Na styku biznesu i IT

Od potrzeby biznesowej do gotowego produktu – rozważania w kontekście inżynierii oprogramowania

# Wstęp

Biznes i IT to dwa z pozoru odległe światy. Pierwszy z nich zainteresowany jest przede wszystkim stopami wzrostu, wskaźnikami wydajnościowymi, sprzedażą, zyskiem, etc.. Drugi fascynują nowoczesne technologie, języki programowania, platformy sprzętowe i systemowe, metodyki realizacji projektów informatycznych oraz tworzenia kodu, itp. Jednocześnie oba te światy wzajemnie się uzupełniają, przenikają i de facto nie mogą bez siebie istnieć.

Biznes potrzebuje IT, bo któż dzisiaj może sobie wyobrazić zarządzanie czymkolwiek bez pomocy wspomagających ten proces programów czy systemów komputerowych. Z drugiej strony nawet najnowocześniejszy i najlepszy program czy aplikacja nie zapewnią jego autorom środków utrzymania i nie wygenerują zysku, jeśli pozostaną w miejscu wytworzenia i nie zostaną sprzedane, gdyż nie będą spełniać wymogów ani realizować potrzeb biznesu.

A zatem oba te światy są na siebie skazane. Czy naprawdę „skazane”? To nie jest dobre słowo na określenie wzajemnych relacji. Na pewno biznes i IT powinny dobrze rozumieć nawzajem swoje potrzeby i wychodzić naprzeciw swoim oczekiwaniom. Prześledźmy zatem drogę od idei i potrzeby biznesowej do gotowego produktu informatycznego, który zespół IT wytwarza w odpowiedzi na potrzebę klienta. Będziemy patrzeć od strony biznesu i od strony IT, a szczególnie będzie nas interesował punkt styku tych dwóch światów.

# Inżynieria oprogramowania

Oprogramowanie komputerowe jest specyficznym produktem, wytwarzanym w procesie wytwarzania oprogramowania. Efektem końcowym realizacji tego procesu jest kod źródłowy, który tworzy funkcje i moduły, z których składa się gotowy produkt – oprogramowanie. Dziedzina, która zajmuje się całokształtem działań związanych z produkcją oprogramowania, czyli: analizą wymagań, projektowaniem, implementacją (pisaniem kodu wg projektu i specyfikacji w konkretnym języku i określonym środowisku programistycznym), testowaniem oraz wdrożeniem gotowego rozwiązania, a także jego utrzymaniem i dalszym rozwojem, to **inżynieria oprogramowania**.

Od strony IT wszystkie rozważania dotyczące produkcji oprogramowania odbywają w kontekście inżynierii oprogramowania. IT jest mocno nastawione na wytworzenie produktu: aplikacji, programu czy systemu informatycznego. W zespołach informatycznych dywaguje się często: może wykorzystać jedną z najnowszych technologii, a może wytwarzać w metodyce, która jest obecnie na topie, może napisać program na tę czy inną platformę systemową. Niestety, podczas tych rozważań niepokojąco często zespołom IT ginie z oczu cel główny pisania oprogramowania: zaspokojenie potrzeby biznesowej, od której wszystko się zaczyna.

# Biznes ma pomysł i środki, a IT może dostarczyć produkt

W pewnym momencie decydent ze świata biznesu dochodzi do wniosku, że potrzebuje oprogramowania komputerowego. Ma środki na jego wykonanie i mniej lub bardziej sprecyzowany pomysł jak to oprogramowanie ma wyglądać i co robić. Rozpoczyna się proces poszukiwania wykonawcy oprogramowania, a po szczęśliwym finale poszukiwań, biznes staje się klientem IT. Relacja ta oznacza, że zleceniodawca oczekuje od zespołu IT wytworzenia oprogramowania o określonych przez siebie cechach, w ramach przeznaczonego na jego wytworzenie budżetu i w uzgodnionym czasie.

