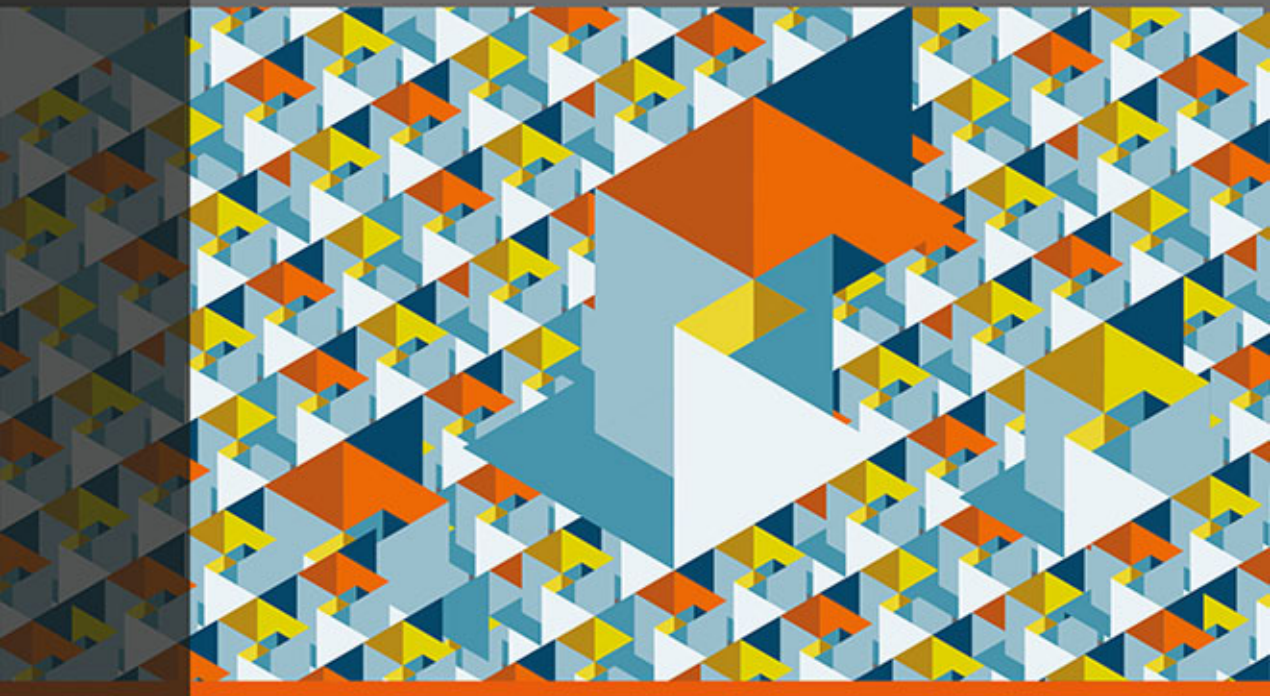
Microsoft

Specyfikacja oprogramowania

Inżynieria wymagań

Wydanie III



Karl Wiegers, Joy Beatty

Helion

Tytuł oryginału: Software Requirements 3

Tłumaczenie: Ireneusz Jakóbik (wstęp, rozdz. 1 – 32, dod. B – C)
Radosław Lesisz (dod. A)

ISBN: 978-83-246-9166-1

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: http://helion.pl (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/speop3
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

DODATKI

ROZDZIAŁ 2

Wymagania z punktu widzenia użytkownika

Gerard, menedżer wyższego szczebla w Contoso Pharmaceuticals, ma umówione spotkanie z Celiną, kierowniczką Działu IT.

— Musimy mieć system informacyjny, który umożliwi śledzenie odczynników chemicznych — zaczął Gerard. — System powinien śledzić wszystkie pojemniki z chemikaliami, które mamy w magazynie i w laboratoriach. Chemicy, zamiast każdorazowo kupować nową skrzynkę odczynników, będą mogli pójść po nie do kogoś po drugiej stronie korytarza. Dzięki temu zaoszczędzimy mnóstwo pieniędzy. Poza tym dział BHP powinien mieć możliwość tworzenia urzędowych raportów na temat zużycia i likwidacji chemikaliów przy mniejszym nakładzie pracy, niż to jest teraz. Czy uda ci się zbudować taki system do czasu przeprowadzenia audytu zgodności za pięć miesięcy?

