

Szczecin, dn. 31 maja 2018 r.

dr hab. inż. Jerzy Pejaś
Katedra Inżynierii Oprogramowania
Wydział Informatyki
Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

**Dr hab. Mirosław Maliński, prof. nadzw.
Prodziekan ds. Nauki
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. Śniadeckich 2
75-453 Koszalin**

Szanowny Panie Dziekanie,

w odpowiedzi na pismo z dnia 21 lutego 2018 roku przesyłam recenzję rozprawy doktorskiej mgr inż. Włodzimierza Wysockiego pt. „Agentowo–obiektowy model wsparcia procesu wytwarzania oprogramowania” w związku z postępowaniem w sprawie nadania stopnia doktora w dziedzinie nauk technicznych w dyscyplinie informatyka.

Z poważaniem
J.G.

Szczecin, dn. 15 maja 2018 r.

dr hab. inż. Jerzy Pejaś

Katedra Inżynierii Oprogramowania

Wydział Informatyki

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

RECENZJA

rozprawy doktorskiej mgr inż. Włodzimierza Wysockiego pt. „Agentowo–obiektowy model wsparcia procesu wytwarzania oprogramowania”

Niniejszą recenzję przedstawiam na prośbę Rady Wydziału Elektroniki Rady Naukowej Politechniki Koszalińskiej w Koszalinie ramach realizowanego przewodu doktorskiego mgr inż. Włodzimierza Wysockiego.

1. Problematyka naukowa oraz przedmiot rozprawy

Przedmiotem rozprawy jest ocena procesu wytwarzania oprogramowania z punktu widzenia gotowości organizacji IT do zwinnej transformacji rozumianej jako zdolność organizacji do realizacji złożonych projektów w oparciu o zwinne metodyki wytwarzania. Stosowanie przez organizacje IT w procesie wywarzania zwinnych metodyk ma wiele zalet, m.in. pozwala na skrócenie czasu wydań kolejnych wersji oprogramowania, wprowadzenie elastyczności w realizacji projektów, polepsza komunikację w zespole wykonawczym. Z tego powodu wiele organizacji jest zainteresowanych szybkim przejściem z metodyk tradycyjnych na metodyki zwinne.

Proces transformacji organizacji i przystosowania jej do wytwarzania oprogramowania w oparciu o metodyki zwinne może być realizowany w oparciu o tzw. modele zwinnej transformacji organizacji. Przykładami takich modeli są m.in. AAİM (Agile Adoption and Improvement Model, SAAF (Strategic Agile pre-Adoption analysis Framework), SAAP (Strategic Analysis for Agile Practices)¹. W swojej pracy Doktorant słusznie zauważa, że modele zwinnej transformacji pozwalają na ocenę zwinności praktyk stosowanych w organizacji, ich selekcję i określenie kolejności ich wdrażania, ale zwykle uniemożliwiają ocenę stopnia gotowości do rozpoczęcia procesu zwinnej transformacji.

¹ Esfahani, H. C., Eric, S., and Annosi, M. C. (2011). Towards the strategic analysis of agile practices. In CAiSE Forum, pages 155–162.

W praktyce przejście organizacji na metodyki zwinne, tzw. zwinna transformacja organizacji, jest obarczona mniejszym lub większym ryzykiem niepowodzenia. Ryzyko to jest mocno zależne od poziomu dojrzałości procesów wytwarzania oprogramowania w organizacji. Stąd też proces zwinnej transformacji organizacji należy rozpoczynać tylko w przypadku uzyskania przez organizację odpowiedniego poziomu dojrzałości.

Według najbardziej znanego modelu oceny dojrzałości CMMI (ang. Capability Maturity Model Integration) z najniższym ryzykiem transformacji mamy do czynienia wtedy, gdy organizacja potrafi w systematyczny sposób przygotować się do zmian (np. zmian związanych z rozwojem metodyk wytwarzania oprogramowania). W metodzie CMMI przypadek ten odpowiada piątemu najwyższemu poziomowi dojrzałości organizacji.

