

Recenzja

rozprawy doktorskiej pana mgr inż. Rafała Wojszczyka pt. „Model oceny jakości implementacji wzorców projektowych”

Niniejsza recenzja została sporządzona w odpowiedzi na decyzję Rady Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej z dnia 19.06.2018, o uruchomieniu postępowania rozprawy doktorskiej pana mgr inż. Rafała Wojszczyka oraz na podstawie Umowy o dzieło nr 476 z dnia 01.07.2018 w związku z powołaniem recenzenta w postępowaniu o nadanie stopnia doktora.

Do recenzji została skierowana praca doktorska pana mgr inż. Rafała Wojszczyka, wykonana pod kierunkiem prof. dr hab. inż. Volodymyra Khadzhynowa, promotor pomocniczy dr inż. Piotr Ratuszniak, w dziedzinie nauk technicznych, specjalność – informatyka. Celem recenzji jest ustalenie stopnia odpowiedniości treści i formy rozprawy, względem wytycznych i wymogów nakładanych na prace doktorskie przez czynne ustawodawstwo oraz przepisy wykonawcze w dziedzinie badań naukowych oraz nadania stopni i tytułów naukowych. Przedmiotem recenzji jest rozprawa doktorska zapisana na 125 stronach, zawierająca 30 rysunków, 17 tabel, oraz spis literatury zawierający 117 pozycje, w tym 15 pozycji to publikacje Autora (5 we współautorstwie z Promotorami). Po wnikliwym zapoznaniu się z pracą możliwe jest, zdaniem recenzenta, udzielenie odpowiedzi na podstawowe pytania w kontekście oceny treści i poziomu jakości rozprawy.

1. Stopień aktualności i ważności naukowo-technicznej tematyki rozprawy.

Tematyka rozprawy leży w zakresie wspólnym dwóch wielkich sub-dziedzin współczesnej informatyki stosowanej – inżynierii oprogramowania i zarządzania projektami informatycznymi. Dawno już skończyły się czasy „sztuki programowania” (z pełnym szacunkiem do prekursorów, pionierów i klasyków programowania!). Współczesna dziedzina technologii systemów informatycznych jest całkowicie oparta na fundamencie systemologii, systemotechniki i teorii projektowania złożonych i bardzo złożonych systemów technicznych i informatycznych. Wynika to z postępu techniki komputerowej i stanowi odpowiedź na wyzwania oraz kryzysy w dziedzinie inżynierii komputerowej, a także technologii informatycznych w wielu podstawowych dziedzinach działalności cywilizacyjnej. Jednym z fundamentalnych rozwiązań sprzyjającym podniesieniu stopnia systemowości i formalizowania procesów projektowania w inżynierii systemów informatycznych, było wprowadzenie technologii programowania obiektowego, co teraz jest powszechnie stosowane w technice projektowania. Jednak wprowadzenie technik programowania obiektowego postawiło nowe wyzwania – przede wszystkim większe wymogi dotyczące kwalifikacji programistów oraz projektantów i, w związku z tym, wzrost nakładów czasowych i finansowych w przedsięwzięciach

