

## MOŻLIWOŚCI POPRAWY PROCESU WYTWARZANIA OPROGRAMOWANIA W MAŁYCH FIRMACH INFORMATYCZNYCH

Jerzy KACZMAREK<sup>1</sup>, Michał WRÓBEL<sup>2</sup>

1. Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska  
tel: (58) 347 26 82 fax: (58) 347 27 27 e-mail: jkacz@eti.pg.gda.pl
2. Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska  
tel: (58) 347 10 37 fax: (58) 347 27 27 e-mail: wrobel@eti.pg.gda.pl

**Streszczenie:** W artykule wykazano, że jest możliwa poprawa jakości procesu wytwarzania oprogramowania poprzez wdrożenie zaleceń standardu CMMI oraz wykorzystywanie doświadczeń z poprzednich udanych i udokumentowanych projektów. Wykazano znaczącą rolę szablonów dokumentacji w poprawie jakości procesu. Przeanalizowano praktyczne konsekwencje nierównomiernego zapotrzebowaniu na zasoby i zmiennej w czasie intensywności prac w cyklu życia oprogramowania. Zjawisko to, przy złym oszacowaniu czasu trwania projektu, może prowadzić do jego nierealizowalności. Omówiono wpływ wielkości zespołu na jego wydajność. Otrzymane wyniki pokazują, że poprawa procesu wytwarzania oprogramowania pomimo kosztów początkowych może być opłacalna. Dane zawarte w tym artykule mogą być interesujące dla projektantów i kierowników projektów informatycznych.

**Słowa kluczowe:** jakość oprogramowania, CMMI

### 1. WSTĘP

W dziedzinie informatyki znane jest obecnie zjawisko braku sukcesów projektów informatycznych. Przekraczany jest planowany czas realizacji projektów oraz budżet, a wiele projektów kończy się niepowodzeniem [1,2].

Problem jest szczególnie istotny dla małych firm, w których zapewnianie czy zarządzanie jakością procesu wytwarzania oprogramowania jest szczególnie utrudnione. Trudności wynikają z kosztów niezbędnych dla realizacji postawionych zadań jakościowych. W firmach średnich, koszt wdrożenia programów zapewniania jakości procesu produkcji może przekraczać 20% budżetu projektu, co jest wielkością nie do zaakceptowania. Problem nie polega na braku informacji co i jak należy wykonać, ponieważ są precyzyjnie opisane standardy dostępne dla wszystkich, ale na tym, że to kosztuje.

Z tych względów sukces w realizacji projektów informatycznych w małych firmach jest nieprzewidywalny a możliwości poprawy jakości procesu wytwarzania oprogramowania są ograniczone.

W artykule przedstawiono rezultaty wprowadzonego z sukcesem, standardu CMMI w przykładowej małej firmie informatycznej.

Z doświadczeń tego przypadku użycia standardu CMMI wynika, że jest możliwe znaczne podniesienie jakości procesu produkcji, nawet w bardzo małej firmie.

Niezbędne jest zwłaszcza zaangażowanie kierownictwa firmy i dobrze zorganizowany zespół do spraw zapewniania jakości, mający dużą wiedzę w dziedzinie inżynierii oprogramowania, znający standardy i modele zapewniania jakości a także sposoby ich wdrażania.

Najczęściej występujące problemy w realizacji projektów informatycznych polegają na złym szacowaniu wielkości projektu i czasu jego realizacji, na trudnościach w zarządzaniu i organizacji zespołów oraz na nierealizowalności zadań w niektórych momentach procesu wytwarzania, z uwagi na niejednorodność zapotrzebowania na zasoby. Co ciekawe, te problemy, które występują często w wielu firmach, są dobrze poznane i szczegółowo opisane w literaturze.

