Spiralne podejście do wytwarzania oprogramowania

Zdzisław Szyjewski
Uniwersytet Szczeciński
Instytut Informatyki w Zarządzaniu
zszyjew@uoo.univ.szczecin.pl

Abstrakt

Proces wytwarzania oprogramowania może być zorganizowany na wiele różnych sposobów. Tradycyjnie stosowane podejście kaskadowe jest coraz częściej zastępowane innymi, szybszymi metodami dochodzenia do rozwiązania. W pracy przedstawiono założenia podejścia spiralnego, wykorzystywanego w firmie Microsoft przy wytwarzaniu oprogramowania.

Wprowadzenie

Wytwarzanie oprogramowania jest jednym z etapów tworzenia systemu informatycznego. W cyklu życia systemu jest to etap, na którym szczególnie koncentruje się uwaga kierownictwa projektu oraz wymagane jest duże zaangażowanie całego zespołu wykonawczego[10]. W procesie realizacji tego etapu prac muszą być usunięte wszelkie usterki i niedoróbki wcześniejszych faz wykonawczych. Programista piszący kod programu musi ostatecznie, jednoznacznie zdecydować o rozwiązaniach systemu informatycznego.

Pracochłonność i skomplikowany charakter prac programistycznych powoduje, że bardzo często ten etap procesu tworzenia systemu informatycznego jest uznawany za przyczynę opóźnień i wzrost kosztów wytwarzania całego systemu informatycznego. Konieczność uzupełnienia wcześniej niekompletnie wykonanych prac, uwarunkowania terminowe wielu wcześniejszych zadań projektowych, powodują, że na etapie wytwarzania oprogramowania problemy eskalują się i często doprowadzają do powstania opóźnień i zwiększonych nakładów na system[1].

Inżynieria oprogramowania jest szeroko rozumianym obszarem badań, których celem jest doskonalenie tej fazy wytwarzania systemu informatycznego[4]. Jednym z obszarów zainteresowań badawczych jest organizacja procesu wytwarzania oprogramowania. Strukturalne metody wytwarzania oprogramowania w połączeniu z kaskadowym układem zadań realizacyjnych w wielu przypadkach okazały się niezadowalające. Prostota tego podejścia i łatwość nadzoru nad realizacją nie rekompensowały problemów wynikających z długotrwałości i małej elastyczności takiej organizacji prac. Jednym z kierunków badań były próby przyspieszenia dochodzenia do rozwiązania, wykorzystując zalety podejścia kaskadowego ale w krótszym czasie. Podział całego procesu wytwarzania oprogramowania na cykle uszczegóławiane iteracyjnie doprowadził do metody spiralnej[2].

Podejście spiralne wytwarzania oprogramowania znalazło zastosowanie w firmie Microsoft, która taką organizację prac programistycznych przyjęła jako obowiązującą. Iteracyjne dochodzenie do rozwiązań końcowych jest podstawą całego procesu tworzenia oprogramowania i rozwiązywania wszystkich problemów powstających w trakcie procesu realizacji. Podejście to uzyskało nazwę MSF (Microsoft Solution Framework) i jest obowiązującą w tej firmie metodyką wytwarzania oprogramowania [3,5].

Model spiralnego dochodzenia do rozwiązania docelowego zakłada cykliczne powtarzanie pewnej sekwencji działań. Idea krokowego dochodzenia do rozwiązania

docelowego jest realizowana poprzez cykliczne wykonywanie tych samych faz projektu. Najczęściej dotyczy to dużych i złożonych przedsięwzięć, dlatego też model ten ma liczne zastosowania w przemyśle lotniczym oraz wytwarzaniu dużych przedsięwzięć informatycznych. Przy krokowym dochodzeniu do rozwiązania docelowego można oceniać efekty prac projektowych w sposób etapowy oraz wprowadzać ewentualne modyfikacje zgodnie ze zmianami celów biznesowych działania organizacji.

