

Modelowanie systemu oceny jakości cyklu życia oprogramowania z wykorzystaniem logiki rozmytej

Adam Nowicki

Jacek Szymański

Celem artykułu jest prezentacja modelu systemu oceny jakości cyklu życia oprogramowania oraz sposobu sterowania tym modelem. Językiem modelowania jest aparat matematyczny z zakresu logiki rozmytej. W ramach modelowanego systemu zidentyfikowano najistotniejsze podsystemy, zgodnie z tradycyjnym podziałem cyklu życia produktu na etapy, wytyczonym przez teoretyków inżynierii oprogramowania (podsystem analizy wymagań, modelowania, kodowania, testowania, wdrożenia oraz serwisowania). W każdym podsystemie wyróżniono kluczowe obszary poprawy jakości i stowarzyszone z nimi wskaźniki jakości, podane w formie rozmytej. Pokazano również, jak za pośrednictwem rozmytych operatorów agregacji wyznaczyć globalny wskaźnik jakości całego systemu. W dalszej części artykułu podano sposób sterowania utworzonym modelem w warunkach narzuconych ograniczeń i celów sterowania, pochodzących ze świata rzeczywistego. Na koniec pokazano, jak wyznaczyć globalny wskaźnik jakości sterowania modelowanym systemem.

1. Wstęp

Projektując system oceny jakości w przedsiębiorstwie, chętnie skupiamy się na badaniu produktu końcowego. Łatwiej jest przecież badać skutki, niż dochodzić ich przyczyn. Ponadto ocena produktu najczęściej opiera się na prostym pomiarze wielkości mierzalnych, a więc zgodności wyrobu ze specyfikacją.

W dziedzinie oprogramowania badamy przykładowo złożoność produktu programowego (ilość linii kodu, klas, metod, punktów funkcyjnych itp.), wydajność (np. czas odpowiedzi bazy danych na zapytanie użytkownika) lub kompatybilność (np. wielość formatów eksportowanych i importowanych plików). Jednak ocena całego przedsiębiorstwa informatycznego przez pryzmat produktu końcowego jest już nieporozumieniem. Dalece bardziej realistyczną koncepcją jest pomiar jakości całego procesu produkcji – cyklu życia oprogramowania, poczynając od etapu zbierania wymagań, aż po serwisowanie wdrożonych produktów. Źródło wysokiej awaryjności produktu zwykle leży w chaotycznie prowadzonych testach lub w pospiesznym kodowaniu pod presją czasu. Z kolei rozmijanie się oferowanej funkcjonalności z oczekiwaniami użytkowników to skutek stojącej na niskim poziomie inżynierii wymagań bądź pominięcia etapu modelowania.

Pomiar z użyciem precyzyjnie zdefiniowanych metryk oprogramowania, takich jak np. ilość linii kodu lub posługiwanie się słynnym już „mitycznym osobomie-

siącem”, może spotkać się wśród programistów z zarzutem spłylenia omawianej problematyki – sprowadzenia procesu twórczego do taśmowej produkcji, właściwej raczej zakładom przemysłowym. Metoda oceny powinna oprzeć się zarzutem subiektywizmu lub instrumentalizacji przedmiotu pomiaru (a więc programisty, który w imię szczytnych idei humanizmu w zarządzaniu ma być przecież podmiotem procesu pracy!). Podstawowy zarzut, którego chcemy uniknąć, to projektowanie wskaźników służących kontroli pracowników, wynikającej z braku zaufania kierownictwa, które nie ma pomysłu na pomiar efektywności wykorzystania ich wiedzy.

Według autorów, idealnym narzędziem pomiarowym będzie tutaj logika rozmyta (zob. np. Dubois, Prade 2000; Zimmermann 1999). Uzasadniamy to z jednej strony „miękkim” charakterem samego pojęcia jakości, a z drugiej istotnym, dominującym wpływem czynnika ludzkiego (a więc wymykającego się ścisłej parametryzacji) na proces produkcji oprogramowania. Logika rozmyta stanowi kompromis między metrykami, prowadzącymi do zbytejnej formalizacji i spłylenia problematyki, a koniecznością precyzyjnego określenia jakości projektu (czy to w celach naukowych, czy praktycznych – np. benchmarkingu). W dalszej części artykułu zaprezentujemy metodologię modelowania cyklu życia oprogramowania, wykorzystującą zestaw rozmytych wskaźników jakości systemu, poszerzonych podczas projektowania procesu sterowania o mierniki stopnia spełnienia ograniczeń i osiągnięcia założonych celów.

