Przemysław Plecka
Krzysztof Bzdyra
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
Ul. Śniadeckich 2, 75-453 Koszalin
tel.: +48602336363,

Wybór metod szacowania kosztów modyfikacji na wstępnych etapach cyklu życia oprogramowania ERP

Keywords: ERP, implementation, cost estimation

Wstęp

W chwili obecnej każdy z liczących się na rynku producentów posiada w swoim portfelu produkt standardowy klasy ERP: SAP - Business Suite, Microsoft - Dynamix AX, JD Edwards - EnterpriseOne, itp. Podczas rozmów handlowych w trakcie sprzedaży systemów strony dochodzą jednakże do wniosku, że organizacja procesów w przedsiębiorstwie nie pokrywa się całkowicie z procesami realizowanymi przez oferowany system informatyczny [1]. Istnieje grupa procesów, której nie odpowiada żadna funkcjonalność w standardowym systemie ERP. Na tym etapie pojawia się potrzeba przystosowania systemu informatycznego (skr. SI) do przedsiębiorstwa. Koszty modyfikacji systemu podnoszą wartość kontraktu. W niektórych przypadkach alternatywą jest przystosowanie przedsiębiorstwa do systemu informatycznego, jednakże koszty zmiany organizacji obciążą wówczas klienta. Dopiero gdy klient pozna koszt, jaki będzie musiał ponieść na wdrożenie systemu (w tym przystosowanie), skłonny jest do rozważenia zmian w swojej organizacji. Aby dostawcy mieli podstawy do negocjacji, istotnym jest szacowanie kosztów na jak najwcześniejszym etapie.

Prowadząc proces wyliczania kosztów modyfikacji dostawcy napotykają na trudności w doborze odpowiedniej metody. Zwykle na początkowym etapie wybierają jedną, najlepiej im znaną metodę i stosują ją przez cały okres wycen. Taka praktyka doprowadza do większych błędów szacowań [2]. Pomocne dla dostawców byłoby narzędzie sugerujące metodę za pomocą, której uzyskane zostanie najdokładniejsze szacowania. Po każdym etapie procesu sprzedaży dostawca mógłby zweryfikować dane jakie zebrał i uzyskać sugestie jakie metody szacowania zastosować. Wiadome są więc etapy cyklu życia oprogramowania [3] oraz metody szacowania oprogramowania. Poszukiwana jest odpowiedź na pytanie, jaka jest procedura wyboru metod szacowania oprogramowania na danym etapie fazy strategicznej procesu wdrożenia SI. Zakres problemu ograniczony

został do systemów informatycznych klasy ERP wdrażanych w średnich przedsiębiorstwach produkcyjnych.

Metody pomagające wycenić koszty wykonania oprogramowania są znane i opisane w literaturze, np. przez McConella [4]. Jednakże z powodu zmian technologii informatycznych powszechność tych metod ciągle ulega zmianie. Zastosowanie algorytmicznych metod w początkowych etapach projektów informatycznych jest trudne. Nie istnieją wówczas jeszcze dokumenty projektowe zawierające dane potrzebne algorytmom estymującym. Mimo że przykłady zastosowania metod algorytmicznych we wczesnych etapach projektów informatycznych można znaleźć w literaturze[5] [6], to praktyka dostawców systemów informatycznych pokazuje stosowanie metod niealgorytmicznych jako szybszych (tzn. tańszych) i łatwiejszych do realizacji. W literaturze możemy znaleźć różne przykłady sugestii zastosowania metod szacowania kosztów dla projektów informatycznych, począwszy od stwierdzeń, że należy stosować dowolne kombinacje technik, poprzez zalecenia, kiedy i jakie metody należy stosować, skończywszy na przepisach „krok po kroku"[7].

Dostawcy systemów ERP negocjują z klientami dwie wielkości: koszty i czas realizacji. Konsultanci szacujący oprogramowania posługują się takimi miarami jak roboczogodzina, roboczodzień czy roboczomiesiąc. Znając szacunek wielkości kosztów w jednostkach czasu pracy potrzebnego na realizacje, można przeliczyć go na wartość kosztu w walucie oraz na czas (terminy) realizacji, uwzględniając możliwość równoczesnej realizacji niektórych prac.

Niniejszy artykuł zawiera w rozdziale 1 opis fazy strategicznej procesu wdrożenia z uwzględnieniem jakości dostępnych informacji do wyceny. Kolejny rozdział jest przeglądem algorytmicznych metod szacowania. Rozdział 3 zawiera opisy metod niealgorytmicznych. Oba rozdziały zwracają uwagę na rodzaj i jakość danych potrzebnych do szacowania. W rozdziale ostatnim zaprezentowano wnioski wynikające z połączenia efektów poszczególnych etapów cyklu życia i danych potrzebnych do wycen. Na tej podstawie zaproponowano algorytm wykorzystujący ankietę do wyboru metody szacowania na poszczególnych etapach fazy strategicznej wdrożenia.

Znaczenie symboli umieszczonych na rysunkach 1,2,4 oraz w Dodatku B jest zgodne z notacją BPMN 2.0 [8].