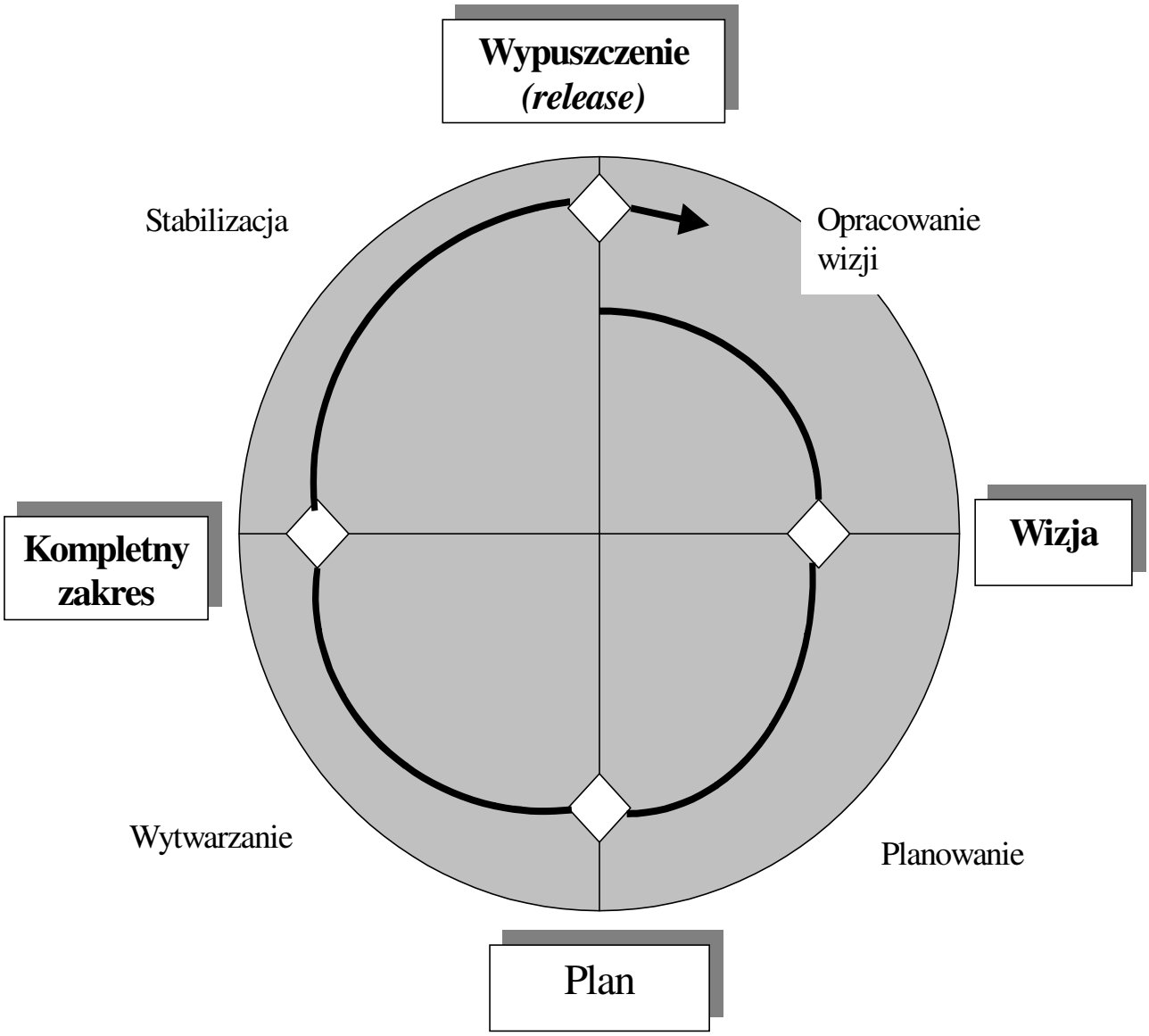
W pierwszej fazie planuje się działania realizacyjne, czyli wyznacza się zakres wprowadzanej zmiany oraz zakres pierwszego etapu realizacji. Następnie ocenia się ryzyko podejmowanego przedsięwzięcia i na podstawie tej oceny modyfikuje się plan. Zmodyfikowany plan poddawany jest realizacji. Najczęściej faza konstrukcji jest realizowana zgodnie z modelem kaskadowym. Zakończenie etapu realizacji stanowi konkretny produkt, który może być poddany ocenie według kryterium użyteczności praktycznej. Etap testowania daje materiał do modyfikacji i określenia zakresu następnego przybliżenia w krokowej realizacji celów projektu. Ponownie realizowana jest faza planowania i spiralnie dochodzimy do wyznaczonych celów projektu.

!. Założenia podejścia spiralnego

Wytwarzanie oprogramowania najczęściej realizowane jest na potrzeby konkretnej specyfikacji wymagań określonej we wcześniejszych etapach prac nad systemem informatycznym. W przypadku dedykowanych systemów informatycznych jest to wynik badań oczekiwań użytkownika i szczegółowych prac projektowych. Specyfiką oprogramowania systemowego oraz uniwersalnego[1] jest brak jednoznacznie zdefiniowanego użytkownika, który mógłby stanowić dla autorów źródło wiedzy o oczekiwaniach. Oprogramowanie takiego typu ma kreować oczekiwania i rolą autorów jest definiowanie funkcjonalności z przekonaniem, że zostanie ona zaakceptowana przez dużą grupę przyszłych, potencjalnych użytkowników tworzonego oprogramowania. Trudne warunki konkurencji rynkowej narzucają nie tylko wysokie wymagania jakościowe, ale również stanowią poważne ryzyko nietrafnego zidentyfikowania oczekiwań użytkownika.

