

Recenzja rozprawy doktorskiej autorstwa mgr. inż. Jarosława Hryszki

pt. „Zastosowanie predykcji defektów w komercyjnych projektach rozwoju oprogramowania – od rozważań teoretycznych do codziennej praktyki”

Podstawa wykonania

Formalną podstawą wykonania niniejszej recenzji jest pismo Przewodniczącego Rady Dyscypliny Informatyka Techniczna i Telekomunikacja Uniwersytetu Jagiellońskiego prof. dr. hab. inż. Macieja Ogorzałka z dn. 19 października 2022 roku dotyczące powołania mnie przez wymienioną Radę do pełnienia funkcji recenzenta rozprawy autorstwa Doktoranta, mgr. inż. Jarosława Hryszki, w prowadzonym przez tę Radę postępowaniu o nadanie stopnia doktora.

Podstawą faktyczną jest rozprawa autorstwa Doktoranta, napisana pod kierunkiem dr. hab. Adama Romana, profesora uczelni na Wydziale Informatyki Uniwersytetu Jagiellońskiego

Kryteria

Zgodnie z art. 13 ust. 1 ustawy z dnia 14 marca 2003 r. o stopniach i tytule naukowym oraz o stopniach i tytule naukowym w zakresie sztuki (Dz.U. 2017 poz. 1789, dalej jako: u.s.n.), zadaniem recenzenta jest dokonanie oceny, czy przedstawiona rozprawa doktorska stanowi oryginalne rozwiązanie problemu naukowego oraz czy dowodzi ogólnej wiedzy teoretycznej doktoranta w zakresie prawa, a także umiejętności samodzielnego prowadzenia przez niego pracy naukowej.

Ocena doboru problematyki badawczej i tematu rozprawy

Rozprawa poświęcona jest predykcji defektów w systemach informatycznych w oparciu o wybrane cechy programów, w tym metryki kodu i metryki procesowe, z wykorzystaniem modeli uczenia maszynowego. Problematyka predykcji defektów stanowi jeden z najważniejszych problemów we współczesnej inżynierii oprogramowania, szczególnie w kontekście przemysłowej produkcji systemów informatycznych. Wynika to z ekonomiki: znajomość choćby przybliżonej liczby defektów oraz ich lokalizacji w kodzie jest czynnikiem mającym pierwszorzędne znaczenie dla prawidłowego oszacowania kosztu budowy i utrzymania oprogramowania, a zatem także dla uzyskania przez producenta takiego systemu przewagi konkurencyjnej.

Problematyka ta jest rozwijana i badana naukowo od wielu lat z zastosowaniem rozmaitych modeli oraz podejść. Należy jednak zwrócić uwagę, że w większości przypadków badania te nie dotyczą jednak systemów produkowanych komercyjnie, w warunkach rynkowych, a przede wszystkim systemów akademickich czy znajdujących się w domenie wolnego oprogramowania, w których jednak mamy do czynienia z innym spektrum celów i ograniczeń niż w przypadku oprogramowania tworzonego przemysłowo. Z tego powodu podjęcie przez Doktoranta badań w oparciu o rzeczywiste dane pochodzące z przedsiębiorstwa produkującego takie systemy jest cenne i zasługuje na podkreślenie. Dlatego wybór tematyki i sposobu jej ujęcia uważam za trafny, odpowiadający rzeczywistym potrzebom gospodarczym, a także odpowiedni dla rozprawy doktorskiej.

Ocena tezy i hipotez badawczych

Rozprawa nie zawiera jednoznacznie postawionej tezy. W zamian Doktorant określił cel ogólny („zapropozowanie nowej metodyki zapewniania jakości dla komercyjnych projektów wytwarzania oprogramowania, wykorzystującej jako jeden z kluczowych elementów predykcję defektów oraz wdrożenie tej metodyki w komercyjnym środowisku wytwarzania oprogramowania i zbadanie jej wpływu na końcową jakość produktu”) oraz dziewięć celów szczegółowych, stanowiących konkretyzację celu ogólnego i ściśle z nim powiązanych. Kolejne cele szczegółowe rozprawy tworzą logiczną sekwencję zadań, dzięki którym możliwe jest osiągnięcie celu ogólnego. W związku z tym, mimo braku formalnej tezy, rozprawa spełnia kryteria stawiane pracy naukowej, dokumentując prowadzone badania i prezentując osiągnięte wyniki, pokazując je również w szerokim kontekście, a także dokonując ich oceny za pomocą uznanych narzędzi i metod badawczych.