Na pierwszy rzut oka mamy w tym momencie do czynienia niemal z sytuacją idealną: zleceniodawca biznesowy jest głęboko przekonany, że za pewną umówioną kwotę, w zaplanowanym terminie, zespół IT wykona oprogramowanie, które będzie dokładnie takim produktem, jaki zleceniodawca sobie wyobraża i jakiego chce. Natomiast zespół IT zwykle z werwą przystępuje do realizacji rozwiązania, mając przekonanie graniczące z pewnością, że jest w stanie wytworzyć oprogramowanie satysfakcjonujące klienta. Skoro na pierwszy rzut oka może być tak dobrze, to dlaczego czasami bywa źle?

# Po co powstaje dane oprogramowanie, czyli cel przedsięwzięcia

Biznes zamawiając oprogramowanie chce zwykle osiągnąć konkretny cel: usprawnić proces, spowodować wzrost wydajności, znaleźć antidotum na szczególnie uciążliwą bolączkę itp. Na etapie pierwotnej idei, pomysłu na oprogramowanie, nie jest jeszcze zwykle określona jego pełna funkcjonalność. Często nie jest nawet znany zakres systemu czy programu. Jednak najczęściej już w tym momencie jest ściśle określony cel biznesowy, który ma być osiągnięty po wdrożeniu tego oprogramowania. Na przykład: wdrożenie systemu monitorowania i zarządzania produkcją ma umożliwić śledzenie produkcji w toku oraz spowodować wzrost wydajności produkcji. Wdrożenie aplikacji monitorującej online flotę samochodową ma na celu obniżenie kosztów zużycia paliwa.

Cel biznesowy jest niezmiernie istotny, gdyż odpowiada na pytanie **po co** powstaje oprogramowanie. Cały proces wytwarzania oprogramowania powinien być podporządkowany temu, aby zapewnić realizację tego celu. Szczególna odpowiedzialność w tym zakresie spoczywa przede wszystkim na analitykach wymagań, którzy identyfikują wymagania odnośnie oprogramowania w kontekście jego celu.

# Nazwa programu i jego zakres

Nazwę przyszłego oprogramowania często proponuje ten, kto zleca jego wykonanie, czyli zleceniodawca (biznes). Warto jednak na samym wstępie, czyli przy pierwszym styku IT z biznesem, przeanalizować adekwatność tej nazwy w stosunku do planowanego zakresu oprogramowania w dziedzinie biznesowej. Te dwa elementy powinny ściśle ze sobą współgrać. Nazwa zbyt szeroko nadana, rodzi pokusę rozszerzania zakresu już w fazie wytwarzania oprogramowania. Z kolei nazwa zbyt wąska w stosunku do przyszłego zakresu, może być podstawą do zarezerwowania budżetu zbyt małego na wykonanie oprogramowania o planowanym zakresie.

Jako przykładem posłużę się autentyczną historią ze swojej praktyki zawodowej. Zdarzyło mi się kiedyś pracować jako zewnętrzny, niezależny konsultant w ramach projektu, którego celem było wytworzenie oprogramowania pod nazwą „System zarządzania produkcją”. Ponieważ dołączyłam do zespołu projektowego już w trakcie realizacji tego projektu, nie miałam wpływu ani na określanie nazwy, ani na precyzyjne ustalanie zakresu systemu. W momencie, kiedy zaproszono mnie do współpracy, projekt był mocno zagrożony. Klient nieustannie rozszerzał funkcjonalność rozwiązania, które było już w zaawansowanej fazie implementacji. Budżet, ustalony i zatwierdzony na początku projektu, kurczył się w zastraszającym tempie, a czas realizacji oprogramowania niebezpiecznie wydłużał, przy czym do końca prac nad projektem nie było jeszcze blisko. Moje zadanie polegało m.in. na tym, aby doprowadzić do konsensusu na linii zleceniodawca – wykonawca, aby projekt mógł być dalej realizowany, a następnie zakończony sukcesem. Dokładanie kolejnych funkcji do oprogramowania, czyli ciągłe powiększanie jego zakresu, klient motywował w tym wypadku stwierdzeniem, że nie wyobraża sobie sytuacji, żeby „System zarządzania produkcją”, nie realizował tej funkcji, o której właśnie mówi, ponieważ jest ona bardzo istotna dla procesu zarządzania produkcją. W opisywanym przypadku nazwa oprogramowania była ewidentnie zbyt szeroka. Jednocześnie zakres rozwiązania niestety nie został precyzyjnie określony na etapie zbierania wymagań, co pozwoliło klientowi na generowanie coraz to nowych żądań w stosunku do wytwarzanego oprogramowania. Na szczęście zleceniodawca dał się przekonać do podziału prac nad systemem na etapy o wyraźnie zdefiniowanych zakresach, harmonogramach realizacji i budżetach. Dzięki temu projekt został zrealizowany pomyślnie.