2. Modelowanie systemu

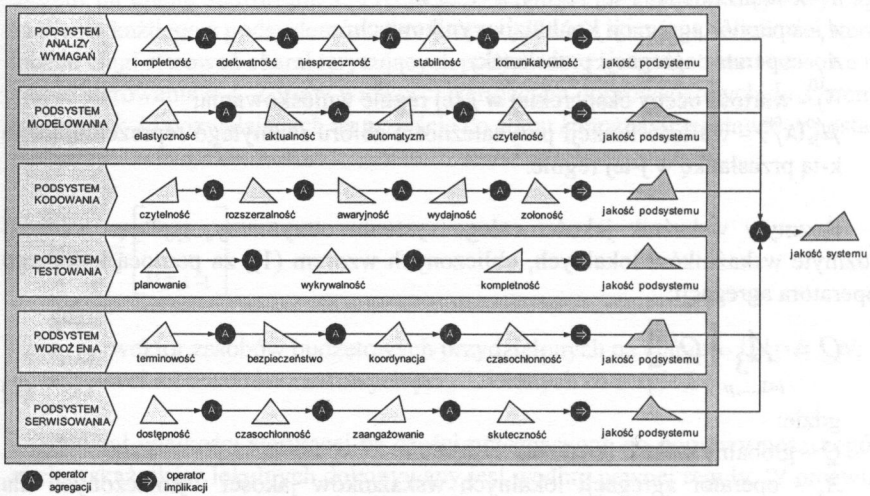
Potraktujmy cykl życia oprogramowania jako system. Przeprowadzając operację dekompozycji, otrzymamy, zgodnie z klasycznymi wytycznymi inżynierii oprogramowania, następujące podsystemy: analizy wymagań, modelowania, kodowania, testowania, wdrożenia i serwisu. W każdym z wymienionych podsystemów możemy z kolei zidentyfikować kluczowe obszary poprawy jakości funkcjonowania, które w dalszej części artykułu będziemy nazywać nośnikami jakości.

Postać modelowanego systemu, wraz z wyszczególnionymi przykładowymi nośnikami jakości i schematycznie przedstawionym procesem wnioskowania rozmytego, przedstawia rys. 1. Oczywiście listę podsystemów możemy poszerzyć, np. o podsystemy zarządzania zmianami, konfiguracją (środowiskiem), ryzykiem czy wreszcie o podsystem zarządzania całym przedsięwzięciem. Również lista nośników jakości jest zapewne niekompletna. Celem artykułu nie jest jednak utworzenie kompletnego modelu, lecz przedstawienie metodologii modelowania.

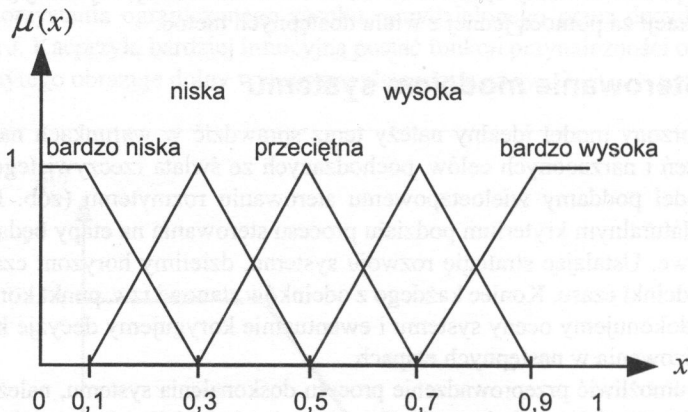
Korzystając z najważniejszego „wynałazku” teorii systemów – dekompozycji, prezentowany model można jednak z łatwością przedstawić na dowolnym poziomie szczegółowości, a dzięki budowie modułowej, zapewniamy jego elastyczność i rozszerzalność. Jedynym warunkiem efektywnej pracy z modelem jest myślenie systemowe – patrzeć na modelowany obiekt przez pryzmat powiązań wejściowo-wyjściowych i relacji przyczyna-skutek.