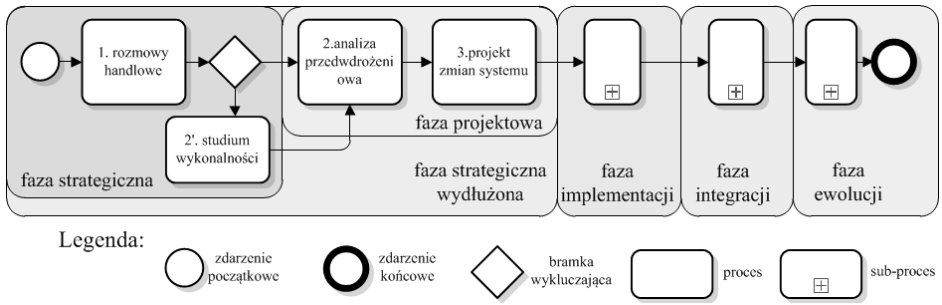
1. Etapy cyklu życia wdrażanego systemu ERP

Liczni autorzy opisują cykle życia oprogramowania, skupiając się na procesie produkcji oprogramowania lub tworzenia oprogramowania na zamówienie [3]. Wśród prezentowanych tam modeli żaden nie odpowiada w pełni procesowi wdrażania oprogramowania klasy ERP w średnim przedsiębiorstwie. Nie uwzględniają one „ruchomości" zakończenia fazy strategicznej (podpisania kontraktu) oraz możliwości wystąpienia dodatkowego etapu - studium wykonalności. Studium wykonalności nie jest istotne dla cyklu życia oprogramowania, ale dostarcza informacji do wyceny projektu.

Etapy fazy strategicznej są następujące:

1. wstępne rozmowy handlowe,
2. analiza przedwdrożeniowa
2'. studium wykonalności,
3. projekt zmian systemu,

Etapy fazy strategicznej oraz pozostałe fazy cyklu życia oprogramowania (implementacje, integrację, ewolucję) przedstawia Rysunek 1.



Rysunek 1. Etapy fazy strategicznej i pozostałe fazy wdrażania systemu ERP

Mając na uwadze szacowanie kosztów, należy pamiętać, że istotny w procesie sprzedaży jest moment podpisania kontraktu. Może to nastąpić po etapie 1., ale nie później niż po zakończeniu etapu 3. Okres ten nazywa się fazą strategiczną. W interesie dostawcy SI jest podpisanie umowy z klientem jak najszybciej, gdyż realizacja kolejnych etapów obciąża go kosztami, a ciągle istnieje ryzyko niepodpisania kontraktu. Jednakże wcześniejsze określenie kosztów w warunkach większej niepewności wiąże się z ryzykiem większego błędu szacowania.

1.1. Wstępne rozmowy handlowe

Dostawca odbywa spotkania z potencjalnym klientem w celu ustalenia zakresu i wartości kontraktu. Zwykle jest to jedno spotkanie wstępne i dwa lub trzy spotkania prezentacyjne. Niektóre z elementów zakresu prac zostają ujawnione szybko i szczegółowo. Dotyczą one przede wszystkim sprzętu komputerowego, wykonania infrastruktury sieciowej, licencji na poszczególne moduły systemu ERP. Niektóre elementy zakresu, np. modyfikacje SI wynikające ze specyficznych wymagań użytkowników, są trudne do określenia. Na tym etapie dostawca w niewielkim zakresie potrafi zidentyfikować wymagania klienta, których nie realizuje wersja standardowa sytemu ERP. Wiedza klienta o SI pochodzi na ogół z prezentacji handlowych, przez co nie zawsze potrafi on precyzyjnie określić te, które nie są standardowe. Wymagania, które dostawca jest w stanie pozyskać od klienta są zwykle niekompletne (brak jest wymagań, które klient uważał za nieistotne) i ogólne (klient nie jest w stanie ocenić istotnego

poziomu szczegółowości).

Jeśli dostawca potrafi wyspecyfikować ujawnione wymagania oraz zasugerować wymagania nieujawnione, może próbować wyceniać zmiany. Na przykład, klient określił wymaganie w obszarze produkcja, dotyczące zamawiania oddzielnie do każdego zlecenia produkcyjnego, surowców z grupy towarowej A. Wymaganie takie sugeruje nieujawnione wymagania dotyczące zmiany procesu tworzenia zamówień towarów w obszarze logistyka, gdzie z ogólnego planów zamówień dostaw wyłączyć należy zarządzanie surowcami z grupy A. Oba wymagania powinny być wycenione, mimo, ze ujawniono przez klienta tylko pierwsze z nich. Na tym etapie sporadycznie pojawiają się pojedyncze, szczegółowe wymagania typu: raporty różnie agregujące te same dane, wydruki w specyficznej formie używanej przez klienta.

1.2. Studium wykonalności lub analiza przedwdrożeniowa

Jeśli dostawca nie był w stanie wycenić dostosowania systemu (modyfikacje), musi przeprowadzić prace, dzięki którym ujawnione i uszczegółowione będą wymagania klienta. Realizuje wówczas studium wykonalności[9] lub analizę przedwdrożeniową. Mimo że oba rozwiązania mają doprowadzić do uszczegółowienia danych do wyceny, to cel utworzenia każdego z nich jest inny.

Studium wykonalności zawiera zestawienie zebranych informacji w postaci uporządkowanego dokumentu opartego na faktach gospodarczych[10]. Informacje dotyczą sfery ekonomicznej, organizacyjnej, technicznej. Celem studium jest określenie zakresu prac oraz kosztów realizacji przedsięwzięcia. Z dokumentu korzystają decydenci dostawcy, rozpatrując ekonomiczne aspekty realizacji projektu.

Analiza przedwdrożeniowa nie zawiera innych informacji niż te, które dotyczą systemu informatycznego w kontekście danego przedsiębiorstwa i realizacji prac. Wynikiem jest raport zawierający następujące elementy: zakres funkcjonalny wdrożenia, wykaz i opis procesów biznesowych, funkcji i danych zalecanych do uwzględnienia w zakresie funkcjonalnym wdrażanego systemu, zakres organizacyjny wdrożenia, proponowane cele wdrożenia, oczekiwane korzyści biznesowe, harmonogram prac [11]. Na tym etapie dostawca zakłada, że wymagania są kompletne i poziom szczegółowości odpowiada wymaganiom projektantów, dla których ten dokument jest podstawą dalszych prac.

