

WYCENA SYSTEMÓW OPROGRAMOWANIA REALIZOWANYCH NA POTRZEBY ADMINISTRACJI PUBLICZNEJ

1. Wstęp

Systemy oprogramowania wspomagające zadania realizowane w obszarze administracji publicznej mają często charakter dedykowanych, nietypowych, unikatowych produktów, które powstają od podstaw na indywidualne zamówienie pojedynczego zleceniodawcy. Wynika to z faktu, iż w tym obszarze mamy zwykle do czynienia z oryginalnymi, niespotykanymi w sferze działalności gospodarczej zadaniami, często realizowanymi przez pojedyncze instytucje, zadaniami, dla których nie zostały wypracowane komercyjne rozwiązania lub są one trudne bądź niemożliwe do przeniesienia z innych obszarów czy krajów.

W związku z tym w przypadku takich systemów występują szczególne trudności w ich wycenie. Jeżeli bowiem planowaniu podlegają nakłady na gotowe systemy oprogramowania („systemy z półki”; ang. *Commercial-Off-The-Shelf* – COTS), to – podobnie jak w przypadku wydatków sprzętowych – ich oszacowanie zwykle nie nastręcza większych trudności, ponieważ zleceniodawca może się przy tym odnieść do cen rynkowych (nierzadko negocjowanych). Natomiast w przypadku dedykowanych systemów oprogramowania, a nawet kastomizowanych „systemów z półki”, nakłady inwestycyjne, a także czas ich realizacji są uzależnione przede wszystkim od nakładów pracy wydatkowanych na działania niezbędne do skonstruowania od podstaw

nowego systemu lub udoskonalenia istniejącego. Właściwe oszacowanie pracochłonności ma tym bardziej istotne znaczenie, że przedsięwzięcia zmierzające do rozwoju takich systemów to często poważne przedsięwzięcia inwestycyjne, których koszty są porównywane nawet z kosztami budowy 50-piętrowego drapacza chmur, zadającego stadionu piłkarskiego czy morskiego statku wycieczkowego o wyporności ok. 70 000 ton¹. Tymczasem kwoty te zlecniodawca nierzadko wydatkuje bez poparcia decyzji o zaangażowaniu w taką inwestycję właściwą analizą kosztów, opartą na racjonalnych, wystarczająco obiektywnych i wiarygodnych podstawach. Wynika to z faktu, iż obiektywne i wiarygodne oszacowanie pracochłonności działań projektowych stanowi ciągle duże wyzwanie dla inżynierii oprogramowania. Toteż od wielu lat na świecie poszukuje się trafnych i wiarygodnych podejść do oceny *ex ante* kosztów omawianych systemów, tak aby istniała możliwość podjęcia wobec nich racjonalnych decyzji inwestycyjnych.

Co więcej, badania przeprowadzone przez autorkę niniejszego artykułu wskazują, że w polskich instytucjach administracji publicznej ciągle nie widzi się w praktyce potrzeby oceny podejmowanych przedsięwzięć rozwoju systemów oprogramowania z perspektywy ekonomicznej, argumentując ów fakt w większości przypadków poza-biznesowymi celami systemów wdrażanych w tego typu organizacjach². Analiza wykorzystania poszczególnych metod szacowania pracochłonności wykazała, że najczęściej wykonawcy takich przedsięwzięć wskazują na technikę „wyceny dla wygranej”, co jest spowodowane przede wszystkim jej powszechnym stosowaniem w przypadku dostarczania dedykowanych systemów oprogramowania na potrzeby instytucji administracji publicznej, ze względu na preferowanie przez takich odbiorców ofert najtańszych w rezultacie uregulowań prawnych. Decyzję zlecniodawcy opartą na takim podejściu trudno uznać za racjonalną: „wycena dla wygranej” sprowadza się bowiem do świadomego zaniżania szacowanych kosztów całkowitych przedsięwzięcia, w rezultacie czego następuje zwykle istotne przekroczenie zakładanych wydatków lub dostosowanie do zaniżonych kosztów funkcji i cech produktu, które są dalekie od spełnienia wymagań zlecniodawcy. Zleceniobiorca wycenia przecież wtedy system jedynie z myślą o wygraniu kontraktu, nie zaś bazując na obiektywnym i wiarygodnym podejściu i abstrahując przy tym od właściwości produktu decydujących o jego pracochłonności.

W polskiej (i nie tylko) praktyce realizacji dedykowanych systemów oprogramowania można także spotkać inne podejścia do wyceny produktu: kontrakty o stałej

¹ T.C. Jones, *Software Project Management in the Twenty-first Century*, Software Productivity Research, Burlington 1999, s. 3.

² Rezultaty tych badań zostały obszernie zaprezentowane w: B. Czarnacka-Chrobot, *Analysis of the Functional Size Measurement Methods Usage by Polish Business Software Systems Providers*, „Lecture Notes in Computer Science” 2009, vol. 5891, *Software Process and Product Measurement*, red. A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, J. Brunekreef, Springer-Verlag, Berlin-Heidelberg, s. 17-34.

cenie oraz kontrakty czasowo-materiałowe. W pierwszym przypadku należność za produkt przedsięwzięcia określa się na bazie założonych stałych nakładów, uzgodnionych na podstawie specyfikacji wymagań, oszacowanych zazwyczaj w oparciu o zasoby lub działania. Oznacza to, że zleceniodawca płaci *de facto* nie za rzeczywiście dostarczoną w produkcie funkcjonalność, a za działania/zasoby wydatkowane przez zleceniodawcę. W kontraktach drugiego typu kalkulacja należności za produkt opiera się na ustalonej stawce za wydatkowane przez jego wykonawcę godziny pracy. Powoduje to, że osobowy koszt jednostkowy jest mierzony w odniesieniu do jednostki czasu pracy, w związku z czym ów czas decyduje o całkowitych kosztach osobowych. W tym przypadku zatem zleceniodawca także nie płaci za faktycznie dostarczoną w produkcie funkcjonalność, a za czas pracy wykonawcy. Realizacja projektu z tzw. kosztorysem powykonawczym, czyli wyceną *ex post* faktycznie dostarczonego produktu, jest ciągle rzadkością, przynajmniej w warunkach polskich, w których mamy do czynienia z niskim (choć rosnącym) poziomem „kultury wymiarowania” w inżynierii oprogramowania.

Podjęcia te sprzyjają także przekroczeniom wydatków zaplanowanych na dostarczenie produktu. W przypadku umów między zleceniodawcą a zleceniobiorcą opartych na godzinowej stawce pracy projektantów dostawca może świadomie wydłużać czas realizacji systemu (badania Standish Group pokazują, że obecnie wśród projektów zakończonych niepowodzeniem częściowym średnie przekroczenie czasu przedsięwzięcia wynosi ok. 80%³). Nie ma też gwarancji, że w rezultacie dostarczy on produkt o wymaganych funkcjach i cechach (według Standish Group produktom dostarczonym w wyniku projektów zakończonych niepowodzeniem częściowym brakuje średnio ok. 35% funkcji⁴). W przypadku zaś kontraktów o stałej cenie, oprócz prawdopodobnej sytuacji, w której rzeczywiście dostarczony system oprogramowania może mieć mniejszy zakres od wymaganego, pojawia się jeszcze inny problem: dostawcy, często nie uwzględniając w sposób obiektywny i wiarygodny wpływu zmian w wymaganiach zleceniodawcy na koszty przedsięwzięcia, bronią się przed jakimikolwiek rozszerzeniami wymagań, które przecież są charakterystyczne dla rozważanych przedsięwzięć.

