

MODEL OCENY JAKOŚCI OPROGRAMOWANIA

Dorota GAWROŃSKA

Wydział Organizacji i Zarządzania, Politechnika Śląska, Gliwice; dorota.gawronska@polsl.pl

Streszczenie: Artykuł przedstawia tematykę jakości oprogramowania, opisuje kryteria oceny jakości zewnętrznej oraz prezentuje wielokryterialny model jej oceny. Ze względu na fakt określania oceny jako stopnia spełnienia oczekiwań względem finalnego produktu uwzględnia również niepewność. W celu reprezentacji niepewnej informacji artykuł przedstawia liczby rozmyte typu *LR*, umożliwiające proste operacje arytmetyczne na niepewnych wartościach ocen.

Słowa kluczowe: jakość oprogramowania, kryteria oceny jakości oprogramowania, liczby rozmyte, programowanie wielokryterialne.

SOFTWARE QUALITY ASSESMENT MODEL

Abstract: The article presents the subject of software quality, describes criteria for external quality assessment and presents multicriteria model of its assessment. Due to the fact of determining the assessment as a degree of meeting expectations for the final product, it also takes into account the uncertainty. In order to represent uncertain information, the article presents LR type fuzzy numbers, enabling simple arithmetic operations on uncertain values of ratings.

Keywords: software quality, software quality assessment criteria, fuzzy numbers, multicriteria programming.

1. Wprowadzanie

Oprogramowanie, jak każdy produkt rynkowy, powinien spełniać wymogi jakościowe określone w normach serii ISO 9000. Wymagania użytkownika mogą dotyczyć funkcjonalności i parametrów technicznych oprogramowania, jak również dotrzymania terminu i budżetu.

Na wstępie odbiorca oprogramowania powinien jednoznacznie i dokładnie określić wymagania oraz specyfikację wyjściowego produktu. Po stronie wykonawcy zlecenia jest dobór personelu wykazującego się wysokim stopniem profesjonalizmu, utworzenie zespołów, które będą prawidłowo realizować powierzone im zadania oraz współpracować ze sobą, stosując dojrzałe standardy techniczne, jak też dobierając adekwatne metody do realizowanych zadań. Finalny produkt powinien spełniać wymagania i potrzeby klienta na oczekiwanym poziomie jakości, bez błędów. Celem niniejszej pracy jest przedstawienie modelu oceny jakości oprogramowania, którego ocena może być wyznacznikiem stopnia spełnienia oczekiwań odbiorcy i wywiązania się wykonawcy z zobowiązań.

2. Jakość oprogramowania

Jakość oprogramowania może być oceniana w kategoriach jakości zewnętrznej (określonej z zewnątrz po wykonaniu oprogramowania), wewnętrznej (postrzeganej ze strony producenta) oraz użytkowej (Bilski, 2007). Niniejsza praca przedstawia ocenę jakości zewnętrznej, określonej przez sześć atrybutów (Begier, 1999; Bilski, 2007; Smilgin)

1. Funkcjonalność - zapewnienie wymagań funkcjonalnych, realizujących potrzeby i wymogi:
 - a) Odpowiedniość - dostępność wszystkich funkcji spełniających wymagania klienta.
 - b) Precyzja uzyskanych wyników.
 - c) Współpraca – funkcje programu współdziałają ze sobą w możliwie najlepszy sposób.
 - d) Zgodność funkcjonalna.
 - e) Bezpieczeństwo - ochrona przed dostępem niepowołanych osób do programów i danych.
 - f) Wielozadaniowość - program może wykonywać wiele zadań prowadzonych równolegle.
2. Niezawodność:
 - a) Dojrzałość – częstość problemów pojawiających się w wyniku błędów w programie: awarie, nieobsługiwane wyjątki czy błędy skryptów.
 - b) Tolerancja błędów – zdolność oprogramowania do utrzymania zadanego poziomu wykonania w przypadku wystąpienia błędu w oprogramowaniu lub naruszenia zdefiniowanego interfejsu, obsługuje przewidziane i nieprzewidziane błędy we właściwy sposób.
 - c) Agnostycyzm danych: wspieranie przez program wszystkich możliwych formatów danych i radzenie sobie z zakłóceniami.
 - d) Odporność: produkt obsługuje przewidziane i nieprzewidziane błędy we właściwy sposób.
 - e) Zdolność do powrotu do normalnej pracy programu, możliwość odzyskania danych i dalszego korzystania z produktu po wystąpieniu błędu krytycznego.