Szacowanie wielkości projektu to problem trudny, ale współczesne metody pozwalają na uzyskiwanie stosunkowo dobrych wyników, o ile nie następują zmiany w specyfikacji wymagań, co niestety ma często miejsce. Szacowanie czasu trwania projektu, to zagadnienie bardzo złożone, zarówno z teoretycznego, jak i praktycznego punktu widzenia, ale możliwe do rozwiązania w pewnym przedziale dokładności.

Problem z organizacją zespołów polega na zachowywaniu optymalnej ich wielkości, ponieważ wydajność zespołu spada przy zbyt dużej liczbie członków.

Znany problem nierównomiernego zapotrzebowaniu na zasoby i zmiennej w czasie intensywności prac wynika ze specyfiki procesu wytwarzania oprogramowania. Dlatego w artykule opisano jeden z modeli wyjaśniający to zjawisko. Wnioski z tego modelu są poważne, ponieważ wynika z nich, że w pewnych okolicznościach, intensywność prac zmierzać może teoretycznie do nieskończoności, co praktycznie oznacza, że projekt nie może być wykonany w przewidzianym czasie nawet przy nieograniczonych zasobach. Zjawisko to jest szczególnie istotne dla małych firm, w których nie ma możliwości przenoszenia pracowników pomiędzy projektami.

Informacje zawarte w tym artykule mogą być zatem interesujące i praktycznie przydatne dla projektantów i kierowników projektów informatycznych.

## 2. MOŻLIWOŚCI POPRAWY PROCESU WYTWARZANIA OPROGRAMOWANIA

Występujące często problemy z jakością produktów informatycznych doprowadziły do opracowania modeli, standardów i zaleceń, jak należy organizować i kontrolować proces produkcji oprogramowania. Powstało wiele takich standardów o różnym stopniu szczegółowości i skuteczności.

Wśród najważniejszych można wyróżnić model CMMI, SPICE czy zalecenia do normy ISO 9000-3 [3,4]. Cechą tych zaleceń jest ich wysoki poziom profesjonalizmu, szczegółowość i ogólna dostępność. W dokumentacji modeli są zawarte szczegółowe zalecenia z podaniem, co należy zrobić i jak.

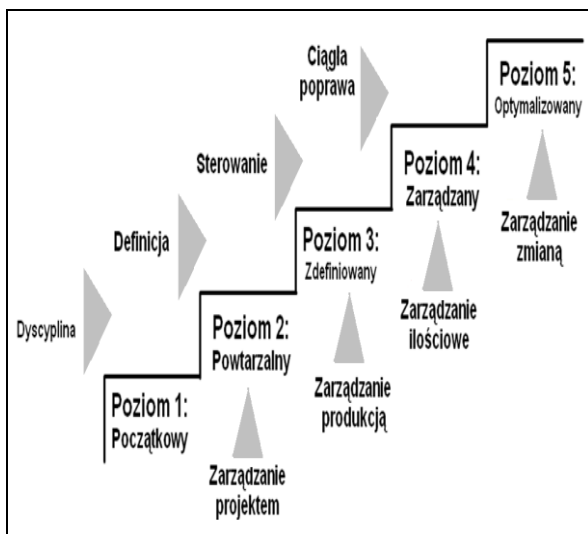
Wdrożenie takiego modelu nie jest jednak proste. Należy wybrać niektóre elementy w zależności od przyjętych priorytetów istotnych dla danej firmy oraz rozwiązać problem nadmiaru danych zawartych w dokumentacji. Oczywiście najważniejszą rzeczą są decyzje o ilości osób zaangażowanych w poprawę jakości procesu oraz oszacowanie kosztu wykonania takiego zadania.

Generalnie poziom dojrzałości procesu wytwarzania oprogramowania charakteryzuje zasada, na jakiej jest oparty. Można wyróżnić poziomy:

- sprawdzanie - testowanie,
- sterowanie - napisz co robisz i rób to co napisałeś,
- zapewnianie - udowodnij, że zrobiłeś to co napisałeś,
- zarządzanie - to co robisz rób coraz lepiej.