informatycznych. Dla rozwiązania tych problemów stosuje się metody i techniki CASE, pozwalające, w pewnej mierze sformalizować, zmechanizować i zautomatyzować procesy tworzenia oprogramowania złożonych systemów informatycznych. Z innej strony, projektowanie złożonych systemów wymaga dalszego usprawnienia procesów zarządzania zmianami w projektach, co z kolei, powiązane jest z wzrostem nakładów pracy na dokumentację projektową i procesy kontroli. Szczególnie obciąża to małe i średnie przedsiębiorstwa informatyczne, stanowiąc wielkie wyzwanie dla firm pracujących w ramach tzw. „zwinnych procesów” projektowania. Nawet duże, zbiurokratyzowane przedsiębiorstwa informatyczne narażone są na znaczne koszty, powiązane z wprowadzaniem zmian w projektach i usuwaniem usterek oraz błędów powiązanych z niską kwalifikacją projektantów i złożonością oprogramowania. Wszystkie wymienione współczynniki (i wiele innych) stawiają przed społecznością naukowo-techniczną i menedżerską swego rodzaju wyzwanie – usprawnić procesy wprowadzenia zmian i wykrywania błędów w projektach oprogramowania w językach obiektowych. To wyzwanie wymaga opracowania pewnych metod i metodyk, opartych w znacznej mierze na technikach sztucznej inteligencji w ramach procesów projektowania. Te metody, z kolei, wymagają stworzenia formalnych modeli procesów wykrywania błędów i usterek, na coraz to kolejnych etapach programowania - analizy, projektowania i implementacji, co stanowi ważne zagadnienie współczesnej technologii projektowania informatycznego. W ujęciu jakościowym, podstawowych elementów najczęściej występujących w programach obiektowych, Autor całkiem słusznie wybrał wzorce projektowe, co pozwala połączyć podejście obiektowe z koniecznością ustalenia pewnych „punktów odniesienia i zbieżności” w analizie strukturalnej kodu programów. Jest to niezbędne do podjęcia decyzji klasyfikacyjnych przy ocenie jakości procesów projektowania, zwłaszcza na etapach implementacji. W związku z powyższym, można stwierdzić, że temat pracy doktorskiej jest aktualny, leży w obszarze współczesnej inżynierii oprogramowania i zarządzania projektami w dziedzinie systemów informatycznych oraz ma charakter teoretyczno-eksperymentalny, ukierunkowany na rozwiązanie ważnego, praktycznego zagadnienia – modelowania procesów zautomatyzowanej kontroli i oceny jakości implementacji programów w środowiskach programowania obiektowego, w ramach projektów prowadzonych wg. metod „zwinnych”.

2. Podstawowy problem naukowy rozprawy i stopień jego sformułowania.

Podstawowy problem, determinujący obszar zagadnień badań naukowych przeprowadzonych w ramach rozprawy, to problem oceny jakości oprogramowania w środowisku języków obiektowych. Ten problem został opisany (w pewnym sensie w sposób rozmyty) w rozdziale 1 pracy doktorskiej. Pozwoliło to Autorowi sformułować cel, zakres i tezę pracy: opracowanie nowej metody oceny jakości implementacji wzorców projektowych, gwarantującej dokładniejsze wyniki i nie kosztowniejszej w użyciu, niż alternatywne metody. Zwłaszcza, że Autor wskazuje na podstawowe zagadnienia (dodatkowe pod-problemy) – wybór sposobów formalizacji kodu źródłowego z oceną wystąpień wzorców projektowych (problem wyszukiwania i rozpoznawania),

możliwość łączenia sposobów formalizacji szablonów wzorców (problem hybrydyzacji metod w ramach kompleksowego modelu). Jako kierunki badań wskazywane są: ustalenie specyfiki wzorców projektowych (charakterystyki), oceny jakości w postaci liczbowej (wprowadzenie metryk), opracowanie dedykowanego modelu – formalnego opisu systemu projektowanego w ramach paradygmatu programowania obiektowego (analiza kodu programu). Autor formułuje tezę pracy jako stwierdzenie, że model „wykorzystujący paradygmat programowania obiektowego do formalizacji kodu programu oraz mieszane reprezentacje do szablonów wzorców projektowych, umożliwia budowę metody oceny jakości implementacji wzorców projektowych, która pozwala uzyskiwać wyniki oceny z lepszą dokładnością i nie większym kosztem niż jest to możliwe przy pomocy alternatywnych metod.” (str. 32 - 33). Jak pokazuje analiza dalszych rozdziałów pracy, teza ta została potwierdzona. Z uwag jakościowych, zdaniem recenzenta, warto zaznaczyć zbyt częste powoływanie się na ogólne sformułowanie „paradygmat programowania obiektowego” i nie wskazanie przy sformułowaniu tezy konkretnych alternatywnych metod, najbardziej konkurencyjnych względem opracowanych przez Autora.

3. Ocena stopnia rozwiązania postawionego problemu i właściwość użytych metod.

Problem oceny jakości oprogramowania jest odwiecznym problemem i coraz pojawia się w nowej interpretacji i nowym kontekście – „w odpowiedzi na wyzwania epoki”. W związku z tym, permanentnym procesem zmagania z wyzwaniami czasu jest pogłębienie wiedzy o problemie jakości, opracowanie nowych standardów i metodyk oceny, analiza i synteza modeli, badanie metod, a na ich podstawie przeprowadzenie testów tych metod i technik w warunkach praktyki. Recenzowana praca, w znacznym stopniu, odpowiada tym wymogom, tak merytorycznie jak i formalnie. Podana niżej krótka analiza treści pokazuje kompleksowość podejścia Autora do rozwiązania postawionego problemu badawczego.

