

WYKORZYSTANIE MIAR ZŁOŻONOŚCI PROJEKTU DO OCENY STANU EWOLUCJI ORGANIZACJI INFORMATYCZNEJ

Cezary ORŁOWSKI, Tomasz DERĘGOWSKI, Miłosz KURZAWSKI,
Katarzyna OSSOWSKA, Artur ZIÓLKOWSKI

Streszczenie: Celem artykułu jest ocena możliwości wykorzystania miar złożoności projektu do oceny stanu ewolucji organizacji informatycznej. Stan ewolucji organizacji informatycznej ma znaczenie dla powszechnego obecnie zjawiska zwinnej transformacji, tak ważnej dla obniżenia ryzyka realizacji projektów informatycznych.

Autorzy pracy przedstawili na wstępie problematykę procesów ewolucji organizacji informatycznych w kontekście zmian transformacyjnych. Prezentacja tej problematyki jest kluczowa dla poszukiwania wskaźników istotnych dla oceny stanu ewolucji organizacji informatycznych. W artykule analizie poddano stosowane w praktyce metody CMMI i SCAMPI aby wyodrębnić w ramach tych metod te obszary procesów, które decydują o procesach zmian stanu organizacji informatycznych (w kontekście dojrzałości procesowej tych organizacji). Dla wyodrębnionego obszaru MA (ang. *Measurements and Analysis*) typowego dla ocen dojrzałości według modelu SCAMPI poddano ocenie możliwość wykorzystania metodę punktów funkcyjnych (FPA) aby wykazać przydatność tej metody do oceny stanu procesu a w konsekwencji do oceny stanu organizacji.

Badanie miary procesu w oparciu o FPA zweryfikowano (w procesach adaptacji miar) w środowisku realizacji ponad 200 projektów informatycznych. Artykuł podsumowuje ocenę przydatności istniejących miar procesów wytwarzania oprogramowania do oceny stanu organizacji informatycznych (ewolucji i transformacji)

Słowa kluczowe: procesy zwinnej transformacji, miary procesów, modele dojrzałości organizacji, metoda punktów funkcyjnych.

Słownik pojęć:

| | |
|-----------------|---|
| CMMI | (ang. <i>Capability Maturity Model Integration</i>) – model oceny dojrzałości organizacji. W ramach modelu CMMI można wyróżnić 5 poziomów dojrzałości organizacji: 1 – początkowy, 2 – Zarządzany, 3 – Zdefiniowany, 4 – Zarządzanie Ilościowe, 5 – Optymalizacja [1]. |
| SCAMPI | (ang. <i>Standard CMMI Appraisal Method for Process Improvement</i>) – metoda oceny dojrzałości organizacji zgodnie z CMMI. W ramach SCAMPI wyróżniono 22 procesy, podzielone na 49 celów oraz 167 praktyk [2]. |
| Proces | Zbiór praktyk, których implementacja i wykonywanie przybliża organizację do spełnienia określonych celów oraz w efekcie poprawę w danym obszarze. |
| Cel | Wyspecyfikowane w ramach każdego procesu (obszaru procesu) w CMMI. Opisują co powinno być spełnione aby uznać obszar procesowy (procesy) za zaimplementowane i spełnione. |
| Praktyka | Działania wyspecyfikowane w ramach każdego celu, których implementacja pozwala na ich osiągnięcie. |

Miara złożoności Miary złożoności oprogramowania pozwalają na oszacowanie wielkości projektowanego (wytwarzanego) systemu.

1. Wprowadzenie

Celem artykułu jest wskazanie przydatności miar złożoności projektu opartych o FPA (ang. *Function Points Analysis*) dla oceny organizacji informatycznych stojących przed koniecznością realizacji procesów zwinnej transformacji. Zjawisko zwinnej transformacji jest złożonym wyzwaniem dla każdej organizacji informatycznej i nie zostało dotychczas szczegółowo zbadane. Nie prowadzi się także badań stanu gotowości organizacji informatycznych do realizacji procesów zwinnej transformacji a same procesy transformacji mają charakter niekontrolowany. Wiele organizacji zawiesza transformację lub po jej przejściu, wraca do stanu przed transformacją. Efektem obserwowanych w takich organizacjach informatycznych procesów transformacji jest trudny do jednoznacznej identyfikacji stan organizacji informatycznej, co prowadzi do zwiększenia liczby nieudanych projektów informatycznych. Dlatego też aby zminimalizować ryzyko realizacji tych projektów, należy monitorować procesy transformacji jak też zapewnić aby procesy transformacji miały charakter ciągły, istnieje potrzeba identyfikacji i badania tych procesów. Doświadczenia autorów wskazują, że o powodzeniu zwinnej transformacji w dużej mierze decyduje dojrzałość organizacji wyrażana zarówno gotowością do realizacji złożonych projektów jak również wynikająca z dojrzałości procesów realizowanych przez te organizacje. Na dojrzałość procesową składa się szereg czynników, które organizacje powinny poddawać ocenie przed podjęciem decyzji o radykalnych (transformacyjnych) zmianach. Jednym z typowych podejść do oceny dojrzałości jest model CMMI, którego znaczenie dla badania organizacji informatycznych zostanie przedstawione w dalszej części. Obserwacje prowadzone przez autorów w organizacjach realizujących procesy transformacji pokazały, że procesy transformacyjne przebiegają w sposób bardziej kontrolowany w tych organizacjach, w których panuje duża umiejętność stosowania mechanizmów mierzenia dojrzałości procesów. Jest to zgodne z podejściami zwinnymi (jak SCRUM), które wymagają stałego, konsekwentnego mierzenia zarówno dojrzałości jak i gotowości do realizacji określonych procesów. Stąd też problematyka mierzenia – zarówno dojrzałości organizacji, jak i złożoności projektów czy dojrzałości zespołów wydaje się istotna w obliczu zwinnej transformacji.