Jakość każdego nośnika będziemy oceniać, podając jedną z rozmytych wartości zmiennej lingwistycznej przedstawionej na rys. 2.

Po ustaleniu postaci zmiennej lingwistycznej badany system możemy modelować z wykorzystaniem klasycznych metod wnioskowania rozmytego. W tym celu należy zbudować bazę wiedzy dla każdego z wyszczególnionych podsystemów.



Rys. 1. Pomiar jakości cyklu życia oprogramowania. Źródło: opracowanie własne.



Rys. 2. Zmienna lingwistyczna „jakość”. Źródło: opracowanie własne na podstawie: Szymański (2004, s. 191).

Jest to zestaw reguł rozmytych modelujących odpowiedź podsystemu na określone oceny, przypisane przez eksperta poszczególnym nośnikom jakości. Rozmytą postać lokalnego wskaźnika jakości dla danego podsystemu, wyznaczoną z wykorzystaniem reguł rozmytych, opisuje wzór:

$$Q^{(i)} = A_1 \left[A_2 \mu_{jk}^{(i)}(x_j^{(i)}) \right] \quad (1)$$

$j=1, \dots, m$ $k=1, \dots, n$

gdzie:

j – j -ta reguła wnioskowania, modelująca zachowanie i -tego podsystemu, $j=1, \dots, m$; $i=1, \dots, p$;

k – k -ta przesłanka j -tej reguły, $k=1, \dots, n$;

A_1 – operator agregacji konkluzji wynikowych;

A_2 – operator agregacji przesłanek;

$x_j^{(i)}$ – wartość oceny eksperckiej w j -tej regule wnioskowania;

$\mu_{jk}^{(i)}(x_j^{(i)})$ – wartość funkcji przynależności zbioru rozmytego reprezentującego k -tą przesłankę w j -tej regule.

Rozmyty wskaźnik jakości całego systemu otrzymamy, scalając wartości rozmyte wskaźników lokalnych, obliczonych wzorem (1), za pomocą kolejnego operatora agregacji:

$$Q = A_3 \left[Q^{(i)} \right]_{i=1, \dots, p} \quad (2)$$

gdzie:

Q – globalny wskaźnik jakości systemu;

A_3 – operator agregacji lokalnych wskaźników jakości wyznaczonych dla poszczególnych podsystemów.

Uzyskany wskaźnik globalny kończy proces rozwiązywania modelu. Jego rozmytą postać możemy sprowadzić do wartości liczbowej, wykonując operację defuzyfikacji za pomocą jednej z wielu dostępnych metod.

3. Sterowanie modelem systemu

Utworzony model idealny należy teraz sprawdzić w warunkach nałożonych ograniczeń i narzuconych celów, pochodzących ze świata rzeczywistego. W tym celu model poddamy wieloetapowemu sterowaniu rozmytemu (zob. Kacprzyk 2001). Naturalnym kryterium podziału procesu sterowania na etapy będą momenty czasowe. Ustalając strategię rozwoju systemu, dzielimy horyzont czasowy na pewne odcinki czasu. Koniec każdego z odcinków stanowi tzw. punkt kontrolny, w którym dokonujemy oceny systemu i ewentualnie korygujemy decyzje konieczne do zrealizowania w następnych etapach.

Aby umożliwić przeprowadzenie procesu doskonalenia systemu, należy zapewnić, oprócz zasobów technicznych, finansowych i ludzkich, odpowiednią wielkość zasobów ekonomicznych. Można je oszacować, korzystając ze wskaźników czasu realizacji zadania, kosztu jego realizacji oraz pracochłonności (Nowicki 2000: 51). Korzystając z rozmytej postaci ograniczeń budżetowych i czasowych, uwzględnimy pozostające do dyspozycji zasoby ekonomiczne w modelowanym systemie. W ogólności każda decyzja sterująca podlega określonym ograniczeniom czasowym i budżetowym. W trakcie tworzenia modelu posługujemy się postacią ograniczenia rozmytego, którego przykład w odniesieniu do środków budżetowych przedstawia rys. 3.