Można zatem postawić hipotezę, że te spotykane najczęściej w praktyce podjęcia do wyceny systemów oprogramowania sprzyjają przekroczeniu wydatków zaplanowanych na dostarczenie produktu spełniającego wymagania zleceniodawcy. Dlatego w obszarze tym należy poszukiwać innych rozwiązań, co z wyżej wymienionych powodów ma szczególne znaczenie w przypadku dedykowanych systemów oprogramowania realizowanych na potrzeby administracji publicznej. Wobec tego celem niniejszego artykułu jest przybliżenie podejścia do wyceny tego typu systemów, jakie

³ Standish Group, *CHAOS summary 2009*, West Yarmouth, Massachusetts 2009, s. 1.

⁴ Ibidem.

w ostatnim czasie na świecie zyskuje coraz większą popularność (punkt 2), a na tym tle przybliżenie zweryfikowanego w praktyce autorskiego modelu służącego wycenie takich systemów w oparciu o to podejście (punkt 3). Artykuł kończą wnioski dotyczące prowadzonych rozważań i zaprezentowanych rozwiązań (punkt 4).

2. Podejście funkcjonalne do wyceny systemów oprogramowania

2.1. Koncepcja i metody wymiarowania funkcjonalnego oprogramowania

Analiza przeprowadzona przez autorkę niniejszego artykułu wykazała, że zasadnicze znaczenie dla trafności ocen szacunkowych pracochłonności niezbędnej do realizacji dedykowanych systemów oprogramowania mają trzy następujące czynniki⁵:

- 1) wykorzystanie metody szacowania opartej na rezultacie przedsięwzięcia, tj. rozmiarze produktu, gdyż to on przede wszystkim decyduje o nakładach pracy, w praktyce natomiast szacowanie kosztów odbywa się najczęściej na podstawie działań/zasobów, co prowadzi do nawet piętnastokrotnych różnic w kosztach dla bardzo zbliżonych systemów oprogramowania⁶;
- 2) dostępność i wykorzystanie odpowiednich, tj. wynikających ze specyfiki funkcjonowania określonej organizacji projektującej, danych historycznych, co pozwala na dostosowanie ogólnych oszacowań do specyfiki wykonawcy;
- 3) wyrażenie rozmiaru produktu przedsięwzięcia we właściwych jednostkach.

Liczne badania wykazują, że w kontekście szacowania kosztów (i nie tylko) najważniejszymi jednostkami rozmiaru produktu programowego są jednostki funkcjonalności produktu⁷. Warto tutaj nadmienić w szczególności o badaniach zleconych przez State Government of Victoria z Australii, które wykazały, że wycena systemów oprogramowania realizowanych na potrzeby tej instytucji administracji publicznej na bazie jednostki funkcjonalności powoduje zmniejszenie średniego przekroczenia

⁵ Problem ten został dokładnie przeanalizowany w: B. Czarnacka-Chrobot, *Wiarygodność metod szacowania pracochłonności przedsięwzięć rozwoju systemów oprogramowania wspomagających zarządzanie*, „Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu” 2010, nr 88, *Informatyka Ekonomiczna. Informatyka w zarządzaniu*, red. J. Sobieska-Karpińska, s. 163–176.

⁶ State Government of Victoria, *southernSCOPE, Reference Manual*, Version 1, Government of Victoria, Melbourne, September 2000, s. 1.

⁷ Szerokie rozważania na ten temat zob. np.: B. Czarnacka-Chrobot, *The Economic Importance of Business Software Systems Size Measurement*, Proceedings of the 5th International Multi-Conference on Computing in the Global Information Technology (ICCGI 2010), 20–25 September 2010, Valencia, Spain, red. M. Garcia, J.-D. Mathias, IEEE Computer Society Conference Publishing Services, Los Alamitos, California–Washington–Tokyo 2010, s. 293–299.

budżetu, wynoszącego obecnie według analiz Standish Group ok. 55%⁸, do mniej niż 10%⁹. Również analizy International Software Benchmarking Standards Group (ISBSG) potwierdzają zasadność takiego podejścia. W ich wyniku ustalono, że przedsięwzięcia, w których produkt programowy wyceniano przy wykorzystaniu jednostki funkcjonalności, charakteryzują się trafnymi szacunkami: dla 90% przypadków oceny szacunkowe kosztów wykazały odchylenie nieprzekraczające 20% ich rzeczywistej wartości, przy czym w przypadku 70% projektów przekroczenie nie było większe niż 10% faktycznych nakładów¹⁰.

To właśnie m.in. wykazane w wyniku wieloletniej weryfikacji wiarygodność i obiektywność ocen szacunkowych pracochłonności i kosztów uzyskiwanych w oparciu o rozmiar systemu oprogramowania, wyrażony w jednostkach funkcjonalności, doprowadziły do standaryzacji w ostatnich latach samej koncepcji, a także opartych na niej metod tzw. wymiarowania rozmiaru funkcjonalnego oprogramowania (ang. *software Functional Size Measurement* – FSM) przez ISO (International Organization for Standardization) i IEC (International Electrotechnical Commission). Rozmiar systemu oprogramowania mierzy się bowiem przede wszystkim po to, aby na jego podstawie móc oszacować kluczowe atrybuty przedsięwzięcia: pracochłonność, koszty i czas jego realizacji.

Zbiór zasad wymiarowania rozmiaru systemu oprogramowania w jednostkach jego funkcjonalności zawarto w sześcioczęściowej normie ISO/IEC 14143¹¹. Na potrzeby takiego wymiarowania w standardzie tym zaproponowano definicję rozmiaru funkcjonalnego (ang. *functional size*) produktu programowego, przez który rozumie się „rozmiar oprogramowania otrzymany przez ilościowe określenie wymagań funkcjonalnych użytkownika”¹². Przy czym przez wymagania funkcjonalne użytkownika (ang. *Functional User Requirements* – FUR) rozumie się z kolei „podzbiór wymagań użytkownika opisujący to, co oprogramowanie ma robić, w kategoriach zadań i usług”¹³.

Wśród metod uznanych przez ISO i IEC za zgodne z zasadami FSM zawartymi w normie 14143 znajdują się obecnie:

⁸ Standish Group, *CHAOS summary*, op.cit., s. 1.

⁹ Sage Technology, *Report on the SCUD Methodology Review*, June 2000.

¹⁰ International Software Benchmarking Standards Group, *The ISBSG Report: Software Project Estimates – How accurate are they?*, ISBSG, Hawthorn, VIC, January 2005, s. 2, 4–5.

¹¹ ISO/IEC 14143 *Information Technology – Software measurement – Functional size measurement* – Parts 1–6, ISO, Geneva 2011.