Przy ocenie stanu dojrzałości organizacji stosuje się szereg technik i metodyk pomiarowych, które dostarczają standardy takie jak ITIL (ang. *Information Technology Infrastructure Library*), COBIT (ang. *Control Objectives for Information and Related Technology*) czy CMMI (ang. *Capability Maturity Model Integration*) [3]. W przypadku tego rodzaju podejść postępowanie dotyczące pomiaru dojrzałości najczęściej ma charakter wstępny, prezentujący stan początkowy (przed uruchomieniem konkretnych projektów). Jednak wraz z ewolucją organizacji i koniecznością adaptowania do zmian, konieczne staje się stałe monitorowanie i audytowanie poziomów dojrzałości. Aby realizować procesy oceny dojrzałości organizacji (jej stanu) konieczne staje się badanie konkretnych obszarów procesowych. Ich liczba jest zależna od standardów (w przypadku CMMI i stosowanej metody oceny SCAMPI wyróżniono 22 obszary procesów).

W niniejszym artykule postanowiono skoncentrować się na jednym z nich, tym którego stosowanie wydaje się kluczowe dla procesów zwinnej transformacji. Wybrany obszar dotyczy stałego i świadomego stosowania miar dojrzałości. Obszar procesowy związany



z konsekwentnym doskonaleniem poprzez stosowanie miar określanym jest mianem MA (ang. *Measurements and Analysis*), i jego celem jest określenie i analiza miar procesów także ważnych z punktu widzenia oceny stanu organizacji. W przypadku organizacji projektowych w ramach tego obszaru procesowego stosuje się szereg technik inżynierskich, które służą mierzeniu np. złożoności projektów [4]. Takie podejście wpisuje się właśnie w obszar MA doskonalenia procesów. Należy także zaznaczyć, że stosowanie miar złożoności projektu, szacowania i estymacji jest kluczowe dla realizacji procesów zwinnej transformacji. Organizacje dokonujące transformacji muszą dokonywać stałej estymacji wytwarzanych funkcjonalności systemów. Dlatego też autorzy niniejszego artykułu pokazali jak w ramach zasad MA (mierzenia i analiz) można stosować określone techniki inżynierii oprogramowania pomagające szacować złożoność wytwarzanych systemów. Stąd też dla obszaru procesowego istotnego z punktu widzenia oceny dojrzałości organizacji, omówiono wybraną miarę estymacyjną opartą na analizie punktów funkcyjnych pozwalających na określenie złożoności funkcjonalności, a tym samym złożoności projektu. Organizacje stosujące takie estymacje mogą podnosić swoją dojrzałość procesową, co może mieć znaczenie w procesie transformacji. Na podstawie miary opartej na punktach funkcyjnych można zarówno definiować stan projektu ale również próbować przenosić tę miarę do oceny stanu organizacji realizującej ten projekt. Stan projektu i stan organizacji są dynamicznymi zmiennymi wpływającymi na ogólną dojrzałość organizacji. Dlatego też w dalszej części artykułu zaprezentowana zostanie ocena punktów funkcyjnych (FPA) jako miary procesu dla potrzeb oceny stanu organizacji zarówno z punktu widzenia jej przydatności ale przede wszystkim z punktu widzenia możliwości weryfikacji. Metoda ta została zweryfikowana w oparciu o bazę danych dotyczących realizacji ponad 200 projektów informatycznych.

Artykuł podzielono na cztery główne części. W pierwszej z nich opisano szczegółowo proces MA i jego znaczenie dla oceny stanu organizacji. Część 2 stanowi wprowadzenie do procesu pomiaru a także ich odniesienie do procesu MA. W części trzeciej przedstawiono weryfikację opracowanych miar. W części czwartej przeprowadzono procesy adaptacji opracowanych miar dla oceny stanu organizacji informatycznej.

2. Procesy pomiaru i analizy stanu organizacji informatycznej

Proces MA (ang. *Measurements and Analysis*) jest jednym z 22 procesów opisanych w ramach modelu CMMI, Jego celem jest określenie miar i analiz ważnych z punktu widzenia organizacji. Zgodnie z metodą oceny SCAMPI w ramach tego procesu można wyróżnić dwa główne cele, do osiągnięcia których powinna dążyć każda dojrzała organizacja oraz 8 praktyk, po 4 w ramach każdego celu [5]. Pierwszym celem jest zapewnienie, że miary i analizy są dopasowane (ang. *Align Measurements and Analysis Activities*). Istotnym jest bowiem wyspecyfikowanie miar i prowadzonych analiz jeszcze przed rozpoczęciem każdego projektu. Zapewnia to, że miary i analizy będą dobrane zgodne z potrzebami i celami organizacji. W ramach tego celu wyróżnić możemy następujące praktyki [6]:

- Ustal cele pomiarowe (ang. *Establish Measurement Objectives*)
- Określ miary (ang. *Specify Measures*)
- Określ procedury zbierania i przechowywania danych (ang. *Specify Data Collection and Storage Procedures*)
- Określ procedury analizy (ang. *Specify Analysis Procedures*)

Stosowanie powyższych praktyk zapewnia, że organizacja dąży do osiągnięcia celu jakim

jest zapewnienie, że miary i analizy w niej stosowane są znane i dostosowane do jej profilu i działalności a także prowadzonych projektów. Drugim celem wyszczególnionym w ramach procesu MA jest dostarczanie wyników pomiarów (ang. *Provide Measurements Results*). Stosowanie procedur związanych z dostarczaniem wyników pozwala bowiem oceniać projekt oraz w patrząc globalnie w skali wszystkich projektów w danej organizacji, całą organizację. Ponadto umożliwia kontrolowanie projektów w całym ich cyklu życia a także przekazywanie informacji decydom, którzy odpowiedzialni są za prowadzenie projektu. W ramach tego celu wyróżnić można następujące cztery praktyki:

- Uzyskaj dane pomiarowe (ang. *Obtain Measurements Data*)
- Analizuj dane pomiarowe (ang. *Analyze Measuremet Data*)
- Magazynuj dane i wyniki (ang. *Store Data and Results*)
- Komunikuj wyniki (ang. *Communicate Results*)

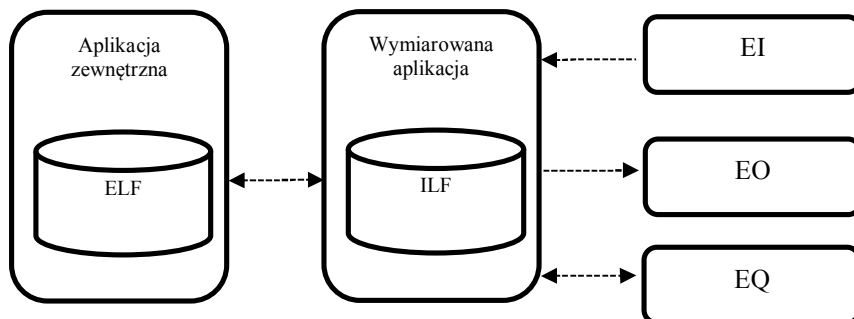
Stosowanie powyższych 4 oraz łącznie w ramach dwóch celów 8 praktyk zapewnia, że proces MA występuje w danej organizacji. W niniejszej publikacji autorzy opracowali metodą szacowania nakładów pracy i czasu trwania projektów przy wykorzystaniu metody punktów funkcyjnych. W opisanej metodzie starano się zaimplementować takie miary i procedury, aby zapewnić zgodność z opisanymi powyżej 8 praktykami, w ramach 2 celów procesu MA.

Przykładem metody pomiaru i analizy która może być zastosowana w organizacji informatycznej może być metoda punktów funkcyjnych (ang. *FPA - Function Point Analysis*). Jest ona wykorzystywana do szacowania złożoności oprogramowania. Rozwijana przez organizację IFPUG (ang. *The International Function Point User Group*) metoda estymacji może być wykorzystywana do oceny wszystkich typów projektów informatycznych, bez względu na język programowania. Co ważne, można ją wykorzystywać do projektów związanych z wytwarzaniem nowego systemu jak i modyfikacją np. rozbudową [7]. Ponadto autorzy w swoich badaniach starali się zweryfikować możliwość wykorzystania metody FPA do oceny procesu MA oraz w efekcie możliwości stosowania metody do oceny stanu organizacji i możliwości rozpoczęcia zwinnej transformacji w organizacji.

Metoda FPA zakłada, że na strukturę każdego systemu IT składa się pięć różnych komponentów [8]:

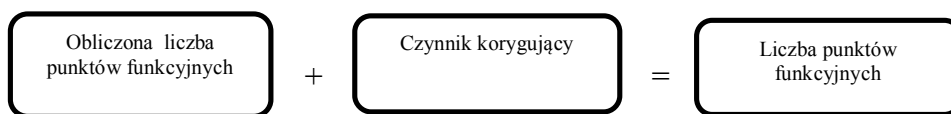
- Wewnętrzne pliki logiczne ILF (ang. *Internal Logic File*) – jest to grupa logicznie powiązanych ze sobą danych znajdujących się w projektowanym systemie.
- Zewnętrzne pliki logiczne ELF (ang. *External Logic File*) – jest to grupa logicznie powiązanych ze sobą danych znajdujących się poza projektowanym systemem, w zewnętrznej aplikacji.
- Wejścia EI (ang. *External Input*) – jest to grupa danych przychodzących spoza projektowanej aplikacji.
- Wyjścia EO (ang. *External Output*) – jest to grupa danych, które są wysyłane poza projektowaną aplikację.
- Zapytania EQ (ang. *External Inquire*) – wykorzystywane są do wysyłania danych poza granice aplikacji.