— Rozumiem, dlaczego ten projekt jest tak ważny, Gerardzie — odpowiedziała Celina. — Ale zanim zobowiążę się do jakiegoś terminu, będziemy musieli ustalić wymagania dla tego systemu.

— Co masz na myśli? Przecież właśnie opowiedziałem ci o moich wymaganiach — Gerard był zdezorientowany.

— W zasadzie to tylko podałeś mi kilka ogólnych celów biznesowych tego projektu — wyjaśniła Celina. — To za mało informacji, abym mogła wiedzieć, jaki program napisać i jak długo może to potrwać. Chciałabym, żeby jeden z naszych analityków biznesowych popracował z kilkoma przyszłymi użytkownikami. Dzięki temu lepiej zrozumiemy, jakiego systemu potrzebują.

— Chemicy są bardzo zajęci — zaprotestował Gerard. — Nie mają czasu na grzebanie się we wszystkich szczegółach, zanim zaczniecie pisać program. Czy twoi ludzie nie mogą domyślić się, co trzeba zrobić?

— Jeśli będziemy zgadywać, co użytkownicy muszą robić w systemie, nie wykonamy porządnie naszej pracy — odpowiedziała Celina. — Jesteśmy twórcami oprogramowania, a nie chemikami. Zdążyłam już się przekonać, że jeśli nie poświęcimy czasu na zrozumienie problemu, nikt nie będzie zadowolony z wyników naszej pracy.

— Nie mamy na to wszystko czasu — upierał się Gerard. — Dostałaś już moje wymagania. A teraz proszę, żebyś zbudowała system. Informuj mnie o postępach.

Tego typu rozmowy są prowadzone regularnie w świecie twórców oprogramowania. Klienci domagający się nowych systemów często nie rozumieją, jak ważne są informacje pochodzące od użytkowników przyszłego systemu oraz od pozostałych interesariuszy. Marketingowcy ze wspaniałymi wizjami produktu są przekonani, że potrafią właściwie reprezentować interesy jego potencjalnych nabywców. Nic jednak nie zastąpi pozyskiwania wymagań bezpośrednio od osób, które będą korzystać z produktu. Niektóre z metod zwinnego tworzenia oprogramowania zalecają, aby przedstawiciel klienta, czasami

nazywany właścicielem produktu, blisko współpracował z zespołem programistów. Jak napisano w jednej z książek poświęconych programowaniu zwinnemu, „projekt jest **kierowany** w stronę sukcesu przez klienta i programistów, pracujących zgodnie ze sobą" (Jeffries, Anderson i Hendrickson, 2001).

Część z problemów związanych z wymaganiami wynika z nieporozumień co do różnych poziomów wymagań, które zostały opisane w rozdziale 1., „Najważniejsze wymaganie dotyczące oprogramowania" — biznesowych, użytkownika i funkcjonalnych. Gerard określił niektóre z celów biznesowych, tj. korzyści, jakie odniesie Contoso po wdrożeniu nowego systemu śledzenia odczynników. Cele biznesowe są głównym elementem wymagań biznesowych. Gerard nie potrafił jednak całościowo omówić wymagań użytkownika, ponieważ nie jest docelowym użytkownikiem systemu. Użytkownicy z kolei mogą opisać zadania, jakie powinni wykonywać w systemie, ale nie potrafią określić wszystkich wymagań funkcjonalnych, które muszą zaimplementować programiści, aby realizowanie tych zadań było możliwe. Analitycy biznesowi powinni współpracować z użytkownikami w celu głębszego zrozumienia ich potrzeb.