3. Użyteczność – zdolność produktu umożliwiająca osiągnięcie konkretnych celów w sposób efektywny, wydajny, dający satysfakcję z użytkowania danego programu:
 - a) Zrozumiałość – obsługa programu jest zrozumiała i nie budząca wątpliwości.
 - b) Łatwość w obsłudze.
 - c) Intuicyjność.
 - d) Dostępność: produkt może być użyty przez możliwie największą grupę użytkowników i jest zgodny ze standardami dostępności.
 - e) Pomoc w obsłudze – dostępność Pomocy, która pomaga i obejmuje zakres funkcjonalności.
4. Wydajność
 - a) Reagowanie: szybkość wykonania zadania.
 - b) Przepustowość: zdolność programu do przetwarzania wielu funkcji.
 - c) Odporność na obciążenia.
5. Utrzymywalność – stopień trudności przy wykonywaniu wymaganych modyfikacji wynikających np. ze zmian środowiskowych, specyfikacji funkcji czy usunięcie wykrytych błędów:
 - a) Łatwość analizy przy diagnozowaniu przyczyn błędów oraz do identyfikacji modyfikowanych składowych.
 - b) Zdolność do wprowadzenia zmian (np. usunięcia błędu, modyfikacja).
 - c) Stabilność – ryzyko wystąpienia nieoczekiwanych problemów na skutek modyfikacji programu.
 - d) Testowalność – wysiłek potrzebny do sprawdzenia zmodyfikowanego programu.
6. Przeność – zdolność oprogramowania do przeniesienia do innego środowiska eksploatacyjnego:
 - a) Zdolność adaptacji do innego środowiska.
 - b) Łatwość instalowania.
 - c) Zgodność przenośności ze standardami i konwencjami.
 - d) Zamienialność – możliwość użycia rozpatrywanego produktu zamiast innego podanego oprogramowania w tym samym środowisku.

3. Rozmyty model oceny jakości oprogramowania

Ze względu na fakt, że jakość oprogramowania może być skojarzona z poziomem usatysfakcjonowania użytkownik, w pracy przyjmuje się możliwość uwzględniania wartości niepewnych. W artykule zaproponowano wykorzystanie liczb rozmytych, które można stosować przy nieprecyzyjności danych wejściowych. Jako reprezentację liczb rozmytych zastosowano liczby typu LR (Kacprzyk, 1986). Funkcje przynależności liczby LR mogą

przyjmować różną postać, ze względu jednak na fakt, że oceny oprogramowania są wynikiem intuicyjnym, przyjmuje się trójkątną postać funkcji przynależności [1]-[2], która wymaga jedynie określenia niepewności w formie przedziału wartości $[m-\alpha, m+\beta]$, gdzie m to ocena najbardziej prawdopodobna (najbardziej zbliżona do odczucia osoby oceniającej), ewentualnie określenia samej wartości m w przypadku pewności co do oceny. W przypadku określenia przedziału niepewności, parametr m przyjmuje średnią wartość z granic tego przedziału. Parametry α i β to rozrzuty odpowiednio w lewo i prawo od wartości m .

$$L(x) = R(x) = \begin{cases} 0 & \text{dla } x < m - \alpha \\ 1 - |x| & \text{dla } m - \alpha \leq x \leq m + \beta \\ 0 & \text{dla } x > m + \beta \end{cases} \quad (1)$$