Nawet w średnim przedsiębiorstwie produkcyjnym ewidencja wszystkich wymagań użytkowników i wszystkich procesów byłaby bardzo czasochłonna i kosztowna (od kilkuset do ponad tysiąca wymagań i procesów). Ponadto w większości pokrywałaby się z zapisami w dokumentacji systemu ERP. Dlatego też dostawcy wykonują analizę różnicową, która zawiera tylko te wymagania i opisy procesów, które nie są realizowane przez wersję standardową SI. Takie postępowanie znacznie skraca czas realizacji etapu, ale powoduje, że klient, żeby mieć obraz całego systemu, musi poznać dokumentację wersji standardowej łącznie z dokumentacją analizy przedwdrożeniowej.

Dostawca powinien zweryfikować jakość wymagań jakie zostały ujawnione na tym etapie pod kątem wykorzystania metod szacowania.

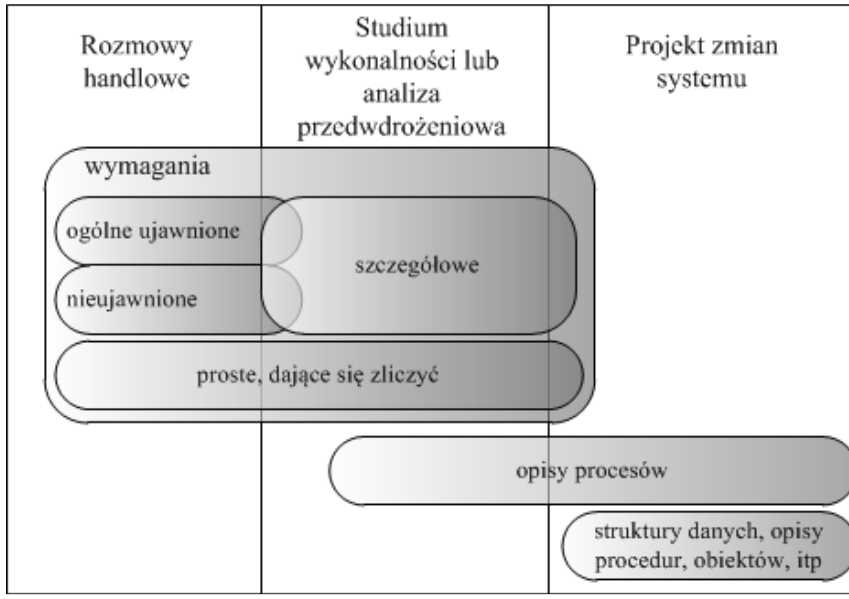
1.3. Projekt zmian systemu

Projekt systemu informatycznego to faza pośrednia między określeniem wymagań a realizacją. Dokumentacja, jaka powstaje, jest przeznaczona wyłącznie na użytek wewnętrzny dostawcy (działy programowania). W zależności od tego, jakimi metodami odbywać się będzie realizacja (programowanie strukturalne, obiektowe, zwinne, itp.), dokumenty projektowe zawierać mogą inne składowe [3]. Niektórzy producenci systemów ERP opracowali własne metodyki i w takim wypadku dokumentacja fazy projektowania będzie specyficzna. Przykładem może być metodyka Select Perspective [12] [13] lub ARIS [14]. Zawsze jednak występować będą wspólne elementy istotne dla procesów szacowania oprogramowania.

Pierwszym elementem prac projektowych jest uściślenie wymagań wynikających ze specyfiki realizacji. Poziom szczegółowości wymagań musi jednoznacznie determinować sposób ich realizacji. Poza tym dokumenty projektowe zawierają składowe opisujące struktury danych i procedury realizujące procesy. Istnieje wiele sposobów prezentacji informacji projektowej: od DFD, diagramy relacji encji, poprzez perspektywy modeli UML. Każdy z nich jest odpowiednim źródłem danych do szacowania realizacji oprogramowania.

1.4. Podsumowanie etapów cyklu życia

Wraz z kolejnymi etapami cyklu życia oprogramowania rośnie wiedza dostawcy o różnicach między procesami realizowanym w przedsiębiorstwie a funkcjonalnością oprogramowania standardowego. Na początku dysponuje on jedynie niekompletnym zbiorem wymagań ogólnych. W kolejnych etapach uzupełnia i uszczegóławia wymagania. Po etapie projektu dostawca może dodatkowo do szacowania użyć elementów projektowych takich jak: obiektów danych (tabel, pól), okien, interfejsów, itp. Z drugiej strony rosną koszty działań po stronie dostawcy. Jeśli dostawca podpisze kontrakt z klientem, koszty te znajdą się w wartości kontraktu, jeśli nie - obciążą w całości dostawcę. Informacje wejściowe, potrzebne do wykonania szacowania na poszczególnych etapach rozszerzonej fazy strategicznej przedstawia Rysunek 2.