¹² ISO/IEC 14143 *Information Technology – Software measurement – Functional size measurement* – Part 1, ISO, Geneva 2011, s. 2.

¹³ ISO/IEC 14143 *Information Technology – Software measurement – Functional size measurement* – Part 1, ISO, Geneva 2011, s. 2.

- metoda punktów funkcyjnych IFPUG (International Function Point Users Group)¹⁴ – najpopularniejsza z technik wymiarowania rozmiaru funkcjonalnego oprogramowania;
- metoda punktów funkcyjnych MkII (Mark II) rozwijana przez UKSMA (United Kingdom Software Metrics Association)¹⁵ – technika popularna w Wielkiej Brytanii, oferująca wyższy poziom szczegółowości wymiarowania w zestawieniu z metodą IFPUG;
- metoda punktów funkcyjnych NESMA (Netherlands Software Metrics Association)¹⁶ – holenderska i uproszczona wersja metody punktów funkcyjnych IFPUG;
- metoda punktów funkcyjnych COSMIC (Common Software Measurement International Consortium)¹⁷ – oferująca inne podejście do wymiarowania rozmiaru funkcjonalnego oprogramowania w zestawieniu z wyżej wymienionymi metodami, skutkujące jej większą uniwersalnością w odniesieniu do różnych kategorii oprogramowania;
- metoda FSM w wersji FiSMA (Finnish Software Measurement Association)¹⁸ – metoda zbliżona do metody COSMIC, popularna głównie w Finlandii.

Metody FSM posiadające akceptację ISO/IEC różnią się możliwościami wymiarowania oprogramowania w odniesieniu do różnych domen funkcjonalnych (kategorii oprogramowania). Dlatego przed wyborem określonej metody należy najpierw ocenić jej adekwatność wobec rodzaju oprogramowania, którego rozmiar funkcjonalny ma podlegać wymiarowaniu. Jednakże w przypadku systemów wspomagających pracę instytucji administracji publicznej wszystkie z metod FSM uznanych przez obie organizacje standaryzacyjne będą adekwatne, gdyż mają one w przeważającej części charakter systemów sterowanych danymi.

Wymiarowanie rozmiaru funkcjonalnego produktu programowego wspomaga¹⁹:

- 1) zarządzanie przedsięwzięciem rozwoju systemu oprogramowania, w obszarze zarówno jego planowania, jak i kontroli, poprzez umożliwienie m.in.: wczesnego prognozowania zasobów niezbędnych do jego realizacji, śledzenia postępów w jego realizacji, zarządzania zmianami w wymaganym rozmiarze produktu przedsięwzięcia czy analizy powykonawczej przedsięwzięcia;

¹⁴ ISO/IEC 20926 *Software and systems engineering – Software measurement – IFPUG functional size measurement method 2009*, ISO, Geneva 2009.

¹⁵ ISO/IEC 20968 *Software engineering – Mk II Function Point Analysis – Counting practices manual*, ISO, Geneva 2002.

¹⁶ ISO/IEC 24570 *Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis*, ISO, Geneva 2005.

¹⁷ ISO/IEC 19761 *Software engineering – COSMIC: a functional size measurement method*, edition 2, ISO, Geneva 2011.

¹⁸ ISO/IEC 29881 *Information Technology – Software and systems engineering – FiSMA 1.1 functional size measurement method*, ISO, Geneva 2010.

¹⁹ ISO/IEC 14143 *Information Technology – Software measurement – Functional size measurement – Part 6*, ISO, Geneva 2011, s. 9–10.

- 2) zarządzanie wytwarzaniem oprogramowania poprzez m.in.: zarządzanie produktywnością procesów budowy, rozbudowy i utrzymania oprogramowania, zarządzanie jakością, zwłaszcza niezawodnością oprogramowania (mierzoną stopą błędów, która wyraża stosunek liczby zidentyfikowanych w kodzie programu błędów do rozmiaru produktu), zarządzanie szybkością realizacji (mierzoną stosunkiem rozmiaru produktu do czasu realizacji), zarządzanie dojrzałością organizacji i możliwościami procesów, umożliwienie wyznaczenia organizacyjnej wartości całego lub części oprogramowania w celu oszacowania kosztu jego ewentualnego zastąpienia, reengineeringu i outsourcingu, umożliwienie prognozowania budżetu niezbędnego do utrzymania oprogramowania czy zarządzanie kontraktami na dostarczanie oprogramowania.

Popularność koncepcji i metod FSM w ostatnim czasie na świecie bardzo dynamicznie rośnie – jest to kwestia ostatnich kilku lat. Wynika to głównie z ich wykazanej w praktyce skuteczności, obiektywności i wiarygodności, zwłaszcza przy ocenie *ex ante* kosztów osobowych realizacji systemów oprogramowania, w tym w szczególności dedykowanych wspomaganie pracy instytucji administracji publicznej. I tak np. metoda IFPUG jest powszechnie wykorzystywana w sferze administracji publicznej w USA, gdzie stanowi metodę rekomendowaną przez Government Accountability Office (GAO) – instytucję kontrolną Kongresu Stanów Zjednoczonych – przy realizacji systemów znajdujących się pod jej kontrolą. Podejście zaproponowane przez COSMIC jest zaś od niedawna standardem narodowym w Japonii i Hiszpanii, a na przełomie lat 2010 i 2011 GAO dodał ją także do listy metod przez siebie rekomendowanych. Metoda ta wywodzi się z Kanady, gdzie jest stosowana w sferze administracji publicznej, np. przez Narodowy Bank Kanady i Narodowy Departament Obrony. Wykorzystuje ją również Komisja Europejska (np. Dyrektoriat Unii Podatkowej i Celnej). W Wielkiej Brytanii stosowanie metody UKSMA stanowi wymóg formalny przy realizacji przedsięwzięć rozwoju systemów oprogramowania na potrzeby administracji publicznej powyżej pewnej wielkości, a dla wszystkich takich przedsięwzięć jest ona rekomendowana przez Central Computer and Telecommunications Agency. Podobnie jest z innymi metodami FSM, zaakceptowanymi przez ISO/IEC w Skandynawii, Włoszech, Australii i innych krajach. Co więcej, metody te wykorzystują nie tylko dostawcy, lecz także coraz częściej zleceniodawcy omawianych systemów²⁰.

Jak pokazują badania autorki niniejszego artykułu, także w Polsce wśród wykonawców dedykowanych systemów oprogramowania rośnie zainteresowanie metodami FSM, zwłaszcza w kontekście szacowania kosztów osobowych realizacji takich systemów. Chociaż poziom zarówno ich znajomości, jak i wykorzystania nie

²⁰ G. Rule, *The most common Functional Size Measurement (FSM) methods compared*, Software Measurement Services, St. Clare's, Mill Hill, Edenbrige, Kent, UK 2010, s. 1–8.

jest imponujący, to jednak rośnie, a przeważająca większość osób znających metody FSM również je wykorzystuje – ze względu na przekonanie o ich skuteczności, wiarygodności i obiektywności²¹.

