

NR 1/2014

REQ MAGAZYN

KWARTALNIK

WERYFIKACJA
WYMAGAŃ

INNOWACYJNOŚĆ WYMAGA
SYNERGII



NA STYKU BIZNESU I IT:
ZAKRES PROJEKTU INFORMATYCZNEGO
I ZAKRES OPROGRAMOWANIA

ISSN 1234-5678
1234567 1234567

Szanowni Państwo

Przedstawiamy Państwu....



Zapraszam do lektury!
Monika Perendyk

Prezes Stowarzyszenia Inżynierii Wymagań

Redaktor Naczelną

REDAKCJA

Opracowanie i redakcja:

Redaktor naczelną:
Monika Perendyk

Zespół redakcyjny:
Włodzimierz Dąbrowski
Artur Kiełbowicz
Rafał Stańczak
Ewa Brzeska

Projekt graficzny i skład:
Dagmara Zawada

Prawa autorskie
Wszystkie opublikowane artykuły są objęte prawem autorskim autora lub przedsiębiorstwa. Wykorzystywanie w innych publikacjach, kopiowanie lub modyfikowanie zawartości artykułu bez pisemnego upoważnienia autora jest zabronione.

Facebook
<http://www.facebook.com/reqmagazyn>

Reklama
czasopismo@wymagania.org.pl

Współpraca
Osoby zainteresowane współpracą w zakresie publikacji prosimy o kontakt:
czasopismo@wymagania.org.pl



wymagania.org.pl



Spis

Treści

2

Słowo wstępu

Inżynieria Wymagań

6
Hanna Wesołowska
Weryfikacja wymagań

Inżynieria Oprogramowania

10
Ewa Brzeska
Na styku biznesu i IT:
Zakres projektu informatycznego i zakres oprogramowania

Zarządzanie projektami

20
Jacek Wach
Innowacyjność wymaga synergii

Działły stałe

Działły ruchome

24

Relacja z wydarzenia:
....





Hanna
Wesołowska

Inżynieria Wymagań

Weryfikacja wymagań

Wstęp

Projekty są niczym bieg sztafetowy. Każdy ma w nim swój czas, swoje ważne zadanie, w którym musi dać z siebie wszystko.

Na końcu odcinka kolejna osoba czeka na przejęcie pałeczki. Chwytą ją mocno i puszczą się pędem przed siebie. My zostajemy z zadyszką i poczuciem dobrze spełnionego zadania. Kiedy, jako analitycy przygotowujący wymagania, możemy z czystym sercem przekazać pałeczkę? Nie wystarczy przedłożyć gotowego zestawu jeszcze ciepłych wymagań. Trzeba jeszcze zapewnić, że są odpowiedniej jakości, czyli podać je procesowi weryfikacji.

Weryfikacja ma za zadanie potwierdzić, że wymagania są odpowiedniej jakości, tj. dobrze napisane czy zamodelowane. W odróżnieniu od walidacji, która to poświadczają, że wymagania są odpowiednie dla odbiorców (klientów, użytkowników), czyli wspierają spełnienie celu.

W świecie programistów znany jest motyw żółtej kaczuszki. Pluszowe lub nadające się do kąpieli, stoją często na biurkach, gotowe do użycia. Tłumaczenie kaczce, co właśnie się programuje, pomaga spojrzeć na własną pracę z innej perspektywy, wypowiedzieć myśle na głos i często znaleźć błędy, które pozostawały niezauważone. W rolę kaczki mogą się też wcielać koledzy z zespołu. Zapewnianiu jakości pomagają też przeglądy kodu przez bardziej doświadczoną lub dedykowaną do tego zadania osobę, pilnując właściwej formy, trzymania się ustalonych w zespole reguł.

Analitykom nie mniej potrzebna jest żółta kaczuszka i przeglądy. Także w tym przypadku potrzebne są inne osoby. Oczywiście nikt nie wykona przeglądu wymagań lepiej, niż ci, na których główny spadnię później ewentualne wynikające z nich niedociągnięcia.

W przypadku wymagań funkcjonalnych dla systemów świetnie sprawdzają się w tej roli:

- Architekci – wykrywają niespójności i problemy z wykonalnością na poziomie struktury systemu,
- Programiści – przeglądają wymagania skrupulatnie, ponieważ na nich spadnie ostateczny ciężar przygotowania oprogramowania spełniającego wymagania i wybranego przed dociekliwymi testerami,
- Testerzy – wyłapują braki wszelkich informacji, które będą utrudniać testowanie, takie jak np. niedookreślone formaty, zakresy danych, zależności,
- Analitycy – oczywiście nikt bardziej skrupulatnie nie będzie pilnował formy zapisu wymagań i wszystkich charakterystyk, jakimi dobre wymaganie powinno się odznaczać.

Wymagania odnośnie wymagań, czyli co mamy sprawdzać

Znamy szereg wymagań do wymagań, tj. jakie powinny być, aby były dobrej jakości (wg BABOK Guide 2.0):

- Kompletne,
- Spójne,
- Poprawne,
- Wykonalne,
- Modyfikowalne,
- Jednoznaczne,
- Testowalne.

Poniżej kilka pomysłów, jak możemy sprawdzić spełnienie tych wymagań jakościowych przez nasze wymagania jako analitycy.

Kompletność

Sprawdzenie kompletności polega na zweryfikowaniu, czy w zakresie są wszystkie istotne wymagania. Możemy to sprawdzić za pomocą odniesienia ich do celu projektu – czy zbiór wymagań pozwala na osiągnięcie celu? Odwołując się do procesów biznesowych weryfikujemy, czy czynności, które chcemy zautomatyzować zostały pokryte wymaganiami. Kolejnym pomysłem jest wykonanie przeglądu wymagań zgłoszonych przez różnych udziałowców (czy te, które miały wejść do zakresu zostały uwzględnione).

Każde wymaganie z osobna także powinno być kompletne. Dlatego w jego zakresie zwracamy uwagę na przewidzenie wszystkich możliwości. W przypadku opisu przypadków użycia będzie to przemyślenie wszystkich kroków, przewidzenie przebiegów alternatywnych czy sytuacji wyjątkowych, a także efektów końcowych. Do tego zadanie bardzo pomocny okazuje się diagram aktywności, na którym łatwiej dostrzec jest możliwości alternatywne do podstawowego przebiegu. Przy każdym kroku pytamy „A co jeśli...“ i tworzymy diagram rozgałęziony na tyle, ile opcji należy wziąć pod uwagę. Wyniki końcowe także powinny uwzględniać wszystkie możliwe warianty.

Przy opisywaniu możliwych stanów obiektu biznesowego i działań wprowadzających zmiany w tych stanach, nieodłączne staje się po prostu narysowanie diagramu stanów. To niesamowicie proste rozwiązywanie na problemy z ewidencją statusów i uchwycenia wszystkich przejść między nimi. Połączenie na diagramie jest, lub go nie ma, więc analiza kompletności staje się prosta.

Modelując procesy czy przepływy aktywności, warto sprawdzić, czy zawarliśmy wszystkie opcje – np. co się stanie, jeśli nie znaleziono żadnego, znaleziono jeden lub więcej niż jeden obiekt, itd.

Weryfikacja kompletności diagramów UML będzie polegała na sprawdzeniu, czy zostały uwzględnione:

- Nazwy elementów,
- Połączenia między elementami,
- Typy połączeń (strzałka o odpowiednim znaczeniu),
- Liczności relacji,
- Inne elementy wymagane przez diagram danego typu.

Z kolei analizując kompletność diagramów w UML za pomocą innych diagramów w UML, możemy posłużyć się następującymi zależnościami:

- kompletność diagramu klas jest sprawdzana przez:

- Diagram stanów,
- Diagram sekwencji,
- Diagram komunikacji,
- Diagram stanów.

Spójność

Badając spójność sprawdzamy, czy różne wymagania nie opisują tego samego, czy któreś nie są sprzeczne między sobą oraz czy poziom szczegółowości jest we wszystkich taki sam. Na część tych wyzwań odpowiadą będzie dobra pamięć i zdrowy rozsądek. Na problem opisywania tego samego wymagania innymi słownami być może pomoże wbudowany w narzędzia CASE słownik pojęć. Jeśli dodawać ważne pojęcia do słownika oraz nimi zarządzać, znacznie łatwiej będzie wykryć niespójność. Używanie słowników w języku polskim jest o tyle ciekawe, że każdy wyraz trzeba odmienić przez liczby i przypadki. Wymaganie, wymagania, wymaganiu, wymaganiem, wymaganiom, wymaganiami, wymaganiach. Ale kiedy się już już człowiek przemęczy, efekt jest ciekawy – śledzone jest każde wystąpienie w całym modelu.