Zadanie sterowania polega na wyznaczeniu wielkości zasobów finansowych i czasowych niezbędnych do osiągnięcia wyznaczonych celów przy nałożonych ograniczeniach. Do każdego ze wskaźników jakości podsystemu, opisanych wzorem (1), dodamy miarę jakości spełnienia ograniczeń i osiągnięcia celu sterowania.

Wskaźniki jakości poszczególnych podsystemów, opisane wzorem (1), możemy traktować jak elementy wektora zmiennych stanu, charakteryzującego

system na etapie sterowania k . Przejście systemu do następnego etapu wymaga zasilenia każdego z podsystemów określoną wielkością sterującą. Wielkości sterujące to ciągi sterowań charakteryzujących podjęte decyzje budżetowe i czasowe na etapie sterowania $k-1$. Zestaw wielkości sterujących doprowadzonych do systemu na etapie $k-1$ i pozwalających na przejście do etapu k można przedstawić w postaci macierzy:

$$\mathbf{u}_{k-1} = \begin{bmatrix} \mathbf{u}_{B,k-1} \\ \mathbf{u}_{T,k-1} \end{bmatrix} \quad (3)$$

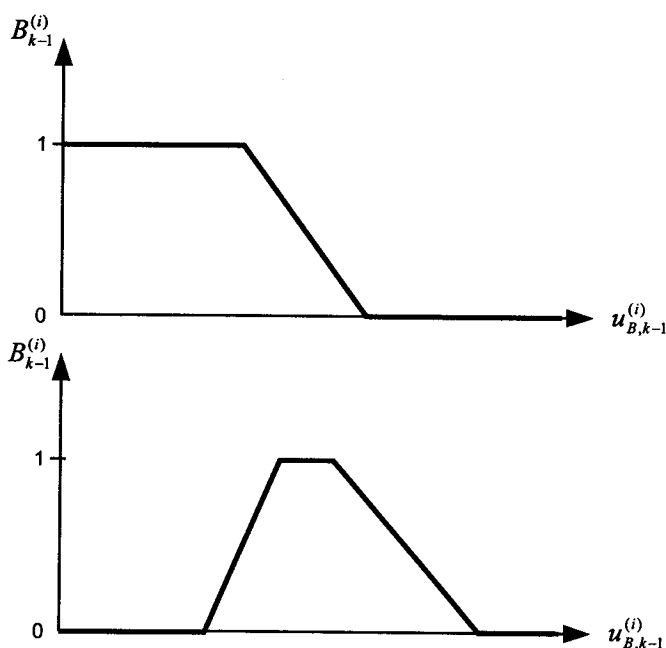
gdzie:

$\mathbf{u}_{B,k-1}$ - wektor zasobów budżetowych przydzielonych na etapie $k-1$, $k=1, \dots, N$;

$\mathbf{u}_{T,k-1}$ - wektor zasobów czasowych przydzielonych na etapie $k-1$.

Podział wielkości sterującej na części przeznaczone do poprawy poszczególnych wskaźników lokalnych dokonywany jest według pewnej reguły. W omawianym modelu zwykle podział jest dany ogólnie, decyzją kierownictwa, na podstawie obserwacji bieżącej sytuacji poszczególnych podsystemów, strategii rozwoju, koniecznych oszczędności itp.

Wskaźnik jakości spełnienia ograniczenia polega na rozmytej ocenie stopnia wykorzystania ograniczonego zasobu, przydzielonego przez decydenta. Jak zauważa J. Kacprzyk, bardziej intuicyjną postać funkcji przynależności ograniczenia rozmytego obrazuje dolny wykres przedstawiony na rys. 3.



Rys.3. Ograniczenie rozmyte. Źródło: opracowanie własne na podstawie (Kacprzyk 2001, s. 319). Legenda: zob. wzór (5).