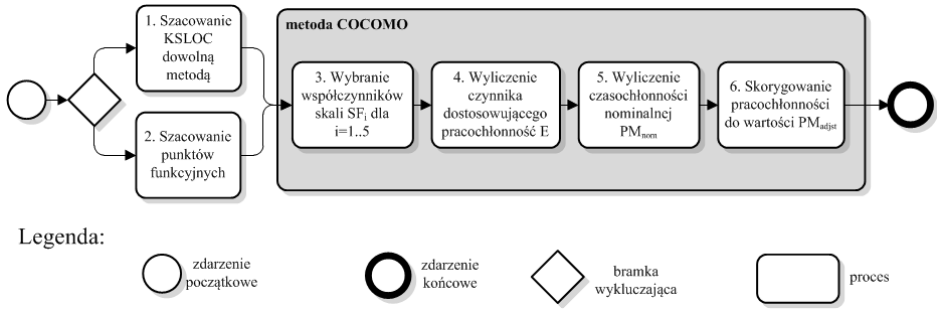


Rysunek 2. Informacje wejściowe dla wycen, na początkowych etapach cyklu życia

2. Algorytmiczne metody szacowania oprogramowania

2.1. Metoda COCOMO II

Metoda Constructive Cost Model (skr. COCOMO) została zaproponowana przez Barry Boehma w 1981 r. [15]. Od tego czasu powstało wiele wersji i odmian tej metody, np.: COCOMO81, COCOMO II [16]. Sekwencję procesów składających się na szacowanie przedstawia Rysunek 3. Za pomocą metody COCOMO można wyliczać czasochłonność (ang. Person per Month, skr. PM) na podstawie wielkości źródłowego kodu programu (ang. Kilo Source Line of Code, skr. KSLOC) (proces 1 na Rysunku 3). Informacje potrzebne do szacowanie ilości kodu pozyskane są z dokumentacji projektu SI. Ilości KSLOC przypisane są do elementów programu takich jak procedury, moduły, obiekty, itp. Ponieważ dla wielu współczesnych zastosowań ilość kodu często nie odpowiada czasochłonności, zmodyfikowano tę metodę, wykorzystując punkty funkcyjne (ang. Function Point, skr. FP) [17] (proces 2 na Rysunku 3) wyliczone na podstawie kompletnych i szczegółowych wymagań użytkownika.



Rysunek 3. Sekwencja procesów w metodzie COCOMO

Pierwszą czynnością jest określenie pięciu współczynników skali (ang. Scale Factor, skr. SF), których wartości wyznaczono empirycznie w pięciu klasach, w zależności od poziomu złożoności (od bardzo niskiej do bardzo wysokiej. Znając współczynniki skali, można wyznaczyć czynnik dostosowujący pracochłonność *E* (ang. Effort) zgodnie ze wzorem:

$$E = B + 0{,}01 \cdot \sum_{i=1}^{5} SF_i \tag{1}$$

gdzie: *B* jest stałą wynoszącą 0,91 dla modelu COCOMO II [17].

Wyliczenie pracochłonności nominalnej PM_{nom} odbywać się będzie zgodnie ze wzorem:

$$PM_{nom} = A \cdot (Size)^{E} \tag{2}$$

gdzie: *Size* - ilość linii kodu w jednostce KSLOC,

A - stała wyznaczona z historycznych projektów wynosząca 2,94 [17].

Dla modeli z pierwszych etapów cyklu życia oprogramowania (Application Composition Model, Early Design Model [17]) nominalny czas powinien zostać skorygowany siedmioma współczynnikami pracochłonności, zgodnie ze wzorem:

$$PM_{adis} = PM_{nom} \cdot \prod_{i=1}^{7} EM_i \tag{3}$$

gdzie: EM_i - (ang. Effort Multiplier) współczynniki pracochłonności.

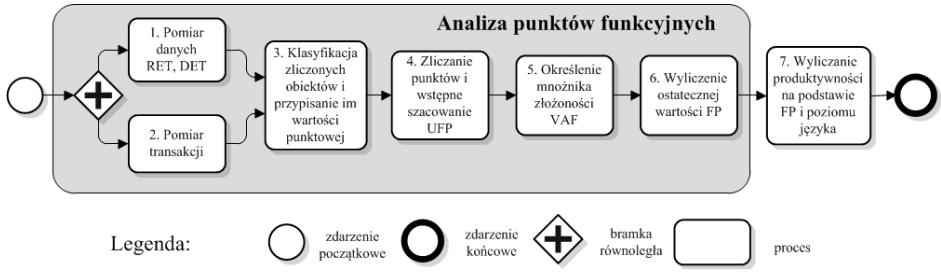
Dla modeli na kolejnym etapie cyklu życia (Post-Architecture Model), wzór na nominalną czasochłonność został rozszerzony o kolejnych 9 współczynników (i=1..16). Podobnie jak *SF* tak i *EM* wyznaczone zostały empirycznie. Dane potrzebne do wyliczeń SF i EM można znaleźć w dokumentacjach metody [17].

W literaturze możemy znaleźć wiele przykładów dostrajania metody *COCOMO* [18] [19] m.in. z wykorzystywaniem logiki rozmytej [20] [21] [22].

2.2. Szacowanie w oparciu o analizę punktów funkcyjnych

Metoda szacowania zaproponowana przez A.J. Albrechta [23] polega na wyliczeniu na podstawie szczegółowych wymagań, ilości punktów funkcyjnych (ang. Function Point, skr. FP) odpowiadających złożoności oprogramowania. Następnie za pomocą metody *COCOMO* lub *Szacowanie przez analogię* można przeliczyć ilość FP na czasochłonność lub koszty. Zbiór wymagań użytkownika będące podstawą obliczeń musi być kompletny a same wymagania szczegółowe.

Proces szacowania za pomocą punktów funkcyjnych przedstawia Rysunek 4.