Spójność na poziomie atrybutów przy przypadkach użycia jednego obiektu biznesowego, np. dodawanie faktury, edycja faktury, użycie faktury, pomaga sprawdzić diagram klas. Jest niezastąpiony w two- rzeniu „jednej wersji wydarzeń“. Opisując wymaganie tekstem możemy pogubić się w tym, co ten obiekt ma, czego nie ma i wprowadzić niespójność. Tworząc klasę i dodając jej atrybuty, przy każdym użyciu (dodaj, edytuj, użyj), odwołujemy się do jednego miejsca, gdzie potwierdzamy spójność i przy okazji też kompletność.

Wykonalność

Wykonalności nikt nie zbada tak żarliwie jak programiści i architekci znający ograniczenia projektu. Tadzież wtajemniczony lider zespołu czy kierownik projektu. Przy wymaganiach jakościowych pomocni będą administratorzy systemowi. Zweryfikują oni, czy wymaganie jest możliwe do spełnienia przy istniejącej infrastrukturze, budżecie, czasie i dostępnych zasobach.

Modyfikowalność

Modyfikowalność kontrolujemy poprzez analizę grupowania wymagań. Jeśli wymagania pozostałe w ścisłym związku ze sobą są blisko, logicznie uporządkowane, wprowadzanie zmian będzie znacznie prostsze.

Jednoznaczność

Przy padaniu jednoznaczności, niezastąpieni są testerzy, programiści, puryści językowi i wszyscy dla czepialscy dla zasady. Daj wymagania do weryfikacji największym malkontentom i złośnikom. Jeśli przejdą przez ich sito, będą jak perełki.

Szczególną uwagę warto zwrócić na ujednolicenie nazewnictwa. Często projekty powstają na styku kilku różnych grup interesariuszy – zamawiającego, odbierającego, użytkowników, dostawców, itd. Każde z pojęć powinno być jednoznacznie zinterpretowane przez wszystkich interesariuszy, dlatego tak ważne są wspólne i dostępne definicje.

Abstrakcyjne, ogólne zasady mogą być niekiedy trudne do zrozumienia. Podanie przykładu może rozwiązać wątpliwości i umożliwić zrozumienie zasady we właściwy sposób.

Testowalność

Bez dwóch zdań, mistrzami w weryfikacji testowalności będą testerzy. To oni najlepiej ocenią, czy na podstawie przedstawionego wymagania są w stanie przygotować sprawdzający je test. Przebywanie z nimi powoduje, że analityk z czasem również wytwarza w sobie małego testera, który siedząc z tyłu głowy podszeptuje mu coraz lepsze formy wymagań.



Wymaganie powinno być sformułowane w taki sposób, aby możliwe było obiektywne, jednoznaczne stwierdzenie, czy jest ono spełnione, czy nie. Pomaga w tym rozbijanie na mniejsze części, z których każdą jest łatwiej ocenić. Warto stosować także priorytety (np. skalę MoSCoW), aby określić, które wymagania muszą być konieczne spełnione, aby rozwiązanie zostało zaakceptowane, a które są mniej ważne. Jeśli wymaganie może być spełnione na różnym poziomie, można wprowadzić skalę i przypisać każdemu z poziomów znaczenie – np. spełnione w 90%, wystarczająco, itp.

Kiedy i jak weryfikować

Weryfikacja może zacząć się, naturalnie, po przygotowaniu przez analityka porcji wymagań. Najlepiej, jeśli są ze sobą powiązane (cały zakres projektu lub cały pakiet), aby sprawdzić także zależności między wymaganiami. Za efekt odpowiedzialny jest analityk – może zlecać prace innym, ale to on ostatecznie odpowiada za poprawność modeli czy dokumentacji.

W jaki sposób przygotować się do wykonywania weryfikacji wymagań w organizacji? Proponuję poniższe sposoby.

Gotowa lista aspektów do weryfikacji

Mogliśmy przygotować gotową listę aspektów, które mogą zostać poddane weryfikacji. Chodzi

o to, aby lista zawierała wszystkie znane sposoby. Wpisy na liście podzielić można na kategorie wg obszaru, jaki jest oceniany. Może to być dokumentacja lub modele. Dalej – procesy, wymagania biznesowe, funkcjonalne, itd. Diagramy klas, sekwencji, stanów, itd. Weryfikacja spójności, kompletności, testowalności, itd. Do każdej metody weryfikacji warto dołączyć opis sposobu przeprowadzenia, np. w formie odnośnika do materiałów w sieci, książce czy firmowym intranecie. Można także przypisać proponowaną rolę w projekcie, która dokona kontroli daną metodą.

Tak przygotowana lista może funkcjonować jako kanon wyznaczający trendy przeprowadzania weryfikacji wymagań w organizacji. Warto, by był to kanon żyjący, to znaczy taki, który zostaje wzbogacony po kolejnych projektach o kolejne propozycje. Nowe pomysły mogą wynikać zauważonych na późniejszych etapach powtarzających się problemów, których nie wychwyciła poprzednia weryfikacja albo ze zdobywania nowych kompetencji i poznawania nowych możliwości.

Oczywiście nie wszystko na raz

Recz jasna, nie chodzi o to, by w najmniejszym projekcie przeprowadzać weryfikację na miarę badań naukowych do doktoratu. Warto oznaczyć na naszej liście te techniki, które są bardzo ważne, wychwytują wiele błędów i powinny być wykonywane w każdym przedsięwzięciu oraz te, które będą pomocne, ale być może w większych projektach, przy większym zapasie czasu lub dużym ryzyku i większych oczekiwaniach co do zapewnienia wysokiej jakości wymagań.

Także, do każdego projektu powinniśmy z głową wybrać sposoby weryfikacji, które będą do niego odpowiednie.

Zapewnienie zasobów

Pamiętaj, że nie jesteś sam. Wybierając metody weryfikacji należy uwzględnić, że w odpowiednim czasie będą potrzebne odpowiednie zasoby. Mam tu na myśli osoby w różnych rolach w pro-

jekcie, dostęp do modelu i narzędzia, w jakim jest wykonany, itp. Warto poinformować zarządu kierownika projektu, lidera zespołu jak i samych zainteresowanych, że będą potrzebni – do czego, kiedy i na jak długo.

Na końcu pamiętajmy też o sobie. Weryfikacja oznacza, że mogą pojawić się błędy brzemienne w skutkach lub drobniejsze uwagi, które tak czy inaczej wymagają poprawek i naszej uwagi. Zarezerwujmy sobie zatem także chwilę na wprowadzenie uwag i zaprowadzenie jeszcze wyższej jakości wymagań.

Dobry czy niedobry

Ocena oceny, czyli chodzi tu podsumowanie jaki wybrana metoda weryfikacji przyniosła efekt. Ile błędów zostało zidentyfikowanych? Jakiej wagi? Czy może skuteczność metody zależy od przeprowadzającego weryfikację? Posiadając takie informacje możemy uwzględnić je przy wyborze najskuteczniejszych metod do projektów, gdzie mamy ograniczone zasoby czy potrzeby w kontekście sprawdzania jakości wymagań.

Śledź efekty

Skoro zadaliśmy sobie już tyle trudu wykonując weryfikację wymagań, wprowadzając propozycje usprawnień, warto zastanowić się, czy to coś dało. Oczywiście poprawiając błędy, uwzględniając uwagi, czujemy, że bez tego mogłyby się pojawić na późniejszych etapach problemy. Dobrze jednak mieć bardziej rzetelne informacje odnośnie skuteczności weryfikacji. Możemy zliczać trudności, które pojawiły się w późniejszych etapach projektu w związku z wymaganiami. Brzmi jak kolejna przesadna formalność? Można na to też spojrzeć jak na wstawienie „+1” w jakimś przygotowanym na te okazje arkuszu w odpowiedniej kolumny (błąd niewielki, spory lub tragiczny w skutkach) za każdym razem, kiedy rozpoczęta się burza rozwiązywania dylematu, którego by nie było, gdyby nie te wymagania... A to przecież nie jest często ;)

Hanna Wesołowska, Analityk biznesowy. Lider Zespołu Analizy Wymagań i Zapewniania Jakości. Obecnie w branży telekomunikacyjnej. Doświadczeni w projektach branży bankowej, ubezpieczeniowej i wielu innych. Poprzednie doświadczenia z projektów dla instytucji rządowych. Absolwentka informatyki na Politechnice Gdańskiej (specjalność Inżynieria Systemów i Bazy Danych) i Psychologii w biznesie. Certyfikaty Agile Project Management Foundation, PSM I (Professional Scrum Master), ITIL Foundation, PRINCE2 Practitioner. Prowadząca zajęcia projektowe z Inżynierii Oprogramowania na Politechnice Gdańskiej. Członek International Institute of Business Analysis, członek Stowarzyszenia Inżynierii Wymagań. Autorka artykułów dla Gazety Ubezpieczeniowej, Gazety Bankowej. Prelegentka na konferencjach NetVision, 3camp, UX Camp, Geek Girls Carrots. Członek Sopockiego klubu Toastmasters - organizacji zajmującej się rozwojem umiejętności mówczych i liderskich. Autorka bloga o analizie biznesowej – www.analizait.pl. W wolnych chwilach biega i podziwia widok na morze.