Spełnienie ograniczeń „z nawiązką” może spowodować w przyszłości obniżenie poziomu środków budżetowych, pozostawionych do dyspozycji lub skrócenie harmonogramów realizacji kolejnych przedsięwzięć (por. Kacprzyk 2001: 319).

Wartość wskaźnika jakości spełnienia ograniczeń czasowych i budżetowych możemy wyrazić wzorami:

$$T_{k-1}^{(i)} = \mu_{k-1}^{(i)}(u_{T,k-1}^{(i)}) \quad (4)$$

gdzie:

$T_{k-1}^{(i)}$ - wskaźnik jakości spełnienia ograniczeń czasowych na $k-1$ etapie sterowania i -tym podsystemem;

$U_{T,k-1}^{(i)}$ - wartość zasobu czasowego, przeznaczanego na przejście i -tego podsystemu z etapu $k-1$ do k .

$$B_{k-1}^{(i)} = \mu_{k-1}^{(i)}(u_{B,k-1}^{(i)}) \quad (5)$$

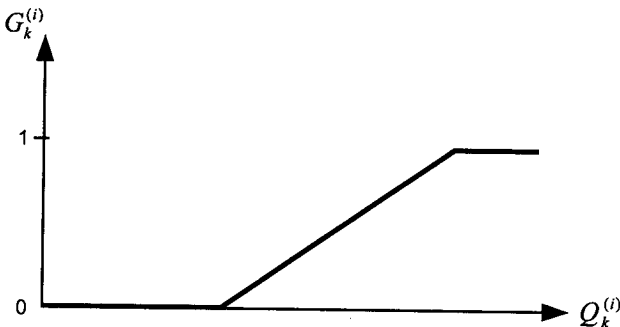
gdzie:

$B_{k-1}^{(i)}$ - wskaźnik jakości spełnienia ograniczeń budżetowych na $k-1$ etapie sterowania i -tym podsystemem;

$U_{B,k-1}^{(i)}$ - wartość zasobu budżetowego, przeznaczanego na przejście i -tego podsystemu z etapu $k-1$ do k .

Generalnie stopień spełnienia ograniczeń uznamy za bardzo niekorzystny, jeśli wielkość sterująca osiągnie lub przekroczy maksymalny dopuszczalny próg wartości. Oczywiście, jeśli przyjmiemy perspektywę wykonawców (pracujących w ramach ograniczeń), a nie decydentów (nakładających ograniczenia), odbiór sytuacji jest diametralnie inny – wtedy jawi się ona jako bardzo korzystna. Udało się przecież uzyskać większe środki do dyspozycji, niż początkowo zakładano. Odwrotna sytuacja zachodzi w przypadku, gdy zużyte zasoby nie osiągnęły nawet dolnego pułapu wyznaczonego przez nakładane ograniczenia.

Na każdym etapie realizacji algorytmu sterowania mamy również do czynienia z zestawem celów pośrednich, których osiągnięcie jest zakładane na danym etapie. Osiągnięcie celu sprecyzowanego dla danego podsystemu zależy



Rys.4. Cel rozmyty. Źródło: opracowanie własne na podstawie (Kacprzyk 2001, s. 322).
Legenda: zob. wzór (6).

od poziomu wskaźnika jakości charakteryzującego ten podsystem (zob. wzór (1)). Przykładowy cel rozmyty przedstawiono na rys. 4.

Proces wyznaczania wskaźników osiągnięcia celu polega na analizie, w jaki sposób zmienia się charakter przesłanek – nośników jakości dla każdego z podsystemów po podaniu na jego wejście określonej wielkości sterującej. Jest to ocena rozmyta stopnia oddalenia wartości wskaźnika jakości systemu, wyznaczonego wzorem (1), od wartości w pełni satysfakcjonującej decydenta. W uproszczeniu możemy wskaźnik osiągnięcia celu dla i -tego podsystemu na danym etapie sterowania zapisać jako:

