnych oraz bardzo wysoka bariera kompetencji, którą musi pokonać potencjalny specjalista z zakresu inżynierii oprogramowania, znacznie wyższa, niż ma to miejsce w innych działach informatyki. Samodzielnych pracowników naukowych zajmujących się inżynierią oprogramowania jest o rząd mniej niż samodzielnych pracowników w zakresie np. sztucznej inteligencji. Tymczasem efekty ekonomiczne badań w zakresie sztucznej inteligencji są w istocie żadne, podczas gdy efekty ekonomiczne zastosowań inżynierii oprogramowania można mierzyć miliardami złotych rocznie.

W tym kontekście rozprawa habilitacyjna dr Barbary Begier wyróżnia się pod względem jej wagi dla rozwoju dziedziny inżynierii oprogramowania w Polsce. Wprowadzenie do rozprawy habilitacyjnej oraz dorobek publikacyjny pokazują, że dr Barbara Begier jest kompetentną specjalistką z tego zakresu, która bardzo dobrze orientuje się w stanie sztuki w zakresie inżynierii oprogramowania i jest w stanie nawiązać do najnowszych badań prowadzonych przez krajowe i światowe ośrodki naukowe. Jest ponadto efektywnym dydaktykiem, kierownikiem wielu projektów badawczo-rozwojowych i może być kompetentnym weryfikatorem działalności i projektów naukowo-technicznych w tej dziedzinie. Podane wyżej okoliczności uważam za istotny czynnik awansu dr Barbary Begier na stopień naukowy doktora habilitowanego.

Spośród wielu tematów, którymi zajmuje się inżynieria oprogramowania, dr Barbara Begier podjęła temat związany z włączeniem przyszłych użytkowników w proces wytwarzania oprogramowania. Nie jest to temat nowy, prace na temat konieczności włączenia użytkowników w procesy wytwórcze pojawiły się już w latach 80-tych. Niemniej problem jest na tyle ważny, że nie istnieje jakakolwiek granica, od której można by mówić, że problem

został dostatecznie zbadany, zaś temat został wyczerpany. Problem polega na tym, że przyszły użytkownik oprogramowania rzadko jest świadomy zarówno swoich potrzeb w zakresie wspomagania ze strony narzędzi informatycznych, jak i możliwości, które może (lub nie może) zaoferować informatyka. Użytkownik zwykle nie jest zainteresowany bezpośrednio informatyką, lecz efektami swojego zawodu, np. jako planisty, marketingowca, kadrowca, księgowego, itp. Informatyka może pomóc w zwiększeniu tych efektów, ale użytkownik znający swoje cele biznesowe zwykle nie potrafi przełożyć ich na wymagania lub specyfikację w stosunku do nowego systemu informatycznego. Natomiast użytkownik będzie doskonale świadomy braków lub błędów w systemie informatycznym po tym, kiedy zostanie on stworzony, zainstalowany i wprowadzony w codzienne działania operacyjne w ramach procesów biznesowych. Ta sprzeczność (w popularnej anegdocie: „użytkownik na początku projektu nigdy nie wie czego chce, ale na jego końcu będzie zawsze wiedział, czego nie chciał") stała się przyczyną krytyki popularnego modelu kaskadowego (*waterfall model*), w którym kontakt z użytkownikiem jest na początku procesu projektowego (zbieranie wymagań), później kontakt ten się urywa, aż do przekazania systemu do testów użytkowych i eksploatacji. Ta długa przerwa w kontakcie wykonawcy z użytkownikami jest przyczyną licznych patologii, w szczególności nieudanych projektów, które nie są użytkowane, których jakość budzi zastrzeżenia lub których funkcjonalność zostaje radykalnie obcięta. Ta sprzeczność jest przyczyną powstawania tzw. metodyk iteracyjnych (np. RUP), w których zakłada się powroty do fazy inżynierii wymagań po każdym znaczącym kamieniu milowym w projekcie. Te powroty mają na celu weryfikację dotychczasowych wyników projektów w kontakcie z użytkownikiem, która ma na celu podniesienie jego świadomości co do celów projektu, jego potencjalnej funkcjonalności oraz wymagań niefunkcjonalnych (np. wydajności, skalowalności, ochrony, bezpieczeństwa, itd.). Podobne motywy kierowały autorami tzw. metodyk zwinnych (*agile*), które negują sens tworzenia obszernej dokumentacji analityczno-projektowej (zakładanej w metodykach „ciężkich") na rzecz jak najszybszego wypracowania działającego prototypu, który jest najlepszym sposobem weryfikacji potrzeb użytkownika.