Do pozostałych zasadniczych powodów wzrostu popularności podejścia funkcjonalnego do wyceny systemów oprogramowania, zarówno w Polsce, jak i na świecie, należy zaliczyć: większą dbałość o środki finansowe w czasach recesyjnych i precesyjnych, coraz większą konkurencję na rynku i rosnący poziom jego globalizacji, coraz wyższą świadomość zlecniodawców, a w związku z tym coraz wyższe wymagania co do konieczności uzasadniania oferowanych kosztów i czasu realizacji przedsięwzięcia przez wykonawców, a także bez wątpienia akceptację koncepcji FSM i kilku z takich metod przez uznane międzynarodowe organizacje standaryzacyjne.

2.2. Metodyki *southernSCOPE* i *northernSCOPE*

Koncepcja i metody FSM stanowią podstawę dwóch metodyk zarządzania tzw. zakresem funkcjonalnym przedsięwzięć rozwoju systemów oprogramowania: metodyki *southernSCOPE* oraz stworzonej na jej bazie metodyki *northernSCOPE*. Metodyka *southernSCOPE*²² powstała w celu wspomagania zlecniodawców przy wycenie systemów oprogramowania budowanych na ich potrzeby, a motywacją do jej opracowania stały się nie tylko olbrzymie wydatki na tego typu produkty, lecz także rezultaty badań Standish Group, świadczące o wysokiej skali niepowodzenia w realizacji przedsięwzięć zmierzających do ich rozwoju²³. Za przyczynę tego stanu rzeczy uznano fakt, iż w obszarze przedsięwzięć rozwoju dedykowanych systemów oprogramowania jeszcze niedawno nie wykorzystywano metod wymiarowania ich rezultatu (produktu), opierając się przy jego wycenie na nieskutecznym podejściu polegającym na pomiarze wejść (zasobów) do tego procesu. Zdaniem autorów metodyki, powstanie metod FSM diametralnie zmieniło tę sytuację, a rozwój repozytoriów z danymi historycznymi, na których są oparte, np. repozytorium International Software Benchmarking Standards Group²⁴ czy Software Productivity Research²⁵, pozwala na ich bazie coraz trafniej szacować koszty realizacji takich przedsięwzięć, dostarcza także nowej wiedzy o charakterze porównawczym, czego efektem jest możliwość coraz bardziej skutecznego zarządzania ich zakresem.

²¹ Więcej szczegółów na ten temat zob.: B. Czarnacka-Chrobot, *Analysis of the Fuctional*, op.cit. Badania w tym obszarze będą kontynuowane.

²² State Government of Victoria, op.cit.

²³ Standish Group, *CHAOS Summary*, op.cit.

²⁴ International Software Benchmarking Standards Group, *Data demographics – release 11*, ISBSG, Hawthorn, Australia, June 2009.

²⁵ T.C. Jones, *Software Estimating Rules of Thumb*, Version 3, Software Productivity Research, 2007.

Toteż metodyka *southernSCOPE* ma na celu zmianę podejścia do wyceny produktu przedsięwzięcia: zamiast należności bazującej na stałej cenie lub należności za czas pracy projektantów, ewentualnie „wyceny dla wygranej”, zaproponowano w niej wycenę na podstawie kosztów jednostkowych liczonych w odniesieniu do jednostki rozmiaru produktu, tj. jednostki funkcjonalności (zwykle punktu funkcyjnego; ang. *function point*). Po kilku latach jej stosowania, w sektorze zarówno publicznym, jak i prywatnym, przeprowadzono niezależne badania mające na celu ocenę efektów wykorzystania metodyki²⁶. Okazało się, że każdy z przeanalizowanych projektów realizowanych przy jej zastosowaniu zakończył się powodzeniem, możliwa była jego kontrola ze strony zleceniodawcy, a dostarczone produkty odpowiadały celom i oczekiwaniom klientów. Z badań tych wynika także, że przy zastosowaniu wyceny w oparciu o rozmiar funkcjonalny produktu programowego średnie przekroczenie budżetu przedsięwzięcia waha się w granicach jedynie 10%. Stanowi to rezultat minimalizacji, dzięki temu podejściu, typowych zasadniczych przyczyn przekroczenia kosztów działań projektowych. Tak wysoka skuteczność metodyki spowodowała duże zainteresowanie nią także w niektórych krajach Europy, w USA i Japonii.

W roku 2007 organizacja FiSMA opublikowała zasady metodyki *northernSCOPE*, opracowanej na bazie podejścia *southernSCOPE* i bardzo do niego zbliżonej²⁷. Opiera się ona na nieco odmiennych krokach postępowania i ich innej kolejności.

Zasadnicze założenia obu metodyk to:

- cena, jaką musi zapłacić klient za system oprogramowania, zależy w sposób bezpośredni od rozmiaru funkcjonalnego danego systemu;
- oceny szacunkowe są wyrowadzane przez cały czas trwania przedsięwzięcia;
- struktura zarządzania zmianami sprzyja właściwemu zarządzaniu wprowadzaniem przez zleceniodawcę zmian do wymagań;
- osobą kluczową, odpowiedzialną za zarządzanie zakresem przedsięwzięcia jest tzw. menadżer zakresu (ang. *scope manager*).

Praktyka pokazuje, iż omawiane metodyki sprawdzają się dla przedsięwzięć mających na celu budowę lub doskonalenie dedykowanych systemów oprogramowania, niezależnie od tego, czy mają one charakter wewnętrzny czy zewnętrzny. Muszą być jednak spełnione warunki skutecznego zastosowania tych podejść, do których należy m.in.:

- zakończenie projektu w ramach zaplanowanego i kontrolowanego budżetu, które ma znaczenie kluczowe, jeśli nie priorytetowe dla zleceniodawcy;
- akceptacja dla metod wymiarowania rozmiaru funkcjonalnego produktu programowego.

²⁶ P.R. Hill, *Some practical uses of the ISBSG history data to improve Project Management*, ISBSG, Hawthorn, VIC 2007, s. 26–27, 30.

²⁷ Finnish Software Metrics Association, *northernSCOPE – customer-driven scope control for ICT projects*, FiSMA, 2007.

3. Model SoftFAM

3.1. Założenia dla modelu SoftFAM

Równoległe do powyższych metodyk autorka niniejszego artykułu zaproponowała i zweryfikowała w praktyce własny model tzw. oceny funkcjonalnej przedsięwzięć rozwoju systemów oprogramowania, o nazwie SoftFAM (ang. *SOFTware projects Functional Assessment Model*). Przez ocenę funkcjonalną przedsięwzięcia autorka rozumie jego ocenę *ex ante* i *ex post* przeprowadzaną w oparciu o koncepcję i metody FSM. Do atrybutów kluczowych takiej oceny (atrybutów funkcjonalnych) zalicza się: rozmiar funkcjonalny produktu (*RF*), nakłady pracy (*NP*), jakie trzeba wydatkować na jego rozwój (budowę/doskonalenie), oraz produktywność funkcjonalną przedsięwzięcia (*P*), rozumianą jako stosunek rozmiaru funkcjonalnego produktu do pracochłonności jego rozwoju (*RF/NP*), lub stanowiące jej odwrotność nakłady pracy konieczne do uzyskania jednostki funkcjonalności (*NP(j)*), które decydują o jednostkowych kosztach osobowych rozwoju produktu mierzonych w odniesieniu do jednostki rozmiaru funkcjonalnego produktu.