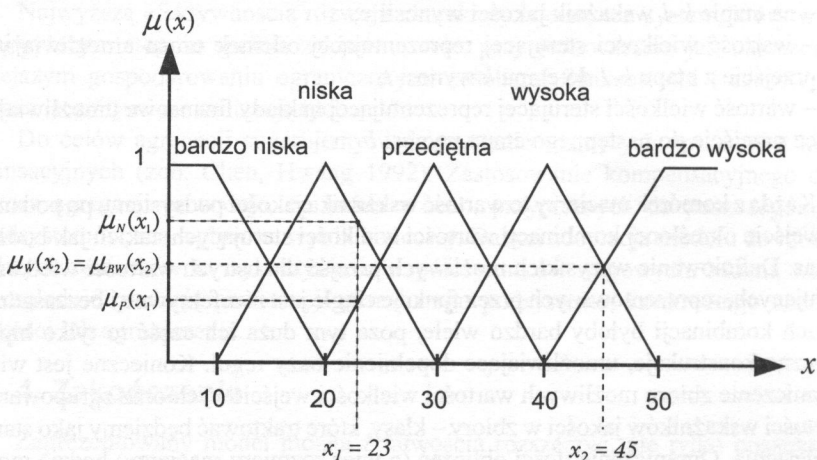
$$G_k^{(i)} = \mu_k^{(i)}(Q_k^{(i)}) \quad (6)$$

gdzie:

$G_k^{(i)}$ - wskaźnik jakości osiągnięcia celu sprecyzowanego dla i -tego podsystemu na k -tym etapie sterowania.

Wykorzystanie zaprezentowanego algorytmu wieloetapowego sterowania rozmytego w praktyce zarządczej pozwoli na projektowanie strategii rozwoju z uwzględnieniem oczekiwań decydentów. J. Kacprzyk proponuje w procesie sterowania uwzględnienie, oprócz celów decydentów (tzw. celów obiektywnych), również subiektywnych celów wykonawców procesów decyzyjnych (w naszym przypadku są to szeregowi pracownicy przedsiębiorstwa) (zob. Kacprzyk 2001: 326). Jest to kolejny sposób większego urealnienia wyników otrzymanych na wyjściu modelu.

Należy jeszcze zwrócić uwagę na to, że zwykle ograniczenia czasowe i budżetowe oraz wielkości sterujące są formułowane jako wartości „ostre” (np. określony budżet lub zakontraktowany termin zakończenia prac). Jeśli chcemy posługiwać się jedynie zbiorami rozmytymi, to konieczna jest fuzyfikacja wartości liczbowych. Przykład takiej operacji obrazuje rys. 5. Na rysunku przedstawiono sposób wyznaczenia wskaźnika jakości spełnienia ograniczeń dla pewnego podsystemu na



Rys. 5. Fuzyfikacja wielkości sterujących. Źródło: opracowanie własne na podstawie (Szymański 2004, s. 261).

określonym etapie sterowania. Ograniczmy dziedzinę problemu do przedziału, którego wartości brzegowe są zdeterminowane nałożonymi ograniczeniami. Powstaje przedział (10, 50), w ramach którego rozmieszczamy wartości rozmyte zmiennej lingwistycznej „*wykorzystanie zasobów*”, podobnie jak to zrobiliśmy w przypadku zmiennej „*jakość*”. Załóżmy, że wielkość umownych jednostek przeznaczonych na użytek sterowania wyniosła 23. Otrzymujemy dwie wartości funkcji przynależności dla tej wielkości sterującej: $\mu_N(x_1)$ oraz $\mu_P(x_1)$. Wybieramy wskazanie o wyższej wartości funkcji przynależności, jako silniej osadzone w rzeczywistości. Tak więc wykorzystanie 23 jednostek będzie decyzją niekorzystną, ponieważ funkcja przynależności $\mu_N(x_1)$ określona jest dla wartości „*niekorzystna*” zmiennej lingwistycznej „*wykorzystanie zasobów*”. Oczywiście subiektywnego wyboru musimy dokonać w sytuacji, gdy wartości funkcji przynależności pokrywają się (np. gdy wartość zmiennej sterującej wynosi 45, zob. rys. 5). Wtedy musimy zdać się na subiektywną opinię eksperta.