Funkcja przynależności liczby LR opisana jest poniższym wzorem:

$$\mu_A(x) = \begin{cases} L\left(\frac{m-x}{\alpha}\right) & \text{dla } x < m \\ 1 & \text{dla } x = m \\ R\left(\frac{x-m}{\beta}\right) & \text{dla } x > m \end{cases} \quad (2)$$

Osoby oceniające oprogramowanie w sytuacji pewności co do oceny danego kryterium podają tylko wartość m , wtedy parametry α i β przyjmują wartość 0. Powodem, dla którego użyto funkcji typu LR jest fakt, że umożliwiają dokonanie oceny wielokryterialnej jednocześnie bazując na danych pewnych, jak też opisujących niepewność oraz oferujących uproszczone operacje algebraiczne (Piegat, 1999). Jeżeli liczby rozmyte A_1 i A_2 przedstawimy w postaci trójek $A_1 = (m_{A_1}, \alpha_{A_1}, \beta_{A_1})$ i $A_2 = (m_{A_2}, \alpha_{A_2}, \beta_{A_2})$ to poszczególne operacje możemy przedstawić poniższymi wzorami:

a) suma

$$(A_1 + A_2) = (m_{A_1} + m_{A_2}, \alpha_{A_1} + \alpha_{A_2}, \beta_{A_1} + \beta_{A_2}), \quad (3)$$

b) różnica

$$(A_1 - A_2) = (m_{A_1} - m_{A_2}, \alpha_{A_1} + \beta_{A_2}, \alpha_{A_2} + \beta_{A_1}), \quad (4)$$

c) iloczyn

$$(A_1 \cdot A_2) = (m_{A_1} \cdot m_{A_2}, m_{A_1} \cdot \alpha_{A_2} + m_{A_2} \cdot \alpha_{A_1} - \alpha_{A_1} \cdot \alpha_{A_2}, m_{A_1} \cdot \beta_{A_2} + m_{A_2} \cdot \beta_{A_1} + \beta_{A_1} \cdot \beta_{A_2}) \quad (5)$$

d) Iloraz dla $A_1 > 0$ i $A_2 > 0$

$$(A_1 / A_2) \cong (m_{A_1}, \alpha_{A_1}, \beta_{A_1}) / (m_{A_2}, \alpha_{A_2}, \beta_{A_2}) = \left(\frac{m_{A_1}}{m_{A_2}}, \frac{m_{A_1} \cdot \beta_{A_2} + m_{A_2} \cdot \alpha_{A_1}}{m_{A_2} \cdot (m_{A_2} + \beta_{A_2})}, \frac{m_{A_1} \cdot \alpha_{A_2} + m_{A_2} \cdot \beta_{A_1}}{m_{A_2} \cdot (m_{A_2} - \alpha_{A_2})} \right), \quad (6)$$

gdzie: $m_{A_2} - \alpha_{A_2} \neq 0$, $m_{A_2} \neq 0$

e) Iloraz dla $A_1 < 0$ i $A_2 > 0$

$$(A_1 / A_2) \cong (m_{A_1}, \alpha_{A_1}, \beta_{A_1}) / (m_{A_2}, \alpha_{A_2}, \beta_{A_2}) = \left(\frac{m_{A_1}}{m_{A_2}}, \frac{-m_{A_1} \cdot \alpha_{A_2} + m_{A_2} \cdot \alpha_{A_1}}{m_{A_2} \cdot (m_{A_2} - \alpha_{A_2})}, \frac{-m_{A_1} \cdot \beta_{A_2} + m_{A_2} \cdot \beta_{A_1}}{m_{A_2} \cdot (m_{A_2} + \beta_{A_2})} \right) \quad (7)$$