Rozdział 1 jest poświęcony w znacznym stopniu przeglądowym i analityczno-krytycznym zagadnieniom oceny jakości oprogramowania, zostały w nim opisane klasyczne modele jakości. Podany jest przykład piramidalnego modelu jakości implementacji wzorców projektowych (opracowanego przez Autora), składający się z 4 stopni jakości, który w dalszych rozdziałach został zastosowany do konkretnych ocen jakości implementacji wzorców projektowych.

Rozdział 2 zawiera część przeglądową z krytyczną oceną stanu wiedzy w temacie rozprawy, (str. 35 - 45 – podrozdział 2.1) dotyczącą technik formalizacji opisów wzorców projektowych na podstawie metod graficznych, struktur drzewa, ontologii, programowania w logice oraz technik hybrydowych. Przedstawione są metody wyszukiwania wystąpień i weryfikacji wzorców projektowych na podstawie analizy tekstów kodów programów. Usystematyzowaną analizę niedostatków istniejących metod (str. 43 - 44), używaną w dalszych rozdziałach rozprawy, przedstawia podr. 2.1.3. Części 2.2 .. 2.7 oraz podsumowanie (podr. 2.8) zawierają wyniki badań własnych Autora o charakterze teoretycznym, w zakresie tworzenia modelu oceny jakości implementacji wzorców projektowych i, zdaniem recenzenta, tworzą rdzeń metodologiczny całej pracy doktorskiej. W związku z tym, w ramach uwag jakościowych o charakterze

metodycznym, można zaznaczyć zbędność połączenia w jednym rozdziale części analityczno-przeglądowej (część 2.1) i części badań własnych (2.2 .. 2.7). To spowodowało rozrost objętości rozdziału i utrudnia analizę merytoryczną jego treści. Ta część pracy wymaga uważnego przeanalizowania.

W podr. 2.2.1 i 2.2.2 Autor formułuje koncepcję trójpodziału modeli na: model programu (MP), model wzorca (MW) i model metryki (MM), wprowadza modele Delnyko (w postaci krotki) dot. konkretnego programu i metamodel Metnyko – abstrakcja przedstawiająca właściwości modelu. Zdaniem recenzenta takie podejście jest uzasadnione i zgodne z metodologią projektowania obiektowego. Warto byłoby przedstawić szczegółowiej kontekst modeli, w ogólnym zakresie procesu (systemu) projektowania na początku, a nie odnosząc te wyjaśnienia na koniec rozdziału (podr. 2.6 i 2.7). Podrozdziały 2.3, 2.4 i 2.5 przedstawiają szczegółowe opisy w ramach wzorów teoretyczno-zbiorowych modeli MP, MW i MM. Autor sumiennie przedstawił bardzo liczne zbiory atrybutów, metod i właściwości identyfikujących modele (wzory (2.2) – (2.113)). Nie można jednak, całkowicie zgodzić się z użytecznością takiego przedstawienia modeli, tj. w postaci listy zbiorów z wyznaczeniem i definicją, każdego członka zbiorów reprezentującego modele w sposób statyczny. Zdaniem recenzenta duże znaczenie również ma przedstawienie swego rodzaju „przestrzenno-czasowej dynamiki” modeli, mając na myśli taksonomię sub-modeli w każdym rodzaju modeli, relacje parametrów (członków zbiorów) w ramach sub-modelu i odwzorowanie modeli w ramach modelu Delnyko (metamodelu Metnyko). W kontekście wymienionego, pomocne byłoby zastosowanie diagramów, przynajmniej w ramach każdego rodzaju modeli MP, MW i MM. Można to odnieść do sposobów przedstawienia metody budowy modeli (podr. 2.6), bowiem z opisów słownych nie do końca wyłania się jasny przebieg procesów budowy – syntezy i identyfikacji modeli. Podany w podr. 2.7 przykład budowy modelu w pewnym sensie przybliży czytelnikowi ten proces, ale też byłby bardziej czytelnym w notacji tekstowo-graficznej.