Ocena dojrzałości organizacji w oparciu o model CMMI jest czasochłonna i kosztowna oraz trudna do zautomatyzowania. Doktorant zaproponował podejście, które bazuje na zastosowaniu do oceny procesu wytwarzania modelu wytwarzania oprogramowania. Model ten wymaga użycia tradycyjnego obiektowego paradygmatu (niezbędnego do opisu cech projektu i metodyki wytwarzania) oraz paradygmatu agentowego, który pozwala na uwzględnienie wpływu struktury zespołu projektowego na proces wytwarzania. Uzyskany model wytwarzania pełni rolę wzorcowego procesu wytwarzania, z którym porównywany jest rzeczywisty badany proces wytwarzania. Wynikiem porównania jest wskaźnik oceny procesu wytwarzania (tzw. wskaźnik zgodności) zdefiniowany jako procent zadań wykonanych zgodnie z modelem.

Biorąc pod uwagę powyższe fakty uważam, że obszar badawczy będący przedmiotem rozprawy został dobrze określony, jest bardzo aktualny i ma duże znaczenie praktyczne.

Autor założył, że głównym celem rozprawy jest *opracowanie metod oceny i planowania procesu wytwarzania i środowiska projektowego, bazujących na zbudowanym agentowo–obiektywnym modelu procesu wytwarzania oprogramowania*. Osiągnięcie tego celu powinno pozwolić Doktorantowi na wykazanie słuszności następującej tezy:

Zastosowanie agentowo–obiektywnego modelu procesu wytwarzania oprogramowania pozwala na ocenę środowiska projektowego² oraz umożliwia planowanie prac w tym środowisku.

W mojej ocenie, tak sformułowana teza pracy została udowodniona. Jednocześnie uważam, że postawiony w rozprawie cel badawczy polegający na opracowaniu modelu wzorcowego procesu wytwarzania oprogramowania i zastosowaniu go do oceny zgodności z rzeczywistymi procesami wytwarzania w organizacji, do generowania listy zadań do zrealizowania w ramach projektu oraz harmonogramu ich realizacji został osiągnięty. Co więcej, rozważane w rozprawie zagadnienia są aktualne i istotne w obszarze informatyki

2. Analiza treści rozprawy oraz uzyskanych wyników

Recenzowana rozprawa doktorska została przygotowana w języku polskim. Jej struktura jest przejrzysta i składa się z sześciu rozdziałów, streszczenia (w języku angielskim i polskim), podsumowania pracy, bibliografii obejmującej 120 pozycji literaturowych (w tym 9 dowiązań do adresów internetowych) oraz czterech dodatków. Całość pracy obejmuje 182 strony i ma charakter praktyczny. Wyniki własne Doktoranta przedstawione są w rozdziałach 3 - 6.

We wprowadzeniu do pracy (rozdz. 1) Autor zwięźle opisał i ocenił podstawowe metodyki wytwarzania oprogramowania, które podzielił na metodyki tradycyjne i zwinne. Ze względu

² Środowisko projektowe składa się z zakresu i planu wytwarzania projektu informatycznego, zespołu projektowego oraz metodyki wytwarzania oprogramowania

na zalety metodyk zwinnych wiele organizacji jest zainteresowanych płynnym i mało ryzykownym przejściem z metodyk tradycyjnych na metodyki zwinne. Nie jest to zadanie łatwe i wymaga oceny dojrzałości organizacji oraz stosowanych przez nią procesów wytwarzania oprogramowania. W większość tradycyjnych metodyk oceny dojrzałości organizacji dostarcza ocen jakościowych i nie poddaje się automatyzacji. W celu wyeliminowania zauważonych wad metodyk tradycyjnych Autor w podsumowaniu rozdziału pierwszego zaproponował nowe ilościowe podejście do oceny procesu wytwarzania oprogramowania w organizacji oparte na odpowiadającym temu procesowi modelowi wytwarzania. Dzięki wprowadzeniu modelu wytwarzania oprogramowania oraz porównaniu go z rzeczywistą realizacją projektu Autor twierdzi, że możliwa jest nie tylko ocena zgodności realizowanego projektu ze wzorcem, ale także zaplanowanie procesu wytwarzania. Uzasadnieniu tego stwierdzenia Autor poświęca kolejne rozdziały pracy, a zwłaszcza rozdz. 4, 5 i 6.