Model SoftFAM może mieć formę zarówno modelu pełnego, jak i jednego z uproszczonych wariantów – ma on więc charakter modułowy. Dla pełnego wariantu modelu ujęto w nim *explicite* następujące założenia, nieuwzględnione w opisanych w poprzednim punkcie metodykach:

1. Ocena funkcjonalna przebiega w trzech etapach:

- 1.1. Wstępnej oceny funkcjonalnej. Może mieć ona miejsce już na etapie inicjacji przedsięwzięcia, dzięki wykorzystaniu wyprowadzonej na podstawie danych historycznych reguły wczesnego oszacowania rozmiaru funkcjonalnego produktu (tzw. obliczenia *Function Points Zero* – FP0). Jednakże trafniejsze oceny szacunkowe uzyskuje się w stadium analizy, kiedy są znane zasadnicze *FUR* – opierają się one na obliczeniach *FP1* (*Function Points One*), dla których zgodnie z zasadami metod FSM przyjmuje się dopuszczalny błąd szacunku do $\pm 30\%$. Do wstępnego zaplanowania kluczowych atrybutów przedsięwzięcia, podjęcia wstępnej decyzji inwestycyjnej, wyboru wariantu realizacyjnego oraz wyboru grupy ofert dostawców z listy chętnych do rozwoju produktu powinno wystarczyć oszacowanie przeprowadzane właśnie w tym stadium. Dalsze prace analityczne pochłaniają znaczne środki, stanowiące według *ISBSG* nawet do ok. 25% nakładów pracy wydatkowanych w ciągu całego cyklu projektowego²⁸, stąd warto wykorzystać możliwość racjonalizacji tych działań i podjęcia decyzji już na tym etapie.

²⁸ International Software Benchmarking Standards Group, *The ISBSG Special Analysis Report: Planning Projects – Project Phase Ratios*, ISBSG, Hawthorn, VIC, March 2007, s. 1–3.

- 1.2. Szczegółowej oceny funkcjonalnej. Po raz drugi oszacowanie powinno się przeprowadzić w momencie, kiedy jest już znana szczegółowa specyfikacja FUR, a zatem po zakończeniu działań analitycznych. Oszacowania na tym etapie opierają się na obliczeniach FP2 (*Function Points Two*), których błąd szacunku zgodnie z zasadami metod FSM nie powinien przekraczać $\pm 10\%$. Należy zatem dokonać korekty wstępnie oszacowanego wymaganego *RF*, a na tej podstawie oczekiwanych nakładów pracy i wymaganej produktywności funkcjonalnej. Korekta ta wynika więc nie tylko ze zmiany FUR od momentu przeprowadzenia obliczeń FP1, ale także ze zmiany wielkości błędu dopuszczalnego na tym etapie dla *RF*, a w konsekwencji także dla szacowanych na jego podstawie atrybutów. Na bazie wyprowadzonych na tym etapie szacunków powinno się jeszcze raz dokonać oceny funkcjonalnej wyselekcjonowanej w poprzednim kroku grupy ofert, aby w efekcie wybrać na podstawie kryteriów takiej oceny najwyżej kilku potencjalnych dostawców produktu. Wybór jednego z nich może zależeć również od innych kryteriów – powinny one mieć charakter merytoryczny i dotyczyć przede wszystkim spełnienia pozafunkcyjnych wymagań zleceniodawcy. Istotne jest, aby wymagany rozmiar funkcjonalny produktu oraz oferowany i zaakceptowany koszt osobowy jednostki funkcjonalności znalazły wyraz w formalnym zobowiązaniu wykonawcy wobec odbiorcy, co oznacza formalną wycenę *ex ante* produktu przedsięwzięcia.
- 1.3. Końcowej oceny funkcjonalnej. Trzeci raz ocenę funkcjonalną powinno się przeprowadzić po zakończeniu działań projektowych w celu pomiaru faktycznie dostarczonego *RF*, co ma prowadzić przede wszystkim do wyceny produktu *ex post* na podstawie tego rozmiaru oraz ustalonego kosztu osobowego jednostki funkcjonalności, a także służyć do weryfikacji stopnia zrealizowania FUR przez dostawcę, który dzięki temu zyskuje możliwość doskonalenia swoich procesów wytwórczych. Uzyskane w ten sposób dane powinny być przez niego przechowywane w organizacyjnym, specjalnie do tego celu przeznaczonym repozytorium danych historycznych. Ma to służyć wyprowadzeniu i weryfikacji specyficznych dla danej organizacji projektującej zależności, ale także doskonaleniu metod FSM i modeli szacowania pracochłonności. Obliczenia na tym etapie powinny uwzględniać fakt, iż od momentu przeprowadzenia obliczeń FP2 wymagania użytkownika mogły ulec zmianie. Należy zatem zaktualizować wartość wszystkich wymaganych atrybutów.
2. Wszystkie wymagane (*RF_w*, *NP_w*, *P_w*), oferowane (*RF_o*, *NP_o*, *P_o*) oraz zrealizowane (*RF_z*, *NP_z*, *P_z*) atrybuty funkcjonalne powinny zawierać się w odpowiednich, zależnych od etapu oceny funkcjonalnej przedziałach tolerancji, normujących zakresy dopuszczalnych dla nich wartości. Potrzeba ich uwzględnienia wynika zarówno z ograniczonych możliwości wyprowadzenia trafnych ocen szacunkowych, zwłaszcza na wstępnym etapie oceny, wynikających z kolei przede wszystkim ze zmienności

w czasie uwarunkowań realizacyjnych przedsięwzięcia, jak i z potrzeb analitycznych. Przedziały tolerancji powinny sprzyjać racjonalnemu wyznaczaniu oczekiwanych i oferowanych wartości atrybutów funkcjonalnych. Są one następujące:

- 2.1. Rozmiar funkcjonalny produktu przedsięwzięcia – zarówno wymagany przez zleceniodawcę (RF_w), jak i oferowany (RF_o) oraz zrealizowany (RF_z) przez wykonawcę musi zawierać się w dopuszczalnym dla RF_w przedziale, tj. $[RF_{min}, RF_{max}]$, gdzie: RF_{min} – minimalny, a RF_{max} – maksymalny wymagany rozmiar funkcjonalny produktu. Zdefiniowanie RF_{max} wynika z powszechnie spotykanej w praktyce nadmiarowości wymagań zleceniodawcy wobec produktu: jak pokazują badania Standish Group, tylko ok. 20% wyspecyfikowanych przez zleceniodawcę funkcji i cech wykorzystuje się zawsze w zaimplementowanym systemie oprogramowania, a ok. 45% z nich nie wykorzystuje się nigdy (sic!)²⁹. Tym samym wyznaczenie maksymalnego oczekiwanego RF produktu zmniejsza ryzyko dostarczenia w produkcie zbędnej funkcjonalności.
- 2.2. Pracochłonność przedsięwzięcia – zarówno wymagana przez zleceniodawcę (NP_w), jak i oferowana (NP_o) oraz zrealizowana (NP_z) przez wykonawcę musi zawierać się w dopuszczalnym dla NP_w przedziale, tj. $[NP_{min}, NP_{max}]$, gdzie: NP_{min} – minimalne, a NP_{max} – maksymalne oczekiwane przez zleceniodawcę nakłady pracy. NP_{min} nie powinny być mniejsze od wynikających z danych historycznych nakładów pracy umożliwiających dostarczenie minimalnego wymaganego rozmiaru funkcjonalnego produktu (RF_{min}).
- 2.3. Produktywność funkcjonalna przedsięwzięcia – zarówno wymagana przez zleceniodawcę (P_w), jak i oferowana (P_o) oraz zrealizowana (P_z) przez wykonawcę musi zawierać się w dopuszczalnym dla P_w przedziale, tj. $[P_{min}, P_{max}]$, gdzie: P_{min} – minimalna, a P_{max} – maksymalna produktywność oczekiwana przez zleceniodawcę. Zdefiniowanie P_{max} jest przydatne do racjonalnego wyboru oferty wykonawcy, tj. ze względu na ograniczenie ryzyka wyboru tej oferty, w której oferowana produktywność byłaby określona jako wartość mało realna do osiągnięcia (zawyżona) dla systemów o wyznaczonym RF_w oraz określonego rodzaju, decydującego o stopniu jego skomplikowania. Sytuacja taka oznaczałaby bowiem, że w rzeczywistości pracochłonność jednostki funkcjonalności zostanie prawdopodobnie przekroczona, co wiązałoby się z ryzykiem dostarczenia produktu o mniejszym niż dopuszczalny rozmiarze funkcjonalnym, jako że wykonawca dążyłby zapewne do tego, aby nie przekroczyć oferowanych nakładów pracy. Ponadto jej wyznaczenie sprzyja zwiększeniu prawdopodobieństwa dostarczenia produktu o wystarczającej jakości.

Spełnienie tych warunków zapewnia:

- racjonalność wymagań zleceniodawcy co do atrybutów funkcjonalnych,

²⁹ Standish Group, *The CHAOS Manifesto*, West Yarmouth, Massachusetts 2009, s. 1.

- zgodność ofert potencjalnych dostawców z racjonalnymi wymaganiami zleceniodawcy co do atrybutów funkcjonalnych,
- zgodność zrealizowanego przedsięwzięcia z wymaganiami zleceniodawcy co do atrybutów funkcjonalnych.

3.2. Weryfikacja modelu SoftFAM

Ze względu na ograniczoną objętość artykułu zostaną w nim przedstawione jedynie wnioski wynikające z weryfikacji modelu SoftFAM³⁰. Dotyczą one:

1. Ograniczania pewnych negatywnych praktyk powszechnie występujących przy realizacji przedsięwzięć rozwoju systemów oprogramowania dzięki zasadom zaimplementowanym w modelu. Zarówno te praktyki, jak i ograniczające je zasady przedstawiono w tabeli 1.
2. Przewagi modelu SoftFAM nad metodykami *southernSCOPE* i *northernSCOPE* – wynika ona z zastosowania w modelu SoftFAM: (a) dopuszczalnych przedziałów tolerancji dla atrybutów funkcjonalnych; (b) dwóch etapów oceny szacunkowej: pierwszego w celu umożliwienia zleceniodawcy podjęcia racjonalnej decyzji inwestycyjnej jak najwcześniej w cyklu życia przedsięwzięcia, a drugiego w celu właściwego wyboru wykonawcy przedsięwzięcia. Co więcej, zaproponowany model ma charakter modułowy – oprócz wariantu pełnego istnieje także pięć jego uproszczonych wariantów, a dwa najprostsze są najbardziej zbliżone do metodyk *southernSCOPE* i *northernSCOPE*.
3. Możliwości oferowanych przez ocenę funkcjonalną, które nie są zaspakajane przez ocenę efektywności ekonomicznej przedsięwzięcia, i odwrotnie. Ocena efektywności ekonomicznej nie pozwala bowiem na ocenę skuteczności przedsięwzięcia, nie przyczynia się także do ograniczenia przedstawionych w tabeli 1 negatywnych praktyk powszechnie występujących w realizacji rozważanych przedsięwzięć, zmniejszając w ten sposób szansę na uzyskanie w rzeczywistości planowanej efektywności. Ocena funkcjonalna natomiast nie pozwala na bezwarunkowe stwierdzenie, **czy** dany wariant realizacyjny jest efektywny ekonomicznie, a jedynie **kiedy** będzie on efektywny ekonomicznie. Dlatego obie oceny powinny się uzupełniać, co zapewni spełnienie dwóch wymiernych kryteriów racjonalności decyzji inwestycyjnej: efektywności i skuteczności³¹.

³⁰ Więcej informacji na ten temat wraz z samym procesem weryfikacji zob.: B. Czarnacka-Chrobot, *Rational Pricing of Business Software Systems on the basis of Functional Size Measurement: a Case Study from Poland*, Proceedings of the 7th Software Measurement European Forum (SMEF 2010), red. T. Dekkers, Libreria Clup, Rome, Italy, June 2010, s. 43–58.

³¹ Więcej informacji na ten temat zob.: B. Czarnacka-Chrobot, *Evaluation of Business Software Systems Development and Enhancement Projects Effectiveness and Economic Efficiency on the basis of Functional Size Measurement*, Proceedings of the 10th International Conference on Software Engineering Research and Practice (SERP'11),

Tabela 1. Negatywne praktyki w realizacji przedsięwzięć rozwoju systemów oprogramowania i zasady sprzyjające ich ograniczaniu

Lp.	Negatywne praktyki w realizacji przedsięwzięć rozwoju systemów oprogramowania	Zasady modelu SoftFAM ograniczające negatywne praktyki
1.	Nieracjonalne wyznaczanie wymagań dla atrybutów funkcjonalnych przez zleceniodawcę – zleceniodawca nie wie, ile powinna kosztować wymagana przez niego funkcjonalność, co sprzyja podejmowaniu niewłaściwych decyzji inwestycyjnych.	Przyjęcie odpowiednich zależności między atrybutami funkcjonalnymi oraz dopuszczalnych przedziałów tolerancji dla tych atrybutów.
2.	Nieuwzględnianie wpływu zmian w wymaganiach funkcjonalnych zleceniodawcy (zwykle ich zwiększania) wprowadzanych w trakcie cyklu projektowego na koszty przedsięwzięcia.	Monitorowanie wpływu każdej zmiany w wymaganiach funkcjonalnych zleceniodawcy na rozmiar funkcjonalny produktu, a przez to na koszty realizacji przedsięwzięcia, co sprzyja wprowadzaniu zmian tylko faktycznie niezbędnych, a to ma istotne znaczenie w kontekście wspomnianych powyżej badań Standish Group, z których wynika niemal pięćdziesięcioprocentowa nadmiarowość wymagań zleceniodawcy wobec produktu przedsięwzięcia.
3.	Świadome zaniżanie przez oferentów kosztów realizacji przedsięwzięcia w celu wygrania kontraktu („wycena dla wygranej”).	Preferowanie przy wyborze ofert na realizację przedsięwzięcia największej dopuszczalnej produktywności funkcjonalnej (najmniejszych kosztów osobowych jednostki funkcjonalności) zamiast ofert najtańszych.
4.	Dostarczanie przez wykonawcę produktu o mniejszej funkcjonalności niż wymagana w ramach kontraktów o stałej cenie.	Wycena <i>ex ante</i> i <i>ex post</i> produktu przedsięwzięcia w oparciu o wymagany (wycena <i>ex ante</i>) i faktycznie dostarczony (wycena <i>ex post</i>) jego rozmiar funkcjonalny oraz obustronnie formalnie uzgodniony na etapie wyboru dostawcy koszt osobowy jednostki funkcjonalności (zasadnicze kryterium wyboru wykonawcy przedsięwzięcia) – dzięki temu zleceniodawca nie jest zobowiązany regulować należności za funkcjonalność wymaganą, która nie została dostarczona, nie reguluje jej także na podstawie działań/zasobów, w tym czasu pracy projektantów. Z tego i powyższego punktu (lp. 3) wynika przewaga modelu SoftFAM nad powszechnymi w praktyce kontraktami między zleceniodawcą a wykonawcą.
5.	Realizacja przez wykonawcę funkcjonalności (nierazko także mniejszej niż wymagana) po większych kosztach całkowitych niż przewidywane w ramach kontraktów czasowo-materiałowych, sprzyjających – często celowemu – wydłużaniu cyklu projektowego.	