Podstawą projektów realizowanych przez firmę Microsoft jest wytwarzanie oprogramowania systemowego i uniwersalnego. Narzuca to istotne uwarunkowania realizacyjne. Czas powstania takiego oprogramowania nie jest limitowany umową z klientem, ale ponoszone koszty wytwarzania oprogramowania zaczną się zwracać dopiero po akceptacji przez klientów i pierwszych sprzedanych egzemplarzach wytworzonego produktu programowego. Innym istotnym uwarunkowaniem jest prawidłowe zidentyfikowanie oczekiwań potencjalnych klientów i zagwarantowanie akceptowalnej przez rynek funkcjonalności wytwarzanego oprogramowania. Wynika z tego potrzeba maksymalnego skracania czasu wytwarzania oprogramowania i konfrontacja oczekiwań klientów z wizją autorów oprogramowania. Spiralne podejście do procesu wytwarzania oprogramowania jest odpowiedzią na te wyzwania. Szybkie wytworzenie prototypu, mającego cechy użyteczności, pozwala na sprawdzenie zainteresowania wytworzonym produktem i zebranie uwag od klientów, tak aby kolejna wersja oprogramowania spełniała oczekiwania klientów w większym stopniu. Cykliczne powtarzanie takiej procedury pozwala na etapowe dochodzenie

[1] Pod pojęciem oprogramowania systemowego rozumiem systemy operacyjne i inne programy gwarantujące funkcjonowanie zestawu komputerowego. Pod pojęciem oprogramowania uniwersalnego rozumiem oprogramowanie, które nie jest tworzone na potrzeby rozwiązania konkretnego zdefiniowanego problemu ale ma charakter otwarty i jest wykorzystywane do realizacji pewnych grup zadań. Przykładem takiego oprogramowania są edytory, arkusze kalkulacyjne itp.



Rys. Model Microsoft Solutions Framework (MSF) (źródło Internet)

do docelowej wersji oprogramowania przy minimalizacji ponoszonych nakładów i zminimalizowanym ryzykiem nietrafnych decyzji.

Firma Microsoft wypracowała podejście całościowe, które bazując na spiralnym, cyklicznym dochodzeniu do rozwiązania uwzględnia wszelkie prace w procesie wytwarzania oprogramowania. Podejście nazwane Microsoft Solution Framework (MSF) zawiera zasady organizacji prac i kierowania procesem wytwarzania oprogramowania. Podstawą tego podejścia jest konsekwentne, spiralne dochodzenie do rozwiązania oraz wiele oryginalnych

rozwiązań organizacyjnych gwarantujących dużą dynamikę procesu realizacji, co powinno skutkować skróceniem czasu wytwarzania. Liczne punkty kontrolne oraz poddawanie kolejnych wersji oprogramowania weryfikacji użytkowej przez klientów ma gwarantować trafności przyjmowanych rozwiązań.

Decyzja o wypuszczeniu na rynek produktu, który z założenia nie jest doskonały, jest dość ryzykowna z punktu widzenia wizerunku firmy, ale okazała się skuteczna biznesowo. Użytkownicy produktów programowych zostali przyzwyczajeni do niedoskonałości dostarczanego oprogramowania, które są usuwane w kolejnych wersjach. Spiralne dochodzenie do rozwiązania posiada liczne zalety, które pozwalają na lepsze dopasowanie wytwarzanego oprogramowania do oczekiwań użytkownika, ale stanowią wyzwanie dla autorów. Wytworzenie kolejnej wersji oprogramowania powinno być wystarczająco szybkie i powinno realizować większość zauważonych usterek i zgłoszonych modyfikacji. To zmusza zespół wykonawczy do wysokiej dyscypliny tworzenia i odpowiedniego oprzyrządowania procesu wytwarzania oprogramowania.

2. Fazy realizacji MSF

Podejście firmy Microsoft z założenia wykorzystuje spiralne rozwiązanie problemu. W kolejnych fazach realizacji wykonywane są cyklicznie grupy działań, które doprowadzają do zadowalającego rozwiązania. Graficznie model tego podejścia obrazuje rysunek. Podział na cztery cyklicznie powtarzane etapy prowadzi od prac koncepcyjnych do wyprodukowania kolejnej wersji wypuszczanego na rynek oprogramowania.

Etapy na jakie dzielony jest jeden cykl wytwarzania oprogramowania według podejścia MSF składa się z:

- opracowania wizji produktu,
- planowania,
- wytwarzania,
- stabilizacji.

Każdy z tych etapów ma ściśle określony podział na dalsze punkty węzłowe i kończy się wytworzeniem produktu, który jest podstawą dalszych prac. Produktami tymi są odpowiednio: wizja wytworzonego oprogramowania, plan prac w danym cyklu wykonawczym, realizacja oprogramowania, testowanie i stabilizacja oprogramowania.