W rozdziale tym omówiono relację na linii klient-twórca oprogramowania, która odgrywa krytyczną rolę w powodzeniu projektu. Zaproponujemy wymaganiową kartę praw klienta oprogramowania oraz odpowiadającą jej wymaganiową kartę obowiązków klienta oprogramowania. Karty te podkreślają istotę zaangażowania klienta, a w szczególności użytkownika końcowego w opracowywanie wymagań. W niniejszym rozdziale omówiono także kluczową kwestię osiągnięcia porozumienia co do zbioru wymagań zaplanowanych do realizacji w konkretnym wydaniu produktu lub iteracji. W rozdziale 6., „Słuchanie głosu użytkownika", opisano różne typy klientów oraz użytkowników, a także omówiono sposoby na zaangażowanie odpowiednich przedstawicieli użytkownika w proces pozyskiwania wymagań.

Odrzucony produkt końcowy

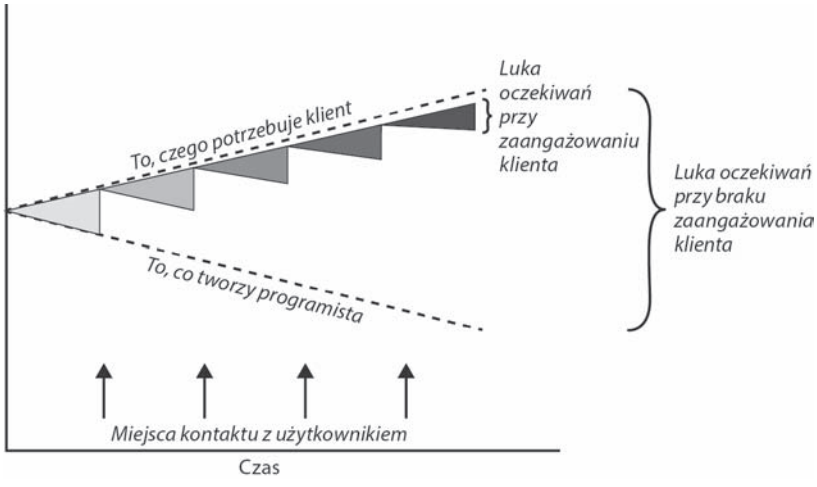
Kiedyś, gdy odwiedzałem dział IT pewnej firmy, usłyszałem smutną historię. Programiści właśnie zbudowali nowy system informacyjny, przeznaczony do wewnętrznego użytku w firmie. Od samego początku wkład ze strony jego użytkowników był znikomy. W dniu, w którym z dumą pokazali swój nowy system, użytkownicy odrzucili go jako zupełnie nieprzydatny. Taka reakcja poruszyła programistów, ponieważ ciężko pracowali, aby sprostać temu, co w ich mniemaniu było wymaganiami użytkowników. Co wobec tego zrobili w tej sytuacji? Naprawili system. Firmy zawsze naprawiają systemy, gdy programiści błędnie zrozumieją wymagania, chociaż koszty takiego działania zawsze są wyższe, niż byłyby w sytuacji, gdyby przedstawiciele użytkowników od samego początku angażowali się w projekt.

Programiści, rzecz jasna, nie planowali, że będą musieli poświęcić czas na wprowadzanie poprawek w wadliwym systemie, w związku z czym ich następny projekt musiał zaczekać w kolejce. W takiej sytuacji przegrana stała się udziałem wszystkich. Programiści byli przygnębieni, użytkownicy zmartwieni, ponieważ ich nowy system nie był gotowy na czas, a zarząd niezadowolony z powodu wydania sporych pieniędzy oraz kosztów utraconych korzyści spowodowanych opóźnieniem w pracach nad innymi produktami. Głębokie i trwające od samego początku zaangażowanie klienta w projekt mogłoby nie doprowadzić do tego niefortunnego, chociaż wcale nie tak rzadko spotykanego końca.

Luka oczekiwań

Bez wystarczającego zaangażowania klienta nie ma możliwości uniknięcia sytuacji, w której pod koniec projektu powstanie luka oczekiwań — różnica między tym, czego w rzeczywistości potrzebują klienci, a tym, co utworzyli programiści, bazując na informacjach, jakie uzyskali na początku prac nad projektem

(Wiegers, 1996), co pokazano na rysunku 2.1 w postaci przerywanych linii. Tak samo jak we wcześniejszej opowieści, luka oczekiwań pojawia się niczym nieprzyjemna niespodzianka, która czekała wszystkich interesariuszy. Z naszych doświadczeń wynika, że niespodzianki w oprogramowaniu nigdy nie niosą ze sobą dobrych wieści. Ze względu na zmiany zachodzące w świecie biznesu wymagania stają się nieaktualne, w związku z czym konieczna jest ciągła współpraca z klientami.