Do często wykorzystywanych i stosowanych modeli należy zaliczyć model CMMI (ang. Capability Maturity Model Integration). Wersję Version 1.2 zaprezentowano w 2006 roku.

Na rysunku 1. przedstawiono poziomy dojrzałości i zasady modelu CMMI. Model składa się z pięciu poziomów: początkowego, powtarzalnego, zdefiniowanego, zarządzanego i optymalizowanego. Interesujący z punktu widzenia tego artykułu jest problem przejścia w małej firmie informatycznej z poziomu początkowego zwanego poziomem chaosu na poziom 2 zwany poziomem powtarzalnym.



Rys. 1. Poziomy dojrzałości modelu CMMI

Poziom 1 cechuje nieprzewidywalność jakości wytwarzanego produktu z uwagi na dominującą rolę testowania oraz personalizacja, silne uzależnienie od indywidualnych możliwości kierownika projektu i członków zespołu.

Poziom 2 oparty o zasadę napisz – co robisz i rób to co napisałeś – wykorzystuje informacje z udanych, uprzednio wykonanych projektów, które zostały szczegółowo udokumentowane w swojej wersji ostatecznej. Poziom ten cechuje przewidywalność, zdyscyplinowanie, istnienie procedur wykonywania czynności, obowiązywanie szablonów dokumentacji oraz zdolność do poprawy.

## 3. PRZYPADK UŻYCIA MODELU CMMI

Postanowiono w małej firmie informatycznej podnieść poziom jakości procesu wytwarzania oprogramowania jako wewnętrzne zadanie firmy, które jednak nie miało prowadzić do uzyskiwania formalnych certyfikatów. Założono, że poprawa będzie odbywać się etapowo, a nie jednorazowo. Niektóre elementy wdrożone w firmie wykonano w oparciu o sugestie autorów artykułu, a szacowanie wielkości produktu dokonano metodą opublikowaną w artykule [5]. Doświadczenia z tego przypadku użycia modelu CMMI mogą być interesujące również dla innych firm informatycznych.

### 3.1. Charakterystyka firmy

Firma jest mała, zatrudnia 16 osób, wytwarza produkty dla stałego dobrze zdefiniowanego segmentu rynku. Cechą tych produktów jest duża ilość odbiorców liczona w tysiącach osób i instytucji. Produkty mają bardzo dobrze rozbudowane, multimedialne interfejsy. Duża ilość odbiorców i cechy produktów generują problemy ze specyfikacją wymagań dla nowych wersji. Firma wytwarza oprogramowanie w klasycznym modelu kaskadowym, projektując systemy obiektowo.

W firmie wykorzystywane jest narzędzie do zarządzania, w tym diagramy Gantt, i identyfikowana jest ścieżka krytyczna. Ponieważ wytwarzane produkty mają podobne cechy, jest szczególnie uzasadnione wykorzystanie zasady drugiego poziomu CMMI, która polega na powtórzeniu udanych uprzednio projektów i korzystaniu z dotychczasowych doświadczeń.

Najważniejszą sprawą w podnoszeniu jakości procesu wytwarzania na każdym poziomie CMMI jest zaangażowanie najwyższych władz firmy. Bez decyzji na najwyższych szczeblach zarządzania poprawa jakości procesu wytwarzania nie powiedzie się.

W omawianym przykładzie za proces poprawy odpowiedzialna była osoba o wykształceniu informatycznym, specjalności Inżynieria Oprogramowania, która jest jednocześnie współwłaścicielem firmy. Taka sytuacja była niezwykle korzystna dla procesu poprawy.

### 3.2. Wybór obszarów poprawy

Model CMMI zakłada konieczność poprawy w 22 obszarach procesu wytwarzania oprogramowania. Wiadomo, że wykonanie wszystkich zaleceń jest kosztowne i często niewykonalne. Jedną z pierwszych i trudnych decyzji jest wybór obszarów poprawy.