gdzie: $m_{A_2} - \alpha_{A_2} \neq 0, \quad m_{A_2} \neq 0$

W modelu oceniającym oprogramowanie przyjmuje się udział I – osób oceniających, którzy należą do skończonego zbioru:

$$O = \{O_1, O_2, \dots, O_i, \dots, O_I\}, \quad i = 1, \dots, I, \quad (8)$$

W prezentowanym modelu uwzględniono określenie stopnia ważności poszczególnych kryteriów. W tym celu wprowadzono zmienne - wagi W_k (k – kryterium). Zmienne te przyjmują wartości z przedziału $[0,1]$, a ich suma w ramach danego poziomu powinna wynosić 1.

Tabela 1.

Opis zmiennych rozmytych poszczególnych kryteriów i podkryteriów

Kryterium	Podkryterium	Zmienna rozmyta	Parametry liczby rozmytej
Funkcjonalność		F_i	$(m_{F_i}, \alpha_{F_i}, \beta_{F_i})$
	Odpowiedniość	F_{-O_i}	$(m_{F-O_i}, \alpha_{F-O_i}, \beta_{F-O_i})$
	Precyzja	F_{-P_i}	$(m_{F-P_i}, \alpha_{F-P_i}, \beta_{F-P_i})$
	Współpraca	F_{-WS_i}	$(m_{F-WS_i}, \alpha_{F-WS_i}, \beta_{F-WS_i})$
	Zgodność funkcjonalna	F_{-Z_i}	$(m_{F-Z_i}, \alpha_{F-Z_i}, \beta_{F-Z_i})$
	Bezpieczeństwo	F_{-B_i}	$(m_{F-B_i}, \alpha_{F-B_i}, \beta_{F-B_i})$
	Wielozadaniowość	F_{-WZ_i}	$(m_{F-WZ_i}, \alpha_{F-WZ_i}, \beta_{F-WZ_i})$
Niezawodność		N_i	$(m_{N_i}, \alpha_{N_i}, \beta_{N_i})$
	Dojrzałość	N_{-D_i}	$(m_{N-D_i}, \alpha_{N-D_i}, \beta_{N-D_i})$
	Tolerancja błędów	N_{-T_i}	$(m_{N-T_i}, \alpha_{N-T_i}, \beta_{N-T_i})$
	Agnostycyzm danych	N_{-A_i}	$(m_{N-A_i}, \alpha_{N-A_i}, \beta_{N-A_i})$
	Odporność	N_{-O_i}	$(m_{N-O_i}, \alpha_{N-O_i}, \beta_{N-O_i})$
	Zdolność do powrotu do normalnej pracy	N_{-Z_i}	$(m_{N-Z_i}, \alpha_{N-Z_i}, \beta_{N-Z_i})$
Użyteczność		U_i	$(m_{U_i}, \alpha_{U_i}, \beta_{U_i})$
	Zrozumiałość	U_{-Z_i}	$(m_{U-Z_i}, \alpha_{U-Z_i}, \beta_{U-Z_i})$
	Łatwy w obsłudze	$U_{-ŁO_i}$	$(m_{U-ŁO_i}, \alpha_{U-ŁO_i}, \beta_{U-ŁO_i})$
	Intuicyjność	U_{-I_i}	$(m_{U-I_i}, \alpha_{U-I_i}, \beta_{U-I_i})$
	Dostępność	U_{-D_i}	$(m_{U-D_i}, \alpha_{U-D_i}, \beta_{U-D_i})$
	Pomoc w obsłudze	U_{-PO_i}	$(m_{U-PO_i}, \alpha_{U-PO_i}, \beta_{U-PO_i})$