Pierwszy etap prac ma na celu określenie wizji przyszłego produktu programowego. Na tym etapie prac wyznaczone są następujące punkty węzłowe:

- ✓ sformułowanie trzonu zespołu wykonawczego,
- ✓ opracowanie pierwszej wersji dokumentu opisującego wizję produktu,
- ✓ pierwsza wersja analizy ryzyka przedsięwzięcia.

Produktem końcowym tego etapu prac jest opracowana wizja produktu programowego, który ma być wytworzony w trakcie dalszych prac. Wizja ta zawiera informacje na temat funkcjonalności oprogramowania, głównych celów i założeń realizacyjnych. Sformułowany jest również trzon zespołu wykonawczego oraz pierwsza analiza głównych zagrożeń i ocena ryzyka całego przedsięwzięcia.

Następny etap prac to faza planowania, na którą składają się:

- ✓ pierwsza wersja specyfikacji funkcjonalnej produktu,
- ✓ pierwsza wersja ogólnego planu realizacji projektu,
- ✓ pierwsza wersja ogólnego harmonogramu realizacji.

Produktem końcowym tej fazy prac jest zatwierdzony plan realizacji projektu. W trakcie tego etapu sformułowana zostanie pierwsza wersja funkcjonalności wytwarzanego oprogramowania, ogólny plan realizacji i ogólny harmonogram prac nad wytwarzanym oprogramowaniem.

Zatwierdzony plan realizacji jest podstawą kolejnego etapu prac, który ma za zadanie wykonać zaplanowane zadania realizacyjne. Realizowane są kolejne wersje oprogramowania aż do otrzymania wersji realizującej założoną funkcjonalność. Na tym etapie prac wykorzystywana jest cykliczność w realizacji tych samych prac, czyli spiralne dochodzenie do celu. W kolejnych cyklach powtarzanych konsekwentnie w ramach tego etapu prac opracowywane są kolejne wersje wewnętrzne aż do osiągnięcia zgodności z założoną w planie funkcjonalnością. Pojedynczy cykl realizacyjny może mieć wymiar jednego dnia roboczego[2]. Standardowy układ dnia roboczego dla całego zespołu wykonawczego narzuca pewien styl pracy oraz wymusza dyscyplinę realizacji zadań. Zespołowa realizacja oprogramowania, gdzie istnieją liczne związki wewnętrzne zmusza do zdyscyplinowanego działania całego zespołu. W wyniku tego etapu prac powstaje oprogramowanie, które realizuje pełen zakres funkcjonalności założonej planem realizacji.

Ostatnia faza realizowana w jednym cyklu wykonawczym, ma na celu stabilizacje wytworzonego produktu, czyli wytworzenie oprogramowania, które może być udostępnione użytkownikowi [8]. Składają się na to według nazewnictwa MSF:

- ✓ wersja „beta”,
- ✓ wersja „zero-błędów”,
- ✓ wersja „kandydat”,
- ✓ wersja „złota”.

Kolejne wersje stanowią kamienie milowe na drodze dojścia do produktu, który może stanowić wypuszczoną na rynek wersje oprogramowania. Wypuszczenie wersji nie oznacza zakończenia prac a jedynie zamknięcie jednego cyklu wytwarzania i bezpośrednio po tym następuje kolejny cykl według takiego samego układu realizacyjnego. Prace w kolejnych cyklach wykonawczych za podstawę mają produkt powstały w poprzednim cyklu.

Taki sposób wytwarzania ma kilka charakterystycznych elementów realizacyjnych. Standardowy układ kolejnych faz wytwarzania powoduje standaryzację prac i możliwość wdrożenia pewnych ujednoliconych zasad organizacji pracy. Podział na określone punkty węzłowe i określoność produktów będących wynikiem realizacji tych punktów węzłowych nie tylko standaryzuje cały proces wytwarzania, ale z uwagi na liczne punkty, nadaje dynamikę procesowi realizacji. Dynamika procesu realizacji jest jednym z podstawowych założeń podejścia MSF. Czas na poszczególne fazy realizacyjne i czas pomiędzy punktami węzłowymi jest ograniczony, tak, aby zachować dynamikę wytwarzania.

Zaleca się, aby jeden cykl wytwarzania nie trwał dłużej niż 6 miesięcy a kolejne fazy odpowiednio 2-3 tygodnie, 1,5 miesiąca, 2 miesiące i 2 miesiące. Zalecenia te zmuszają zespół wykonawczy do pełnego zaangażowania i dużej dynamiki procesu realizacji. W warunkach, gdy nie ma niecierpliwie oczekującego na produkt użytkownika jest to bardzo ważne posunięcie organizacyjne. Dynamika całego projektu pozwala również na uniknięcie zmian wynikających ze zmieniającego się otoczenia. Okres 6 miesięcy gwarantuje, ze zmiany otoczenia nie będą tak duże, że ich oddziaływanie na wytwarzany produkt nie zakłócą procesu wykonania. Zachodzące zmiany w otoczeniu nie powinny być ignorowane, ale powinny być uwzględnione w kolejnym cyklu wytwarzania.