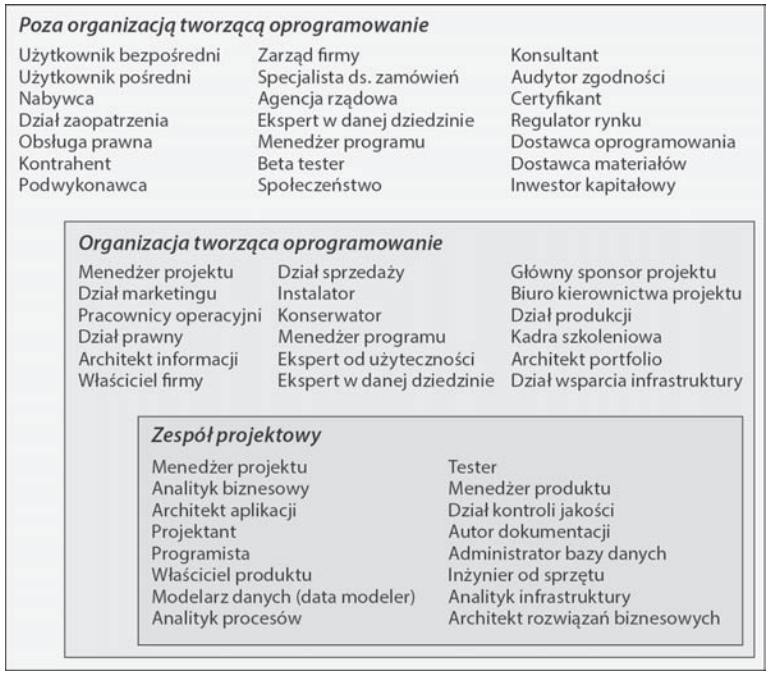
RYSUNEK 2.1. *Częste angażowanie się klienta zmniejsza lukę oczekiwań*

Najlepszym sposobem na ograniczenie luki oczekiwań jest aranżowanie częstych kontaktów z odpowiednimi przedstawicielami użytkowników. Kontakty te mogą przyjmować formę wywiadów, rozmów, przeglądania wymagań, testowania interfejsu użytkownika, oceniania prototypów oraz — w przypadku programowania zwinnego — informacji zwrotnych otrzymywanych od użytkownika na temat kolejnych przyrostów działającego oprogramowania. Każde spotkanie daje możliwość likwidowania luki oczekiwań, ponieważ to, co utworzy programista, będzie w większym stopniu zbieżne z potrzebami klienta.

Oczywiście zaraz po każdym kontakcie z klientem luka oczekiwań zacznie rosnąć, gdy tylko ponownie ruszy proces programowania. Im częstsze jednak będą kontakty, tym łatwiej będzie pozostać na właściwej ścieżce rozwoju produktu. Jak pokazują coraz bardziej kurczące się szare trójkąty na rysunku 2.1, cykl spotkań doprowadzi pod koniec projektu do znacznie mniejszej luki oczekiwań oraz uzyskania rozwiązania, które będzie o wiele bliższe rzeczywistym potrzebom klienta. To właśnie dlatego jedną z głównych zasad programowania zwinnego jest prowadzenie częstych rozmów między programistami a klientami. Reguła ta znakomicie sprawdza się w dowolnym projekcie.

Kim jest klient?

Zanim porozmawiamy o klientach, powinniśmy zająć się interesariuszami. **Interesariusz** to osoba, grupa albo organizacja, która aktywnie uczestniczy w projekcie, na które mają wpływ procesy odbywające się w projekcie lub wynik realizacji tego projektu albo które mogą mieć wpływ na proces lub jego wynik. Interesariusze mogą pozostawać wewnątrz lub na zewnątrz zespołu projektowego oraz organizacji tworzącej produkt. Na rysunku 2.2 zidentyfikowano wielu potencjalnych interesariuszy należących do tych kategorii. Rzecz jasna, nie wszyscy z nich będą występować we wszystkich projektach i sytuacjach.