cd. tabeli 1

Kryterium	Podkryterium	Zmienna rozmyta	Parametry liczby rozmytej
Wydajność		W_i	$(m_{W_i}, \alpha_{W_i}, \beta_{W_i})$
	Reagowanie	W_{-R_i}	$(m_{W_{-R_i}}, \alpha_{W_{-R_i}}, \beta_{W_{-R_i}})$
	Przepustowość	W_{-P_i}	$(m_{W_{-P_i}}, \alpha_{W_{-P_i}}, \beta_{W_{-P_i}})$
	Odporność na obciążenia	W_{-OO_i}	$(m_{W_{-OO_i}}, \alpha_{W_{-OO_i}}, \beta_{W_{-OO_i}})$
Utrzymywalność		UT_i	$(m_{UT_i}, \alpha_{UT_i}, \beta_{UT_i})$
	Łatwość analizy	$UT_{-ŁA_i}$	$(m_{UT_{-ŁA_i}}, \alpha_{UT_{-ŁA_i}}, \beta_{UT_{-ŁA_i}})$
	Zdolność do wprowadzenia zmian	UT_{-ZWZ_i}	$(m_{UT_{-ZWZ_i}}, \alpha_{UT_{-ZWZ_i}}, \beta_{UT_{-ZWZ_i}})$
	Stabilność	UT_{-S_i}	$(m_{UT_{-S_i}}, \alpha_{UT_{-S_i}}, \beta_{UT_{-S_i}})$
	Testowalność	UT_{-T_i}	$(m_{UT_{-T_i}}, \alpha_{UT_{-T_i}}, \beta_{UT_{-T_i}})$
Przeność		P_i	$(m_{P_i}, \alpha_{P_i}, \beta_{P_i})$
	Zdolność adaptacji do innego środowiska	P_{-ZA_i}	$(m_{P_{-ZA_i}}, \alpha_{P_{-ZA_i}}, \beta_{P_{-ZA_i}})$
	Łatwość instalowania	$P_{-ŁI_i}$	$(m_{P_{-ŁI_i}}, \alpha_{P_{-ŁI_i}}, \beta_{P_{-ŁI_i}})$
	Zgodność przeność	P_{-ZP_i}	$(m_{P_{-ZP_i}}, \alpha_{P_{-ZP_i}}, \beta_{P_{-ZP_i}})$
	Zamienialność	P_{-Z_i}	$(m_{P_{-Z_i}}, \alpha_{P_{-Z_i}}, \beta_{P_{-Z_i}})$

Na podstawie określonych zmiennych rozmytych przedstawionych w Tabeli 1 określona zostaje ocena łączna oprogramowania w ramach poszczególnych kryteriów:

a) Kryterium Funkcjonalności:

$$F_i = \frac{W_O \cdot F_{-O_i} + W_P \cdot F_{-P_i} + W_{WS} \cdot F_{-WS_i} + W_Z \cdot F_{-Z_i} + W_B \cdot F_{-B_i} + W_{WZ} \cdot F_{-WZ_i}}{W_O + W_P + W_{WS} + W_Z + W_B + W_{WZ}} \quad (9)$$

gdzie:

$$m_{F_i} = W_O \cdot m_{F_{-O_i}} + W_P \cdot m_{F_{-P_i}} + W_{WS} \cdot m_{F_{-WS_i}} + W_Z \cdot m_{F_{-Z_i}} + W_B \cdot m_{F_{-B_i}} + W_{WZ} \cdot m_{F_{-WZ_i}} \quad (10)$$

$$\alpha_{F_i} = W_O \cdot \alpha_{F_{-O_i}} + W_P \cdot \alpha_{F_{-P_i}} + W_{WS} \cdot \alpha_{F_{-WS_i}} + W_Z \cdot \alpha_{F_{-Z_i}} + W_B \cdot \alpha_{F_{-B_i}} + W_{WZ} \cdot \alpha_{F_{-WZ_i}} \quad (11)$$

$$\beta_{F_i} = W_O \cdot \beta_{F_{-O_i}} + W_P \cdot \beta_{F_{-P_i}} + W_{WS} \cdot \beta_{F_{-WS_i}} + W_Z \cdot \beta_{F_{-Z_i}} + W_B \cdot \beta_{F_{-B_i}} + W_{WZ} \cdot \beta_{F_{-WZ_i}} \quad (12)$$