Ewa
Brzeska

Inżynieria Oprogramowania

Na styku biznesu i IT

Zakres projektu informatycznego i zakres oprogramowania

Wstęp

W pierwszym numerze REQMAGAZYN ukazał się mój artykuł „Na styku biznesu i IT: od potrzeby biznesowej do gotowego produktu – rozważania w kontekście inżynierii oprogramowania”. Poruszyłem w nim kilka aspektów, które są istotne, aby proces twórczy oprogramowania realizowany przez IT zakończył się pełnym sukcesem, czyli dostarczył oczekiwanyą wartość biznesowi. Zainteresowanych zapraszam serdecznie do przeczytania całego artykułu. Dziś chciałbym kontynuować poruszanie się po styku biznesu i IT i zająć się szerzej pojęciem „zakres”. Pojęcie to występuje w kontekście tworzonego oprogramowania - produktu, który IT ma dostarczyć biznesowi oraz w kontekście projektu informatycznego, w ramach którego oprogramowanie jest wytwarzane. Zakres oprogramowania definiuje konkretne funkcje, jakie mają być zawarte w produkcie, zakres projektu natomiast – prace, które mają być wykonane, aby powstał produkt posiadający zdefiniowane funkcje. Zasadne jest zatem rozważanie pojęcia „zakres” zarówno z perspektywy zarządzania projektem informatycznym, jak i z perspektywy twórczej - inżynierii oprogramowania.

Zakres systemu informatycznego, bądź w ujęciu projektowym - zakres prac, jakie mają być zrealizowane, jest pojęciem niezmiernie istotnym. Definiuje dokładnie co ma być wykonane oraz stanowi podstawę do odbioru wykonanych prac przez biznes. Dla IT precyzyjnie określony zakres projektu jest podstawą do szacowania jego pracochności oraz terminów i kosztów realizacji. Tymczasem, jak pokazuje praktyka, zakres projektu z różnych powodów bywa nieprecyzyjnie ustalony we wstępnych fazach prac nad projektem oraz zmienia się nagminnie w fazie implementacji oprogramowania. Jakie to niesie za sobą reperkusje? Na co należy zwrócić uwagę podczas realizacji projektu? Na czym polega zarządzanie zakresem? Spójrzmy

na to, co jest istotne zarówno z punktu widzenia biznesu, jak i IT, ze wspólnej perspektywy prowadzącej do powodzenia w realizacji projektu, w rozumieniu dostarczenia przez IT produktu oczekiwanej przez biznes, zgodnie z postawionym na wstępie celem biznesowym przedsięwzięcia.

Zakres jako jeden z parametrów projektu informatycznego

Wszyscy pewnie znają pojęcie trójkąta ograniczeń projektu. W ujęciu klasycznym tzw. żelazny trójkąt projektu przedstawiany jest jako trójkąt o trzech wierzchołkach: zakres, czas, koszty. Jeszcze lepiej tradycyjną rzeczywistość projektową odzwierciedla trójkąt obrazujący zrównoważony system, w którym występuje pięć wzajemnie zależnych zmiennych:

- zakres
- koszt
- czas
- jakość
- zasoby

Budowa tego trójkąta jest w zasadzie dowolna. Ważna jest odpowiednia interpretacja pomiędzy poszczególnymi zmiennymi, które budują trójkąt oraz brak elastyczności wzajemnych powiązań. Do poniższych rozważań przyjęłam model, w którym czas, budżet i zakres umieściłam po bokach trójkąta, natomiast jakość i zasoby znalazły się w jego wnętrzu (Rys. 1)



Rys. 1 Trójkąt ograniczeń projektu

Wzajemna zależność w/w zmiennych – parametrów projektu informatycznego - oznacza, iż zmiana jednego z nich pociąga za sobą konieczność zmiany innych zależnych parametrów tak, aby cały układ wrócił do równowagi. Ponieważ te właśnie parametry mają decydujący wpływ na sukces lub porażkę projektu, myślę że warto poświęcić im chwilę uwagi.

Zakres definiuje dokładnie granice projektu, czyli co w projekcie ma być wykonane oraz to, co w ramach tego projektu zrealizowane nie będzie. Określanie zakresu projektu odbywa się na podstawie wymagań klienta w stosunku do produktu – oprogramowania, które ma zostać wytworzone w ramach projektu. Jasno i precyzyjnie zdefiniowane wymagania są podstawą definicji projektu informatycznego. Zbiór kompletnych wymagań, zarówno funkcjonalnych, jak i niefunkcjonalnych, definiuje również pełen zakres produktu. Wymagania określone są we wstępnej fazie prac projektowych, co nie znaczy, że pozostają stałe w całym okresie realizacji projektu. Im projekt jest bardziej rozbudowany, a czas jego realizacji dłuższy, tym bardziej prawdopodobne, że wymagania w stosunku do wytwarzanego oprogramowania ulegną zmianie, co pociągnie za sobą zmianę zakresu samego produktu, jak i projektu, w ramach którego produkt jest wytwarzany.



Zakres definiuje dokładnie granice projektu, czyli co w projekcie ma być wykonane oraz to, co w ramach tego projektu zrealizowane nie będzie

Koszt, czy budżet projektu, to parametr wyrażony w jednostkach monetarnych, który ma decydujący wpływ na cały projekt. Często zdarza się, że zakres projektu, a także funkcjonalność oprogramowania, które ma w ramach tego projektu powstać, są wręcz dopasowywane do środków (górnjej granicy budżetu), które klient biznesowy może przeznaczyć na ten projekt. A chyba nie znam klienta, który podczas prezentacji oferty nie zapytałby „dlaczego tak drogo i czy da się zrealizować ten projekt taniej?”. Warto pamiętać, że koszt jest dla klienta często najważniejszym, bo z jego punktu widzenia najbardziej odczuwalnym, parametrem projektu. Z własnej praktyki projektowej wiem, że jeśli już zostanie ustalony budżet, w ramach którego ma powstać produkt informatyczny, nagminnie zdarza się, że zamawiający chce zwiększać ustalony razem z budżetem zakres projektu, przy zachowaniu pozostałych warunków (czas wykonania oraz jakość produktu) i jednocześnie zupełnie nie widzi potrzeby zwiększenia budżetu na to przedsięwzięcie.

Czas – zmienna określająca ramy czasowe projektu, które najczęściej wyznacza termin zakończenia realizacji zadań projektowych. Czas jest zazwyczaj mocno związany z kosztami, co oznacza, że np. jeśli w trakcie realizacji zadań projektu znacznie zwiększy się jego zakres, to da się dotrzymać wcześniej wyznaczonego terminu realizacji jedynie wówczas, jeśli znacząco zostanie zwiększyły budżet przedsięwzięcia.

Jakość – w kontekście produktów i projektów informatycznych parametr jakości występuje w dwóch kategoriach:

- Jakość produktu końcowego, rezultatu projektu
- Jakość procesu realizacji tego projektu

Jeśli chodzi o jakość produktu końcowego - przyjmuje się, że produkt ma właściwą jakość, jeśli spełnione są kryteria jakościowe określone w dokumentacji projektu, w ramach którego produkt jest wytwarzany. W informatycznych projektach wytwarzających produktem jest oprogramowanie. Z racji różnorodności typów oprogramowania i jego przeznaczenia, nie można mówić o jednoznacznej definicji jakości, uniwersalnej dla wszystkich produktów. Inne normy jakościowe będzie miało oprogramowanie sterujące maszynami lub procesami, inne – oprogramowanie czasu rzeczywistego wspomagające procesy podejmowania decyzji czy systemy ban-

kowe, a całkowicie inne – proste programy i aplikacje desktopowe czy mobilne służące rozrywce użytkownika. Kryteria jakościowe produktu powinny zatem każdorazowo zostać jednoznacznie

zdefiniowane w fazie zbierania wymagań. Kryteria te mają bezpośredni wpływ na zakres produktu i projektu.