Rysunek 4. Proces szacowania w oparciu o punkty funkcyjne

Metoda punktów funkcyjnych polega na wydzieleniu w wymaganiach lub gotowym programie pięciu klas obiektów (procesy 1 i 2 na Rysunku 4):

- logiczne wewnętrzne pliki (ang. Internal Logic File, skr. ILF),
- zewnętrzne interfejsy plików (ang. External Interface File, skr. EIF),
- zewnętrzne wejścia (ang. External Inputs, skr. EI),
- zewnętrzne wyjścia (ang. External Outputs, skr. EO),
- zewnętrzne zapytania (ang. External Inquires, skr. EQ.

Dwie pierwsze klasy opisują obiekty związane z danymi, trzy pozostałe - transakcje. Do szacowania na tym etapie (proces 3 na Rysunku 4) wykorzystuje się następujące wskaźniki:

- RET (ang. Record Element Type) - unikalna rozpoznawalna przez użytkownika podgrupa elementów danych w ILF lub EIF,
- DET (ang. Data Element Type) - unikalne, możliwe do zidentyfikowania przez użytkownika pole w ILF lub EIF,
- FTR (ang. File Type Referenced) - rozpoznawalny przez użytkownika zbiór logicznie powiązanych danych.

Następnie należy zidentyfikować wszystkie obiekty w klasach i przypisać im odpowiednią wartość wskaźników. ILF i EIF opisane są za pomocą RET i DET, natomiast EO, EI i EQ za pomocą FTR i DET. Na tej podstawie odczytujemy z tabeli wag [24] ilość nieostatecznych (nieskorygowanych) punktów funkcyjnych *UFP* (ang. Unadjustment Function Point) danego obiektu. Sumując wartości *UFP* dla wszystkich obiektów, we

wszystkich klasach, otrzymujemy wartość sumaryczną nieostatecznych punktów funkcyjnych.

Czynnik korygujący (ang. Value Adjustment Factor - *VAF*) uwzględnia wewnętrzną złożoność systemu niezwiązaną z jego funkcjonalnością. Wyznaczenie go polega na podaniu wartości wpływu dla 14 czynników, które mogą spowodować zwiększenie stopnia skomplikowania systemu (proces 5 na Rysunku 4). Listę czynników można znaleźć w dokumentacji metody [24]. Wartość *VAF* oblicza się wg wzoru:

$$VAF = B + 0{,}01\sum_{i=0}^{14} C_i \tag{4}$$

gdzie: B - empirycznie wyznaczona stała o wartości 0,65 [24],

C_i - wartość wpływu i-tego czynnika.

Znając VAF możemy obliczyć ostateczną wartość punktów funkcyjnych FP korygując wartość nieostatecznych punktów funkcyjnych UFP zgodnie ze wzorem:

$$FP = VAF \cdot UFP \tag{5}$$

Znając wartość FP, produktywność można wyznaczyć dwoma metodami:

- przeliczyć na KSLOC za pomocą empirycznie wyznaczonych wartości z tabeli przeliczeniowej [25] i dalej skorzystać z metody *COCOMO* do wyznaczenia czasochłonności,
- wykorzystując metodę *Szacowania przez analogię* może przeliczyć wartości FP bezpośrednio na czasochłonności, jeśli organizacja posiada odpowiednie dane historyczne,

Źródłem kompletnej i aktualnej dokumentacji *Analizy Punktów Funkcyjnych* jest strona WWW organizacji International Function Point Users Group [26].

3. Niealgorytmiczne metody szacowania oprogramowania

3.1. Dekompozycja i rekonstrukcja

Dekompozycja i rekonstrukcja to z powodu intuicyjności i uniwersalności bardzo popularna metoda. Stosowana jest w sytuacjach, kiedy szacowanie całego zakresu stanowi trudność, np. wynikającą z niejednorodności prac. W praktyce realizacji projektów informatycznych nieczęsto zdarzają się projekty, które można szacować z pominięciem tej metody.

Polega ona na podzieleniu szacowanego zakresu na wiele części. Sposób podziału może być dowolny i uzależniony od specyfiki projektu. Często dostawcy dokonują podziału za pomocą metody Work Breakdown Structure (skr. WBS) [13]. Po dokonaniu podziału części obiektu są szacowane lub dalej dzielone tą samą lub inną metodą. Głębokość podziału zależy od tego, jaką metodą odbędzie się szacowanie na kolejnym etapie. Mimo że w literaturze, metoda ta jest wymieniana na równi z innymi [4], to jej

rola w procesie szacowania jest różna od pozostałych. Szacowania projektów zaczyna się tą metodą, ale po dekompozycji wybierane są inne metody do szacowania cząstkowego. Szczegółowy opis metody dekompozycji zgodnie z WBS można znaleźć w wielu pozycjach literatury [27] [28] [29] [30].

3.2. Indywidualna ocena eksperta

Metoda szacowania poprzez *Indywidualną ocenę eksperta* to najczęściej stosowana metoda, nie tylko w praktyce tworzenia oprogramowania [31], ale i w innych przedsięwzięciach informatycznych, takich jak implementacje czy modyfikacje. Badania przeprowadzone w USA w 2002 roku pokazały, że 72% szacowań odbywa się tą metodą [32]. W pierwszym etapie polega na wytypowaniu ekspertów posiadających odpowiednią do zadań projektowych wiedzę i doświadczenie. W kolejnym etapie wyceny eksperci oceniają przydzielone im zakresy. W celu zmniejszenia błędów szacowania zmodyfikowano metodę o wykonanie szacowań parokrotnie dla różnych wariantów realizacji. Technika taka o nazwie PERT (ang. Program Evaluation and Review Technique) [27] [33] zakłada szacowanie dla: najbardziej korzystnego przypadku, najbardziej prawdopodobnego przypadku, najgorszego przypadku. Różni się jednak od znanej z analiz ścieżki krytycznej (m.in. CPM [34]) tym, że wykorzystywana jest jedynie do szacowania pojedynczych zadań. Po wcześniejszych procesach dekompozycji utracona została informacja o powiązaniach między zadaniami. Oczekiwana wartość szacowania wygląda wówczas następująco:

$$f(x) = \sum_{i=1}^{N} (Cp(x_i) + 4 \cdot Co(x_i) + Ck(x_i))/6 \tag{6}$$

gdzie: Cp – najbardziej korzystna wartość i-tego zadania,
Co – najbardziej prawdopodobna wartość i-tego zadania,
Ck – najmniej korzystna wartość i-tego zadania.