Strukturę systemu oraz powiązania między opisanymi powyżej elementami przedstawiono graficznie na rys. 1.



Rys. 1 Struktura każdego systemu w metodzie FPA

Wymiarowanie systemu informatycznego za pomocą metody punktów funkcyjnych składa się z dwóch faz. W pierwszej szacuje się rozmiar funkcjonalny systemu, który jest sumą punktów funkcyjnych i wynika z funkcjonalności jakich oczekuje użytkownik (nieskorygowana liczba punktów funkcyjnych). W drugiej fazie, obliczona liczba FP (ang. *Function Points*) ulega skorygowaniu o czynnik korygujący, który wynika z elementów, które mogą mieć wpływ na wielkość systemu. Suma nieskorygowanych punktów funkcyjnych i czynnika korygującego daje ostateczną liczbę punktów funkcyjnych.



Rys. 2 Schemat obliczania skorygowanej liczby punktów funkcyjnych

Oszacowana, zgodnie ze schematem przedstawionym na rys. 2 liczba punktów funkcyjnych stanowi bazę na oszacowania miar, które mogą być wykorzystywane przy ocenie procesu MA oraz gotowości do rozpoczęcia zwinnej transformacji. Metoda harmonogramowania w zależności od wybranego języka programowania przedstawiona została w kolejnym punkcie niniejszego artykułu.

3. Miary stanu organizacji informatycznej

W niniejszym rozdziale opisano miary oraz sposób ich obliczania, które mogą być wykorzystywane przy ocenie procesu MA oraz w efekcie stanu organizacji i jej gotowości do rozpoczęcia zwinnej transformacji. Opisywana metoda zakłada uwzględnienie tylko złożoności (rozmiaru) oprogramowania oraz ram czasowych na zamknięcie projektu. Oznacza to pominięcie innych, także istotnych z punktu widzenia estymacji projektów, elementów takich jak np. czynniki nieprzewidzianie (nieoczekiwana niedostępność zasobów, awarie itp.).

Na podstawie FPA możemy oszacować np. liczbę stron dokumentacji, przypadków testowych czy też liczbę osób potrzebnych w projekcie [9]. Poza przedstawionymi powyżej przypadkami umożliwia też w prosty sposób oszacowanie wstępnego harmonogramu



(czasu trwania) projektu, określenie prawdopodobieństwa dotrzymania założonego czasu czy też nakładów pracy.



Rys. 3 Model szacowania harmonogramu w oparciu o FPA

Zaproponowana przez autorów metoda szacowania miar MA projektów IT opiera się na wykorzystaniu dwóch elementów wejściowych – liczby punktów funkcyjnych oraz wybranego języka programowania. Konkurencyjną dla zaproponowanej jest metoda wykorzystująca jako elementy wejściowe liczbę punktów funkcyjnych oraz wskaźnik linii kodu w zależności od języka programowania – LOC (ang. *Lines of Code*) [10]. Metoda ta obciążona jest jednak dużym wskaźnikiem błędów ze względu na nierozróżnianie np. części kodu łatwiejszych od trudniejszych czy też stylu programowania przez danego programistę. Ze względu na to jest ona coraz rzadziej wykorzystywana w szacowaniu projektów. Inną z metod jest wykorzystanie wskaźnika liczby godzin na punkt funkcyjny w zależności od języka programowania. Metoda ta opiera się na wykorzystaniu średniego czasu potrzebnego na „wyprodukowanie” jednego punktu funkcyjnego. Jej przewagą nad metodą LOC jest fakt, że nie koncentruje się ona na jednym wyznaczniku (liniach kodu) a wartościach uśrednionych, dzięki czemu uwzględnia zarówno łatwiejsze jak i trudniejsze części kodu czy też styl programowania.

Do wyliczenia łącznego czasu potrzebnego na wykonanie aplikacji możemy posłużyć się czasem uśrednionym w zależności od wybranego języka programowania. Wartości te zostały przez autorów wyznaczone na podstawie danych o projektach pochodzących z bazy ISBSG – ang. *The International Software Benchmarking Standards Group Limited (Release 12 February 2013)*. W analizie uwzględniono sześć, następujących języków programowania, dla których wyznaczono medianę: ASP, C++, C, C#, Java, SQL oraz Visual Basic. Zamiast średniej arytmetycznej zdecydowano się na medianę ze względu na zbiór przypadków z przypadkami odstającymi. Spośród 6006 projektów wybrano łącznie 1582 ważnych przypadków. Za przypadek ważny uznano taki projekt, w którym wykorzystywano jeden z wymienionych powyżej języków programowania, dane były informacje na temat liczby punktów funkcyjnych, liczbę FP wyznaczono metodą IFPUG oraz podana była informacja na temat nakładu pracy wyrażonego w godzinach. W tabeli 1 przedstawiono otrzymane wyniki w podziale na język programowania.