Za najważniejsze i pierwszoplanowe zadanie uznano poprawienie procesu w 7 obszarach:

- Planowanie Projektu
- Zarządzanie Wymaganiami
- Zarządzanie Konfiguracją
- Pomiar i Analiza Jakości
- Monitoring i Kontrola Projektu
- Zapewnianie Jakości Procesu i Produktu
- Zarządzanie Współpracą z Dostawcą

W każdym z tych obszarów wybrano dwa cele (ang. Specific Goal) i precyzyjnie w punktach określono elementy (ang. Specific Practice), które należy wykonać, aby cel osiągnąć. Określono również praktyki ogólne, zwłaszcza dotyczące dokumentacji, które dotyczą wszystkich obszarów.

### 3.3. Rola dokumentacji projektowej

Jednym z ważniejszych wniosków, jaki wynika z tego przypadku użycia standardu CMMI, jest znaczenie szablonów dokumentacji projektowej. Można postawić tezę, że sprawą najważniejszą w firmie o nieustabilizowanych dotąd procedurach jest posiadanie procedur do wykonywania dokumentacji.

Zastosowano ciekawe i oryginalne rozwiązanie. Wykonano szczegółową, zgodną z zasadami inżynierii oprogramowania dokumentację projektową końcowego produktu już wyprodukowanego, który był wykonywany na poziomie początkowym bez szablonów dokumentacji. Takie podejście pozwoliło na identyfikację błędów i nieprawidłowości w procesie wytwarzania oraz doprowadziło do uzyskania poprawnych szablonów dla większości etapów wytwarzania, możliwych do wykorzystania w przyszłych projektach.

Wykonano następujące szablony:

- Statut projektu
- Specyfikacja wymagań systemowych
- Plan projektu
- Szacowanie ryzyka
- Projekt systemu
- Specyfikacja testów
- Tygodniowy raport finansowy
- Tygodniowy raport projektu
- Raport oceny jakości

Spośród zidentyfikowanych i przeznaczonych do poprawy elementów procesu zarządzania projektami, można wymienić konieczność identyfikacji zagrożeń, kontrolę i wręcz wymuszanie wykonywania dokumentacji zgodnie z przyjętymi szablonami, wykorzystywania i raportowania czasu pracy, oraz przeprowadzania okresowej kontroli na tak zwanych liniach bazowych. Z uwagi na specyfikę produktów wytwarzanych przez firmę, ważnym elementem było też zbieranie wymagań od licznych użytkowników dla planowanych kolejnych wersji produktów oraz przydzielanie odpowiedzialności konkretnym pracownikom przy wykonywaniu zadań.

Jest interesujące, że docenianie znaczenia procedur, szablonów dokumentacji projektowej i kontroli procesu produkcji następuje po pewnym czasie, po kilku latach istnienia firm.

### 3.4. Planowanie projektu

Planowanie projektu to jedno z najtrudniejszych zagadnień w Inżynierii Oprogramowania. Wynika ono z niepowtarzalności projektów, nawet w projektach o

podobnej specyfikacji trudno określić złożoność produktu w początkowej fazie wytwarzania. W firmie zastosowano planowanie optymistyczne, pesymistyczne i optymalne.

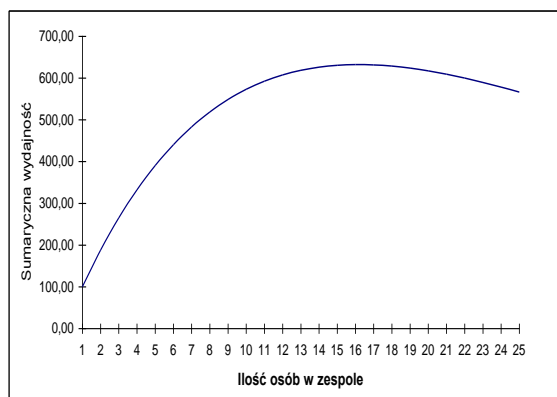
Z przeprowadzonej analizy popełnionych błędów wynikało, że należy dokonywać modyfikacji planów w dwóch momentach procesu wytwarzania, w oparciu o diagram przypadków użycia i po zaprojektowaniu diagramu klas.