Rozdział 3 zawiera wyniki opracowania autorskiej metody oceny jakości implementacji wzorców projektowych Danyko, funkcjonującej w ramach modelu Delnyko (rozdział 2), z zastosowaniem modeli programu, wzorca i metryki. Przebieg procesu zastosowania metody ma charakter iteracyjny i polega na ocenie modelu programu względem charakterystyk i metryk z modelu wzorca. W rozdziale również przedstawiono prototypowe narzędzia wspierające metodę oceny Danyko, opracowane w środowisku bazodanowym Microsoft SQL Server, zbudowane w technologii Microsoft.NET (podr. 3.2). Również w rozdziale 3 zostały przedstawione wyniki eksperymentów z skuteczności pozyskiwania struktury obiektowej w procesach analizy kodu źródłowego, a także eksperymentu akademickiego z grupą studentów posiadających różne umiejętności programistyczne. Eksperymenty zilustrowały możliwości usprawnienia procesów kontroli błędów i usuwania usterek, z poprawą jakości kodów do poziomu zgodnego z 3 stopniem modelu jakości. Eksperymenty wykazały również zależności wyników oceny jakości – czasochłonność i dokładność – efektywność, od zestawów danych wejściowych, rodzaju narzędzi pomocniczych i poziomu kwalifikacji programistów wykorzystujących opracowane narzędzia. Niewielka objętość eksperymentów wyznacza ich charakter raczej

ilustratywny, niż statystyczno-dowodowy, w związku z czym trudno skomentować charakter i naturę przyczyn rozbieżności wskaźników efektywności metody analizy kodu (Tab. 3.1 i 3.2). Autor wskazuje na ważną rolę jaką odgrywa dobra znajomość kodu źródłowego przez projektantów do zwiększania efektywności pozyskiwania struktur obiektowych (akapit 5 str. 85). W pewnym sensie to stwierdzenie wchodzi w sprzeczność z koncepcją rozprawy – maksymalna automatyzacja procesów oceny jakości w ramach stworzonych modeli i obniżenie przy tym roli człowieka.

Rozdział 4 zawiera opis eksperymentów weryfikacyjnych przeprowadzonych przez Autora we współpracy z firmą informatyczną Quick-Solution. Do praktycznej weryfikacji zostały wybrane trzy wzorce projektowe, które pracownicy firmy zaimplementowali w swoim programie. Weryfikacja jakościowa – efektywność metody oceny jakości została zbadana za pomocą czterech metod – autorskiej i trzech alternatywnych. Wyniki eksperymentów (Tab. 4.2 – 4.6) wskazują przekonująco na przewagę modelu autorskiego w kontekście dokładności i kosztów użycia. W ramach uwag jakościowych pracy, należy zaznaczyć brak informacji o właściwościach odrębnych wystąpień elementów (0 w tabeli), co zaniża mierzoną charakterystykę i wnosi pewien błąd w ocenę jakości metodami alternatywnymi. Również trzeba zaznaczyć zbyt krótką nazwę rozdziału – Weryfikacja.

Rozdział 5 zawiera podsumowanie przebiegu i rezultatów pracy doktorskiej, wnioski i kierunki dalszych badań. Uwagę zwracają rozważania Autora w sprawie dalszych rozwinięć modeli i metody oceny jakości oprogramowania, powiązanych z opracowaniem nowych metryk uwzględniających zachowanie – analizę dynamiki oprogramowania (akap. 2 str. 116). Opinia recenzenta całkowicie zbiega się z tym stwierdzeniem. Jako kierunki dalszych badań zostały wyznaczone modele ontologiczne (semantyczne) i modele oparte na wiedzy ukrytej z zakresu dobrych praktyk (przypuszczalnie Autor miał na myśli wykorzystanie metod matematyczno-lingwistycznych i systemów ekspertowych). Z jednej strony jest to sprzeczne ze stwierdzeniem Autora o dużym koszcie używania modeli semantycznych (p.2.1.1.4 str.38 – 39), z drugiej – odpowiada współczesnym trendom w projektowaniu informatycznym.