Tab. 1 Czas w godzinach przypadający na 1FP dla różnych języków programowania [11]

| Języka programowania | N ważnych | Min. | Q1 | Q2 (mediana) | Q3 | Max. | Średnia |
|----------------------|-----------|------|-----|--------------|------|-------|---------|
| ASP | 26 | 0,4 | 2,7 | 6,3 | 9,2 | 29,1 | 7,3 |
| Visual Basic | 327 | 0,1 | 4,4 | 8,2 | 18,3 | 256,1 | 15,4 |
| Java | 682 | 0,1 | 5,7 | 9,3 | 15,1 | 259,7 | 13,0 |
| SQL | 135 | 0,4 | 4,8 | 9,0 | 15,5 | 67,4 | 12,3 |
| C++ | 159 | 0,3 | 6,6 | 14,6 | 29,1 | 211,8 | 24,3 |
| C | 191 | 0,1 | 5,6 | 11,4 | 21,8 | 141,1 | 18,0 |
| C# | 62 | 1,9 | 9,9 | 17,3 | 26,0 | 54,0 | 20,4 |

Znając liczbę punktów funkcyjnych oraz posługując się danymi z tabeli 1 możemy oszacować czas trwania projektu. W pierwszej kolejności należy określić wymagany nakład



pracy wyrażony w godzinach (1).

$$TWE = FPs \times HpFP \quad (1)$$

gdzie:

- TWE – całkowity nakład pracy (ang. *Total Work Effort*)
- FPs – całkowita liczba punktów funkcyjnych (ang. *Funtion Points*)
- HpFP – liczba godzin przypadających na punkt funkcyjny (ang. *Hours per Function Point*)

Mając policzony całkowity nakład pracy można przeliczyć go na tygodnie bądź miesiące. Zakładając pięciodniowy tydzień pracy oraz ośmiogodzinny dzień, w celu obliczenia czasu wyrażonego np. w miesiącach należy TWE podzielić przed odpowiednią ilość godzin (2).

$$TWE \text{ w miesiącach} = \frac{TWE}{5 \times 4 \times 7,5} \quad (2)$$

We wzorze 4.2. założono dzienny czas pracy każdego z programistów na poziomie 7,5 godzin. Jest to optymistyczny czas, dlatego też aby maksymalnie urealnić wyniki, autorzy proponują zastosowanie metody PERT do policzenia rzeczywistego, dziennego czasu pracy programisty (RDWT – *Real, Daily Working Time*) (3), zakładając odpowiednio:

- Czas optymistyczny (O) = 7,5h/dziennie
- Czas najbardziej prawdopodobny (M) = 6,5h/dziennie
- Czas pesymistyczny (P) = 5,5h/dziennie

Wartości te mogą być indywidualnie dostosowywane w zależności od zespołu, jego dojrzałości oraz stylu pracy.

$$RDWT = \frac{O+4M+P}{6} = \frac{7,5 + 4 \times 6,5 + 5,5}{6} = 6,5h/dziennie \quad (3)$$

Zastosowanie metody PERT zapewnia dodatkowy bufor bezpieczeństwa podczas szacowania miar projektowych. W szacowaniu istotnym elementem jest także uwzględnienie, oprócz czasu pracy, wielkości zespołu deweloperskiego. Współczynnik TWE pozwala na obliczanie łącznego czasu pracy tj. łącznej ilości godzin pracy programisty do wykonania danego projektu. W zależności od zespołu, kalendarzowy czas wykonania ulegnie zatem zmianie. Niezbędnym jest zatem policzenie jednostkowego TWE, który jednocześnie stanowić będzie czas trwania projektu wyrażony w miesiącach (PT – *Project Time*) (4.4).

$$PT = \frac{TWE}{(4 \times 5 \times RDWT) \times NoD} \quad (4)$$

gdzie:

- TWE – całkowity nakład pracy (ang. *Total Work Effort*)
- RDWT – rzeczywisty dzienny czas pracy programisty (and. *Real, Daily Working Time*)
- NoD – liczba deweloperów (ang. *Number of Developers*)

Wynikiem wyliczenia PT jest czas, wyrażony w miesiącach, niezbędny na wykonanie danego projektu przez nasz zespół IT. W ostatnim kroku należy zestawić obliczony PT z zakładanym czasem projektu obliczając tym samym ryzyko niedotrzymania terminu



(RoF – *Risk of Failure*) (5)

$$\text{RoF} = \frac{\text{PT}}{\text{Czas zakładany}} \quad (5)$$

Współczynnik RoF przyjmuje wartości z przedziału od 0 do nieskończoności jednak w rzeczywistości przekroczenie wartości 4 oznacza, że projekt będzie prowadzony nieefektywnie np. ze względu na zbyt dużą ilość programistów, którzy mogliby ten czas poświęcić na inny projekt. Poniżej przedstawiono interpretację współczynnika RoF:

- RoF < 1 – projekt obciążony dużym ryzykiem niedotrzymania terminu
- RoF < 1 ; 2) - projekt obciążony średnim ryzykiem niedotrzymania terminu
- RoF < 2 ; 4> - małe prawdopodobieństwo niedotrzymania terminu
- RoF > 4 – bardzo małe ryzyko niedotrzymania terminu, mało efektywne wykorzystanie zasobów.