b) Kryterium Niezawodności

$$N_i = \frac{W_D \cdot N_{-D_i} + W_T \cdot N_{-T_i} + W_A \cdot N_{-A_i} + W_O \cdot N_{-O_i} + W_Z \cdot N_{-Z_i}}{W_D + W_T + W_A + W_O + W_Z} \quad (13)$$

gdzie:

$$m_{N_i} = W_D \cdot m_{N_{-D_i}} + W_T \cdot m_{N_{-T_i}} + W_A \cdot m_{N_{-A_i}} + W_O \cdot m_{N_{-O_i}} + W_Z \cdot m_{N_{-Z_i}} \quad (14)$$

$$\alpha_{N_i} = W_D \cdot \alpha_{N_{-D_i}} + W_T \cdot \alpha_{N_{-T_i}} + W_A \cdot \alpha_{N_{-A_i}} + W_O \cdot \alpha_{N_{-O_i}} + W_Z \cdot \alpha_{N_{-Z_i}} \quad (15)$$

$$m_{\beta_i} = W_D \cdot \beta_{N-D_i} + W_T \cdot \beta_{N-T_i} + W_A \cdot \beta_{N-A_i} + W_O \cdot \beta_{N-O_i} + W_Z \cdot \beta_{N-Z_i} \quad (16)$$

c) Kryterium Użyteczności

$$U_i = \frac{W_Z \cdot U_{-Z_i} + W_{LO} \cdot U_{-LO_i} + W_I \cdot U_{-I_i} + W_D \cdot U_{-D_i} + W_{PO} \cdot U_{-PO_i}}{W_Z + W_{LO} + W_I + W_D + W_{PO}} \quad (17)$$

gdzie:

$$m_{U_i} = W_Z \cdot m_{U-Z_i} + W_{LO} \cdot m_{U-LO_i} + W_I \cdot m_{U-I_i} + W_D \cdot m_{U-D_i} + W_{PO} \cdot m_{U-PO_i} \quad (19)$$

$$\alpha_{U_i} = W_Z \cdot \alpha_{U-Z_i} + W_{LO} \cdot \alpha_{U-LO_i} + W_I \cdot \alpha_{U-I_i} + W_D \cdot \alpha_{U-D_i} + W_{PO} \cdot \alpha_{U-PO_i} \quad (20)$$

$$\beta_{U_i} = W_Z \cdot \beta_{U-Z_i} + W_{LO} \cdot \beta_{U-LO_i} + W_I \cdot \beta_{U-I_i} + W_D \cdot \beta_{U-D_i} + W_{PO} \cdot \beta_{U-PO_i} \quad (21)$$

d) Kryterium Wydajności

$$W_i = \frac{W_R \cdot W_{-R_i} + W_P \cdot W_{-P_i} + W_{OO} \cdot W_{-OO_i}}{W_R + W_P + W_{OO}} \quad (22)$$

gdzie:

$$m_{W_i} = W_R \cdot m_{W-R_i} + W_P \cdot m_{W-P_i} + W_{OO} \cdot m_{W-OO_i} \quad (23)$$

$$\alpha_{W_i} = W_R \cdot \alpha_{W-R_i} + W_P \cdot \alpha_{W-P_i} + W_{OO} \cdot \alpha_{W-OO_i} \quad (24)$$

$$\beta_{W_i} = W_R \cdot \beta_{W-R_i} + W_P \cdot \beta_{W-P_i} + W_{OO} \cdot \beta_{W-OO_i} \quad (25)$$