W rozdziale drugim Autor przedstawia w formie czarnej skrzynki układ eksperymentu, który jest próbą opisu rzeczywistego procesu wytwarzania oprogramowania. Celem przedstawionego opisu jest zidentyfikowanie zmiennych wejściowych (zakres projektu, metodyka wytwarzania, plan projektu, zespół projektowy) i wyjściowych (lista i harmonogram prac) procesu wytwarzania oprogramowania niezbędnych do jego oceny i planowania. Zestaw zmiennych wejściowych silnie zależy od zastosowanej metodyki wytwarzania. W praktyce daje to możliwość opracowania układu eksperymentu pozwalającego na planowanie i ocenę procesów wytwarzania realizowanych według różnych metodyk. Istotną wartością tego rozdziału jest szczegółowy opis struktury elementów składowych zmiennych wejściowych i wyjściowych.

Opracowany w rozdz. 2 układ eksperymentu posłużył Autorowi do opracowania w rozdziale trzecim tzw. modelu podstawowego, w którym czarną skrzynkę układu eksperymentu zastąpiły wewnętrzne elementy odpowiadające podstawowym składowym procesom wytwarzania oprogramowania i ich interakcjom. Składowe te stanowią rdzeń modelu, który uzupełniony o rozszerzenia (tj. zbiór podmodeli, elementy i interakcje) powinien odzwierciedlać zachowanie systemu rzeczywistego dla wszystkich dopuszczalnych układów eksperymentu. Uzyskany model jest bardzo obszerny i pozwala na zastosowanie go do różnych celów. Autor uzasadnił jednak, że w przypadku oceny i planowania procesu wytwarzania model można uprościć wykorzystując w tym celu zasady modelowania bazujące na podejściu agentowo-obiektowym z dyskretnym czasem symulacji. Istotną częścią tego rozdziału jest także propozycja replikatywnej i predykcyjnej walidacji zasadności modelu uproszczonego.

Najwartościowsze wyniki są zawarte w rozdziale czwartym. W rozdziale tym najpierw zaproponowano formalny uproszczony model procesu wytwarzania oprogramowania (AGOMO), a następnie przedstawiono jego implementację w systemie wieloagentowym. W przypadku modelu formalnego trudno nie odnieść wrażenia, że wiele formalizmów z tego rozdziału jest powieleniem rozdziałów 2 i 3 (np. równania (2.2), (3.1) i (4.1)). Oczywiście, merytorycznie poprawniej jest najpierw zaproponować model bazowy z rozszerzeniami, a następnie uproszczyć go, ale z punktu widzenia celu pracy jest to zbędne i prowadzi do zbyt obszernej pracy. Zdecydowanie pomogłoby w tym przedstawienie w pracy tylko sformalizowanego modelu uproszczonego. Model formalny bazuje na trzech typach agentów, z których pierwszy symuluje zachowanie członków zespołu projektowego, drugi kontroluje i zarządza planem realizacji projektu, zaś ostatni synchronizuje prace i zachowania agentów. Jest to ciekawy pomysł, który w połączeniu z opisem zachowania agenta jako automatu skończonego daje duże możliwości modelowania procesu wytwarzania oprogramowania i w szczególności pozwolił Autorowi na przedstawienie metody AGOMAS przeznaczonej do

oceny procesów RUP i w efekcie do oceny poziomu gotowości organizacji do zmiany wykorzystywanej metodyki wytwarzania.