Wykorzystując model COCOMO II oszacowano, że wykonany produkt powinien mieć 5000 wierszy kodu, a nie zawierać 20 000 wierszy i trwać 14 miesięcy, a nie trzy lata jak to miało miejsce [6]. Otrzymane wyniki pokazują, że poprawa jakości procesu wytwarzania oprogramowania pomimo kosztów początkowych może być opłacalna dla firmy.

Z omawianego przypadku użycia modelu CMMI w analizowanej firmie można wyróżnić ponadto dwa dodatkowe problemy. Jednym z nich jest konieczność podziału firmy na zespoły o optymalnej wielkości, a drugi problem to nierównomierna intensywność prac i zmienne w czasie zapotrzebowania na zasoby w trakcie trwania projektu.

### 3.5. Optymalna wielkość zespołu

Znane jest zjawisko spadku wydajności zespołu w zależności od jego wielkości, które wynika z konieczności komunikacji wewnątrz zespołu. Na rysunku 2. przedstawiono funkcję ilustrującą to zjawisko.



Rys. 2. Sumaryczna wydajność zespołu

Można zauważyć, że zespoły liczące powyżej 10 osób mają coraz mniejszą wydajność. W omawianej firmie dokonano optymalizacji wielkości zespołów, tak by liczyły nie więcej niż siedem osób, a kierownik zespołu odpowiadał przed dwoma członkami zarządu.

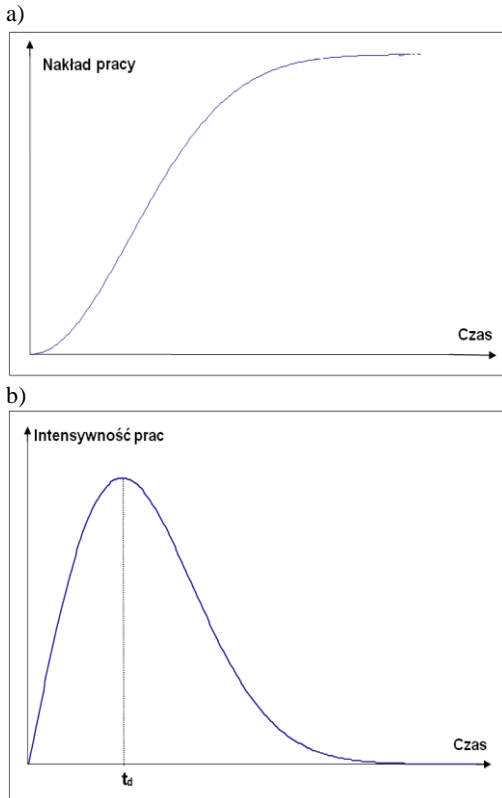
### 3.6. Intensywność prac projektowych

Podstawowym celem, dla którego postanowiono wdrożyć w firmie standard CMMI było osiągnięcia stanu, gdy projekty kończyć się będą w zaplanowanym terminie, trwać krócej i wykonywane będą sprawniej. Osiągnięcie takiego stanu w małej firmie nie jest rzeczą łatwą.

Szacowanie czasu realizacji projektu jest możliwe, na przykład metodą COCOMO, jak to zrobiono, ale błąd oszacowań przekraczał 150%. Specyfiką procesów produkcji oprogramowania jest duża niejednorodność intensywności prac w czasie cyklu życia oprogramowania [7].

Zjawisko to dotyczy poszczególnych faz wytwarzania oprogramowania jak również całego procesu. Istnieje teza, która głosi, że nie można wykonać złożonego produktu w dowolnie krótkim czasie, nawet przy nieograniczonych zasobach.