Problem włączenia użytkowników w procesy wytwórcze oprogramowania nie jest łatwy. Wymaga przede wszystkim zbudowania świadomości u zleceniodawcy projektu co do wagi włączenia się użytkownika w proces projektowy i – w związku z tym – traktowanie tego włączenia się użytkownika jako nieodzownego warunku powodzenia projektu oraz jako niezbędnej inwestycji finansowej, kadrowej i czasowej. Z doświadczenia autora recenzji, dość wielu decydentów nie ma tej świadomości, zaś przyszli użytkownicy nie mają żadnej wiedzy ani doświadczeń współpracy z wykonawcami – informatykami. W jednym z projektów, który prowadziłem dla instytucji centralnej w Warszawie, na wywiady była kierowana za każdym razem inna osoba, kompletnie nie orientująca się w przedmiocie wywiadu oraz w poprzednich ustaleniach. To zamieniło ten projekt w koszmar, „marsz ku klęsce". Powstaje zatem zadanie, w jaki sposób można budować taką świadomość i odpowiednie do niej procedury formalne. To jest właśnie przedmiotem rozprawy dr Barbary Begier. W tym względzie rozprawa jest dziełem z pogranicza informatyki oraz psychologii i badania zachowań społecznych. Dr Barbara Begier proponuje określone procedury w ramach nakreślonej przez nią metodologii UID (*User-Involved Development*). Metodologia, czyli sformalizowany przepis postępowania, jest najlepszym sposobem przekonania decydentów o konieczności dodatkowych inwestycji ze strony zamawiającego, gdyż sformalizowana metodologia nie podlega większej dyskusji co do konieczności lub braku konieczności niektórych kroków i działań. Ma także odpowiedni urzędniczy autorytet wymuszający określone zachowania ze strony użytkowników i ich przełożonych. Użytkownicy i decydenci, świadomi wagi swoich

akcji dla powodzenia i jakości projektu, chętnie zastosują się do metodologicznych rygorów oraz poczują się twórcami nowego systemu. Z doświadczeń recenzenta, jest to klucz do powodzenia projektu.

**Omówienie wyników badawczych przedstawionych w rozprawie habilitacyjnej**

W rozprawie habilitacyjnej Autorka proponuje nowe podejście (UID) do włączania użytkowników w procesy wytwórcze oprogramowania i jest ramą koncepcyjną lub wyspecjalizowaną metodyką tworzenia oprogramowania, której zasadniczym celem jest poprawa jakości oprogramowania wyrażana w terminach lepszej specyfikacji wymagań funkcjonalnych i niefunkcjonalnych. Proponuje w tym celu odpowiednie pojęcia, procedury i miary.

W rozdziale 1 Autorka wprowadza czytelnika w podstawowe pojęcia, tematy i metody inżynierii oprogramowania, ze szczególnym uwzględnieniem kwestii standardów w tej dziedzinie, zarządzania jakością oprogramowania, oceną dojrzałości procesów wytwórczych oprogramowania, oraz przyczyn niepowodzeń projektów programistycznych. Omawia także zagadnienia metodologiczne, w tym twarde i zwinne metodyki wytwórcze.