3. Kierowanie projektem

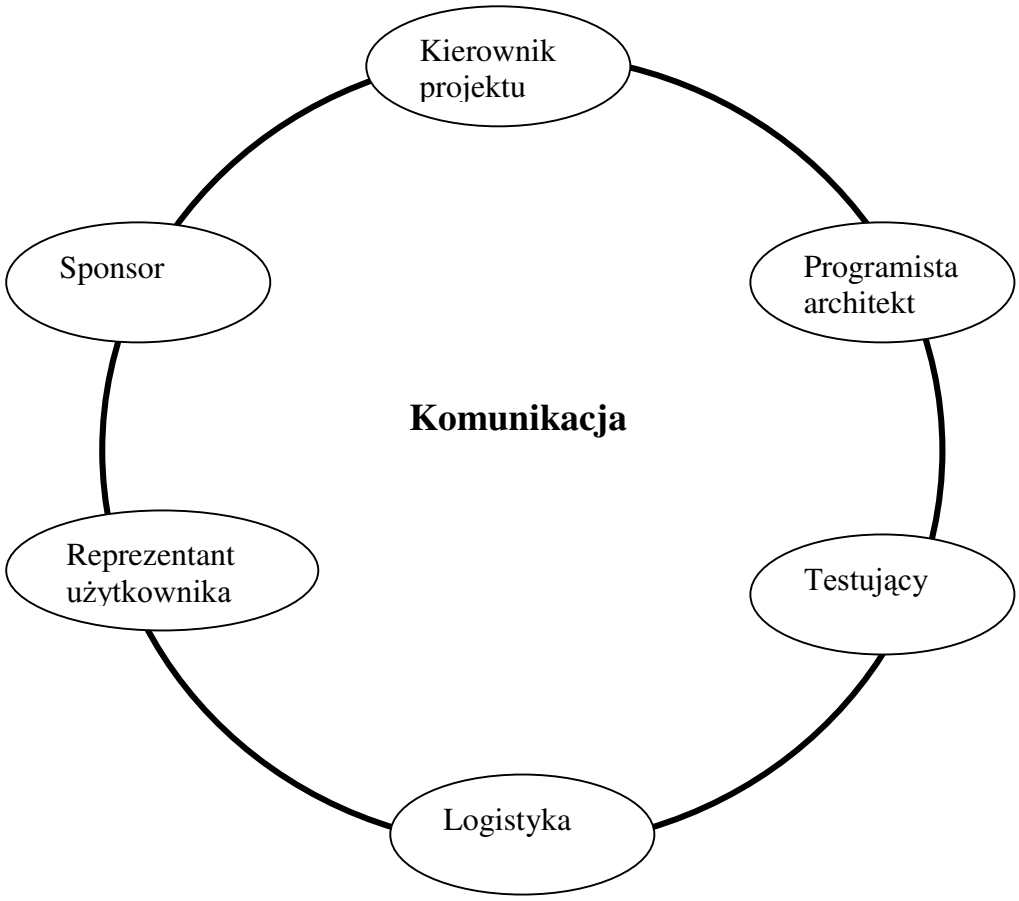
Istotą realizacji projektu programowego według podejścia MSF jest zagwarantowanie dynamiki realizacji. Dynamika ta osiągana jest poprzez odpowiednio gęste rozmieszczenie punktów węzłowych oraz standardy postępowania przez cały współpracujący zespół

[2] Przez dzień roboczy można rozumieć dobę z uwagi na to, że pewne prace wykonywane są bez udziału pracowników. Procesy kompilacji i wykonywania przebiegów testowych mogą być realizowane w godzinach nocnych. W przypadku wytwarzania dużych produktów programowych są to procesy bardzo czasochłonne.

wykonawczy, wymusza pewien styl organizacji prac i zarządzania zespołem. Rozwiązania organizacyjne w pracy zespołów wykonawczych są oryginalne i dostosowane do specyfiki podejścia MSF.

Podstawowym założeniem organizacyjnym jest odrzucenie hierarchicznej zależności członków zespołu. Wszyscy członkowie zespołu wykonawczego są jednakowo ważni, chociaż każdy ma jasno określoną rolę do realizacji. Rola realizowana na danym etapie prac określa pozycje w zespole a nie miejsce w hierarchii organizacyjnej. Podział władzy w projekcie jest więc rozproszony pomiędzy wszystkich członków zespołu i ulega zmianie w zależności od etapu prac. Konsekwentnie stosowana jest zasada, że wraz z władzą przekazywana jest odpowiedzialność za podejmowane decyzje. Kluczowe decyzje podejmowane są kolektywnie na zasadzie konsensusu wszystkich uczestników procesu. Taki sposób podejmowania decyzji wymusza dzielenie odpowiedzialności i konsoliduje zespół.