Zasoby, czyli przede wszystkim ludzie – wykonawcy zadań związanych z realizacją projektu oraz sprzęt i infrastruktura, dzięki którym realizacja tych zadań staje się możliwa. Zapewnienie właściwych zasobów i zarządzanie nimi jest zazwyczaj kwestią strony wykonawczej projektu. Klienta biznesowego nie interesuje zwykle, kto personalnie wytwarza dla niego produkt, na jakim spręcie produkt jest wytwarzany i testowany oraz jaka infrastruktura jest do tego potrzebna. Nieco inaczej sprawa może wyglądać jeśli produkt wytwarzany jest w metodyce zwinnej, która z definicji zakłada ścisłą współpracę wykonawcy z zamawiającym, ale chwilowo zostawmy na boku tę kwestię – do metodyki realizacji projektu i wytwarzania oprogramowania wróć jeszcze w dalszej części artykułu.

Powyżej omówiłem krótko każdy z pięciu parametrów projektu, parametrów które mają bezpośredni wpływ na produkt informatyczny wytwarzany w ramach danego projektu. Przy okazji omawiania w/w zmiennych myślę, że warto poruszyć jeszcze jedną kwestię. Otóż **szybko, tanio i jednocześnie dobrze to mission impossible projektu informatycznego**. Nie da się wytworzyć oprogramowania wysokiej jakości, jeśli budżet będzie niewystarczający, a czas zbyt krótki. Warto o tym pamiętać, gdyż przy zdefiniowanym zakresie projektu i z góry narzuconym przez zamawiającego terminie jego realizacji, zbyt często jedynym kryterium wyboru oferenta jest cena za jaką oferuje on swoje usługi. Rażąco niska wycena zamiast cieszyć klienta, powinna tak naprawdę od razu budzić jego niepokój co do przyszłej jakości produktu za tak niską cenę wytworzonego.



Szybko, tanio i jednocześnie dobrze to mission impossible projektu informatycznego.

Nie da się wytworzyć oprogramowania wysokiej jakości, jeśli budżet będzie niewystarczający, a czas zbyt krótki.

Zmiana zakresu produktu w trakcie realizacji projektu

Układ wzajemnie zależnych parametrów projektu pozostaje w równowadze na początku prac nad projektem, pod koniec fazy planowania, kiedy to zatwierdzany jest budżet, harmonogram realizacji

projektu i planowane są zasoby. Praktyka pokazuje jednak, że w trakcie realizacji projektu, dawno po zakończeniu fazy planowania oraz zatwierdzeniu budżetu i harmonogramu, zamawiający bardzo często prosi o wprowadzenie różnego rodzaju zmian we wcześniej określonym zakresie projektu. Najczęściej są to dodatkowe funkcje oprogramowania, które nie były brane pod uwagę na etapie planowania, a które w mniejszym lub większym stopniu zwiększą zakres produktu i projektu. Powody chęci zmian mogą być różne: najczęściej jest to zmiana oczekiwania zamawiającego w stosunku do realizowanego produktu, która może być spowodowana np. nowymi uwarunkowaniem rynkowymi, korektą celu biznesowego przedsięwzięcia, działaniami konkurencji, czy nawet zmianami technologicznymi, jakie zaszły w międzyczasie. Nie ma nic pewniejszego podczas realizacji projektu informatycznego niż to, że w trakcie wytwarzania oprogramowania trzeba będzie reagować na różnego rodzaju zmiany zakresu. Przy czym najczęściej zamawiający sądzi, że rozszerzenie zakresu, którego właśnie chce dokonać jest tak znikome, że nie powinno mieć wpływu na pozostałe parametry projektu, czyli przede wszystkim na budżet oraz czas realizacji (termin oddania projektu). Spójrzmy tymczasem na Rysunek 2 i zobaczymy co dzieje się z układem, który uprzednio był w równowadze:



Rys. 2 Trójkąt ograniczeń projektu - niezrównoważony

Jak widać, nie da się zwiększać zakresu produktu i projektu bez wpływu na pozostałe parametry. Co prawda można sobie wyobrazić, że jeśli zmiany zakresu są niewielkie i dotyczą na przykład kilku dodatkowych funkcji oprogramowania, którego funkcjonalność jest bardzo złożona, zmiana taka będzie na tyle mało istotna, że wykonawca zrealizuje ją na własny koszt, w imię powodzenia całego

projektu. A co robić, jeśli na skutek kolejnych żądań klienta zmiany zakresu stanowią 10 lub 20% zakresu pierwotnego? Wtedy na pewno nie da się już zrealizować przedsięwzięcia z zachowaniem stałości pozostałych parametrów projektu. Co zatem możemy zrobić, aby wówczas cały projekt zakończył się sukcesem, klient był zadowolony z produktu, który otrzyma, a wykonawca nie poniosł straty realizując projekt?

Zarządzanie zmianą zakresu w projekcie

Klient zamawiający potrzebne mu oprogramowanie nie musi absolutnie znać zasad zarządzania projektem, w tym projektem IT. Nie musi także wiedzieć o zależnościach pięciu parametrów projektu pozostających ze sobą w ścisłym związku, o których pisałam wyżej. Dlatego aby ułatwić późniejsze zarządzanie zmianami zakresu w projekcie, warto na etapie planowania projektu uświadomić zamawiającemu te zależności oraz uzgodnić z nim, jakie priorytety mają dla niego poszczególne parametry, a następnie umieścić te informacje w specyfikacji wymagań projektu. Przykładową macierz możliwych kompromisów projektowych - tabelę zawierającą zmienne projektu oraz nadane im priorytety umieszciliem w Tabeli 1.

Parametr \ Priorytet	Najwyższy (krytyczny, ustalony)	Optymalizowany	Optymalizowany	Optymalizowany	Najniższy (elastyczny, negocjalny)
	5	4	3	2	1
Zakres			x		
Koszt				x	
Czas	x				
Jakość		x			
Zasoby					x

Tabela 1 Przykładowa macierz kompromisów projektowych

Dzięki zdefiniowaniu priorytetów odnośnie poszczególnych parametrów projektu, każdorazowo gdy zamawiający zgłasza potrzebę zmiany zakresu projektu, należy przygotować dokument, w którym zawarty jest opis wpływu oczekiwanych zmian na projekt oraz co najmniej jedna propozycja praktycznego wprowadzenia zmian w projekcie, uwzględniająca wcześniej nadane priorytety. Dla przykładu priorytetów zawartego w Tabeli 1, ze względu na to, że projekt jest mocno zorientowany na czas i jakość, można zaproponować klientowi wprowadzenie zmian kosztem zwiększenia budżetu na jego realizację oraz zrealizowanie zmiany za pomocą większej ilości zasobów. Takie podejście pozwala podać wytwórcy oprogramowania za oczekiwaniemi klienta, jednocześnie uzmysławiając mu, że oczekiwane przez niego zmiany zakresu

projektu generują najczęściej konieczność zmian pozostałych zmiennych definiujących projekt.

Jak może skutkować w projekcie zmiana propozowana przez klienta? Możliwych jest kilka wariantów, z których najczęściej występują poniższe:

- zmiana mała - nie pociąga za sobą żadnych skutków w pozostałych zmiennych projektu
- zmiana możliwa do przeprowadzenia, lecz konieczne jest wydłużenie czasu realizacji projektu, aby można ją było zrealizować przy obecnych zasobach
- zmiana możliwa do realizacji, lecz żeby projekt mógł zostać zrealizowany we wcześniej ustalonym terminie, konieczne będą dodatkowe zasoby
- zmiany można dokonać jedynie wówczas, jeśli zostanie wydłużony czas realizacji projektu oraz zostaną przydzielone dodatkowe zasoby.
- proponowana zmiana ma na tyle silny wpływ na projekt, że nie można jej dokonać bez znacznej modyfikacji założeń całego projektu.

Jak to wygląda w praktyce? Wyobraźmy sobie, że w trakcie realizacji projektu klient silnie nalega na dołożenie kilku czy kilkunastu funkcji do wcześniej ustalonej funkcjonalności. Zamawiający chce przy tym (co jest bardzo częste), aby zmieścić się w ustalonym wcześniej budżecie i dotrzymać wcześniej ustalonego terminu realizacji, natomiast

wykonawca z pewnych przyczyn chce sprostać tym żądaniom. Czy jest to w ogóle możliwe?

Teoretycznie tak – można na przykład skierować do pracy nad tym projektem większy zespół ludzi przy zachowaniu tego samego budżetu. Oznacza to, że przy realizacji zadań projektu będzie pracować więcej, ale za to mniej opłacanych pracowników. Być może uda się tą drogą rzeczywiście zrealizować większy zakres projektu mieszcząc się we wcześniej ustalonym budżecie i dotrzymując uprzednio ustalonego terminu realizacji, lecz jakość oddanego produktu z dużym prawdopodobieństwem niestety spadnie, gdyż mniej opłacani pracownicy to z reguły pracownicy o nizszych kwalifikacjach zawodowych. Jeśli ta niższa jakość będzie nadal spełniała wcześniej ustalone kryte-

ria jakościowe produktu, to wszyscy powinni być zadowoleni: klient, ponieważ bez zwiększenia budżetu i w oczekiwany czasie otrzyma produkt o zakresie większym niż pierwotnie ustalony, ze strony wykonawcy zaś nadal takie przedsięwzięcie będzie opłacalne, pomimo zmian wprowadzonych do pierwotnego kształtu produktu i projektu.