Rozdział 2 przedstawia wizję Autorki dotyczące modeli wytwarzania oprogramowania ukierunkowanego na ciągłe doskonalenie wyrobu i procesu. Omówione są klasy wyrobów programowych, pojęcia dotyczące użytkowników oprogramowania oraz zagadnienia metodologiczne. W tym rozdziale Autorka wprowadza czytelnika w proponowane przez nią podejście UID (*User-Involved Development*). Porównuje także UID do koncepcji SOA (*Servis Oriented Architecture*), ostatniej „modzie" w dziedzinie inżynierii oprogramowania, co do której istnieje wiele nadziei i jeszcze więcej kontrowersji. W rozdziale tym podjęto także próbę sformułowania definicji matematycznych dotyczących opisu procesu składowego w UID. Moim zdaniem, taka „matematyka" nie ma większego sensu, ale to oczywiście rzecz gustu i moich doświadczeń z metodami formalnymi.

W rozdziale 3 Autorka proponuje kryteria i miary oceny jakości oprogramowania przez użytkowników. Jest to bardzo ważny rozdział, gdyż mierzalność elementów procesów wytwórczych jest podstawa przejścia na odpowiedni poziom ich dojrzałości. Lista miar, w zależności od rodzaju projektu oraz mierzonego aspektu jest bardzo długa. Jest dość wątpliwe, że wszystkie one znajdą zastosowanie, ponieważ niektóre pomiary mogą być kosztowne, inne zaś niewykonalne z powodu braku danych wejściowych. Niemniej rzetelna praca badawcza powinna przedstawiać taką listę, zaś wybór konkretnych miar będzie i tak należał do zarządu projektu. Wśród tych miar znalazło się także spełnienie norm etycznych i oczekiwań społecznych, co jest rzadkością w podręcznikach inżynierii oprogramowania, zwykle bardziej skoncentrowanych na kwestiach technokratycznych na niekorzyść humanistycznych.

Rozdział 4 proponuje nową strategię doskonalenia procesu i wyrobu. Na podkreślenie zasługuje fakt, że rozważania te są poparte 18-toma projektami prowadzonymi przez Autorkę w latach 1998-2007. Rozważane jest wykorzystanie sprężenia zwrotnego podczas współpracy z użytkownikami. Omawiany jest także kontekst ustanowienia, prowadzenia i oceny procesu wytwórczego oraz analizę jakości celów przedsięwzięcia. Rozdział ten zajmuje się także problematyką czynnika ludzkiego w procesie wytwórczym, zarządzaniem kompetencjami osób oraz postuluje miary oceny dojrzałości współpracy z użytkownikami.

W rozdziale 5 Autorka przedstawia zalecane procedury postępowania i sposób ich realizacji. Są to konkretne wytyczne dotyczące poznania użytkowników, modelowania i analizy procesów biznesowych klienta, zarządzanie relacjami z klientem i sprzężeniem zwrotnym. Omawia także sposoby opracowania z udziałem użytkowników wymaganych cech oprogramowania oraz kryteriów i miar ich jakości. Przedstawia także metody oceny jakości wyrobu programowego, specyfikacji ulepszeń wyrobu, rozwijanie kompetencji personelu, zarządzanie dokumentami uwzględniające zaangażowanie użytkowników, wspomaganie inspekcji projektów, analizę etyczną, zarządzanie ryzykiem, włączanie pracowników do ulepszania procesu, wewnętrzną oceną jakości procesu wytwórczego oraz kilkoma dalszymi tematami. Jest to rozdział zasadniczy dla rozprawy habilitacyjnej, przedstawiający zarówno erudycję Autorki, umiejętność kompleksowego myślenia o licznych aspektach zadanego tematu. Autorka proponuje własną procedurę oceny dojrzałości relacji z użytkownikami, gdzie wyróżnia 5 poziomów tej dojrzałości i podaje kryteria przynależności procesu wytwórczego do danego poziomu.

Rozdział 6 jest poświęcony praktycznej weryfikacji proponowanych strategii oraz ocenie ich przydatności na podstawie badań. Wykonano szereg działających prototypów dających możliwość weryfikacji proponowanego podejścia na bazie przygotowanych ankiet. Przedstawiono wykonano prototyp systemu CRM dla firmy programistycznej, systemu wspomagającego zarządzania dokumentami, prototyp wspomagający inspekcje oprogramowania oraz zarządzanie ryzykiem. Przedstawiono wnioski i propozycje w zakresie poprawy jakości. Zaproponowano także automatyzację wprowadzania zmian użyciem agentów programowych. Całość sprawia bardzo dobre wrażenie dobrze wykonanej pracy badawczej o znacznym potencjale praktycznym.