Dokładność wyników zależy wyłącznie od doświadczenia eksperta. Kryteria wyboru eksperta są nieprecyzyjnie określone. Wpływ osobowości jest na tyle duży, że większe doświadczenie nie gwarantuje dokładniejszych wycen. Zdarzają się eksperci zwykle zawyżający szacowania, zaniżający lub nieprzewidywalni.

Metoda ta możliwa jest do stosowania od etapu pierwszych kontaktów dostawcy z klientem. Dzięki odpowiedniemu doborowi ekspertów szacować można nawet na podstawie niekompletnego zbioru ogólnych wymagań użytkownika.

3.3. Ocena eksperta w grupach

Metoda polega na przedstawieniu do wyceny tego samego zakresu prac więcej niż jednemu ekspertowi. W wersji niestrukturalnej (recenzje grupowe) eksperci wspólnie ustalają wartość wyceny lub zakres wyceny. W wersji strukturalnej zwanej Wideband Delphi [35] [15] ustalenia ekspertów dokonuje się w sposób sformalizowany i efektem jest

wycena punktowa.

Praca w grupach jest kosztowniejszą metodą niż praca pojedynczych ekspertów, ale przewagą nad *Indywidualną oceną eksperta* jest zmniejszenie wpływu składnika osobowościowego. Mimo różnych doświadczeń, charakterów i skłonności, albo eksperci dojdą do wspólnych ustaleń, albo jak w odmianie Wideband Delphi rozstrzygnięcie spornych wycen nastąpi poprzez przypisanie umownych punktów.

Metoda szacowania stosowana jest szczególnie często w początkowych etapach projektów informatycznych, w sytuacjach dużej niepewności i nieprecyzyjności wymagań.

3.4. Zliczanie, obliczanie i ocenianie

Metoda polega na odszukaniu w projekcie obiektów, które dają się zliczyć, takich jak wymagania, funkcje, przypadki użycia, historyjki, raporty, okna dialogowe, tabele baz danych, klas. Do każdego zidentyfikowanego obiektu wymagane jest przypisanie wielkości składowej szacowania (kosztu lub czasu). Wartości szacowane są funkcją obiektów składających się na projekt informatyczny:

$$f(x) = \sum_{i=1}^{N} C(x_i) \tag{7}$$

gdzie: x - zliczony obiekt,

N - ilość zliczonych obiektów,

C - oceniany koszt zliczonego obiektu.

Metoda może być stosowana na każdym z etapów powstawania lub modyfikacji oprogramowania. Nie jest skomplikowana pod warunkiem, że w dokumentacji źródłowej, np. w studium wykonalności lub analizie przedwdrożeniowej, można wyodrębnić zliczane obiekty. Wadą jest duże ryzyko pominięcia obiektów lub zakresów prac, które mają wpływ na wartość całego projektu, na przykład pominięcie tabel pomocniczych lub nieuwzględnienie kosztów przygotowania zapytań filtrujących w trakcie szacowania kosztów okien interfejsów. Ważnym etapem tej metody jest ocena poszczególnych obiektów. Dokonać tego można, wspomagając się metodą *Indywidualnej oceny eksperta* lub *Oceny eksperta w grupach*. Metoda jest efektywna w projektach, w których udało się wyodrębnić niewiele rodzajów obiektów za to występujących licznie, na przykład 30 raportów, 25 zapytań SQL i 18 okien interfejsów.

3.5. Szacowanie przez analogię

Metoda polega na podzieleniu projektu na takie części, jakie występują w już zrealizowanym projekcie. Szacując wybrane części, można obliczyć stosunek wielkości obu projektów (nowego i zrealizowanego). Znając relacje między wielkościami i koszty zrealizowanego projektu, można oszacować wartość nowego projektu.

Trudność polega na zebraniu danych historycznych z projektów o podobnym charakterze i strukturze jak szacowny projekt. Dodatkowym problemem jest wybór reprezentatywnej części zdekomponowanego projektu, na podstawie której obliczany jest współczynnik krotności. Pominięcie istotnych obiektów może zwiększyć błąd szacowania.

Danymi wejściowymi dla tej metody powinny być obiekty danych i programów (okna, zapytania SQL, FP). Wykorzystanie wymagań, nawet szczegółowych, nie pozwoli na obliczenie współczynnika krotności, a co za tym idzie całego szacowania. Stąd metoda może być stosowana dopiero na etapie, na którym znane są efekty fazy projektowania.