Reasumując przedstawioną wyżej analizę można stwierdzić, że przebieg badań i otrzymane przez Autora wyniki są całkowicie zgodne z założeniami i tezą rozprawy doktorskiej. Nowe wyniki mają wartość naukową w kontekście pogłębiania wiedzy o metodach modelowania i ewaluacji (oceny) jakości oprogramowania oraz w znacznym stopniu potwierdzają wstępne założenia pracy i przedstawiają wartość praktyczną, bowiem ukierunkowane są na możliwość ulepszenia jakości kodu źródłowego. Użyte do badań metody i narzędzia informatyczne są współczesne i adekwatne postawionym zagadnieniom.

Oczywiście, nie ma granic coraz to lepszych rozwiązań, ale na niniejszym poziomie praca doktorska odpowiada wymogom merytorycznym rozwiązując postawiony problem badawczy w planie teoretycznym i weryfikując te rozwiązania w procesie analizy i aprobacji praktycznej.

4. Stopień oryginalności, znaczenie poznawcze i wartość praktyczna dorobku Autora.

Sformułowanie problemu oceny jakości oprogramowania w ramach modelu jakości implementacji wzorców projektowych, z podziałem oceny na 4 poziomy (0 – 3) i zastosowaniem dwukryterialnego wskaźnika jakości – dokładność i koszty użytkowania, jak wykazała analiza krytyczna literatury jest elementem oryginalności danej rozprawy, determinującym możliwość tworzenia autorskich modeli i metamodeli procesów oceny. Dodatkowo wprowadzenie do modelu autorskiej metody oceny jakości implementacji wzorców projektowych, również jest elementem oryginalnym dorobku naukowego Autora. Trzecim elementem oryginalności jest opracowanie technologii badań eksperymentalnych w procesach symulacyjnych weryfikacji i badań porównawczych zaproponowanych metod i metod alternatywnych. W związku z powyższym, można stwierdzić, że stopień oryginalności jest istotnie wymiernym. Bez wątpienia wykonana praca ma wartość poznawczą, bowiem pozwala poszerzyć możliwości badania właściwości modeli programów z wskazaniem ich stopnia doskonałości i tym samym wydobywać nową wiedzę. Jednakże, zdaniem recenzenta, stwierdzać z przekonaniem i odpowiedzialnością o bezwzględnej wartości praktycznej metodyki i wszystkich opracowanych komponentów technologii, jest na razie trudno. To wynika z relatywnie młodego wieku opracowań, jak w dziedzinie kwalimetrii oprogramowania, tak samo w podejściu do automatyzacji zarządzania zmianami w projektach informatycznych. Rozmaitość języków programowania, środowisk deweloperskich, jak również zadań projektowych i struktur danych do obróbki, pomnożone przez rozpiętość kwalifikacji samych projektantów, zdaniem recenzenta, na wstępie skazują na niepowodzenie próbę ujednolicenia na jednakowo wysokim poziomie ocen w ramach jakichkolwiek, nie tylko zaproponowanych przez Autora, modeli i metod oceny jakości. Potrzebne są długie lata i wielka przestrzeń operacyjna przy współpracy z firmami informatycznymi i zespołami praktyków, do zebrania wiarygodnych statystyk i ostatecznego zrozumienia, na ile wysoki jest stopień praktycznej użyteczności opracowywanych w środowiskach akademickich rozwiązań teoretycznych i symulacyjnych. Więc, nie ze zlej woli Autor nie wskazał w tabelach porównawczych zbyt wielu wyników oceny uzyskanych za pomocą alternatywnych metod (rozdz. 4), bowiem ich po prostu nie ma.

5. Ocena stopnia wiedzy Autora i znajomości z współczesną literaturą w zakresie rozprawy.

Analizując treść rozprawy dochodzimy do wniosku o wysokiej kwalifikacji Autora w zakresie tematycznym. Autor posługuje się technikami programowania obiektowego, narzędziami i technologiami inżynierii systemów informacyjnych i oprogramowania oraz współczesnymi metodami analizy kodu źródłowego. Autor wykazał się również dobrą znajomością literatury i elektronicznych zasobów informacyjnych w dziedzinie badawczej. Przeprowadzona analiza krytyczna (Rozdz.1, podr. 2.1) świadczy o dojrzałości Autora jako praktyka w sferze inżynierii oprogramowania. Warto podkreślić, że z 117 pozycji literaturowych ujętych w spisie literatury, aż 78 źródeł (67 %) wydane jest w okresie ostatnich 10 lat. Zdaniem recenzenta, znajomość Autora z współczesną

literaturą w zakresie dziedziny badawczej jest dostateczna i świadczy o jego kompetencjach.