Źródło: opracowanie własne.

The 2011 World Congress in Computer Science, Computer Engineering & Applied Computing (WORLDCOMP'11), vol. 2, red. H.R. Arabnia, H. Reza, L. Deligiannidis, CSREA Press, Las Vegas, Nevada, USA, July 2011, s. 401–407.

Reasumując, weryfikacja modelu SoftFAM wykazała, że jego wykorzystanie ogranicza ryzyko:

- realizacji nieskutecznego i nieefektywnego przedsięwzięcia,
- wyboru niewłaściwego wykonawcy przedsięwzięcia,
- nadmiarowości wymagań zleceniodawcy wobec produktu,
- zaniżonej wyceny *ex ante* produktu,
- zawyżonej wyceny *ex post* produktu.

Ocena funkcjonalna stanowi zatem niewątpliwie właściwą, chociaż niekoniecznie jedyną, podstawę do podejmowania racjonalnych decyzji inwestycyjnych przez zleceniodawców następujących kategorii przedsięwzięć:

- polegających na konstrukcji systemów oprogramowania od podstaw,
- polegających na rozbudowie istniejącego systemu oprogramowania,
- polegających na nabywaniu gotowego systemu oprogramowania i jego dostosowaniu do specyficznych potrzeb organizacji,
- mających na celu utrzymanie funkcjonującego systemu oprogramowania w stanie gotowości do realizacji niezbędnych funkcji.

4. Wnioski

Celem niniejszego artykułu było przybliżenie podejścia do wyceny zwykle dedykowanych systemów oprogramowania realizowanych na potrzeby administracji publicznej, które zyskuje na świecie w ostatnich latach coraz większą popularność, a na tym tle przybliżenie zweryfikowanego w praktyce autorskiego modelu SoftFAM, służącego wycenie systemów oprogramowania w oparciu o takie podejście.

Rozważania zawarte w artykule prowadzą do następujących wniosków:

1. Spotykane najczęściej w praktyce podejścia do wyceny systemów oprogramowania sprzyjają przekroczeniom wydatków zaplanowanych na dostarczenie produktu spełniającego wymagania zleceniodawcy. Dlatego w obszarze tym, a zwłaszcza w przypadku dedykowanych systemów oprogramowania realizowanych na potrzeby administracji publicznej, należy poszukiwać innych rozwiązań.
2. Podstawę wyceny, zarówno *ex ante*, jak i *ex post*, systemów oprogramowania powinny stanowić nie oferowane przez potencjalnych dostawców najniższe koszty całkowite, często celowo zaniżane („wycena dla wygranej”) bądź szacowane w sposób nieobiektywny i/lub niewiarygodny (na podstawie działań/zasobów), ale najniższe koszty jednostkowe, które jednak powinny być mierzone nie w odniesieniu do czasu pracy projektantów, a w odniesieniu do jednostki rozmiaru produktu. Rozmiar produktu zatem także musi być w takiej wycenie uwzględniany: przy wycenie *ex ante* powinien być to rozmiar wymagany (oszacowany), natomiast przy wycenie *ex post* – faktycznie zrealizowany (zmierzony). Takie podejście do

- wyceny wymaga właściwej miary (jednostki) rozmiaru oprogramowania. Jednakże dopiero wówczas wycena produktu będzie miała obiektywny i wiarygodny charakter. Zleceniodawca zyska bowiem możliwość zaplanowania kosztów realizacji przedsięwzięcia w zależności od oczekiwanego rezultatu, a w wyniku jego realizacji zapłaci za rzeczywiście dostarczony rozmiar produktu, a nie za swoje niezrealizowane przez zleceniobiorcę wymagania.
3. Badania wykazują, że najwłaściwszą jednostką rozmiaru systemu oprogramowania, zwłaszcza w kontekście jego wyceny, stanowi jednostka funkcjonalności. Zostało to potwierdzone m.in. przez standaryzację – na podstawie gruntownej weryfikacji – koncepcji i niektórych metod tzw. wymiarowania rozmiaru funkcjonalnego oprogramowania przez międzynarodowe organizacje standaryzacyjne ISO i IEC. Jednostka funkcjonalności stanowi jak dotąd jedyną zaakceptowaną w ten sposób miarę wielkości produktu programowego. W związku z tym jest ona coraz częściej wykorzystywana na świecie do wyceny systemów oprogramowania realizowanych na potrzeby administracji publicznej – jako co najmniej rekomendacja, jeśli nie wymóg formalny.
 4. Rozmiar funkcjonalny systemu oprogramowania stanowi podstawę metodyk *southernSCOPE* i *northernSCOPE*, służących do zarządzania tzw. zakresem funkcjonalnym przedsięwzięcia, w tym m.in. do wyceny ich produktów. Metodyki te nie uwzględniają jednak dwóch istotnych założeń, które przyjęto w zaproponowanym przez autorkę modelu SoftFAM: (1) potrzeby zastosowania górnych dopuszczalnych przedziałów tolerancji dla wymaganego, oferowanego i zrealizowanego rozmiaru funkcjonalnego produktu i produktywności funkcjonalnej oraz dolnego dla nakładów pracy; (2) potrzeby zastosowania co najmniej dwóch etapów oceny szacunkowej – pierwszego w celu właściwej oceny racjonalności decyzji inwestycyjnej, natomiast drugiego w celu trafnego wyboru wykonawcy produktu. W zestawieniu z tymi metodykami zastosowanie modelu SoftFAM ogranicza zatem ryzyko wyboru niewłaściwego dostawcy oraz ryzyko zaniżonej wyceny produktu, a w konsekwencji ogranicza możliwość niedostarczenia wymaganej funkcjonalności i/lub dostarczenia produktu o niewystarczającej jakości. Modułowy charakter modelu SoftFAM daje natomiast możliwość wyboru jego najodpowiedniejszej w danej sytuacji wersji – może to być wersja oparta na najprostszych kryteriach, najbardziej zbliżona do metodyk *southernSCOPE* i *northernSCOPE*.
 5. Rezultaty weryfikacji zaproponowanego modelu SoftFAM dowodzą, że pozwala on na racjonalizację określonych działań praktycznych oraz podejmowanych na podstawie jego kryteriów decyzji biznesowych. Do działań praktycznych należą: wyznaczanie racjonalnych wymagań zleceniodawcy dotyczących atrybutów funkcjonalnych, ocena pod ich kątem ofert potencjalnych dostawców, porównanie wariantów realizacyjnych z perspektywy oszacowanych kosztów osobowych, racjonalna wycena *ex ante* i *ex post* produktu przedsięwzięcia oraz doskonalenie