Inna sytuacja powstanie, jeśli uzyskana jakość produktu nie będzie spełniała wcześniej ustalonych kryteriów jakościowych – produkt będzie wymagał poprawek, które mogą być zarówno czasochłonne, jak i kosztowne, co w rezultacie z perspektywy wykonawcy uczyńi projekt nieopłacalnym. Natomiast produkt nie spełniający kryteriów jakościowych zamawiającego, zostanie przez niego odrzucony i wówczas projekt zakończy się niestety porażką.

Jest jeszcze jedna sytuacja, kiedy zmiana zakresu projektu, przy założeniu zachowania wcześniej ustalonego budżetu oraz czasu może być bardzo trudna od strony wykonawczej. Wyobraźmy sobie, że zamawiający chce dodać do zakresu produktu tylko jedną czy dwie funkcje, które z perspektywy ilościowej całego projektu wydają się pomijalnie małe (częsty argument zamawiającego: cóż znaczą dwie kolejne funkcje, w stosunku do tak dużego systemu, który zamówiliśmy). Z perspektywy jedynie ilościowej tak może to rzeczywiście wyglądać. Lecz dodatkowe funkcje, bez względu na ich ilość, zawsze mogą mieć znaczący wpływ na całą, lub dużą część architektury oprogramowania, które jest już obecnie w zaawansowanej fazie wytwarzania. Z drugiej strony, realizacja tych funkcji może wymagać pewnych specyficznych zasobów, które są aktualnie trudno dostępne. Na przykład zmiany może zrealizować tylko jeden z dwóch programistów systemowych, którzy aktualnie pracują nad innym projektem i pierwszy możliwy termin ich pozyskania dla tego projektu nie gwarantuje, że zmiana zakresu będzie mogła być przeprowadzona w pierwotnie zakładanym terminie.

Tak więc nawet z pozoru niewielkie zmiany zakresu projektu mogą w praktyce niesć za sobą poważne konsekwencje organizacyjne lub finansowe i dlatego powinny być świadomie zaakceptowane przez obie strony: zarówno zamawiającą, jak i wykonawczą. Jeśli zaś wykonawca projektu zdecyduje się zrealizować zmianę na własny koszt, powinien w jasny sposób powiedzieć zamawiającemu, że właśnie robi mu prezent. Klient zwykle nie docenia tego, co łatwo mu przychodzi i jest całkowicie za darmo, a to może rodzić niebezpieczeństwo, że zmiany w tak realizowanym projekcie staną się regułą i de facto mogą bardzo opóźnić lub nawet uniemożliwić jego zakończenie.

Zarządzanie zmianą zakresu w procesie twórczym oprogramowania

A więc stało się – klient zgłosił w projekcie chęć zmiany zakresu produktu, zmiana została zatwierdzona na szczeblu kierownictwa projektu i przychodzi ten dzień, kiedy ze zmianą musi zmierzyć się zespół deweloperski: projektanci i architekci, którzy muszą uwzględnić zmianę w dotychczasowej architekturze rozwiązania i programiści, których zadaniem będzie zmianę tę zaimplementować w kodzie źródłowym. Zwykle nie są oni zachwyceni koniecznością takich zmian. Jeśli praca nad oprogramowaniem trwa już jakiś czas, to najczęściej zmiany zakresu powodują nie tylko konieczność napisania nowych linijek kodu, aby oprogramować nowe funkcje, lecz także generują konieczność modyfikacji kodu już napisanego i dokładnie przetestowanego. Należy przy tym zwrócić uwagę na to, że wiele projektów informatycznych jest dużych i skomplikowanych, czas ich realizacji jest długi, a w ich wyniku powstaje często oprogramowanie liczące dziesiątki i setki tysięcy linii kodu, składające się z wielu bibliotek, modułów i plików, realizowane w zależności od potrzeb, przez mniejsze i większe zespoły specjalistów.

Żądanie zmiany w projekcie nie trafia zwykle w realizacyjną pustkę. Najczęściej deweloperzy ścigają się z czasem (czy ktoś kiedykolwiek widział, żeby harmonogram realizacji projektu był stworzony z jakimkolwiek zbędnym zapasem czasowym?) i w poście czola implementują kolejne funkcje oprogramowania. W atmosferze gorączkowego kodowania, aby zdążyć zrealizować projekt na czas, w zależności od estymowanej pracochłonności realizacji nowego zadania, jeden lub kilku programistów dostaje od swojego przełożonego zlecenie zaimplementowania zmiany. Jaki to ma wpływ na pozostałe prace programistyczne? Czy prace te muszą być przerwane, aby mogły zostać wprowadzone zmiany? Jako, że z procesem twórczym oprogramowania związana jestem od lat, mogę z czystym sumieniem powiedzieć, że absolutnie nie wyobrażam sobie takiej sytuacji. Prace uprzednio zaplanowane zgodnie z harmonogramem powinny być kontynuowane, natomiast oprócz nich muszą być zrealizowane nowe zadania, związane z implementacją zmiany. Owszem, mogą być potrzebne pewne korekty w stosunku do poprzednio założonego harmonogramu, mogą być konieczne przesunięcia deweloperów do innych zadań niż te, które były poprzednio przydzielone, może być wreszcie konieczne pozyskanie nowych zasobów, czyli rozszerzenie zespołu deweloperskiego – nie wyobrażam sobie jednak całkowitego wstrzymywania wszelkich prac programistycznych tylko po to,

aby wprowadzić do kodu konieczne zmiany. Tym bardziej, że jak pokazuje praktyka, konieczność zmian pojawi się bardzo często.

Jak zatem pracować na wspólnym kodzie w wieloosobowym zespole programistycznym, w którym każdy z deweloperów może mieć przydzielone odrębne zadania? Pracę taką, a przede wszystkim zarządzanie zmianami w kodzie oprogramowania oraz jakością kodu źródłowego, umożliwiają systemy kontroli wersji.

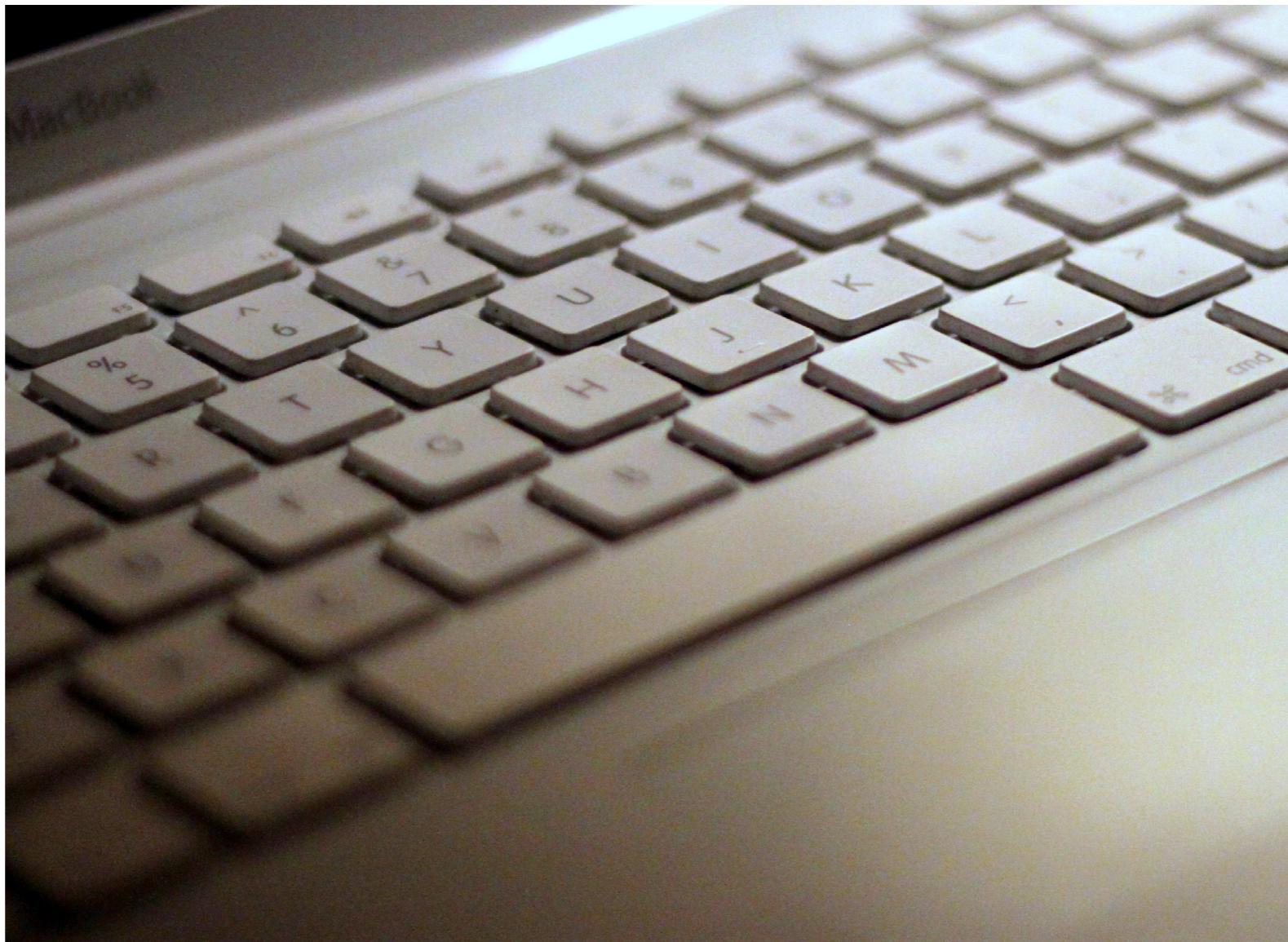
Narzędzie do kontroli wersji oprogramowania pozwala blokować poszczególne pliki zawierające kod źródłowy, czy inne zasoby oprogramowania (pliki tekstowe, dźwiękowe, graficzne, dokumentację, itp.) na wyłączność tego członka zespołu deweloperskiego, który ma zrealizować na nich swoje zadanie. W tak zablokowanych plikach nie mogą być przeprowadzane równolegle inne modyfikacje. System kontroli wersji śledzi ponadto wszelkie zmiany dokonane przez wszystkich programistów i inne uprawnione do zmian osoby, a także w razie potrzeby (np. stwierdzenia trudnego do znalezienia i usunięcia błędu w kodzie źródłowym) umożliwia powrót do wcześniejszych wersji oprogramowania.