e) Kryterium Utrzymywalności

$$UT_i = \frac{W_{LA} \cdot UT_{-LA_i} + W_{ZWZ} \cdot UT_{-ZWZ_i} + W_S \cdot UT_{-S_i} + W_T \cdot UT_{-T_i}}{W_{LA} + W_{ZWZ} + W_S + W_T} \quad (26)$$

gdzie:

$$m_{UT_i} = W_{LA} \cdot m_{UT-LA_i} + W_{ZWZ} \cdot m_{UT-ZWZ_i} + W_S \cdot m_{UT-S_i} + W_T \cdot m_{UT-T_i} \quad (27)$$

$$\alpha_{UT_i} = W_{LA} \cdot \alpha_{UT-LA_i} + W_{ZWZ} \cdot \alpha_{UT-ZWZ_i} + W_S \cdot \alpha_{UT-S_i} + W_T \cdot \alpha_{UT-T_i} \quad (28)$$

$$\beta_{UT_i} = W_{LA} \cdot \beta_{UT-LA_i} + W_{ZWZ} \cdot \beta_{UT-ZWZ_i} + W_S \cdot \beta_{UT-S_i} + W_T \cdot \beta_{UT-T_i} \quad (29)$$

f) Kryterium Przenośności

$$P_i = \frac{W_{ZA} \cdot P_{-ZA_i} + W_{LI} \cdot P_{-LI_i} + W_{ZP} \cdot P_{-ZP_i} + W_Z \cdot P_{-Z_i}}{W_{ZA} + W_{LI} + W_{ZP} + W_Z} \quad (30)$$

gdzie:

$$m_{P_i} = W_{ZA} \cdot m_{P-ZA_i} + W_{LI} \cdot m_{P-LI_i} + W_{ZP} \cdot m_{P-ZP_i} + W_Z \cdot m_{P-Z_i} \quad (31)$$

$$\alpha_{P_i} = W_{ZA} \cdot \alpha_{P-ZA_i} + W_{LI} \cdot \alpha_{P-LI_i} + W_{ZP} \cdot \alpha_{P-ZP_i} + W_Z \cdot \alpha_{P-Z_i} \quad (32)$$

$$\beta_{P_i} = W_{ZA} \cdot \beta_{P-ZA_i} + W_{LI} \cdot \beta_{P-LI_i} + W_{ZP} \cdot \beta_{P-ZP_i} + W_Z \cdot \beta_{P-Z_i} \quad (33)$$

Na podstawie przedstawionych powyżej ocen łącznych można dalszej kolejności określić ostateczną ocenę oprogramowania:

$$OL_i = \frac{W_F \cdot F_i + W_N \cdot N_i + W_U \cdot U_i + W_W \cdot W_i + W_{UT} \cdot UT_i + W_P \cdot P_i}{W_F + W_N + W_U + W_W + W_{UT}}. \quad (34)$$

gdzie:

$$m_{OL_i} = W_F \cdot m_{F_i} + W_N \cdot m_{N_i} + W_U \cdot m_{U_i} + W_W \cdot m_{W_i} + W_{UT} \cdot m_{UT_i} + W_P \cdot m_{P_i} \quad (35)$$

$$\alpha_{OL_i} = W_F \cdot \alpha_{F_i} + W_N \cdot \alpha_{N_i} + W_U \cdot \alpha_{U_i} + W_W \cdot \alpha_{W_i} + W_{UT} \cdot \alpha_{UT_i} + W_P \cdot \alpha_{P_i} \quad (36)$$

$$\beta_{OL_i} = W_F \cdot \beta_{F_i} + W_N \cdot \beta_{N_i} + W_U \cdot \beta_{U_i} + W_W \cdot \beta_{W_i} + W_{UT} \cdot \beta_{UT_i} + W_P \cdot \beta_{P_i} \quad (37)$$