Ocena merytoryczna pracy

Predykcja defektów stanowi jeden z najważniejszych problemów w obrębie inżynierii oprogramowania, ponieważ pozwala na oszacowanie kosztu realizacji i utrzymania systemu informatycznego, a także umożliwia zidentyfikowanie obszarów o zwiększonym ryzyku obecności defektów. Nawet pobieżny przegląd literatury wskazuje, że problem ten stanowi aktywne pole poszukiwań badawczych z wykorzystaniem różnych narzędzi i metod. Zastosowanie w tym celu metod uczenia maszynowego nie jest innowacją, jednak warto podkreślić dwa walory rozprawy, które w największym stopniu decydują o jej unikatowej wartości: zastosowaniu proponowanej metody predykcji w rzeczywistym środowisku przemysłowym, z uwzględnieniem jego specyficznych ograniczeń, uwarunkowań oraz orientacji na zastosowania, oraz silnej orientacji na opłacalność ekonomiczną rozwiązania.

Na uwagę zasługuje dogłębna analiza literatury przedmiotu, obejmująca nie tylko zagadnienia kluczowe dla problemu predykcji defektów, ale również wiele wątków pobocznych, uwzględniających m.in. informacje o wykorzystywanych narzędziach, alternatywnych metodach oraz dotychczas uzyskanych opublikowanych wynikach. Dzięki temu nawet

czytelnik nie zaznajomiony ze stanem wiedzy w obszarze, którego dotyczy rozprawa, jest w stanie nie tylko zrozumieć jej treść, ale również samodzielnie zastosować jej wyniki w praktyce. W związku z tym walor edukacyjny, często pomijany w innych pracach naukowych, szczególnie zasługuje tutaj na podkreślenie.

Zdaniem Doktoranta najważniejszym osiągnięciem zaprezentowanym w rozprawie jest metodyka zapewniania jakości (w praktyce obejmująca dużą część pełnego cyklu rozwojowego oprogramowania) o nazwie DPDQA (ang. Defect Prediction-Driven Quality Assurance). Stanowi ona podsumowanie rozważań i wyników badań przedstawionych w początkowych rozdziałach pracy i faktycznie wnosi samoistną wartość do rozprawy. Metodyka ta polega na możliwie jak najwcześniejszym (zgodnie z ideą *shift left*) rozpoczęciu działań zmierzających do przewidywania, identyfikowania oraz usuwania defektów. Ponieważ wymagany nakład pracy wynikający z realizacji tych czynności jest tym niższy, im wcześniej ta czynność w cyklu rozwojowym występuje, działanie takie, choć sumaryczna liczba defektów w systemie informatycznym pozostaje bez zmian, pozwala na obniżenie całościowych kosztów realizacji systemu informatycznego, na co wskazują liczne dowody oraz wyniki przedstawione w rozprawie. W związku z tym uzyskany wynik jest istotny nie tylko z naukowego, ale również praktycznego punktu widzenia.