Model zespołu wykonawczego przedstawia poniższy rysunek.



Rys. Model zespołu wykonawczego

Trzon zespołu wykonawczego formułowany jest już na początku pierwszego cyklu wykonawczego w fazie opracowywania wizji produktu. W kolejnych fazach realizacji poszczególne grupy pracowników mogą mieć różny skład i wielkość w zależności od intensywności prac danego etapu. Pokazane na rysunku role są reprezentowane przez cały czas realizacji projektu. Są to osoby lub zespoły odpowiedzialne za określoną rolę w składzie

zespołu projektowego i posiadające odpowiednie uprawnienia i odpowiedzialność. Cechą charakterystyczną jest niezależność tych ról i ich interdyscyplinarność, co ma gwarantować utrzymywanie odpowiedniego balansu odpowiedzialności i władzy w całym zespole projektowym.

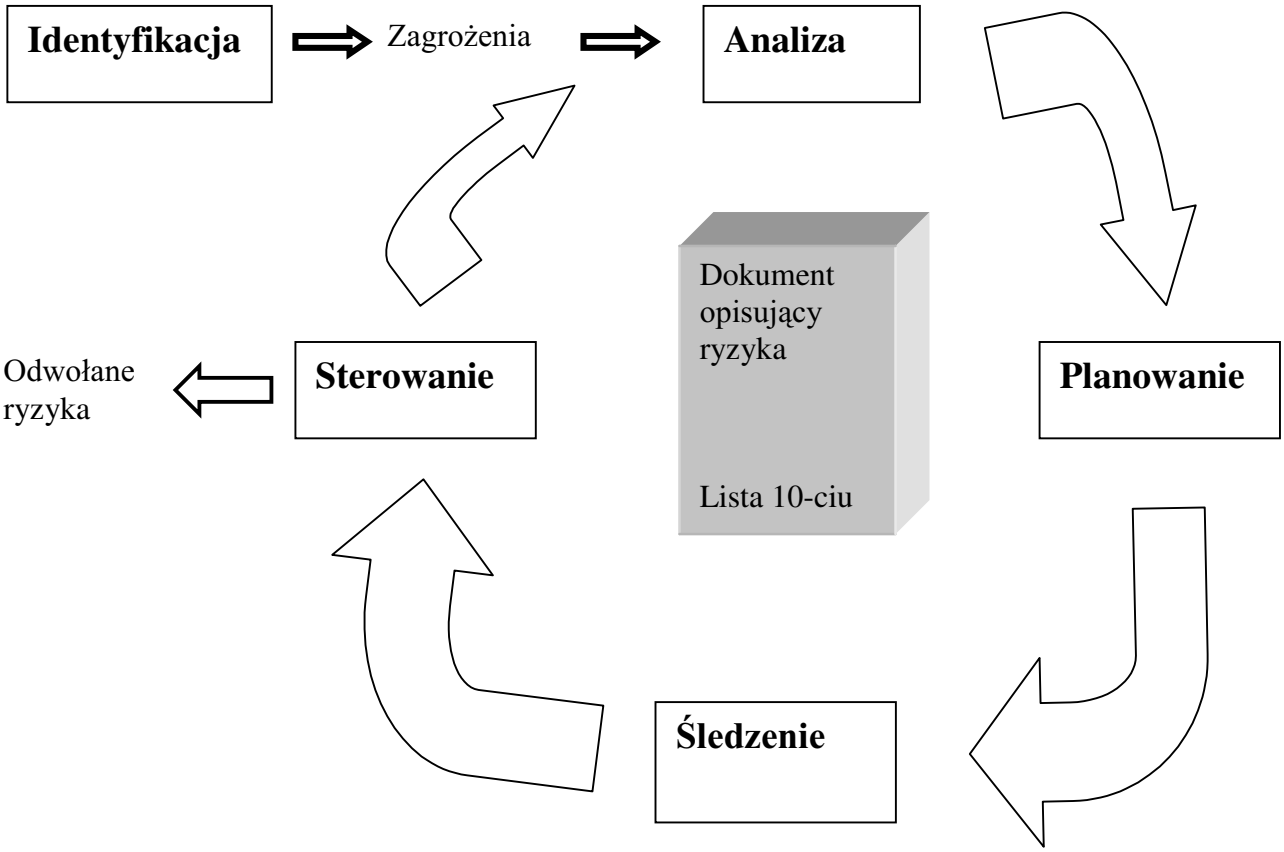
Odrzucenie hierarchicznej zależności pomiędzy poszczególnymi członkami zespołu wykonawczego gwarantuje ich niezależność w podejmowanych decyzjach. Innym ciekawym rozwiązaniem jest cykliczne przekazywanie kierowania zespołem. Niezależność poszczególnych ról nie oznacza, że panuje anarchia w pracy zespołu. W zależności od fazy realizacji projektu inna osoba przejmuje kierownictwo nad pracą całego zespołu. Każda z faz realizacji w kolejnych cyklach ma określony cel cząstkowy i w zależności od tego celu inna osoba z zaznaczonych w modelu przejmuje rolę kierownika. Rozwiązanie takie, niespotykane w innych metodykach realizacji systemów informatycznych [9], ma gwarantować utrzymanie dynamiki procesy wytwarzania. Uzyskuje się efekt podobny, jak w biegu sztafetowym, gdzie poprzez przekazywanie pałeczki, łączny czas zespołu jest lepszy niż suma wyników indywidualnych. Sztafetowe przekazywanie kierownictwa zespołu jest konsekwentnie realizowane w każdym kolejnym cyklu w całym spiralnym procesie dochodzenia do rozwiązania.