Metoda AGOMAS została wykorzystana w rozdz. 5 do weryfikacji replikatywnej zasadności modelu AGOMO. Eksperymenty Autor przeprowadził dla wybranej organizacji IT oraz realizowanego tam w oparciu o metodykę RUP projektu UBEZP. Eksperymenty zostały podzielone 5 etapów, z których najważniejszy polegał na takim dostrojeniu modelu AGOMO, aby model jak najdokładniej odtwarzał proces projektowania UBEZP i mógł być użyty jako wzorzec do porównania z badanym procesem rzeczywistym. W wyniku przeprowadzonych eksperymentów Autor uzyskał zgodność harmonogramów na poziomie 85%, co interpretuje jako równoważność z osiągnięciem poziomu 3 w pięciopoziomowej skali w modelu CMMI. Wynik ten według Autora potwierdza przydatność metody AGOMAS do oceny procesów wytwarzania oprogramowania, i tym samym do oceny stopnia gotowości do rozpoczęcia przez organizację procesu zwinnej transformacji.

Powyższy wniosek budzi jednak parę wątpliwości.

1. Wartości zmiennych wejściowych do modelu powinny być wartościami wzorcowymi dla danego typu projektu zależnymi od m.in. liczebności i ról członków zespołu projektowego, planu pracy, rozmiaru projektu i zależeć od rozmiaru projektu rzeczywistego. Z treści pracy wynika jednak, że model był najpierw dostrajany do rzeczywistego projektu, a dopiero później generowany wzorcowy harmonogram. Jeśli tak, to uzyskana zgodność świadczy o poprawnym odzwierciedleniu przez model rzeczywistego projektu a nie o gotowości do rozpoczęcia przez organizację procesu zwinnej transformacji.
2. Dostrajanie, o którym mowa wyżej, dotyczy wewnętrznych parametrów modelu. Opis dostrajania jest bardzo ogólny i uniemożliwia ocenę sposobu jego realizacji. Jeśli przyjąć dosłowne znaczenie słowa „dostrajanie”, to oznacza to, że parametry wewnętrzne były dobierane w pętli zamkniętej, tj. dobór parametrów – symulacja - obserwacja wyników i ich ocena – korekta parametrów – symulacja – itd.

Takie rozumienie przez Autora pojęcia „dostrajanie” wydaje się znajdować odzwierciedlenie w treści rozdz. 5.2.5: *„Celem dostrojenia modelu było odtworzenie za pomocą modelu, jak najdokładniej procesu projektu UBEZP, aby mógł być użyty jako wzorzec do porównania z badanym procesem.”*. Co więcej w tym samym rozdziale Autor napisał: *„Dostrojenie modelu polega na dobraniu wartości parametrów wewnętrznych modelu tak, aby uzyskać zgodność struktury pracochłonności projektu referencyjnego i harmonogramu utworzonego przez model AGOMO”*. Treści te nie przekładają się na rys. 5.1, w którym dostrajanie nie jest realizowane jest w pętli ze sprzężeniem zwrotnym.

3. Wątpliwości wskazane w punktach 1 i 2 stają się nieaktualne jedynie w dwóch przypadkach:
 - (a) gdy analizowany projekt rzeczywisty jest projektem referencyjnym wykonanym przez badaną organizację o odpowiednim poziomie dojrzałości (np. poziomie 3); tylko, że wtedy z góry wiadomo, iż badana organizacja już osiągnęła odpowiedni poziom dojrzałości; stąd wniosek, że przeprowadzone eksperymenty potwierdziły jedynie *zasadność replikatywną modelu*, ale niekoniecznie gotowość do rozpoczęcia przez organizację procesu zwinnej transformacji;
 - (b) gdy „dostrajanie” będziemy rozumieć jako kalibrowanie modelu, o czym słusznie wspomina Autor w rozdz. 4.4, a później konsekwentnie zapomina w całej pracy; ponieważ kalibrowanie oznacza nadawanie modelowi właściwych parametrów, np. obliczonych na podstawie parametrów rzeczywistego modelu, stąd po

skalibrowaniu modelu można go traktować jako wzorzec i porównywać z rzeczywistą realizacją projektu.