Istnieje jednak kilka kwestii, które wymagają doprecyzowania i uzupełnienia. Choć trudno nie zgodzić się z przedstawioną w rozprawie argumentacją, sam opis metodyki DPDQA jest bardzo ogólny i pozostawia pewien niedosyt. Na przykład, Doktorant wskazuje, że jej elementami są m.in. przeglądy kodu, stosowanie wzorców projektowych, zaleceń SOLID czy tzw. pryncypiów architektonicznych. Wprawdzie są to praktyki stosowane i cenione we współczesnej produkcji oprogramowania, jednak brak niezbędnej precyzji oraz dowodów wskazujących na rolę, jaką pełnią w metodyce, powoduje, że postulat ich wykorzystania w DPDQA ma charakter raczej publicystyczny niż naukowy. Dlatego w mojej ocenie największą wartość rozprawy stanowi nie sama metodyka, a jej opis jej umiejętnego i przemyślanego wdrożenia w praktyce komercyjnego tworzenia oprogramowania oraz przedstawienie wyników tego wdrożenia. Zebrane i opracowane przez Doktoranta dane jasno wskazują, że przyjęte założenie o wczesnym zastosowaniu predykcji defektów jest wysoce opłacalne dla producenta i daje wysoką stopę zwrotu przy stosunkowo niewielkiej inwestycji. W ten sposób rozprawa ma szansę stać się ważnym argumentem dla zarządzających organizacjami tworzącymi oprogramowanie; wskazuje ona nie tylko na ogólnie rozumianą „pożyteczność” określonych działań związanych z predykcją, ale również ich bezpośrednią i wysoką opłacalność finansową.

W rozprawie do predykcji defektów wykorzystano model zbudowany w oparciu o uczenie maszynowe. W uczeniu takich modeli istotnym zagadnieniem jest kwestia jakości danych. Ma ona kluczowe znaczenie w przypadku, gdy ilość dostępnych danych wykorzystanych w procesie uczenia jest stosunkowo niewielka, a taka sytuacja ma miejsce właśnie w analizowanym projekcie. Odnosząc się do tego problemu, doktorant w całości

uzasadniony sposób wiele uwagi poświęca ważnym czynnikom, np. zbalansowaniu danych oraz ich oczyszczeniu. Ponadto, wielokrotnie w rozprawie podkreśla rolę właściwego oznaczenia defektów w repozytorium kodu oraz w narzędziu do śledzenia tych defektów. W związku z tym pojawia się pytanie, jaki realny wpływ na dokładność predykcji mają błędne, niedokładne i brakujące informacje w tym zakresie. Czy przeprowadzono symulacje pozwalające na ocenę tego wpływu oraz konsekwencji z niego wynikających? Czy brakujące dane można uzupełnić w sposób sztuczny (np. poprzez ich imputację), czy też należy je usunąć ze zbioru uczącego? Kwestia dostępności dużej ilości wysokiej jakości danych ma duże znaczenie w praktyce, na co wskazuje choćby liczba projektów wykluczonych przez Doktoranta z analizy właśnie z tego powodu.

Trzecia sprawa dotyczy wrażliwości wyników predykcji na niedokładność pomiaru. Doktorant szczegółowo prezentuje sposób obliczenia relacji nakładów i zysków. W tym kontekście należy zauważyć, że obliczenie ilorazu dwóch liczb, z których dzielnik ma niedużą wartość w porównaniu do dzielnej, powoduje, że nawet niewielki błąd jego pomiaru spowoduje duże wahania wyniku. Dlatego ciekawym pobocznym problemem jest właśnie ocena wrażliwości wyniku obliczeń na nieuchronne, szczególnie w badaniach empirycznych, niedokładności. Czy możliwe jest oszacowanie tego wpływu?

Szczególne zainteresowanie budzi rozdział 6, opisujący podjętą próbę uwzględnienia w modelu predykcyjnym informacji o obecności tzw. przykrych zapachów (ang. code smells). Wyniki uzyskane przez Doktoranta wskazują, że w tym przypadku poszerzenie zestawu cech nie powoduje poprawy jakości predykcji, a pewnym przypadkiem nawet jej pogorszenie. W kontekście innych opublikowanych prac jest to wynik dość zaskakujący. Wprawdzie przykre zapachy nie mają bezpośredniego związku z defektami (dotyczą one pielęgnowalności oprogramowania, a nie jego poprawności funkcjonalnej), jednak wiele prac o charakterze empirycznym (np. Piotrowski i Madeyski, 2020; Sotto-Mayor, 2022; Lavazza, 2021) wskazują, że kod obarczony takimi zapachami faktycznie posiada większą gęstość defektów. Dlatego warto byłoby pokusić się o głębszą analizę obserwowanego zjawiska i podjęcie próby wyjaśnienia jego możliwych przyczyn.