# Użytkownicy oprogramowania

Ważną czynnością początkowej fazy prac nad realizacją oprogramowania jest poprawna i kompletna identyfikacja wszystkich przyszłych użytkowników tego oprogramowania.

Zamawiający oprogramowanie, czyli ktoś ze świata biznesu, może, lecz nie musi, być jednocześnie jego przyszłym użytkownikiem. Grono użytkowników może być dość szerokie, a w jego skład może wchodzić wiele grup o różnych potrzebach i interesach. Czasami interesy te mogą być wzajemnie sprzeczne.

Jako przykład posłuży nam aplikacja monitorująca on-line flotę samochodową. Oprogramowanie takie dostarcza najczęściej informacji o przebiegu tras poszczególnych samochodów, zatrzymaniach, tankowaniu paliwa, kierowcach pojazdów itp. Użytkownikami tego typu oprogramowania są zazwyczaj m.in.: dyrektor przedsiębiorstwa, kierownik floty oraz każdy z kierowców. Kierownik floty, jako osoba bezpośrednio nadzorująca pracę kierowców jest zainteresowany dokładnymi raportami tras, tankowań, zużycia paliwa itp. Dyrektor przedsiębiorstwa, który najczęściej jest zamawiającym tego typu aplikację (biznes) potrzebuje syntetycznych raportów, dzięki którym będzie mógł określać zbiorcze koszty zużycia paliwa w jednostce czasu. Natomiast grupa kierowców nie będzie zupełnie zainteresowana wdrożeniem takiego oprogramowania (konflikt interesów), lecz będzie zobligowana do korzystania z niego w zakresie swoich kompetencji, np. wprowadzając dane dotyczące przyczyn postoju lub ceny jednostkowej zatankowanego paliwa.

Przed inżynierami wymagań, analitykami biznesowymi, stoi niełatwe zadanie: powinni oni jednoznacznie i poprawnie zidentyfikować wszystkie grupy użytkowników, a także ich potrzeby oraz wymagania odnośnie oprogramowania. Jeśli wymagania są niespójne lub sprzeczne, zadaniem analityków jest doprowadzenie do ich spójności i niesprzeczności. Jednak, aby cały proces odkrywania potrzeb i pozyskiwania wymagań zakończył się sukcesem, konieczna jest dobra współpraca pomiędzy IT a biznesem, w ramach całego zespołu projektowego.

# Potrzeby a wymagania

Potrzeby użytkowników często są artykułowane w bardzo ogólny sposób: „program ma działać szybko”, „ma być prosty w obsłudze”, „ma robić raporty miesięczne”, itd. Proces zbierania wymagań polega na określeniu na podstawie potrzeb użytkowników, kompletnych wymagań funkcjonalnych odnośnie oprogramowania (czyli **co** program ma robić). Dodatkowo określa się również tzw. wymagania niefunkcjonalne, w skład których wchodzą np. kryteria wydajnościowe i jakościowe, jakie ma spełniać oprogramowanie oraz ograniczenia, przy jakich oprogramowanie musi pracować.