Opisana metoda opiera się na wykorzystaniu obliczonych przez autorów wartości przypadających na jeden punkt funkcyjny w zależności od języka programowania. Zapewnia to lepsze wyniki oraz dopasowanie do charakterystyki organizacji niż w przypadku stosowania ogólnych wartości. Ponadto dostarcza wiele istotnych, z punktu widzenia decydentów w projekcie oraz organizacji, informacji. Pozwala bowiem na zwiarytowanie wytwarzanego systemu, określenie potencjalnego czasu trwania projektu, wielkość nakładów pracy oraz w efekcie odpowiedź na pytanie: czy mam odpowiednie zasoby i kompetencje aby zakończyć projekt z sukcesem. W kolejnym punkcie niniejszej publikacji przedstawiono otrzymane wyniki podczas przeprowadzania procesu weryfikacji przedstawionej metody.

4. Weryfikacja miar stanu organizacji informatycznej

Na potrzeby weryfikacji wyznaczonych przez autorów nakładów pracy na punkt funkcyjny oraz przedstawionej metody, wykorzystano bazę projektów ISBSG (Release 12 Februrary 2013). Wykorzystanie projektów zgromadzonych przez ISBSG pozwala na zweryfikowanie metody na podstawie informacji o rzeczywiście zrealizowanych projektach. Baza zawiera zbiór informacji na temat projektów informatycznych realizowanych na przestrzeni wielu lat, przy wykorzystaniu różnych metodyk zarządzania. Zestawienie wyników realnych (tj. przedstawionych w bazie ISBSG) z obliczonymi pozwala na weryfikację opracowanej metody. Autorzy założyli porównanie nakładów pracy wyrażonych w godzinach dla 1582 projektów wyznaczonych za pomocą opisanej metody z rzeczywiście poniesionym w projekcie nakładem pracy. Założono, że należy zweryfikować obliczone wartości przy błędzie szacowania na poziomie $\pm 0,2$ oraz $\pm 0,8$. Otrzymane wyniki przedstawiono w tabeli 2.

Tabela 2 zawiera zestawienie nakładów rzeczywistych tj. pobranych z bazy projektów ISBSG oraz szacowanych tj. obliczonych za pomocą przedstawionej metody, dla poszczególnych języków programowania (kolumna 1). Założono dwa poziomy błędów szacowania $\pm 0,2$ oraz $\pm 0,8$, tj. próg tolerancji przy zestawianiu nakładów rzeczywistych i szacunkowych. Projekty niedoszacowane oznaczają takie, których rzeczywisty nakład był większy niż wyliczony. Projekty w zakresie oznaczają takie, dla których nakłady rzeczywiste i szacowane są zbliżone, przeszacowane natomiast takie, dla których rzeczywiste nakłady były mniejsze niż wyliczone. Na podstawie danych zawartych w tabeli 2 można policzyć procent projektów, w których szacunkowy nakład pracy



wyrażony w godzinach mieścił się w odpowiednim przedziale akceptowanego błędu szacowania. Wyniki te przedstawiono w tabeli 3.

Tab. 2 Zestawienie nakładów szacowanych i rzeczywistych

| Język prog. | N ważnych | Odchylenie = $\pm 0,2$ | | | Odchylenie = $\pm 0,8$ | | |
|--------------|-----------|------------------------|------------|---------------|------------------------|------------|---------------|
| | | Niedoszacowane | W zakresie | Przeszacowane | Niedoszacowane | W zakresie | Przeszacowane |
| ASP | 26 | 0 | 6 | 17 | 0 | 17 | 6 |
| Visual Basic | 327 | 134 | 57 | 136 | 27 | 212 | 88 |
| Java | 682 | 256 | 142 | 284 | 16 | 527 | 139 |
| SQL | 135 | 37 | 55 | 43 | 3 | 97 | 35 |
| C++ | 159 | 53 | 42 | 63 | 10 | 98 | 50 |
| C | 191 | 55 | 68 | 68 | 10 | 128 | 53 |
| C# | 62 | 13 | 28 | 21 | 0 | 48 | 14 |

Tab. 3 Procent projektów, których szacunkowy czas mieścił się w założonym zakresie

| Język prog. | N ważnych | Odchylenie = $\pm 0,2$ | Odchylenie = $\pm 0,8$ |
|--------------|-----------|------------------------|------------------------|
| ASP | 26 | 23,08% | 65,38% |
| Visual Basic | 327 | 17,43% | 64,83% |
| Java | 682 | 20,82% | 77,27% |
| SQL | 135 | 40,74% | 71,85% |
| C++ | 159 | 26,42% | 61,64% |
| C | 191 | 35,60% | 67,02% |
| C# | 62 | 45,16% | 77,42% |

Przy założeniu błędu szacowania na poziomie $\pm 0,2$, średnio 30% projektów, zweryfikowanych za pomocą przedstawionej metody i wyznaczonych nakładów na punkt funkcyjny, mieściło się w zakresie akceptacji w stosunku do realnego nakładu. Odpowiednio dla obłędu wynoszącego $\pm 0,8$ procent ten wzrósł do 69. Kolejny punkt niniejszej publikacji stanowi rozważania autorów na temat poziomu wsparcia opisanej metody dla procesu MA a także w efekcie przydatność w procesie analizy zwinnej transformacji.

5. Adaptacja opracowanych miar w procesie zwinnej transformacji organizacji informatycznej

Stosowana metoda pozwala na obliczenie trzech podstawowych miar: nakładów pracy, czasu trwania projektu oraz ryzyka porażki. Cała metoda natomiast była projektowana tak, aby być zgodną z praktykantami i celami opisanymi w ramach procesu MA. W tabeli 4 przedstawiono zestawienie praktyk i celów oraz poziomu spełnienia ich poprzez zastosowanie opisanej metody.