Rolą kierownika projektu, zaznaczoną na rysunku, jest reprezentowanie interesu zespołu wykonawczego oprogramowania, czyli jest to reprezentant programistów tworzących oprogramowanie. Sponsor natomiast reprezentuje potencjalnego klienta wytwarzanego oprogramowania i jego cele biznesowe. Reprezentant użytkownika ma za zadanie dbać o rozwiązania akceptowalne przez użytkownika i reprezentuje jego pragmatyzm. Programista czy architekt rozwiązania jest odpowiedzialny za poprawność programistyczną rozwiązania w sensie metod i zastosowanych algorytmów. Poprawność wykonania założeń, gwarantuje osoba lub zespół testujący, wymieniony w modelu zespołu. Wdrażanie, wspieranie i administrowanie pracą zespołu, to rola osoby odpowiedzialnej za logistykę w zespole. W innych metodykach role te występują, ale w układzie podporządkowania hierarchicznego.

Odrzucenie zależności hierarchicznych i konsekwentne realizowanie przypisanych ról poszczególnym członkom zespołu, gwarantuje nie tylko dynamikę działania i odpowiednia jakość rozwiązania ale również utrzymuje zespół w stanie równowagi. Zakładane jest zdyscyplinowane dążenie do sukcesu całego zespołu. Jasne rozdzielenie ról i sztafetowe przekazywanie kierownictwa w kolejnych cyklach realizacyjnych ma gwarantować dynamiczna realizacje projektu. Rozmiar całego zespołu i jego poszczególnych, pokazanych na rysunku ról, jest uzależniony od konkretnego projektu. Niezależnie od wielkości zespołu i ewentualnych dalszych jego podziałów, cały zespół stanowi jedną drużynę dążącą do wspólnego celu, zrozumiałego przez wszystkich członków zespołu.

4. Zarządzanie ryzykiem

Spiralne podejście do rozwiązywania problemów szczegółowych zastosowane jest również przy zarządzaniu ryzykiem projektu. Opracowana w początkowej fazie pierwszego cyklu realizacyjnego, pierwsza wersja analizy ryzyka projektu, jest podstawa dalszych cyklicznie realizowanych analiz. Graficznie przedstawia to rysunek.



Rys. Model zarządzania ryzykiem (według [6])

Proces zarządzania ryzykiem jest standardowy, składa się z fazy identyfikacji zagrożeń, analizy ich wpływu na nasz projekt i po ocenie przyjmujemy plany zapobiegawcze. Następnie śledzimy zidentyfikowane zagrożenie oraz oddziaływanie podjętych działań zapobiegawczych sterując ryzykiem projektu. Zagrożenia, które ustały lub plany zapobiegawcze wyeliminowały dane zagrożenie powodują odwołanie ryzyka.

Procedura zarządzania ryzykiem, przedstawiona na rysunku powtarzana jest cyklicznie w kolejnych cyklach realizacji projektu. Charakterystyczne jest stworzenie listy 10-ciu najważniejszych, w sensie zagrożenia, ryzyk i koncentracja na tej liście. Lista ta ułożona według najsilniejszego negatywnego oddziaływania pozwala na odpowiednim skoncentrowaniu uwagi na zagrożeniach. W pierwszej kolejności rozwiązywane są problemy najtrudniejsze, co oznacza, że zadania najbardziej ryzykowne planowane są do realizacji w pierwszej kolejności. Taki sposób podejścia do ryzykownych zadań pozwala na wczesne podejmowanie krytycznych decyzji dla całego projektu. Rozpoczęcie od najbardziej ryzykownych zadań, minimalizujemy ewentualne straty w przypadku nieudanych projektów.

Inaczej oceniane i obsługiwane jest zagrożenie, które dotyczy podstawowej funkcjonalności tworzonego oprogramowania a inaczej zagrożenie dotyczące funkcjonalności pomocniczej. Podział na funkcjonalność główną i pomocniczą, pozwala na lepszym planowaniu i harmonogramowaniu prac, jak również na odpowiednim zarządzaniu ryzykiem projektowym. Jeśli na etapie testowania błędy wykryte zostaną w obszarze funkcjonalności pomocniczej to nie podejmuje się działań zmierzających do usunięcia tych błędów tylko eliminowana jest z aktualnej wersji cała funkcjonalność. Oczywiście nie rezygnuje się z tej funkcjonalności całkowicie, tylko przenosi do następnego cyklu realizacji.