Zakładam, że przypadek pierwszy nie był celem Autora. Stąd dalej przyjmuję, że „dostrajanie” Autor rozumie jako „kalibrowanie”.

W rozdz. 6 Autor pokazał, że *opracowany model model AGOMO może być stosowany we wstępnych fazach projektu do planowania przebiegu procesu wytwarzania, odpowiedniego dobrania składu zespołu projektowego, dostosowania metodyki wytwarzania do projektu, oszacowania terminu zakończenia i kosztów projektu*. Zastosowanie modelu AGOMO w tej roli wymaga wskazania projektu wzorcowego o parametrach zbliżonych do realizowanego projektu rzeczywistego, skalibrowanie modelu, wykonanie symulacji i wygenerowanie planowanego harmonogramu prac i ich kosztów. W tym momencie model odzwierciedla właściwości projektu wzorcowego, według którego realizowany powinien być projekt. Harmonogram realizacji projektu rzeczywistego może jednak odstawać od harmonogramu prognozowanego. Stąd też porównanie ze sobą obu harmonogramów może posłużyć do oceny zasadności predykcyjnej modelu AGOMO.

W pracy Autor opisał eksperyment z wykorzystaniem projektu wzorcowego ISBSG do predykcji harmonogramu realizacji projektu rzeczywistego UBEZP. Zaplanowany proces wytwarzania został następnie porównany z rzeczywistym procesem wytwarzania projektu UBEZP. Porównanie zostało wykonane w oparciu o trzy kryteria: czas trwania procesu, pracochłonność sumaryczną w kategoriach i przebieg prac w poszczególnych kategoriach. Uzyskane wyniki są interesujące i można je uznać za potwierdzenie dobrych własności predykcyjnych modelu AGOMO i tym samym dużej praktycznej wartości zaproponowanego podejścia.

Zakończeniem pracy jest podsumowanie, w którym Autora przedstawił nie tylko istotę swoich osiągnięć, ich znaczenie, ale wskazał także silne i słabe strony zaproponowanego modelu. Bardzo interesujące są także propozycje Autora dotyczące kierunków dalszych prac nad rozwijaniem modelu, oraz opracowaniem komercyjnego produktu wspierającego zarządzanie wytwarzaniem oprogramowania. Według mojej oceny pojawienie się takiego produktu znacznie poprawiłoby planowanie realizacji projektów, a także ocenę dojrzałości organizacji jeszcze przed podjęciem decyzji o przejściu na zwinne metody wytwarzania oprogramowania.

3. Ogólna merytoryczna ocena zaprezentowanego osiągnięcia naukowego

Rozprawa doktorska mgr inż. Włodzimierza Wysockiego zawiera nowe, oryginalne wyniki dotyczące tworzenia modeli opisujących proces wytwarzania oprogramowania w organizacji IT. Co prawda, w swojej propozycji modelu Autor nie odniósł do istniejących sformalizowanych modeli wytwarzania oprogramowania, co utrudnia ocenę na ile zaproponowany model jest oryginalnym pomysłem doktoranta. Jednak z całą pewnością oryginalne jest połączenie w modelu paradygmatu obiektowego (odpowiada za uwzględniający cechy projektu i metodykę wytwarzania) i agentowego (uwzględnia wpływ struktury zespołu projektowego, w tym współpracę i komunikację członków zespołu na proces wytwarzania).