Korzyści z wdrożenia w zespole deweloperskim narzędzi do kontroli wersji jest znacznie więcej, lecz ich omawianie przekracza ramy tego artykułu. Ważnym jest to, że narzędzie kontroli wersji wdrożone w zespole wytwarzającym oprogramowanie pozwala właściwie zarządzać procesem twórczym, poszczególnymi wymaganiami, zmianami zakresu i kodu źródłowego oprogramowania oraz jakością kodu, a przez to jakością wytwarzanego produktu.

Zakres a metodyka prowadzenia projektu

W moich powyższych rozważaniach skupiłem się przede wszystkim na założeniu, że oczekiwany zakres produktu i projektu są określone we wstępnej fazie realizacji projektu. Oznacza to, że po fazie zbierania wymagań wszystkie wymagania w odniesieniu do produktu - oprogramowania, które ma powstać w wyniku realizacji właśnie rozpoczętego projektu informatycznego są zdefiniowane i zebrane w dokumencie specyfikacji wymagań oprogramowania (ang. Software Requirements Specification, SRS), który stanowi podstawę realizacji projektu oraz podstawę prowadzenia prac twórczych prowadzących do powstania oprogramowania.

Nie zawsze jednak tak jest. Nie zawsze także projekt informatyczny powinien być aż tak sformalizowany, nie zawsze ma określony z góry zakres, a nawet nie zawsze już na wstępnie zdefiniowany precyzyjny cel biznesowy. Do takich nieokreślonych projektów nie pasują tradycyjne metodyki



zarządzania projektami informatycznymi, ani tradycyjne metodyki wytwarzania oprogramowania. Tym bardziej, że z perspektywy lat wiadomo, iż metodyki te okazały się w wielu przypadkach zbyt mało elastyczne, za bardzo sformalizowane i w rezultacie za „ciężkie”. W wielu projektach, aby zmaksymalizować wartość biznesową wynikającą z realizowania projektu, warto postawić na ścisłą współpracę zamawiającego i wytwórcy oprogramowania, na

szynkowe i naturalne reagowanie na zmiany kodem oprogramowania, bez potrzeby tworzenia rozbudowanej dokumentacji tych zmian zanim zmiany zostaną zaimplementowane, na częste przedstawianie klientowi kolejnych iteracji oprogramowania lub też oprogramowania, które jest wynikiem realizacji kolejnego cyklu wytwórczego. Warto wówczas stosować iteracyjny model wytwarzania lub zwinne metodyki wytwarzania oprogramowania i zarządzania projektem.

Ważnym jest, aby zarówno metodykę zarządzania projektem, jak i metodykę wytwarzania oprogramowania każdorazowo dobierać właściwie do projektu, który ma być realizowany. W zależności od celu tworzonego rozwiązania, stopnia zdefiniowania wymagań (pełnego lub tylko częściowego określenia zakresu rozwiązania), a także rodzaju samego oprogramowania, które ma zostać wytworzone w trakcie realizacji prac projektowych, powinna być dobrana odpowiednia me-

todyka: tradycyjna, iteracyjna, czy zwinna, z pełną świadomością zalet i ograniczeń każdej z nich.

Metodyka tradycyjna jest niewygodna i zbyt sformalizowana do prowadzenia wielu projektów, które z natury swojej muszą być prowadzone dynamicznie. Na przykład oprogramowanie mobilne, które musi nadążać bardzo szybko za zmianami rynkowymi i poczynaniami konkurencji, w związku z czym proces jego wytwarzania

powinien być ukierunkowany na ciągłą zmianę zakresu projektu, jest wprost stworzone do tego, aby wytwarzać go w metodykach „lekkich”, zwinnych. Natomiast skomplikowane oprogramowanie np. do sterowania procesem produkcji energii elektrycznej w elektrowni ma tak wiele sztywnych, z góry ustalonych wymagań wstępnych, które bezwzględnie muszą być zdefiniowane, a następnie zaimplementowane w oprogramowaniu, że wręcz predestynuje to go do wytwarzania w metodyce klasycznej. W praktyce przy tak dużych projektach informatycznych często stosuje się metodyki mieszane: duże części oprogramowania, które zależne są od różnego rodzaju praktycznie stałych uwarunkowań i ograniczeń tworzy się metodyką klasyczną, natomiast części, które powinny szybko reagować na zmiany wymagań (np. obsługa interfejsu użytkownika), mogą i nawet powinny być realizowane w metodyce zwinnej, przy założeniu ciągłej interakcji zespołu wytwórczego z przyszłymi użytkownikami oprogramowania.

Koncentracja na kliencie i jego potrzebach daje właściwe ramy realizowanym procesom i wydatnie zwiększa prawdopodobieństwo dostarczenia klientowi satysfakcjonującego produktu.

projektu, duża częstotliwość zmian zakresu oraz niewłaściwe zarządzanie tymi zmianami, to częste przyczyny niepowodzeń projektów informatycznych. Dlatego tak ważne jest skuteczne zebranie wymagań, zaakceptowanie przez cały zespół projektowy nieuchronności zmian oraz odpowiednie zarządzanie tymi zmianami i ich skutkami, zarówno w sferze zarządzania projektem, jak i w sferze twórczej.

Bardzo ważne jest również dobranie odpowiednich metod wytwórczych projektu i wytwarzania oprogramowania. W zależności od stopnia wstępnego określenia celu i zakresu projektu oraz rodzaju oprogramowania, jakie ma zostać wytwarzane w ramach tego projektu, można stosować metodyki tradycyjne, lekkie / zwinne lub mieszane.

Najważniejszym jest jednak, aby w całym cyklu wytwórczym oraz cyklu zarządzania projektem informatycznym nie stracić z oczu celu biznesowego tego projektu. Koncentracja na kliencie i jego potrzebach daje właściwe ramy realizowanym procesom i wydatnie zwiększa prawdopodobieństwo dostarczenia klientowi satysfakcjonującego produktu.