przez jego wykonawcę prognoz dotyczących przyszłych przedsięwzięć. Wśród wspomaganych przez nią decyzji biznesowych należy wymienić: decyzję inwestycyjną zleceniodawcy o zaangażowaniu w realizację przedsięwzięcia o oczekiwanych atrybutach funkcjonalnych, wybór oferty najlepiej dopasowanej do jego wymagań dotyczących atrybutów funkcjonalnych, a także decyzję o oferowanym przez potencjalnych dostawców koszcie osobowym jednostki funkcjonalności. Co więcej, rezultaty weryfikacji modelu SoftFAM wskazują także, iż formalna wycena systemu oprogramowania powinna się opierać na rozmiarze wymagany (wycena *ex ante*) i rzeczywiście zrealizowanym (wycena *ex post*) tego produktu wyrażonym za pomocą odpowiednich jednostek i na jednostkowych kosztach osobowych mierzonych w stosunku do jednostki rozmiaru produktu. Dzięki wymienionym możliwościom model SoftFAM sprzyja ograniczeniu niektórych negatywnych zjawisk powszechnie występujących w praktyce realizacji takich przedsięwzięć, niekorzystnie wpływających na skuteczność ich realizacji.

Literatura

1. Czarnacka-Chrobot B., *Analysis of the Functional Size Measurement Methods Usage by Polish Business Software Systems Providers*, „Lecture Notes in Computer Science” 2009, vol. 5891, *Software Process and Product Measurement*, red. A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, J. Brunekreef, Springer-Verlag, Berlin-Heidelberg, s. 17–34.
2. Czarnacka-Chrobot B., *Evaluation of Business Software Systems Development and Enhancement Projects Effectiveness and Economic Efficiency on the basis of Functional Size Measurement*, Proceedings of the 10th International Conference on Software Engineering Research and Practice (SERP'11), *The 2011 World Congress in Computer Science, Computer Engineering & Applied Computing (WORLDCOMP'11)*, vol. 2, red. H.R. Arabnia, H. Reza, L. Deligiannidis, CSREA Press, Las Vegas, Nevada, USA, July 2011, s. 401–407.
3. Czarnacka-Chrobot B., *Rational Pricing of Business Software Systems on the basis of Functional Size Measurement: a Case Study from Poland*, Proceedings of the 7th Software Measurement European Forum (SMEF 2010), red. T. Dekkers, Libreria Clup, Rome, Italy, June 2010, s. 43–58.
4. Czarnacka-Chrobot B., *The Economic Importance of Business Software Systems Size Measurement*, Proceedings of the 5th International Multi-Conference on Computing in the Global Information Technology (ICCGI 2010), 20–25 September 2010, Valencia, Spain, red. Miguel Garcia, Jean-Denis Mathias, IEEE Computer Society Conference Publishing Services, Los Alamitos, California-Washington-Tokyo 2010, s. 293–299.

5. Czarnacka-Chrobot B., *Wiarygodność metod szacowania pracochłonności przedsięwzięć rozwoju systemów oprogramowania wspomagających zarządzanie*, „Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu” 2010, nr 88, *Informatyka Ekonomiczna. Informatyka w zarządzaniu*, red. J. Sobieska-Karpińska, s. 163–176.
6. Finnish Software Metrics Association, *nothernSCOPE – customer-driven scope control for ICT projects*, FiSMA, 2007.
7. Hill P.R., *Some practical uses of the ISBSG history data to improve Project Management*, ISBSG, Hawthorn, VIC 2007.
8. International Software Benchmarking Standards Group, *Data demographics – release 11*, ISBSG, Hawthorn, Australia, June 2009.
9. International Software Benchmarking Standards Group, *The ISBSG Report: Software Project Estimates – How accurate are they?*, ISBSG, Hawthorn, VIC, January 2005.
10. International Software Benchmarking Standards Group, *The ISBSG Special Analysis Report: Planning Projects – Project Phase Ratios*, ISBSG, Hawthorn, VIC, March 2007.
11. *ISO/IEC 14143 Information Technology – Software measurement – Functional size measurement – Parts 1–6*, ISO, Geneva 2011.
12. *ISO/IEC 19761 Software engineering – COSMIC: a functional size measurement method*, edition 2, ISO, Geneva 2011.
13. *ISO/IEC 20926 Software and systems engineering – Software measurement – IFPUG functional size measurement method 2009*, ISO, Geneva 2009.
14. *ISO/IEC 20968 Software engineering – Mk II Function Point Analysis – Counting practices manual*, ISO, Geneva 2002.
15. *ISO/IEC 24570 Software engineering – NESMA functional size measurement method version 2.1 – Definitions and counting guidelines for the application of Function Point Analysis*, ISO, Geneva 2005.
16. *ISO/IEC 29881 Information Technology – Software and systems engineering – FiSMA 1.1 functional size measurement method*, ISO, Geneva 2010.
17. Jones T.C., *Software Estimating Rules of Thumb*, Version 3, Software Productivity Research, 2007.
18. Jones T.C., *Software Project Management in the Twenty-first Century*, Software Productivity Research, Burlington 1999.
19. Rule G., *The most common Functional Size Measurement (FSM) methods compared*, Software Measurement Services, St. Clare’s, Mill Hill, Edenbrige, Kent, UK 2010.
20. Sage Technology, *Report on the SCUD Methodology Review*, June 2000.
21. Standish Group, *The CHAOS Manifesto*, West Yarmouth, Massachusetts 2009.
22. Standish Group, *CHAOS Summary 2009*, West Yarmouth, Massachusetts 2009.
23. State Government of Victoria, *southernSCOPE, Reference Manual*, Version 1, Government of Victoria, Melbourne, September 2000.

Summary

Pricing of the software systems executed for the needs of public administration

Software systems executed for the needs of public administration often hold a character of dedicated, individual solutions that entail particular problems with regard to their pricing. Thus for many years now objective and reliable approaches to the *ex ante* assessment of the costs of such systems have been sought out so that they could provide the possibility to make rational investment decisions concerning those systems. The purpose of this paper is to introduce approach to the pricing of dedicated software systems that in the area of public administration has been recently growing in global popularity and to introduce author's own practically verified model of pricing based on that approach.