Niezależnie od powyższego stwierdzenia warto przytoczyć silne i słabe strony pracy Autora:

Silne strony. Wyniki uzyskane przez Autora są istotne zarówno z punktu widzenia aktualnych problemów badawczych dotyczących tworzenia sformalizowanych modeli opisu procesów wytwarzania oprogramowania, jak również ich zastosowania do ilościowej oceny poziomu dojrzałości procesów wytwarzania organizacji (zwłaszcza w kontekście oszacowania ryzyka zwinnej transformacji) oraz planowania procesu realizacji projektu w oparciu o podobne projekty referencyjne. Silną stroną pracy jest także wstępny plan

dalszych prac badawczych, mających na celu uzyskanie stabilnego produktu komercyjnego oferowanego organizacjami wytwarzającymi oprogramowania zgodnie z różnymi metodkami.

Słabe strony. Autor nie wskazał innych prac badawczych dotyczących formalizowania modeli wytwarzania oprogramowania lub wyraźnie nie napisał, że istnieje mało prac związanych z tą tematyką. Co prawda we wprowadzeniu do pracy Autor wspomina o znalezionej luce badawczej, dotyczącej braku badań w zakresie ilościowej oceny dojrzałości organizacji, proponuje ideę rozwiązania tego braku w oparciu o model procesu wytwarzania oprogramowania, ale nie skupia się na poszukiwaniu podobnych modeli. Brakuje zwartej postaci modeli opisywanych w pracy wraz z opisem ich składowych oraz wspierających je struktur danych i algorytmów. Słabo uzasadniona jest także propozycja interpretacji wartości wskaźnika zgodności działań przedstawiona w rozdz. 1.2.2 i rozdz. 4.4.5. Autor wykonał zbyt mało eksperymentów i z tego powodu zbyt trudno jest uogólnić uzyskane wyniki na inne projekty i organizacje.

4. Uwagi formalne

Autor bardzo dobrze porusza się w obszarze nowoczesnych metodyk wytwarzania oprogramowania, zna ich dobre i słabe strony, widzi także praktyczne problemy związane z przejściem organizacji na nowe metodyki wytwarzania. Co więcej dobrze orientuje się w pracach badawczych prowadzonych w obszarze inżynierii oprogramowania, a w szczególności w pracach z zakresu metodyk wytwarzania. Połączenie dużej wiedzy praktycznej oraz znajomości współczesnej literatury naukowej pozwoliło Autorowi na sformułowanie ważnego problemu badawczego i rozwiązanie go w oryginalny sposób.

Rozprawa jest napisana na dobrym poziomie językowym, zdania są poprawnie formułowane, nie znalazłem istotnych błędów stylistycznych. Tym niemniej w pracy można znaleźć i wytknąć drobne usterki formalne. Są to między innymi:

- (a) Praca jest bardzo obszerna, zawiera sporo rozbudowanych i przez to zawitych opisów (dotyczy to zwłaszcza opisów w rozdz. 1, 2 i 3), z których część powtarzana jest w kolejnych rozdziałach (np. formalizmy z rozdz. 2 i 3 są powtarzane w rozdz. 4).
- (b) Autor zamiast na opisowych detalach powinien bardziej skupić się na podaniu informacji o algorytmach, ich powiązaniu oraz stosowanych strukturach danych używanych w oprogramowaniu. Opisy typu „*Zadania projektu UBEZP zostały zaimportowane do bazy danych Oracle i przetworzone automatycznie za pomocą procedur napisanych w języku PL/SQL*” niczego nie wyjaśniają i są z pewnością banalne.
- (c) W części eksperymentalnej pracy jest sporo mało wyraźnych rysunków zawierających wyniki symulacji. Zdecydowanie lepiej byłoby umieścić na nich tylko wybrane i powiększone fragmenty.
- (d) Podczas kalibrowania modelu AGOMO na potrzeby badania jego własności replikatywnych i predykcyjnych Autor nie podaje obliczonych wewnętrznych wartości parametrów modelu. Jak były obliczane te parametry i jeśli to miało miejsce, to jak były korygowane?
- (e) Na str. 67 jest napisane: „Obydwie relacje uproszczono do jednej – relacji pochodzenia.” Brak uzasadnienia takiego uproszczenia.
- (f) Str. 43: Autor napisał, że „*W poprzednim rozdziale przedstawiono problem oceny procesów wytwórczych organizacji IT oraz koncepcję jego rozwiązania dzięki zastosowaniu agentowo-obiektowego modelu procesu wytwarzania.*”. W rzeczywistości koncepcje modelu wytwarzania opartego na podejściu agentowo-obiektowym Autor

przedstawił dopiero w rozdz. 3.