Podsumowując, całość propozycji jest spójna i dobrze osadzona w realiach rzeczywistych projektów programistycznych. Pracę czyta się dość lekko, jakkolwiek bardzo specjalistyczna terminologia oraz liczne odniesienia do obecnych trendów i pojęć inżynierii oprogramowania mogą utrudnić jej czytanie dla nowicjuszy. Nie można jednak z tego czynić zarzutu, gdyż praca ta nie jest podręcznikiem akademickim, lecz specjalistyczną monografią, którą można zrozumieć dopiero wtedy, gdy pozna się wiele technik stosowanych w inżynierii oprogramowania, szczególnie metod proponowanych ostatnio przez zwinne metodyki tworzenia oprogramowania.

**Krytyka podejścia i niektórych tez rozprawy habilitacyjnej**

Wydaje się, że habilitantka zbyt optymistycznie traktuje kwestię dodatkowych nakładów związanych z wdrożeniem proponowanych przez nią metod. Niestety, jest to problem dla wielu nowo proponowanych metodyk. Zwykle wdrożenie nowych technik wytwórczych wiąże się z inwestycjami w infrastrukturę oraz niezbędne procesy biznesowe. Nowe metodyki nie udaje się głównie dlatego, że zarządcy firm nie mają pewności, czy te dodatkowe koszty przyniosą wymagany efekt. W przypadku metody proponowanej przez habilitantkę trudno sobie wyobrazić, że jej wdrożenie obędzie się bez znacznych kosztów. Zmiana procesów projektowych będzie wymagać znacznych nakładów na przygotowanie odpowiednich procedur i formularzy, przeprowadzenie szkoleń zarówno przyszłych użytkowników jak i wytwórców oprogramowania, oraz prawdopodobnie szereg dodatkowych czynności. Bez kilkuletnich eksperymentów w realnym środowisku projektowym nie da się jednoznacznie stwierdzić, że proponowana przez habilitantkę metoda przyniesie pożądany zwrot tych inwestycji.

W rozprawie nie dopatrzyłem się rozważań na temat tego, czy zawsze, czy dla każdego projektu warto włączać użytkownika w proces wytwórczy. Dla prostych rutynowych projektów i zakładanym doświadczeniu firmy wykonawczej włączenie użytkownika w procesy projektowe mija się z celem. Np. w jednym z projektów miałem do wykonania system, który przy założonej strukturze bazy danych musiał posiadać moduł wprowadzania danych (w odpowiednich, typowych ekranach graficznych) i miał produkować ok. 100 różnych raportów, których przeznaczenie i struktura były znane, większość z nich wynikała z odpowiednich przepisów. W tego rodzaju projekcie nie widziałbym miejsca dla włączenia użytkowników.

Rozprawa całkowicie pomija ważną klasę projektów, w których użytkownika w ogóle nie ma, ponieważ użytkownik jest tworzony przez zaoferowany produkt. Ten rodzaj projektów dotyczy ogromnej sfery oprogramowania narzędziowego (edytory tekstu, arkusze kalkulacyjne, kompilatory, systemy operacyjne, systemy zarządzania bazą danych, przeglądarki Internetowe, wyszukiwarki stron Internetowych, studia graficzne, narzędzia CASE, itd. itp.). Jakkolwiek w Polsce ten segment informatyki jest bardzo słabo rozwinięty, ten stan powinien się zmienić, gdyż jak dotąd informatyka narzędziowa produkowana na masową skalę przynosi największe efekty ekonomiczne (patrz wyniki finansowe firm IBM, Sun, Microsoft, Apple, Oracle, SAP, Google, Facebook i tysięcy innych – w momencie powstania odpowiednich produktów ich potencjalni użytkownicy nie mieli pojęcia, ze niedługo będą ich bardzo potrzebować, więc nikt ich o zdanie nie pytał). Powstaje pytanie, czy metodyka proponowana przez habilitantkę może w jakimś stopniu mieć zastosowanie do tej klasy produktów informatycznych.