3.6. Szacowania oparte na zastępstwie

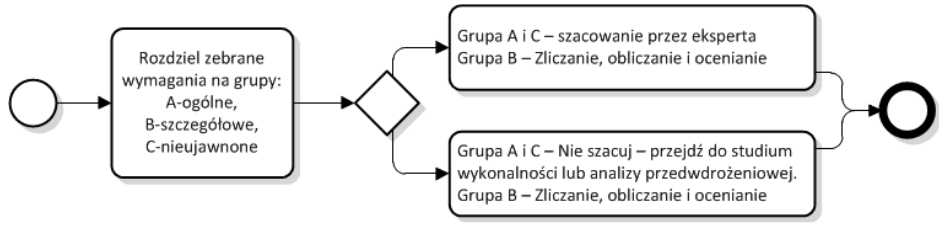
Metoda ta, podobnie jak poprzednia, wymaga znajomości kosztów wcześniej zrealizowanych w danej organizacji obiektów standardowych (interfejsy, raport, itp.). W zależności od wersji metody obiekty mogą być różnie grupowane. Na przykład, Putnam [33] i Humphrey [36] wyodrębnili klasy obiektów: bardzo małe, małe, średnie, duże, bardzo duże. Innym sposobem klasyfikacji jest metoda standardowych składników [4] używana do szacowania oprogramowania obiektowego. Jeśli dostawca SI wykorzystuje programowanie ekstremalne lub bliskie metodom Agile [37], standardowym składnikiem mogą być tzw. historyjki.

Następnie grupom obiektów przypisuje się średnie historyczne miary kosztów, np. liczby linii kodu (skr. ang. LOC), roboczogodziny lub roboczodni. W podobny sposób trzeba sklasyfikować obiekty z nowego projektu. Wówczas, na tej podstawie można obliczyć ich sumaryczną wartość.

Podobnie jak poprzednia metoda, ta również powinna być wykorzystywana. gdy znane są klasy obiektów programistycznych. Wyjątkiem są organizacje używające programowania ekstremalnego, czy zwinnego. W tym przypadku, koszty „historyjek" jakie udokumentowane zostały na etapie rozmów z klientem mogą być zastępowane danymi historycznymi. Praktyka wycen [2] wskazuje jednak, że wykorzystywana może być na etapie wcześniejszym (analiza przedwdrożeniowa), kiedy znane są tylko wymagania szczegółowe.

4. Wnioski

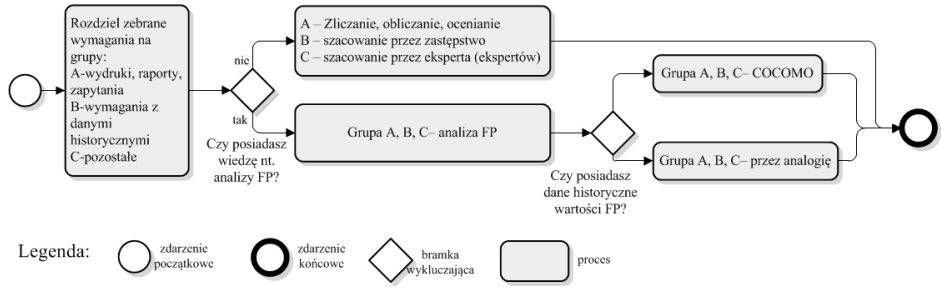
Podsumowując, należy zauważyć, że z jednej strony realizacja pierwszych etapów cyklu życia oprogramowania dostarcza coraz więcej informacji o planowanym wdrożeniu, a z drugiej strony istnieje zbiór metod szacowania, które na wejściu potrzebują różnego typu informacji (w zależności od metody).



Rysunek 5. Wybór metod szacowania na etapie rozmów handlowych

Na podstawie dokonanej analizy metod szacowania, autorzy proponują własną metodę wyboru najtrafniejszego sposobu oceny kosztów i czasu realizacji wdrożenia (modyfikacji oprogramowania w trakcie wdrożenia).

Dla etapu rozmów handlowych algorytm postępowania przedstawiony jest na Rysunku 5. Jak widać dla wszystkich grup wymagań szacować można czas i koszty tylko w przypadku gdy wymagania te będą już kompletne. W praktyce należy dla większości wymagań, przejść do studium wykonalności (analizy przedwdrożeniowej). W taki przypadku mamy do wyboru inne metody szacowania co przedstawiono na Rysunku 6.



Rysunek 6. Wybór metod szacowania na etapie analizy przedwdrożeniowej (studium wykonalności)

Etap projektu zmian systemu dostarcza wyceniającym, oprócz wymagań dodatkowych, informacji związanych z realizacją prac - struktury danych, informacje o procedurach realizujących procesy, obiektach, itp. Podobnie, jak na poprzednich etapach, dostawca powinien dokonać klasyfikacji dostępnych danych:

A. dane umożliwiające szacowanie obiektów w KSLOC,
B. dane, o których lub podobnych dostawca posiada informacje historyczne (koszty realizacji),
C. dane, o których dostawca posiada sklasyfikowane informacje historyczne (koszty realizacji),

Następnie wymagania typu A dostawca szacuje metodą *COCOMO* po wcześniejszym oszacowaniu KSLOC, typu B - *Metodą przez Analogię*, typu C - *Metodą przez Zastępstwo*.

Powyższe propozycje postępowania pozwalają wykorzystywać te metody, które w warunkach poszczególnych etapów będą najefektywniejsze.