I wreszcie pozostaje kwestia dotyczy możliwości ekstrapolacji uzyskanych wyników na inne przypadki. W rozprawie Doktorant przedstawia wiele założeń wpływających na sposób realizacji predykcji oraz na końcowy wynik, dotyczący oszczędności, jakie może przynieść metodyka DPDQA. Ponieważ rozprawa w dużym stopniu ma charakter analizy przypadku, a nie typowego badania empirycznego, ekstrapolacja w dużej mierze jest utrudniona (bądź nawet niemożliwa) z powodu specyficznych, często niejawnych uwarunkowań konkretnego badanego projektu informatycznego. Warto jednak rozważyć, czy jednak istnieje możliwość przeniesienia części uzyskanych wyników także na inne przypadki. Ponieważ wątek ten nie jest szerzej omówiony w rozprawie, interesującym jej rozszerzeniem byłaby analiza wskazująca założenia wspólne dla pewnej klasy projektów informatycznych, oraz elementy różnicujące, które każdorazowo powinny być na nowo opracowane i rozważone.

Ocena kwalifikacyjna

Dokonując oceny kwalifikacyjnej całości pracy stwierdzam, że Doktorant wykazał się dobrym poziomem wiedzy w zakresie predykcji defektów w systemach informatycznych, szczególnie z wykorzystaniem metod uczenia maszynowego. Dogłębny i wielokierunkowy przegląd dostępnej literatury nie tylko dowodzi erudycji Doktoranta, ale może również stanowić wartościowe i przystępne źródło wiedzy dla czytelników rozprawy. Ponadto, sposób prowadzenia wywodu i trafnego doboru argumentów stanowią dowód, że Doktorant posiada także umiejętność samodzielnego prowadzenia pracy naukowej, umiejętnie i trafnie identyfikując ważne problemy, dobierając do nich właściwe metody badawcze oraz rozwiązując te problemy za ich pomocą. Uzyskany wynik, choć nie może być bezpośrednio uogólniony na inne przypadki, stanowi oryginalne rozwiązanie istotnego problemu badawczego i praktycznego oraz dowodzi (zarówno analitycznie, jak i empirycznie), że wczesna konstrukcja procesu budowy oprogramowania, w którym predykcja defektów następuje możliwie wcześnie i jest oparta na danych historycznych, przynosi trudny do przecenienia zwrot z inwestycji.

Wyniki cząstkowe przedstawione w rozprawie zostały opublikowane w kilku recenzowanych publikacjach, w tym Foundations of Computing and Decision Sciences oraz materiałach pokonferencyjnych wydanych przez Springer Verlag. Wprawdzie liczba tych publikacji nie jest duża, a ich zasięg mierzony współczynnikiem IF nie jest wysoki, jednak ten dorobek publikacyjny mieści się w spektrum typowym dla tego etapu kariery naukowej.

Całościowa ocena recenzowanej rozprawy jest zdecydowanie pozytywna. Zadane wcześniej przeze mnie pytania i poczynione komentarze nie wpływają na ogólną ocenę pracy, a jedynie stanowią głos w dyskusji, do której chciałbym zaprosić Doktoranta podczas publicznej obrony.

Konkluzja

Stwierdzam, że rozprawa doktorska autorstwa mgr inż. Jarosława Hryszki w wystarczającym stopniu spełnia wymagania określone w art. 13 ust. 1 *Ustawy z dnia 14 marca 2003 roku o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki w związku z art. 179 Ustawy z dnia 3 lipca 2018 r. Przepisy wprowadzające ustawę Prawo o szkolnictwie wyższym i nauce* (Dz. U. z 2018 r., poz. 1669).

Na tej podstawie wnioskuję do Rady Dyscypliny Informatyka Techniczna i Telekomunikacja Uniwersytetu Jagiellońskiego w Krakowie o dopuszczenie Pana mgr. inż. Jarosława Hryszki do dalszych etapów postępowania o nadanie stopnia doktora.