Bibliografia:

Książki:

Frączkowski Kazimierz, Zarządzanie projektem informatycznym; Projekty w środowisku wirtualnym; Czynniki sukcesu i niepowodzeń projektów, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2003, ISBN: 83-7085-731-0

Wysocki K. Robert, dr, Efektywne zarządzanie projektami; Tradycyjne, zwinne, ekstremalne, Wydanie 6, Gliwice, Helion, 2013, ISBN: 978-83-246-6891-5

Praca zbiorowa:

A Guide to the Project Management Body of Knowledge: (PMBOK® Guide), Project Management Institute Staff, Edition: 5th, Newtown Square; PA USA, Project Management Institute, 2013, ISBN 13: 978-1-935589-67-9; ISBN: 1-935589-67-9



Zakończenie

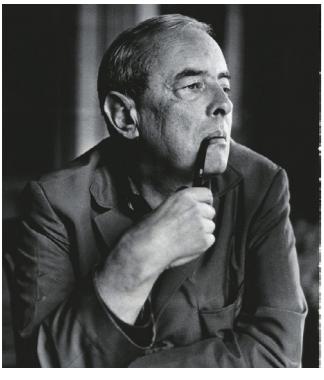
W projektach informatycznych, których celem jest wytworzenie oprogramowania zakres projektu definiuje co należy wykonać, aby zrealizować specyfikowane wymagania klienta. Jednocześnie zakres określa także czego w projekcie realizować nie trzeba, a także jakie są kryteria odbioru gotowego produktu (specyfikacja tych wymagań, które muszą być zrealizowane, aby produkt mógł być zaakceptowany przez klienta). Zakres wyznacza granice projektu. Nie są to jednak granice stałe podczas całego czasu realizacji projektu. Częstotliwość i skala zmian silnie zależą od tego jakiego rodzaju oprogramowania projekt dotyczy.

Niepoprawnie zdefiniowany zakres

Ewa Brzeska, Doświadczony Manager i Konsultant IT. Od 25 lat związana czynnie z branżą informatyczną. W tym czasie uczestniczyła w realizacji wielu projektów, pełniąc rolę analityka, projektanta, architekta i programisty oraz kierownika projektu. Wśród zrealizowanych przez nią projektów są rozwiązania wspomagające zarządzanie różnymi sferami biznesu i produkcji. Ma ponad 15 lat doświadczenia w tworzeniu systemów klasy MES (ang. Manufacturing Execution System) dla energetyki i przemysłu. Zarządza projektami i produktami informatycznymi oraz wytwarzaniem oprogramowania. Zajmuje się konsultingiem i doradztwem IT, wykorzystując przy tym bogate doświadczenie praktyczne oraz wiedzę teoretyczną, zdobytą m.in. w Szkole Głównej Handlowej w Warszawie.

W 2009 roku wraz ze wspólnikiem założyła firmę Ebitech, która zajmuje się głównie tworzeniem oprogramowania biznesowego na platformy systemowe Apple: iOS i Mac OS X.

Współzałożycielka oraz Członek Zarządu Stowarzyszenia Inżynierii Wymagań.



Autor
Autorski



Tytuł artykułu tytuł artykułu tytuł artykułu tytuł artykułu tytuł artykułu

Streszczenie



Jacek
Wach

Zarządzanie Projektami

Innowacyjność wymaga synergii

Wdrożenie strategii zarządzania innowacjami wewnątrz organizacji

Wstęp

W wielu przedsiębiorstwach wciąż po-kutuje przekonanie, że monopol na dobre pomysły znajduje się wyłącznie w głowach jej właścicieli. To przekona-nie trwa dopóty, dopóki nie okaże się, że pomysły naczelnego kierownictwa, które z większym lub mniejszym sukcesem zrealizowano, przestają być wystarczające do zwiększenia efektywności, a tym samym konkurencyjności tych organizacji.

W takich sytuacjach, często w asyście okrzyków „wszystkie ręce na pokład”, rozpoczyna się mo-bilizacja. Mobilizacja kreatywności i wytężonej pracy nad usprawnieniem procesów, rozwojem produktów i wsparcia posprzedażowego. Czy jednak popularne wówczas skrzynki pomysłów i burze mózgów przynoszą wystarczające rezul-taty? Czy aby nie jest tak, że cała organizacja oczekuje jedynego, słusznego złotego pomysłu z góry? Przecież tak było zawsze. Przecież zarząd z pewnością coś wymyśli, dlaczego więc mamy zmieniać status quo? Otóż dlatego właśnie, że status quo w turbulentnym otoczeniu organizacji zwyczajnie nie istnieje. Podobnie zresztą jak i wspomniany już monopol na dobre pomysły. To zdolność do systemowego kreowania kultury innowacyjności staje się podstawą tworzenia nowej jakości zarządzania organizacją.

Innowacyjność wymaga zaangażowania

Proces stałego doskonalenia systemu zarządzania jest nierozerwalnie związany z koniecznością kreowania zaangażowania pracowników w codzien-nej troskę o przyszłość przedsiębiorstwa. Umie-jetne przekazanie współodpowiedzialności za

przetrwanie i rozwój, pozwala bowiem zwiększyć morale zespołu, przyczyniając się do tworzenia nowych, oczekiwanych postaw wobec celów orga-nizacji. Wspomniane cele przestają być wówczas celami firmy X, przestają być celami postawionymi przez Top Management, przestają być nieosi-galne pomimo dołożenia wszelkich starań w ich uSMARTowienie. W nomenklaturze wewnętrznej rozpoczynają funkcjonowanie jako „nasze cele”, co do których realizacji jesteśmy wszyscy tak samo mocno przekonani, w których sens tak samo wierzymy i dołożyliśmy wszelkich starań, aby je osiągnąć. Jak jednak tego dokonać? Co zrobić,

aby zaangażowanie pracowników nie było tylko krótkotrwałym „pospolitym ruszeniem”, nastawionym na re-alizację celów oso-bistycznych, w myśl za-sady – „co JA będę z tego miał?”. Jak zbudować przekonanie wszystkich współpracowników do tego, że ich wie-dza i doświadczenie oraz pomysły, które generują, w znaczący sposób wpływają na przy-słość organizacji? Naturalną odpowiedzią wyda-je się stworzenie w ramach organizacji przestrze-ni otwartej na wymianę poglądów, doświadczeń, dzielenie się pomysłami i ich konsultowanie – przestrzeń otwartą na innowacje, wspieraną przez narzędzia informatyczne.

Innowacyjność wymaga zarządzania

Wdrożenie strategii zarządzania pomysłami (Idea Management) w organizacji wymaga indywidualnego podejścia. Tak bowiem, jak różnią się po-między sobą zarządzający organizacjami tak róż-nią się zarządzane przez nich przedsiębiorstwa. Tymczasem wizje organizacji kreowane przez jej

liderów mają decydujący wpływ na sukces wdrożenia koncepcji zarządzania innowacjami. Jak wobec tego rozpocząć wdrażanie idei organizacji otwartej na nowe pomysły?

„
Na wstępnie należy zadać sobie zatem kilka kluczowych pytań. Po co chcę kreować innowacyjność w mojej organizacji? Jakie cele chcę osiągnąć? Jakie cele chcę, aby osiągnęła organizacja?

kilka kluczowych pytań. Po co chcę kreować innowacyjność w mojej organizacji? Jakie cele chcę osiągnąć? Jakie cele chcę, aby osiągnęła organizacja? Czy chcę rozwijać dotychczas oferowane produkty, a może wprowadzić do oferty zupełnie nowe? Czy chcę doskonalić procesy wewnętrzne? Czy też chcę zbierać wszystkie pomysły, które rozwiązają obecne problemy i/lub wykorzystać po-tencjał organizacji? Odpowiedzi na te zasadnicze pytania, pozwolą nam zaprojektować narzędzia i wskazać sposób ich zastosowania w procesach zbierania pomysłów oraz zarządzania nimi. Ułatwia nam również rozdzielenie pomysłów dotyczących codziennego funkcjonowania organizacji od tych, które decydują o jej przyszłości. To właśnie celo-wość budowania repozytorium pomysłów, stanowiąca podstawę Idea Management, wpływa na kształt wykorzystywanych do tego celu narzędzi. Chcąc skutecznie wdrożyć koncepcję zarządzania innowacjami musimy zbudować w ramach organi-zacji przekonanie, że jesteśmy do tego przygo-to-wani zarówno procesowo, jak i narzędziowo. Przy-gołotowanie to, nie powinno więc ograniczać się jedynie do wyposażenia organizacji w narzędzia informatyczne wspomagające zarządzanie pomy-slami, ale powinno także uwzględniać rozwiąza-nia systemowe, obejmujące procesy idea man-agement, project management i lessons learned / best practices składające się na cykl innowacji.

Innowacyjność wymaga cykliczności

Umożliwienie pracownikom ewidencji ich pomy-slów jest tylko zarzemem wdrożenia cyklu innowa-cyjności wewnętrzorganizacyjnej. Aby skutecznie realizować koncepcję zarządzania innowacjami nalezy przyjąć, że pomysły o znaczeniu strate-gicznym, rozwojowym, wymagające definiowania budżetu w oparciu o wstępnie zarysowany za-kres, powinny zostać przekształcone w projekty zarządzane w sposób metodyczny. Narzędzia, które będziemy wykorzystywać w procesie zarządzania pomysłami, powinniśmy zatem wyposażyć w możliwość pełnego i szczegółowego przedsta-wienia pomysłu, który stanowić będzie zarys uza-





sadnienia biznesowego jego realizacji oraz trzon karty projektu. Na podstawie opisu powinniśmy być w stanie wstępnie zdefiniować cel, zakres i budżet projektu.