- (g) Str. 68: jest “*aoi*”, powinni być raczej *oai*.
- (h) Str. 87: “Model podstawowy umożliwia określenie dokładnej liczby błędów w wytworzonym oprogramowaniu”. Jak przebiega takie określenie liczby błędów? W oparciu o testy nie da się raczej wykryć wszystkich błędów.
- (i) Str. 125, tab. 5.2: brak legendy dotyczącej oznaczeń (oczywiście można je odtworzyć na podstawie wcześniejszych rozdziałów, ale utrudnia to czytanie i analizowanie tekstu).
- (j) Str. 167: błędne odwołanie do “podrozdziale 0”
- (k) Str. 172: jest “Wybranie pliku parametrów następuje po wciśnięciu przycisku [...]” Co to za przycisk?

Należy zauważyć, że wskazane usterki formalne nie wpływają w znaczny sposób na merytoryczną końcową ocenę pracy.

6. Konkluzja recenzji

Przedstawione powyżej ogólne uwagi merytoryczne i formalne nie umniejszają osiągnięć doktoranta, ani nie podważają zaproponowanych modeli wytwarzania oprogramowania oraz idei ich zastosowania do oceny dojrzałości organizacji wytwarzającej oprogramowanie i planowania procesu realizacji nowych projektów. Przedstawioną mi do oceny rozprawę oceniam pozytywnie, zarówno z uwagi na aktualność i ważność tematyki rozprawy, staranność jej przygotowania, zastosowany warsztat metodyczny, jak również dużą wiedzę Autora i znajomość literatury z zakresu metod projektowania i modelowania procesów wytwarzania oprogramowania. Autor uzyskał wartościowe i oryginalne z naukowego punktu widzenia wyniki zawarte zwłaszcza w zaproponowanych modelach wytwarzania oprogramowania oraz propozycjach ich zastosowań. Wartość tą częściowo obniża zbyt mała liczba eksperymentów wykonana na rzeczywistych i referencyjnych projektach, niepozwalająca na zbyt daleko idące uogólnienie uzyskanych wyników. Dostęp do danych projektów jest skrzętnie jednak chroniony przez organizacje wytwarzające oprogramowanie, stąd Autor z pewnością nie miał łatwego zadania w ich pozyskaniu. Uzyskane wyniki były opublikowane przez Autora w materiałach trzech konferencji międzynarodowych oraz są przedmiotem dwóch artykułów. Wszystkie te artykuły są indeksowane w bazie Web of Science, zaś trzy z nich w bazie Scopus. Treści tych artykułów nie są jednak zbyt mocno rozpoznawalne w środowisku naukowym.

Uważam, że Autor zrealizował cel rozprawy oraz wykazał się umiejętnościami i odpowiednim przygotowaniem do samodzielnej pracy naukowej w dyscyplinie informatyka. Na tej podstawie stwierdzam, że przedstawiona do oceny rozprawa doktorska mgra inż. Włodzimierza Wysockiego pt. „Agentowo–obiektowy model wsparcia procesu wytwarzania oprogramowania” **spełnia wymagania stawiane rozprawom doktorskim w Ustawie o stopniach naukowych i tytule naukowym z dnia 14 marca 2003 roku (Dz. U. 2017, poz. 1789) i wnoszę o dopuszczenie jej Autora do publicznej obrony.**