W tabeli 4 przedstawiono, wsparcie których praktyk w ramach procesu MA jest możliwe dzięki opisanej metodzie. Umożliwia ona wsparcie we wszystkich 8 praktykach opisanych w ramach tego procesu. Ponadto dzięki zaproponowanej metodzie można oszacować, na podstawie danych historycznych, procent dobrze oszacowanych projektów w ramach organizacji. Wysoki poziom dobrze oszacowanych projektów (tj. takich, dla których nakłady szacunkowe i rzeczywiste były zbliżone) oznacza wyższy poziom uporządkowania projektów, dobre metody szacowania oraz poprawnie prowadzoną analizę.

Tab. 4 Poziom spełnienia praktyk w ramach procesu MA dzięki zaproponowanej metodzie

| Proces: MA – Measurement and Analysis | | |
|--|--------------------|-------------|
| Cel 1: Align Measurements and Analysis Activities | | |
| Praktyka | Stopień spełnienia | |
| | Spełnia | Nie spełnia |
| Ustal cele pomiarowe | X | |
| Określ miary | X | |
| Określ procedury zbierania i przechowywania danych | X | |
| Określ procedury analizy | X | |
| Cel 2: Provide Measurements Results | | |
| Uzyskaj dane pomiarowe | X | |
| Analizuj dane pomiarowe | X | |
| Magazynuj dane i wyniki | X | |
| Komunikuj wyniki | X | |

Może to wskazywać na, że proces MA jest na wysokim poziomie co z kolei może świadczyć o wyższym poziomie dojrzałości organizacji. Niski procent projektów w zakresie oznacza, że prowadzone przez nas szacunki są niedokładne a co za tym idzie procesy takie jak zarządzanie projektem i ryzykiem nie są zoptymalizowane, co wskazuje na niższy poziom dojrzałości organizacji. Na taki stan rzeczy wpływać może wiele elementów takich jak np. wysoki poziom innowacyjności projektu (wytworzenie czegoś pierwszy raz, brak podobnych projektów w przeszłości) przez co szacunki, szczególnie dotyczące złożoności systemu w FPA, mogą być obciążone dużym błędem szacowania. Taki stan rzeczy wymaga przeanalizowania zasadności stosowania miar opartych o FPA do szacowania stanu ewolucji organizacji. Niski procent projektów dobrze oszacowanych może stanowić jeden z argumentów w rozważaniach nad rozpoczęciem zwinnej transformacji, jednak nie może stanowić elementu świadczącym o gotowości organizacji do jej rozpoczęcia. Miary oparte o FPA mogą zatem być pomocne w szacowaniu projektów i zwiększaniu poziomu dojrzałości organizacji, jednak z punktu widzenia zwinnej transformacji nie pozwalają na określenie gotowości organizacji do rozpoczęcia zwinnej transformacji. Można jednak założyć, że zespoły wytwórcze wykorzystujące metody szacowania (np. opisaną przez autorów) będą do niej lepiej przygotowane niż inne. Wynika to z faktu, iż szacowanie jest istotnym elementem prac zespołów wytwarzających zwinnie - np. SCRUM i Story Points [12].

6. Analiza wyników

W artykule przedstawiono przykład oparty na metodzie szacowania złożoności projektów informatycznych przydatnych dla oceny złożoności wytwarzanego oprogramowania. Stosowanie takiej metody jest zgodne z oczekiwaniami standardów zarządzania procesowego, które (jak SCAMPI) zalecają ocenę organizacji w 22 obszarach, w tym w obszarze „mierzenia i analizowania” (MA). Przedstawiona metoda pozwala na szacowanie nakładów pracy, ram czasowych projektów informatycznych oraz ryzyka porażki. Dodatkowo podczas prac badawczych autorzy pokazali jak wykorzystać dostępne informacje o projekcie do stosowania dodatkowych, pomocniczych miar (jak nakłady pracy wyrażone w godzinach przypadające na punkt funkcyjny w zależności od wybranego języka programowania). Takie podejście zgodne jest z zaleceniem stosowania miar do oceny postępów w projektach oraz do doskonalenia tych projektów. Organizacje stosujące miary mają większe możliwości doskonalenia swojego funkcjonowania oraz są bardziej

świadome swoich procesów, co wydaje się kluczowe dla procesów zwinnej transformacji. Organizacje, które realizują projekty bez stosowania miar postępów, miar dojrzałości czy gotowości do realizacji projektów będą miały problemy ze stosowaniem zaleceń typowych dla technik zwinnych.

Przedstawiony przykład zastosowania metody oceny złożoności projektu w oparciu o punkty funkcyjne i opracowane w jej ramach miary oceny złożoności projektów informatycznych mogą być stosowane we wczesnej fazie projektu informatycznego i być pomocne w procesie decyzyjnym, w którym rozważa się podjęcie danego przedsięwzięcia. To też jest znaczące z punktu widzenia procesów zwinnej transformacji, gdzie oczekuje się od zespołów definiowania np. gotowości do realizacji zadań („definition of ready” w zespołach Scrumowych). Dzięki umiejętności szacowania zespoły projektowe mogą podnosić swoją dojrzałość a tym samym zwiększać szanse powodzenia procesów zwinnej transformacji.