Powyższa ocena łączna oprogramowania jest liczbą rozmytą OL_i uzyskaną na podstawie ocen szczegółowych i -tej osoby oceniającej oprogramowanie. W celu określenia ostatecznej oceny można określić średnia z tych ocen, przy jednakowym zaufaniu do wszystkich osób oceniających oprogramowanie:

$$LO = \frac{\sum_{i=1}^I LO_i}{I}. \quad (38)$$

Chcąc uwzględnić stopień zaufania do osób oceniających oprogramowanie, należy wprowadzić dodatkową zmienną, określającą to zaufanie. Przyjmując zmienną Z_i za stopień zaufania do i -tej osoby oceniającej oprogramowanie oraz przyjmując jej wartości z przedziału $[0,1]$, otrzymujemy ostateczną ocenę oprogramowania:

$$LO = \frac{\sum_{i=1}^I Z_i \cdot LO_i}{\sum_{i=1}^I Z_i}. \quad (39)$$

W celu określenia wartości rzeczywistej oceny oprogramowania należy dokonać defuzyfikacji, przypisującej liczbie rozmytej LO wartość rzeczywistą R_{LO} :

$$R_{LO} = \frac{3 \cdot m_{R_{LO}} - \alpha_{R_{LO}} + \beta_{R_{LO}}}{3}. \quad (40).$$

4. Podsumowanie

Niezwykle istotną kwestią jest ocena jakości oprogramowania na każdym etapie projektowania, tworzenia, wdrażania czy testowania oprogramowania. Zaprezentowany w artykule model umożliwia ocenę poziomu wykonania zadania przez zleceniobiorcę i realizacji celów określonych przez potencjalnych użytkowników. W pracy przedstawiono model wielokryterialny, uwzględniający ocenę jakości zewnętrznej oprogramowania,

przy czym model ten może być modyfikowany zgodnie z zapotrzebowaniem na tego typu ocenę. Uzyskana ocena służy określeniu poziomu zrealizowanych założeń, jak też być wskazówką, które aspekty powinny być poprawione.

Bibliografia

1. Begier, B. (2007). *Doskonalenie jakości oprogramowania przez włączenie użytkowników w proces jego wytwarzania*. Poznań: Wydawnictwo Politechniki Poznańskiej.
2. Begier, B. (1999). *Inżynieria oprogramowania – problematyka jakości*. Poznań: Wydawnictwo Politechniki Poznańskiej.
3. Bilski, E., Dubielewicz, I. (2007). *Cykl życia oprogramowania – modele, procesy, jakość w normach ISO*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej.
4. Dubois, D., Prade, H. (1980). *Fuzzy set and systems – theory and applications*. New York: Academic Press.
5. Kacprzyk, J. (2001). *Wieloetapowe sterowanie rozmyte*. Warszawa: WNT.
6. Kacprzyk, J. (1986). *Zbiory rozmyte w analizie systemowej*. Warszawa: PWN.
7. Łachwa, A. (2001). *Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji*. Warszawa: AOW Exit.
8. Piegat, A. (1999). *Modelowanie i sterowanie rozmyte*. Warszawa: AOW Exit.
9. Przechlewski, T. (2012). *Modelowanie satysfakcji, użyteczności i wykorzystania oprogramowania Open Source*. Gdańsk: Wydawnictwo Uniwersytetu Gdańskiego.
10. Roma, A., Zmitrowicz, K. (2018). *Testowanie oprogramowania w praktyce*. Warszawa: PWN.
11. Stephen, K. (2006). *Metryki i modele w inżynierii jakości oprogramowania*. Warszawa: Wydawnictwo naukowe PWN SA.
12. Wiegiers, K., Beatty, J. (2014). *Specyfikacja oprogramowania. Inżynieria wymagań*. Gliwice: Helion.
13. Zmitrowicz, K. (2015). *Jakość projektów informatycznych. Rozwój i testowanie oprogramowania*. Gliwice: Helion.
14. Smilgin R. http://testerzy.pl/materialy/TheTestEye_SoftwareQualityCharacteristicsV1_PL.pdf