6. Analiza wad i słabych stron rozprawy. Rekomendacje w kierunku rozwoju dalszych badań.

W procesie analizy rozprawy uwagę zwraca szereg mankamentów, które, w pewnym sensie, obniżają poziom formalno-metodologiczny przedstawienia wyników badań. Wśród nich:

- 6.1. Tytuł pracy zawęża oczekiwania wobec jej zakresu i treści. Tak naprawdę, w rozprawie są ujęte zagadnienia analizy i syntezy modeli systemów oceny jakości (strukturalna analiza, synteza i identyfikacja) jak również przedstawione metody oceny jakości i technologie ich realizacji. Zdaniem recenzenta bardziej pasowałby tytuł rozprawy „Metodyki oceny jakości...”;
- 6.2. Swego rodzaju słowem kluczowym występującym w pracy jest termin „zwinność...”. W pracy nie podjęto próby dokładnego opisu, definicji i formalizacji tego dość wieloznacznego pojęcia, natomiast często występują odwołania do niego (zwinna metoda, zwinny zespół, możliwości zwinnego zespołu...);
- 6.3. Na str. 13 – 14 Autor opisuje normę ISO/IEC 9126, ale bardziej nowoczesna norma ISO 25010:2001 została pominięta;
- 6.4. Na str. 26 – 27 są podane przykłady trzech poziomów jakości implementacji wzorca Singleton. W tekście kodu Singleton3 (wiersz 5) wystąpił błąd;
- 6.5. Na str. 27 jest użyte, zdaniem recenzenta nie całkiem trafnie, określenie „...grupa alternatywnych metod to badania naukowe...”;
- 6.6. Na str. 32 wspomniane jest o sposobach formalizacji szablonu, przecież szablon jest sam w sobie formalizacją. Również formułowana jest definicja modelu, który jest formalnym opisem systemu – całkiem oczywista definicja;
- 6.7. Przegląd literatury powinien uprzedzać sformułowanie i uzasadnienie celu, zagadnień i tezy pracy (podr. 2.1 str. 34 -45);
- 6.8. Stwierdzenie że „...macierze są typowe dla zastosowań inżynierskich...” (str. 36, akapit 4) nic nowego do nauki nie wnosi;
- 6.9. Nie całkiem jasne jest wyjaśnienie terminu „ziarnistość” (podr. 2.1.3, str. 42 – 43);
- 6.10. Sposób nazewnictwa skróconego modeli i metody oceny jakości jest, zdaniem recenzenta, nietypowy (zazwyczaj, w skrótach są ujęte pierwsze litery lub sylaby słów pełnej nazwy);
- 6.11. Podrozdziały 2.2 – 2.5 (str. 45 – 66) są przeładowane wzorami matematycznymi – modele formalne teoretyczno-zbiorowe – co mocno utrudnia czytanie, odwleka i rozprasza uwagę od głównego wątku. Często w wyliczeniach indeksów zmiennych – elementów modeli – górne granice są zaznaczone małymi literami, co może wprowadzić w błąd czytelnika. Zdaniem recenzenta, lepiej byłoby większą część sub-modeli umieścić w dodatkach do rozprawy, a w tekście rozdziału zastosować notacje graficzne w postaci diagramów;

- 6.12. Na str. 54 w pierwszym akapicie podr. 2.4.1 powtórzony jest opis hierarchii modeli – charakterystyki, metryki itp., który jest wyjaśniony wcześniej w podr. 1.2.1 Rys. 1-1;
- 6.13. Autor wprowadza terminy „model referencyjny” i „wartość kwantyfikacyjna” bez dostatecznego wytłumaczenia i wskazania ról tych bytów w ogólnej strukturze modeli oceny jakości (podr. 2.5);
- 6.14. W podsumowaniu rozdz. 2 (podr. 2.8) użyto zwrot „Model...spełnia niedostatki...” – w opinii recenzenta właściwy zwrot to „...uwzględnia lub eliminuje...”;
- 6.15. W Rys. 3.1 brakuje etapu IV;
- 6.16. Niejasny zwrot „odpowiednia interpretacja parametru...” (str. 82, akapit 1);
- 6.17. Akapit 2 na str. 83 jest ogólnym rozważaniem nie mającym merytorycznej wagi;
- 6.18. Autor używa dwie notacji graficzne – ERD (Oracle/Barker) i UML (rozdz. 3 i 4);
- 6.19. Nie opisana metodyka oszacowania kosztów wykonania zmian kodu programu w bieżącej iteracji przez firmę Quick-Solution (rozdz. 4, Tab. 4.2, str. 107);
- 6.20. Indeks skrótów i symboli (str. 8 – 10) został sporządzony w kolejności niealfabetycznej, co utrudnia korzystanie przy czytaniu tekstu pracy.