Osobny problem stanowi powiązanie wskaźnika jakości systemu z podawanymi na jego wejście wielkościami sterującymi. To zadanie sprowadza się do skonstruowania tzw. macierzy przejść stanów. W ramach rozpatrywanej problematyki stan obiektu sterowania charakteryzowany będzie przez jego wskaźnik jakości. Korzystając z macierzy, możemy określić, jak zmieni się wskaźnik jakości danego podsystemu przy przejściu z etapu na etap pod warunkiem podania na wejścia podsystemu określonych wartości wielkości sterujących. Wyznaczenie macierzy przejść jest więc warunkiem koniecznym obliczenia wskaźnika jakości osiągnięcia celu, ponieważ jego wartość jest uzależniona od wskaźnika jakości podsystemu. Ogólną postać macierzy przejść stanów dla *i-tego* podsystemu przedstawia wzór:

$$M_{(k-1,x)}^{(i)} = \left[Q_{(t,b)/(k-1,x)}^{(i)} \right] \quad (7)$$

Komórka macierzy, określana jako $Q_{(t,b)/(k-1,x)}^{(i)}$, reprezentuje wartość wskaźnika jakości podsystemu przy następujących założeniach:

- system przechodzi z etapu $k-1$ do etapu k ;
- na etapie $k-1$ wskaźnik jakości wynosił x ;
- wartość wielkości sterującej reprezentującej odcinek czasu umożliwiającą przejście z etapu $k-1$ do etapu k wynosi t ;
- wartość wielkości sterującej reprezentującej nakłady finansowe umożliwiające przejście do następnego etapu wynosi b .

Każda z komórek macierzy to wartość wskaźnika jakości podsystemu po podaniu na wejście określonej kombinacji wartości wielkości sterujących, takich jak budżet i czas. Definiowanie wszystkich możliwych przejść dla ostrych wartości wielkości sterujących, reprezentowanych przez funkcje ciągłe jest nieefektywne i bezzasadne. Takich kombinacji byłoby bardzo wiele, poza tym duża ich część to tylko hipotetyczne konstrukcje, umożliwiające dopełnienie bazy reguł. Konieczne jest więc ograniczenie zbioru możliwych wartości wielkości wejściowych oraz zgrupowanie wartości wskaźników jakości w zbioru – klasy, które traktować będziemy jako stany odniesienia. Ograniczenie ilości obliczeń (a więc rozmiaru macierzy) będzie możliwe przy zaakceptowaniu zbiorów rozmytych jako sposobu modelowania wiedzy ekspertów. Przykładowe zdanie wchodzące w skład bazy wiedzy reprezentującej macierz przejść stanów będzie wtedy wyglądało następująco:

„Jeśli wartość dostępnych środków budżetowych jest *wysoka* oraz oddano do dyspozycji *przeciętną* ilość czasu, to przy przejściu ze stanu $k-1$, w którym jakość podsystemu była *bardzo niska*, do stanu k podsystem osiągnie *niski* poziom jakości”.

Wskaźnik jakości sterowania dla danego podsystemu na etapie k będzie miał postać:

$$E_k^{(i)} = A_4 (B_{k-1}^{(i)}, T_{k-1}^{(i)}, G_k^{(i)}) \quad (8)$$

gdzie:

$E_k^{(i)}$ - wskaźnik jakości sterowania i -tym podsystemem na k -tym etapie sterowania;

A_4 - operator agregacji wskaźników spełnienia ograniczeń oraz wskaźnika osiągnięcia celu sterowania.

Agregując wskaźniki jakości sterowania dla poszczególnych podsystemów możemy wyznaczyć wskaźnik jakości dla całego systemu na etapie sterowania k . Wykonując kolejną operację agregacji otrzymamy wskaźnik jakości całego procesu sterowania:

$$E_k = A_5 \left[A_6 (E_k^{(i)}) \right]_{\substack{k=1, \dots, N \\ i=1, \dots, p}} \quad (9)$$

gdzie:

E_k - wskaźnik jakości sterowania systemem;

A_5 - operator agregacji wskaźników jakości sterowania systemem na poszczególnych etapach sterowania;

A_6 - operator agregacji wskaźników jakości sterowania dla poszczególnych podsystemów na k -tym etapie sterowania.