Na rysunku 3a. przedstawiono szkic funkcji opisującej znormalizowany przyrost pracochłonności w czasie trwania procesu produkcji.



Rys. 3. Zmienność intensywności prac projektowych

Na rysunku 3b. przedstawiono pochodną funkcji nakładu pracy. Można zauważyć, że istnieje w procesie wytwarzania produktu moment czasu  $t_d$ , w którym dochodzi do spiętrzenia prac. Jeżeli planowany czas wykonania projektu został źle obliczony, to chwilowy nakład pracy nad projektem może osiągnąć bardzo duże wartości.

Znajomość istnienia zjawiska niejednorodnego zapotrzebowania na zasoby nie jest powszechna, co może być jedną z przyczyn niepowodzeń wielu projektów informatycznych.

W małych firmach nie ma możliwości przenoszenia pracowników z innych projektów do projektu, w którym w danym momencie występuje spiętrzenie prac. Gwarancją sukcesów projektów informatycznych jest zatem poprawne szacowanie złożoności systemu i obliczanie realnego czasu trwania projektu według znanych w inżynierii oprogramowania metod [8].

#### 4. WNIOSKI KOŃCOWE

Przykłady przypadków użycia modeli, standardów i procedur są często wykorzystywane w informatyce dla ilustracji ich przydatności i skuteczności.

Jak wykazano, wdrożenie w małej firmie informatycznej, bardzo dobrze opisanego i często stosowanego modelu CMMI jest możliwe i nie wymaga nadzwyczajnych działań, zwłaszcza dużych zmian w organizacji firmy. Najważniejszą rzeczą jest jednak zaangażowanie władz firmy i wiedza z dziedziny inżynierii oprogramowania na najwyższych szczeblach zarządzania. Wykazano, znaczącą rolę szablonów dokumentacji projektowej, wymuszanie ich stosowania oraz korzystanie z informacji o poprawnie wykonanych uprzednio projektach. Na podkreślenie zasługuje też fakt, że nie zawsze uwzględniany jest w zarządzaniu projektami informatycznymi, problem wielkości zespołów oraz niejednorodności intensywności prac w trakcie trwania projektu.

Opisany w artykule przypadek wdrożenia modelu CMMI w małej firmie informatycznej, wykorzystany do poprawy jakości procesu wytwarzania i przejścia z poziomu chaosu na poziom powtarzalny pokazuje, że jest to możliwe i przynosi wymierne korzyści.

#### 5. BIBLIOGRAFIA

1. Sommerville I.: Software Engineering, Wesley 1992.
2. Górski J.: Inżynieria Oprogramowania, MIKOM 1999.
3. CMMI for Development, Version 1.2, 2006.
4. PN ISO 9000- 3:1994.
5. Kaczmarek J., Kucharski M.: Size and effort estimation for applications written in Java, Information and Software Technology, Elsevier 2004
6. COCOMO II Model Definition, University of Southern California 1998.
7. Putnam L.H., W. Myers W.: Measures for Excellence: Yourdon Press 1992.
8. Boehm B.: Software Engineering Economics, Prentice Hall 1981, ISBN 978-0138221225

## IMPROVEMENT AREAS OF SOFTWARE DEVELOPMENT PROCESS IN SMALL IT COMPANIES

**Key-words:** software quality, CMMI

The paper shows that the software development process may be improved by implementation of CMMI standard recommendations and application of experiences from successfully realized and documented previous projects. Significant importance of documentation templates in the improvement process was shown. Practical consequences of irregular resource demand and time-dependant workload were analyzed in the context of software development lifecycle. The occurrence of irregularities may lead to project infeasibility if development time requirements are calculated incorrectly. Additionally, the dependence between team size and team efficiency was analyzed. The results show that improvement of the software development process may be profitable though it requires initial costs. Presented data may be of interest for software designers and software project leaders.