W odróżnieniu od swobodnie artykułowanych potrzeb, wymagania zebrane przez analityków muszą być m.in.:

* kompletne,
* spójne,
* jednoznaczne,
* niesprzeczne z innymi wymaganiami,
* weryfikowalne.

Rolą analityka wymagań jest nie tylko pogodzenie sprzecznych interesów, a co za tym idzie często wymagań, różnych grup użytkowników. Wszystkie ogólne potrzeby typu „program ma działać szybko” muszą zostać zamienione w konkretne wymagania czasowe, np. „Oczekiwany czas odpowiedzi systemu dla raportu X – poniżej 3 sekund”. Przy czym podczas całego procesu zbierania i analizy wymagań należy pamiętać o tym, po co oprogramowanie ma zostać wytworzone, a następnie wdrożone. Wszystkie wymagania powinny być pozyskiwane oraz analizowane w kontekście celów określonych przez biznes.

# Właściwa komunikacja podstawą wytworzenia dobrego oprogramowania

Członkowie zespołu projektowego, którego zadaniem jest wytworzenie oprogramowania, komunikują się ze sobą nieustannie. Analitycy biznesowi podczas rozmów (wywiadów analitycznych) z użytkownikami zbierają wymagania. Na podstawie specyfikacji wymagań architekci i projektanci systemów projektują rozwiązanie, osadzając go w konkretnej technologii informatycznej. Często konsultują oni projekt z analitykami z jednej strony, a z programistami z drugiej. Następnie programiści implementują zaprojektowane rozwiązanie, przekazując go testerom. Testerzy testują zgodność oprogramowania z wymaganiami, a w przypadkach wątpliwych zwracają się o wyjaśnienia do analityków. Na koniec wdrożeniowcy wdrażają gotowe oprogramowanie, szkoląc użytkowników ze świata biznesu. Niezmiernie istotnym jest, aby wszyscy interesariusze projektu mówili tym samym językiem. Tymczasem dialogi często przebiegają w taki mniej więcej sposób:

**A.** „Chciałbym, aby dostęp do poszczególnych funkcji systemu był autoryzowany za pomocą nazwy użytkownika i hasła. System może pamiętać hasła użytkowników, ale tylko tych, który nie mają dostępu do newralgicznych funkcji systemu, które za chwilę określę”

**B**. „Dobrze, zrobimy zatem ekran logowania z dwoma entry fieldami: Login i Password. Na ekranie będzie Checkbox do ustawienia, czy hasło ma być zapamiętane. Ale Checkbox będziemy wyświetlać tylko dla tych User’ów, którzy będą mieli poziom uprawnień większy niż 3”

Obie osoby mówią o tej samej funkcji systemu, lecz język ich jest zupełnie inny. Pierwsza kwestia wypowiedziana jest językiem ogólnym, druga zaś to typowy żargon informatyczny. Przy tak różnych językach trudno o dokładne i wzajemne zrozumienie wypowiadanych kwestii, szczególnie jeśli dotyczyłyby one skomplikowanych zagadnień z dziedziny biznesowej. Dlatego w pierwszej fazie prac nad tworzeniem oprogramowania, w fazie zbierania wymagań, powinien powstać słownik pojęć, zawierający kompletny zbiór pojęć z dziedziny oprogramowania, którym będą posługiwać się wszystkie zainteresowane strony.

# Projektowanie i implementacja, czyli niech technologia i metodyka będą dopasowane do wymagań i nie przysłonią celu biznesowego

Czy w ramach projektu zrealizować klientowi aplikację internetową, czy rozwiązanie mobilne? Czy duży system, monitorujący produkcję ciągłą w elektrowni 24 godziny na dobę przez 7 dni w tygodniu, mający bardzo wysokie wymagania odnośnie wydajności oraz niezawodności, projektować i implementować na platformę Windows, czy AS/400? Pisać w C ++, czy w Javie? Czy to oprogramowanie realizować za pomocą prototypu, czy może jednak odpowiedniejsza byłaby metoda Agile, a może pozostać przy metodyce klasycznej?