Podstawowym założeniem realizacji w podejściu MSF jest utrzymanie dynamiki działań, co skutkuje nie tylko eliminowaniem funkcjonalności, w których stwierdzono błędy ale również zasada „good enough", co oznacza, że proces testowania nie ma wyeliminować wszystkie błędy a jedynie najważniejsze. Produkt „wystarczająco dobry" jest wypuszczany na rynek i poddawany ocenie użytkowników. Odgłosy rynku są zbierane i uwzględniane w kolejnym cyklu wytwarzania [7].

W analizie ryzyka w podejściu MSF nie uwzględnia się wzajemnego oddziaływania zagrożeń. Każde ryzyko rozważane jest oddzielnie i analiza nie uwzględnia przenoszenia zagrożeń. Brak mechanizmu oceny ryzyka globalnego dla całego projektu. Niezbyt jasna jest procedura obsługi „listy 10-ciu", gdzie zapisane są zagrożenia zidentyfikowane w początkowej fazie prac. Zagrożenia spoza listy mogą w trakcie prac nabrać nowych wartości i stać się poważnym zagrożeniem dla projektu.

Wnioski

Spiralne podejście do tworzenia oprogramowania jest bardzo atrakcyjne, chociaż wymaga dużej dyscypliny działań zespołowych oraz odpowiedniego oprzyrządowania, umożliwiającego szybkie reagowanie na zgłaszane modyfikacje. Podejście MSF stosowane przez firmę Microsoft jest bardzo interesującym rozwiązaniem wykorzystującym podejście spiralne w połączeniu z wieloma oryginalnymi rozwiązaniami organizacyjnymi. Sukces rynkowy firmy Microsoft w zestawieniu z innymi, niezadowalającymi wynikami prac programistycznych, skłania do głębokiej analizy tego podejścia.

Każda metodyka wytwarzania oprogramowania ma swoją specyfikę, wynikającą z uwarunkowań realizacyjnych i możliwości organizacyjnych. Wydaje się, że w warunkach wytwarzania oprogramowania uniwersalnego, podejście spiralne jest najlepszym rozwiązaniem minimalizującym ryzyko nietrafnego ocenienia zapotrzebowania na wytwarzaną funkcjonalność. Stała presja czasu i niezadowolonego w pełni użytkownika, zmusza do poszukiwania nowatorskich rozwiązań organizacyjnych w połączeniu ze znanymi metodykami wytwarzania systemów informatycznych. Wydaje się, że podejście MSF jest udaną próbą takiego rozwiązania.

Sukces rynkowy firmy Microsoft oraz popularność oprogramowania wytwarzanego przez tą firmę, wskazują na wielką skuteczność podejścia MSF. W praktyce korzystania z oprogramowania Microsoft, dość często spotykamy się z drobnymi usterkami wykorzystywanego oprogramowania, które są denerwujące. Może jest to najlepsza droga do wyprodukowania oprogramowania tego typu, gdzie nie ma jednoznacznie zdefiniowanego użytkownika i jego określonych potrzeb, które można przeanalizować i wytworzyć oprogramowanie zgodnie z zaakceptowaną przez klienta specyfikacją. W praktyce wytwarzania oprogramowania warto skorzystać z niektórych rozwiązań zawartych w podejściu MSF.

Literatura

[1] Capers J. : *Patterns of Software Systems Failure and Success*, International Thomson Computer Press 1996.
[2] William R. Duncan, *A Guide To The Project Management Body Of Knowledge*, PMI Standards Committee, Project Manamement Institute, PA 19082 USA
[3] http://www.microsoft.com/solutionsframework
[4] Andrzej Jaszkiewicz, *Inżynieria oprogramowania*, Helion, Gliwice 1997

[5] *Managing Software Milestones at Microsoft*, American Programmer, Volume 8, Number 2, February 1995, 28–37.
[6] *MSF Risk Management Process,* Microsoft Corp., 1999
[7] Roger W. Sherman, *Can We Ship It Yet ?,* Microsoft Corp., April 1996
[8] Roger W. Sherman, *Shipping the Right Product at the Right Time,* Microsoft Corp., March 1994
[9] Smith, Stanley A., and Michael A. Cusumano. "Beyond the Software Factory: A Comparison of 'Classic' and PC Software Developers," *MIT Sloan School WP#3607-93\BPS*. September 1, 1993
[10] Zdzisław Szyjewski, *Zarządzanie projektami informatycznymi,* Wydawnictwo Placet, Warszawa, 2001