Scenariusz włączenia użytkowników w procesy wytwórcze oprogramowania wydaje się także nieco zbyt optymistyczny. Często użytkownicy nie mają najdrobniejszej woli zmiany swojego profilu zawodowego, uważają, że są dobrymi specjalistami w swoich dziedzinach i nie widzą potrzeby włączania się w procesy projektu informatycznego tylko dlatego, że powstaje dla nich system informatyczny. Jako przykład podam jeden z projektów który prowadziłem dla instytucji centralnej w Warszawie (Komisji Papierów Wartościowych i Giełd). Pracownikami tej instytucji byli prawnicy, specjaliści z zakresu funkcjonowania giełdy, maklerzy, ekonomiści, itp. Osoby te, będąc wybitnymi fachowcami w swoich dziedzinach, z trudnością odczytywały diagramy przypadków użycia, zaś kompletnie nie radziły sobie z innymi diagramami, np. diagramem klas. Włączenie tych użytkowników w procesy projektowe okazało się niemożliwe. Dopiero powstanie komórki informatycznej (3-ch pracowników – informatyków) na terenie Komisji, której zadaniem było rozpoznanie procesów informacyjnych zachodzących na tym terenie celem właściwego dialogu w kwestii wymagań z zespołem wykonawców, doprowadziło do w miarę gładkiego wykonania odpowiedniego systemu.

**Ocena dorobku publikacyjnego dr Barbary Begier**

Oprócz rozprawy habilitacyjnej, dorobek ten obejmuje 71 pozycji napisanych po uzyskaniu przez Autorkę stopnia naukowego doktora. Spośród nich 31 pozycji ukazały się w wydawnictwach punktowanych przez MNiSW, w większości o dość wysokiej randze z punktu widzenia prestiżu w środowisku naukowym. Popularna bibliografia internetowa DBLP, http://www.informatik.uni-trier.de/~ley/db/ (uznawana za najbardziej wiarygodną, jeżeli chodzi międzynarodowy zasięg publikacji) , zawiera 5 pozycji habilitantki, z czego 4 są bezpośrednio związane z tematyką rozprawy. Jest to wynik niezbyt wysoki, ale dość typowy,

szczególnie w dziedzinie inżynierii oprogramowania. Z osobistych doświadczeń autora recenzji, publikowanie w wydawnictwach międzynarodowych prac z zakresu inżynierii oprogramowania jest znacznie trudniejsze niż publikowanie w takich dziedzinach jak bazy danych, technologie Internetu lub sztuczna inteligencja. Wynika to stąd, że inżynieria oprogramowania jest dziedziną bardzo tradycyjną, liczne tematy były już poruszane dziesiątki lat temu i bardzo trudno dodać coś nowego, szczególnie jeżeli miałoby to mieć rangę wyniku badawczego. Zasadniczym powodem jest jednak to, że w inżynierii oprogramowania wyniki mają charakter „miękki", spekulacyjny, dyskusyjny i często arbitralny. „Twardy" wynik przybiera postać teorii, algorytmu, implementowanej komputerowo metody, nowego wynalazku. „Twardy" wynik uzyskuje się np. poprzez zaimplementowanie prototypu nowego języka zapytań do baz danych o nieznanych dotąd własnościach lub prototypu nowatorskiego systemu procesów pracy. Taki wynik jest praktycznie natychmiast weryfikowalny przez recenzentów, zaś liczba forów publikacyjnych jest ogromna. W inżynierii oprogramowania weryfikacja zaproponowanej nowej metody lub metodyki w zasadzie nie ma szans na realizację: wymagałaby znalezienia środowiska wytwórców, które chciałoby ją wdrożyć oraz lat eksperymentów porównawczych. W warunkach akademickich jest to prawie niewykonalne.