Na takie i podobne pytania odpowiadają zwykle projektanci oprogramowania i kierownicy projektów. Podejmują oni decyzje, które mają wpływ zarówno na ostateczny kształt oprogramowania, jak i na łatwość jego wdrożenia oraz późniejszego utrzymania. Decyzje te jednak przede wszystkim są istotne dla realizacji celu biznesowego przedsięwzięcia. To przede wszystkim ten aspekt powinien być brany pod uwagę podczas odpowiedzi na powyższe pytania.

Tymczasem niepokojąco często zdarza się, że głównym, jeśli nie jedynym czynnikiem, mającym wpływ na decyzje projektowe jest to, jakiego typu specjalistów mamy aktualnie w zespole. Czyli, jeśli w zespole IT mamy programistów PHP i C#, to sprzedajmy klientowi aplikację webową. Za specjalistów od Javy, czy Objective-C, którzy mogliby zaimplementować aplikację mobilną, musielibyśmy dodatkowo zapłacić, co obniży nasz zysk. Jeśli nasz zespół implementuje „od zawsze” w środowisku Windows, to nawet nie rozpatrujmy, że w tym szczególnym przypadku powinniśmy wziąć pod uwagę inną platformę sprzętową (AS/400), choć tak naprawdę to ona lepiej spełnia kryteria opisane w wymaganiach.

Reasumując: dobór technologii, w której zostanie zrealizowane oprogramowanie oraz metodyki realizacji projektu nie może być ani przypadkowy, ani też głównie zgodny z aktualnym trendem, modą. Dobór ten zawsze powinien być przeprowadzony w taki sposób, aby przede wszystkim zapewnić realizację celu projektu narzuconego przez biznes oraz realizację wszystkich wymagań, które zostały wyspecyfikowane w stosunku do tworzonego oprogramowania.

# Podsumowanie

Co jest istotne, aby przedsięwzięcie polegające na wytworzeniu oprogramowania, będącego odpowiedzią na potrzeby biznesu i spełniającego określone kryteria, zakończyło się sukcesem?

Poniżej najważniejsze, choć nie jedyne czynniki i elementy, z jakimi mamy do czynienia na styku biznesu i IT, na które należy zwrócić baczną uwagę w procesie wytwórczym oprogramowania:

1. Jasno określone cele, jakie biznes chce osiągnąć wdrożeniem oprogramowania (**cel biznesowy przedsięwzięcia**, czyli **po co** oprogramowanie ma być zrealizowane i wdrożone).
2. **Nazwa** programu adekwatna do zakresu oprogramowania w  dziedzinie biznesowej.
3. Jednoznacznie określony **zakres** programu. Stanowi on podstawę rozliczeń pomiędzy wykonawcą oprogramowania i zleceniodawcą, czyli pomiędzy IT i biznesem. Jasno i precyzyjnie określony zakres pokazuje, które funkcje należą do programu, a które są funkcjonalnością dodatkową, pozostającą poza jego zakresem. W przypadkach konfliktowych ułatwia rozstrzygnięcie kto płaci za rozbudowę oprogramowania: wykonawca (IT), ponieważ funkcja jest we wcześniej ustalonym zakresie, czy zleceniodawca (biznes), ponieważ funkcja wykracza poza ramy uzgodnionego zakresu.
4. Poprawna identyfikacja wszystkich **użytkowników** oprogramowania i ich wymagań.
5. Właściwa komunikacja w całym zespole projektowym i **wspólny język**, którym mówią wszystkie zainteresowane strony, zarówno po stronie zamawiającego, jak i wykonawcy oprogramowania.
6. Najważniejszy jest cel biznesowy przedsięwzięcia. Zatem technologia, w jakiej oprogramowanie zostanie wytworzone oraz metodyka realizacji projektu powinny być podporządkowane realizacji tego celu, a nie być celem samym w sobie.