Unikatowość organizacji determinuje jednak konieczność ponownego przeanalizowania zdefiniowanych wcześniej celów zarządzania innowacjami. Organizacje, które generowanie pomysłów realizują w formie kampanii poszukiwania rozwiązań zdefiniowanych problemów lub które są nastawione wyłącznie na pozyskiwanie pomysłów o znaczeniu strategicznym, mają stosunkowo prostą drogę. Idee, które uznają za wartościowe, zrealizują stosując podejście projektowe. Organizacje, które w pierwotnych założeniach zdecydowały, że nie ograniczają kreatywności swojego zespołu muszą dokonać wyboru, które z pomysłów zakwalifikowanych do realizacji, wdrożą w formie projektów, które staną się zadaniami realizowanymi w ramach zdefiniowanych wcześniej projektów, ze względu na silne korelacje pomiędzy pomysłami, a które zostaną wykonane w formie zadań bieżących.

Odmienne „DNA” organizacji, przejawiające się w różnym podejściu do zarządzania pomysłami,

będzie miało również ogromne znaczenie w procesie zarządzania projektami, wyłonionymi z procesu idea management. To organizacja musi bowiem podjąć decyzję w jakiej formie będzie realizować projekty – czy będzie to podejście zwinne, tradycyjne czy też hybrydowe, łączące elementy zwanego i tradycyjnego zarządzania projektami.

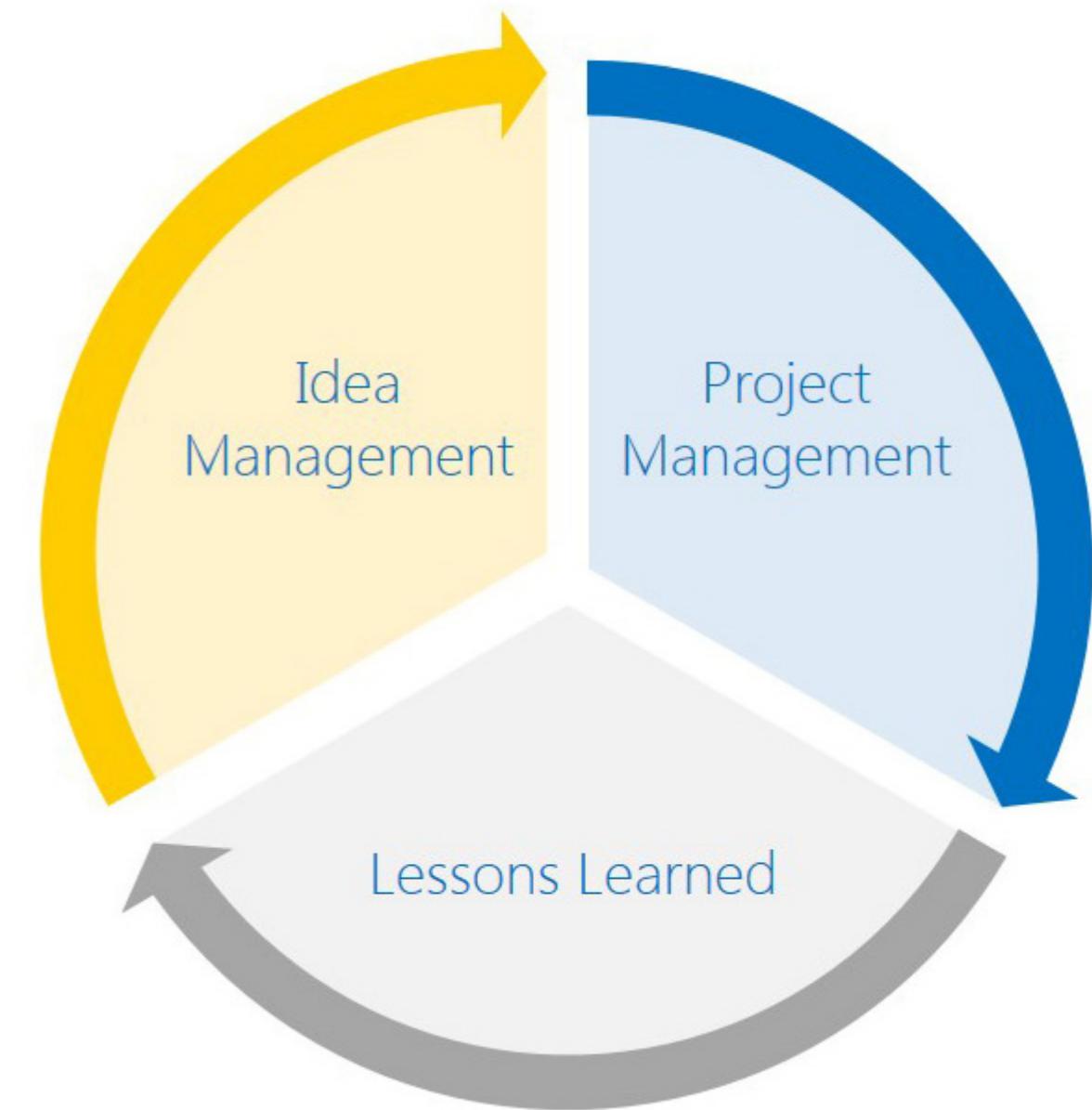
Dopełnieniem całego cyklu zarządzania innowacjami jest lessons learned / best practices, polegające na gromadzeniu zarówno dobrych praktyk, jak i nauczek płynących z błędów popełnionych w projektach oraz wykorzystywaniu tych doświadczeń w przyszłości. Mogą one być źródłem kolejnych innowacyjnych pomysłów, interesujących projektów i... nowych doświadczeń projektowych.

Innowacyjność wymaga zmiany

Zakończenie wdrożenia koncepcji kreowania innowacyjności wewnętrz organizacji jedynie na dostarczeniu narzędzi i zaprojektowaniu procesów zarządzania pomysłami jest niewystarczające. Model cyklicznego zarządzania innowacjami wymaga często przekształcenia tradycyjnej formy zarządzania w zarządzanie projektowe. Dopiero zastoso-

wanie podejścia projektowego, na etapie realizacji pomysłów, pozwala przenieść zarządzanie innowacjami na wyższy poziom. Ta zmiana podejścia pociąga za sobą z kolei konieczność wyciągania wniosków oraz wymaga stałej dbałości o gromadzenie i przetwarzanie doświadczeń przeszłości w przyszłości. Powiązanie idea management, pro-

ject management i lessons learned / best practices pozwala budować organizację wiedzy, otwierającą na nowe wyzwania w świecie, w którym jedyną pewną rzeczą jest... zmiana.



Rys. 1 Cykl zarządzania innowacjami

Jacek Wach, Kierownik Działu Marketingu w BMM Sp. z o.o., Certified Project Management Associate Level D, absolwent Wydziału Zarządzania i Marketingu Politechniki Rzeszowskiej. Trener, konsultant, współtwórca i projektant systemów informatycznych wspomagających idea management, project management i lessons learned. Kierownik projektu wdrożenia strategii zarządzania wiedzą i kreowania innowacyjności wewnętrz organizacji, wyróżnionego w konkursie „Innowator Podkarpacia 2014”. Człowiek, który wierzy, że światy marketingu, Public Relations i Zarządzania Projektami da się pogodzić, a nawet zapewnić pomiędzy nimi synergię. Kierując się na co dzień zawodową dewizą „rób to co kochasz, a nigdy nie będziesz pracować” kreuje, inspiruje i zarządza projektami marketingowymi. Pisał dla portali: Nowy Marketing, Marketing przy kawie, newsline.pl i Day&Night. Członek IPMA Polska.

Relacja z wydarzenia:

....

Wydarzenie



REQ MAGAZYN

chcesz układać puzzle w
inżynierii wymagań
razem z nami?



Facebook

<http://www.facebook.com/reqmazgyn>

Linkedin

[www.linkedin.com/groups/
Stowarzyszenie-Inżynierii-
Wymagań-7464107](http://www.linkedin.com/groups/Stowarzyszenie-Inżynierii-Wymagań-7464107)

Reklama

czasopismo@wymagania.org.pl

Współpraca

Osoby zainteresowane współpracą
w zakresie publikacji prosimy o kontakt:
czasopismo@wymagania.org.pl