Wymienione uwagi w znacznym stopniu mają charakter metodyczno-formalno-redakcyjny i nie wpływają na ogólną dobrą ocenę merytoryczną jakości rozprawy. Praca doktorska ma swoje silne strony – elementy oryginalności: dobry teoretyczny poziom formalizacji procesów modelowania oraz opracowanie eksperymentalnych aspektów weryfikacji modeli i metod oceny jakości implementacji wzorców projektowych w programach obiektowych. Do słabszych stron pracy, oprócz uwag, zaznaczonych w procesie analizy dorobku naukowego w p.p.3, 4 recenzji, można zaliczyć jedną próbę podjętą przez Autora w zakresie opracowania teoretycznych podstaw tworzenia i modelowania metamodelu i modelu oceny jakości oraz badania eksperymentalne efektywności metody oceny jakości implementacji wzorców projektowych – jako elementów struktury programów obiektowych – bez wyłonienia dość wąskich klas (rodzajów) analizowanego oprogramowania, co, za pewne, zmniejszyło by rozpiętość wskaźników efektywności i podniosłoby statystyczną jednorodność oraz stabilność metodyki oceny jakości oprogramowania.

7. Podsumowanie, wnioski, ocena ogólna.

Wyżej podane wyniki analizy treści rozprawy doktorskiej pana mgr inż. Rafała Wojszczyka pozwalają na sformułowanie końcowych wniosków i ocenę pracy w następujący sposób.

Tematyka rozprawy jest aktualna, a jej rezultaty są ważne i mają znaczenie dla rozwoju metodologii inżynierii oprogramowania i zarządzania projektami informatycznymi. Cele rozprawy sformułowane są w sposób konkretny, w miarę jasny i pozwalają określić miejsce tej pracy w dziedzinie badawczej w obszarze alternatywnych rozwiązań. Rozprawa zawiera nowe wyniki, istotne dla dyscypliny „Informatyka” w zakresie kwalimetrii oprogramowania i zarządzania zmianami w projektach systemów informatycznych. Rozprawa ma charakter teoretyczno-eksperymentalny. Wyniki teoretyczne zostały użyte przy analizie, syntezie i identyfikacji modelu oceny jakości

implementacji wzorców projektowych, jako podstawowych fragmentów programów obiektowych. W ramach w/w modelu zostały przeprowadzone procesy weryfikacji rozwiązań teoretycznych i przeprowadzone symulacje metod oceny. Opracowana metoda oceny jakości oprogramowania może być zastosowana przez zwinne zespoły w praktyce projektowania informatycznego oraz w procesach wprowadzenia i kontroli zmian w kodach źródłowych. Logiczny i systemowy układ pracy doktorskiej, zaplanowany w uzasadniony sposób charakter przebiegu eksperymentów, wystarczająca przejrzystość i przewidywalność wyników weryfikacyjnych, pozwalają stwierdzić pozytywnie o poprawności merytorycznej rozprawy. Choć istnieją pewne uwagi w stosunku do kwestii językowych i redakcyjnych. Aprobata wyników badań potwierdza 15 publikacjach naukowych.

W świetle wymienionego, recenzent ocenia przedstawioną pracę doktorską pozytywnie, jako w pełni spełniającą wymogi Ustawy o nadawaniu stopni naukowych i tytułów naukowych oraz w zakresie sztuki stawiane do prac doktorskich, z ogólną oceną dobrą, i wnioskuje o dopuszczenie Autora do publicznej obrony rozprawy i dalszych czynności w toku postępowania w sprawie przewodu doktorskiego.

Recenzent

Viktor Vlasenko, prof. dr hab. inż.,

Profesor zwyczajny, kierownik Zakładu Technologii

Uniwersytetu Opolskiego