Bibliografia

1. M. Burns, „How to select and implement an ERP System," 2005. [Online]. Available: http://www.180systems.com/ERPWhitePaper.pdf.
2. P. Plecka, „Selected Methods of Cost Estimation of ERP Systems' Modyfications.," *Zarządzanie Przedsiebiorstwem,* p. w druku, 2013.
3. I. Sommerville, Software Engineering, Edingurgh: Pearsom Education Limited, 2007.
4. S. McConell, Software Estimation: Demystifying the Blac Art., Microsoft Press, 2006.
5. R. Meli, „Early Function Points: a new estimation method for software projekct," w *WSCOM97*, Berlin, 1997.
6. L. Santillo, M. Conte i R. Meli, „Early &Quick Function Point: Sizing More with Less," w *Metrics 2005, 11 th IEEE Intl Software Metrics Symposium*, Como, Italy, 2005.
7. B. Boehm, C. Abts i S. Chulani, „Software Development Cost Estimation Approaches – A Survey," *Annals of Software Engineering,* tom 10, nr 1-4, pp. 177-205, 2000.
8. „BPMN," Object Management Group, 2013. [Online]. Available: http://www.bpmn.org/.
9. K. Frączkowski, Zarządzanie projektem informatycznym., Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2003.
10. J. Philips, IT Project Management. On Track from Start to Finish., Osborne, 2004.
11. K. Justynowicz, „Analiza przedwdrożeniowa coraz popularniejsza," wrzesień 2007. [Online]. Available: http://www.bcc.com.pl/akademia-lepszego-biznesu/analiza-przedwdrozeniowa-coraz-popularniejsza.html.
12. P. Allen i S. Frost, Component-Based Development for Enterprise Systems, Applying the Select Perspective, Cambridge: Cambridge University Press, 1998.
13. „Select Business Solution," [Online]. Available: http://www.selectbs.com.
14. „ARIS," [Online]. Available: http://www.softwareag.com.
15. B. Boehm, Software Engeneering Ecomonics, New York: Englewood Clifs, 1981.
16. B. W. Boehm, Software Cost Estimation with COCOMO II, Prentice Hall, 2000.
17. J. Baik, COCOMO II, Model Definition Manual, Version 2.1, Center for Software Engineering at the University of Southern California, 2000.
18. J. Hele, A. Parrish, B. Dixon i R. Snith, „Enhancing the Cocomo estimation models," *Software, IEEE, Volume:17, Issue: 6* , pp. 45-49, 2000.

19. S. Aljahdali i A. Sheta, „Software effort estimation by tuning COOCMO model parameters using differential evolution," w *Computer Systems and Applications (AICCSA), IEEE/ACS International Conference on*, Hammamet, 2010.
20. Z. Fei, „f-COCOMO: fuzzy constructive cost model in software engineering," w *Fuzzy Systems, 1992., IEEE International Conference on*, San Diego, CA, 1992.
21. R. C. Satyananda, „An Improved Fuzzy Approach for COCOMO's Effort Estimation using Gaussian Membership Function," *JOURNAL OF SOFTWARE,* tom VOL. 4, nr NO. 5, pp. 452-459, July 2009.
22. I. Attarzadeh, „Improving estimation accuracy of the COCOMO II using an adaptive fuzzy logic model," w *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, Taipei, 2011.
23. A. Albreht, „Measuring Application Development Productivity," w *Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium*, Monterey, California, USA, 1997.
24. IFPUG, Function Point Counting Practices: Manual Release 4.1, Westerville, OH: IFPUG, 1999.
25. „The QSM Function Points Languages Table," QSM, 2013. [Online]. Available: http://www.qsm.com/resources/function-point-languages-table.
26. „International Function Point Users Group," [Online]. Available: http://www.ifpug.org/.
27. R. D. Stutzke, Estimation Software-Intensive Systems, Upper Saddle River, New York: Addison-Wesley, 2005.
28. R. Tausworthe, „The work breakdown structure in software project management," *Journal of Systems and Software,* tom 1, 1984.
29. E. Norman, S. Brotherton i R. Fried, Work Breakdown Structures: The Foundation for Project Management Excellence, John Wiley & Sons, 2010.
30. G. Haugan, Effective Work Breakdown Structures, Project Management Institute, 2002.
31. M. Jorgensen, „A Review of Studies on Expert Estimation of Software Development Effort," *Journal of Systems and Software,* tom 70, nr 1-2, p. 37–60, Fabruary 2004.
32. B. Kitchenham, S. L. Pfleeger, . B. McColl i S. Eagan, „An empirical study of maintenance and development estimation accuracy.," *Journal of Systems and Software,* tom 64, nr 1, pp. 57-77, 2002.
33. P. L. H. i. W. Myers, Measures for Excellence: Reliable Software on Time, Within Budget, Englewood Cliffs, NY: Yourdon Press, 1992.
34. J. W. Fondahl, „The History of Modern Project Management Precedence Diagramming Methods: Origins and Early Development," *Project Management Journal,* tom XVIII., nr 2, 1987.
35. NASA, „ISD Wideband Delphi Estimation," 09 2004. [Online]. Available:

http://software.gsfc.nasa.gov/assetsapproved/PA1.2.1.2.pdf.
36. W. S. Humphrey, A Discipline for Software Engineering, Addison Wesley, 1995.
37. M. Cohn, Agile Estimating and Planning, Upper Side River, NY: Prentice Hall PTR, 2005.

Streszczenie

W pracy poruszono problem doboru metod szacowania kosztów modyfikacji systemu ERP podczas wdrożenia. Przeprowadzono przegląd dostępnych metod opisanych w literaturze oraz scharakteryzowano etapy fazy strategicznej procesu wdrożenia. Na podstawie analizy zakresu danych wymaganych przez poszczególne metody oraz danych uzyskiwanych na różnych etapach, zaproponowano algorytmy doboru metod do tychże etapów.

Słowa kluczowe: ERP, projekt informatyczny, szacowanie kosztów, analiza punktów funkcyjnych, wdrożenia systemów informatycznych.

Summary

The work discusses the problem of selecting methods for valuing the costs of modifying ERP systems during implementation process. The methods presented in literature have been reviewed and the stages of strategic phase of implementation have been characterised. On the basis of the analysis of data required by each method and the data obtained at different stages, algorithms of method selection for each stage ware proposed.

Key words: ERP, IT project, cost valuation, function points analysis, IT systems implementations.