Najbardziej płodnym „generatorem" publikacji naukowych w informatyce są metody formalne z zastosowaniem zaawansowanej matematyki, udowadnianiem twierdzeń, budową systemów aksjomatycznych, itd. Metody matematyczne odegrały i nadal odgrywają istotną rolę w licznych dziedzinach inżynierskich (nie tylko informatycznych), ale nie dotyczy to inżynierii oprogramowania. W tej dziedzinie praktycznie wszystkie teorie matematyczne, których największy rozwój przypadł na lata 1970-te i 1980-te, poniosły sromotną klęskę (1-sza metoda wiedeńska, VDM, Lotos, Notacja Z, logika algorytmiczna itd.) i obecnie praktycznie zniknęły z podręczników (chociaż nadal są nisze akademickie próbujące robić na nich kariery „naukowe"). Wynika to z faktu, że inżynieria oprogramowania z różnych stron jest silnie uwarunkowana czynnikiem ludzkim (analitycy, projektanci, wykonawcy, użytkownicy, bezpieczeństwo, prywatność, psychologia, zachowania społeczne, skłonność do błędów, lenistwa i nadużyć itd.) który całkowicie nie podlega matematycznej formalizacji. Popularne hasło „od teorii do praktyki" w inżynierii oprogramowania przybrało postać „od teorii do składu makulatury". Obecna inżynieria oprogramowania jest syntezą dobrych praktyk pochodzących z tysięcy ośrodków produkujących oprogramowanie. Tylko wieloletnia praktyka może być podstawą dobrej publikacji. Stąd także wynika trudność publikowania w tej dziedzinie.

Liczba forów publikacyjnych w dziedzinie inżynierii oprogramowania jest stosunkowo niewielka i są opanowane przez osoby znane („ewangelistów"), z reguły pochodzące z dobrych firm w USA lub w krajach anglosaskich. Polski autor, szczególnie ze środowiska akademickiego, nie jest postrzegany przez te fora jako równorzędny partner. Polska jest często postrzegana (nie bez racji) jako informatyczna pustynia, o czym wielokrotnie się przekonałem podczas lat spędzonych za granicą. To oczywiście obniża dodatkowo i tak niezbyt wielkie szanse na renomowaną zagraniczną publikację. Jestem jednak zdania, że waga publikacji dr Barbary Begier jest bardzo znacząca, szczególnie dla rozwoju tej tematyki w Polsce.

Najważniejsze prace opublikowane w recenzowanych wydawnictwach krajowych i międzynarodowych zostały dołączone do dokumentacji przewodu habilitacyjnego. Materiał zawarty w tych publikacjach tematycznie pokrywa się z materiałem zawartym w rozprawie habilitacyjnej, ale zawiera liczne rozszerzenia. Podsumowując, uważam, że ten dorobek

znacznie przekracza powszechnie stosowane w Polsce minima wymagane dla uzyskanie stopnia doktora habilitowanego.

**Konkluzja końcowa**

Rozprawa habilitacyjna dr Barbary Begier zawiera wiele elementów całkowicie oryginalnych i bardzo ciekawych. Moje uwagi krytyczne są bardziej polemiką merytoryczną niż wskazaniem zasadniczych wad rozprawy. Ponieważ zarówno ja, jak i habilitantka poruszamy się w podobnym kręgu tematycznym podchodząc do problemów z różnej strony, jest naturalne, że powstają rozbieżności w koncepcji, które mogą być przedmiotem krytyki i różnych poglądów. Nie mam jednak najmniejszych wątpliwości, że mimo tych rozbieżności rozprawa dr Barbary Begier zasługuje na wysoką ocenę od strony naukowej. Na uwagę zasługuje również to, że habilitantka wykazała w niej dobrą znajomość wielu trudnych tematów i zaproponowała własne podejście do ich opanowania. Autorka rozprawy wystarczająco dobrze zaznaczyła swoją obecność na forum naukowym, co rokuje nadzieje na przyszłość. Biorąc pod uwagę wszystkie aspekty, jestem zdania, że rozprawa dr Barbary Begier spełnia wymogi odpowiedniej ustawy o stopniach i tytułach naukowych i zasługuje na bardzo pozytywną ocenę. **Wnoszę o dopuszczenie habilitantki do dalszych faz przewodu habilitacyjnego.**

K Surdeta