Najwyższą efektywnością rozwoju charakteryzuje się ten system, który w jak najwyższym stopniu spełnia zakładane cele, przy jednocześnie jak najoszczędniejszym gospodarowaniu ograniczonymi zasobami finansowymi i czasowymi, przeznaczonymi na realizację tych celów.

Do celów agregacji sugerujemy wykorzystanie rozmytych operatorów kompensacyjnych (zob. Chen, Hwang 1992). Zastosowanie kompensacyjnego operatora agregacji powoduje, że każda z ocen, przypisanych do poszczególnych podsystemów, ma wpływ na wynik końcowy. Dalsza adaptacja modelu do rzeczywistych warunków to wykorzystanie współczynników wagowych. Można wtedy uwzględnić gradację preferencji określonego podsystemu, konkretnego nośnika jakości lub ograniczenia.

4. Zakończenie

Zaprezentowany model można z łatwością rozszerzyć, nie tylko poszukując nowych nośników jakości, ale również wychodząc poza tradycyjnie rozumiany cykl życia oprogramowania. Przykładowo, dodając podsystem zarządzania wiedzą, zarządzania innowacjami, zarządzania kadrami lub ich kombinację, uży-

skamy model zarządzania jakością, obejmujący całe przedsiębiorstwo, a nie tylko jego sferę produkcyjną. Co więcej, wymieniając podsystem produkcji, którym jest w prezentowanym artykule cykl życia oprogramowania, na dowolny cykl życia produktu, uzyskamy elastyczny model zarządzania jakością w dowolnej organizacji uczącej się. Dzięki takim zabiegom możemy próbować prognozować wpływ „miękkich” składników przedsiębiorstwa, takich jak „wiedza”, „kultura”, „klimat”, „satisfakcja”, na jakość procesu produkcji i w rezultacie – jakość produktu końcowego. W tym miejscu warto podkreślić, że włączenie takich niemierzalnych liczbowo cech do procesu oceny jakości jest możliwe dzięki wykorzystaniu aparatu wnioskowania rozmytego.

Na zakończenie zaznaczmy, że warunkiem koniecznym użytkowania prezentowanego modelu jest przestrzeganie zasady ciągłego uczenia systemu, poprzez uwzględnianie w bazie wiedzy nowych reguł lub dokonywanie korekt proponowanych przez praktyków zarządzania i szeregowych pracowników, na co dzieją borykających się z problemami wynikającymi z realizacji projektów programistycznych.

Informacje o autorze

Prof. zw. dr hab. Adam Nowicki – Kierownik Katedry Inżynierii Systemów Informatycznych Zarządzania, Akademia Ekonomiczna im. Oskara Langego we Wrocławiu.
E-mail: adam.nowicki@ae.wroc.pl.

Dr inż. Jacek Szymański - absolwent Studium Doktoranckiego AE we Wrocławiu, pracownik Departamentu Rozwoju Systemów AIG Credit SA.
E-mail: Jacek.Szymanski@aigcredit.pl

Bibliografia

- Chen, S-J. i C-L. Hwang. 1992. *Fuzzy Multiple Decision Making. Methods and Applications*. Berlin: Springer-Verlag.
- Dubois D., i H. Prade. 2000. *Fundamentals of Fuzzy Sets*. Boston, MA: Kluwer Academic Publishers.
- Kacprzyk, J. 2001. *Wieloetapowe sterowanie rozmyte*. Warszawa: WNT.
- Nowicki, A. 2000. *Systemowe ujęcie doskonalenia systemu informacyjnego w zarządzaniu przedsiębiorstwem*. w: Nowicki A. (red.) *Zarys teorii doskonalenia systemów informacyjnych w zarządzaniu*. Wrocław: Wyd. AE Wrocław.
- Szymański, J. 2004. *Zarządzanie jakością w procesie tworzenia oprogramowania. Podejście modelowe*. Rozprawa doktorska. Wrocław: Wyd. AE Wrocław.
- Zimmermann, H. J. 1999. *Practical applications of fuzzy technologies*. Boston, MA: Kluwer Academic Publishers.