Należy zaznaczyć, że w artykule starano się wskazać powiązanie między procesem MA oraz opisaną techniką szacowania według punktów funkcyjnym. Zweryfikowano przy tym, że opisana technika szacowania miar złożoności może być stosowana dla każdej z opisanych 8 praktyk dla obszaru procesowego MA w CMMI (SCAMPI). Opisany przykład pokazał także, że stosując metodę dla projektów historycznych można uzyskać procent projektów oszacowanych prawidłowo. To z kolei pozwala stwierdzić, że organizacje przygotowujące się do procesów zwinnej transformacji mogą dokonać także ocen ex-ante swoich projektów uzyskując miarę swojej dojrzałości. Mając wiedzę np. o niskim procencie projektów prawidłowo szacowanych może stanowić jeden z argumentów przy podejmowaniu decyzji o realizacji zwinnej transformacji. Taka miara może mieć dualny charakter – z jednej strony pokazywać, że skoro szacowania dokonujemy niewłaściwie, to stosowanie technik zwinnych może przebiegać niewłaściwie. Z drugiej strony może pokazywać, że organizacja musi podnosić swoje umiejętności analizy i mierzenia zanim przystąpi do zmian (ocena daje szansę na doskonalenie). To z kolei potwierdza w pewnym stopniu przyjętą we wstępie tezę o istotności procesów mierzenia i analiz dla weryfikacji gotowości organizacji informatycznych do rozpoczęcia zwinnej transformacji.

Należy zaznaczyć, że niniejszy artykuł wpisuje się w program badawczy całego zespołu nad zjawiskiem zwinnej transformacji i stanowi wstęp do badań prowadzonych przez autorów nad modelami oceny gotowości organizacji do realizacji procesów zwinnej transformacji. Pierwszy etap stanowiła próba weryfikacji możliwości wykorzystania miar projektów stosowanych w ramach procesu MA do oceny gotowości do rozpoczęcia zwinnej transformacji. Kolejnym etapem badań nad zwinną transformacją są: weryfikacja pozostałych procesów SCAMPI, weryfikacja czynników mających wpływ na zwinną transformację oraz finalnie budowa narzędzia do oceny organizacji pod kątem możliwości przejścia przez proces transformacji.

Literatura

1. Ziółkowski A., Deręgowski T., Hybrid approach in project management – mixing capability maturity model integration model with agile practices, Social Science(Socialiniai Mokslai), Nr. 3 (85), 2014
2. Beth Chrissis M.,Konrad M., Shrum S., CMMI for Development: guidelines for process integration and product improvement, Third Edition, Pearson Education, 2007



3. Kowalczyk Z., Orłowski C., Modelowanie procesów zarządzania technologiami informatycznymi, Pomorskie Wydawnictwo Naukowo – Techniczne, 2012
4. Ziółkowski A., Orłowski C., Wysocki W., Knowledge Management in the Processes Of Project Requirements Analysis, Studies & Proceedings of Polish Association for Knowledge Management, 106 – 120, 2012
5. Beth Chrissis M., Konrad M., Shrum S., CMMI for Development: guidelines for process integration and product improvement, Third Edition, Pearson Education, 2007, s. 289
6. CMMI Product Team, CMMI for Acquisition, Version 1.3, Carnegie Mellon University, 2013, s. 171
7. Longstreet D., Fundamentals of Function Point Analysis, Longstreet Consulting Ing., 2005
8. The International Function Point Users Group, Function Point Counting Practices Manual, Release 4.1.1., 2000
9. Mirecki J., Szacowanie wdrożeń, Computer World, 28.10.2008
10. Koteswara Rao K., Raju G.S.V.P., Madhusudhana Rao T.V., Effort Estimations Based on Lines of Code and Function Point in Software Project Management, International Journal of Computer Science and Network Security, VOL. 8 NO.6, June 2008
11. Opracowanie na podstawie bazy ISBSG Data, Release 12 February 2013
12. Kaczor K., Scrum i nie tylko, Wydawnictwo Naukowe PWN, 2014

Prof zw. dr hab. inż. Cezary ORŁOWSKI
 Dr inż. Artur ZIÓŁKOWSKI
 Instytut Zarządzania i Finansów
 Katedra Zastosowań Informatyki w Zarządzaniu
 Wyższa Szkoła Bankowa w Gdańsku
 Aleja Grunwaldzka 238A, 80-266 Gdańsk
 e-mail: corlowski@wsb.gda.pl
 aziolkowski@wsb.gda.pl

Mgr inż. Miłosz KURZAWSKI
 Mgr inż. Katarzyna OSSOWSKA
 Wydział Zarządzania i Ekonomii
 Politechnika Gdańska
 ul. Narutowicza 11/12
 80-233 Gdańsk
 e-mail: mkurzawski@zie.pg.gda.pl
 kossowska@zie.pg.gda.pl

Mgr inż. Tomasz DERĘGOWSKI
 Acxiom Corporation
 Aleja Grunwaldzka 186
 80- 266 Gdańsk
 e-mail: tomasz.deregowski@acxiom.com