RYSUNEK 2.2. *Potencjalni interesariusze w zespole projektowym, w organizacji tworzącej oprogramowania oraz poza nią*

Analiza interesariuszy jest ważną częścią opracowywania wymagań (Smith, 2000; Wiegers, 2007; IIBA, 2009). Podczas identyfikowania potencjalnych interesariuszy w danym projekcie zarzuć szeroką sieć, abyś mógł uniknąć przeoczenia jakiejś ważnej ich grupy. W następnej kolejności możesz zredukować listę kandydatów na interesariuszy do osób, których opinia istotnie będzie potrzebna, dzięki czemu zrozumiesz wszystkie wymagania oraz ograniczenia projektu, a Twój zespół dostarczy właściwe rozwiązanie.

Klienci stanowią podzbiór interesariuszy. **Klient** to osoba lub organizacja, która osiąga bezpośrednie lub pośrednie korzyści z użytkowania produktu. Klienci oprogramowania mogą domagać się określonych danych wyjściowych generowanych przez produkt, płacić za nie, wybierać je lub określać, korzystać z nich albo je uzyskiwać. Do klientów pokazanych na rysunku 2.2 należą użytkownicy bezpośredni i pośredni, główny sponsor projektu, nabywca i dział zaopatrzenia. Niektórzy interesariusze nie są klientami, jak na przykład dział prawny, audytorzy zgodności, dostawcy, kontrahenci oraz inwestorzy kapitałowi. Gerard, menedżer, którego już poznaliśmy, jest reprezentantem głównego sponsora projektu, który płaci za produkt. Klienci tacy jak on przedstawiają wymagania biznesowe, które tworzą bazę projektu oraz stanowią biznesowe uzasadnienie jego rozpoczęcia. Jak to zostanie omówione w rozdziale 5., „Określanie wymagań biznesowych", wymagania biznesowe dokumentują cele biznesowe, jakie chcieliby zrealizować klienci, firma oraz inni interesariusze. Wszystkie pozostałe wymagania dotyczące produktu muszą być zgodne z tymi celami.

Wymagania użytkowników powinny pochodzić od osób, które rzeczywiście będą używać — bezpośrednio bądź pośrednio — danego produktu. Użytkownicy ci (często nazywani **użytkownikami końcowymi**) stanowią podzbiór grupy użytkowników. Bezpośredni użytkownicy będą osobiście korzystać z produktu. Z kolei użytkownicy pośredni mogą korzystać z systemu bez styczności z nim, jak na przykład kierownik magazynu, który za pośrednictwem poczty elektronicznej otrzymuje automatyczne raporty na temat dziennych ruchów magazynowych. Użytkownicy mogą opisywać zadania, które muszą wykonywać za pomocą produktu, określać potrzebne im dane wyjściowe oraz charakterystyki jakościowe, jakich oczekują od produktu.

Przypadek brakującego interesariusza

Znam projekt, w którym już prawie skończono pozyskiwać wymagania, gdy podczas przeglądania przepływu procesów analityk biznesowy zapytał interesariusza:

— Czy jesteś pewien, że etap obliczania podatku w tym przepływie jest poprawny?

— Nie wiem — odpowiedział interesariusz. — Naliczanie podatków nie należy do mnie. Od tego jest dział podatków.

Podczas trwających kilka miesięcy prac nad projektem nikt z zespołu programistycznego nigdy nie rozmawiał z żadną osobą z działu podatków. Nikt nawet nie wiedział, że taki dział w ogóle **istnieje**. Jak tylko analitycy biznesowi spotkali się z przedstawicielem działu podatków, okazało się, że można sporządzić długą listę brakujących wymagań związanych z prawnymi następstwami przyjętego sposobu implementacji funkcji odpowiedzialnych za naliczanie podatków. W rezultacie projekt opóźnił się o kilka miesięcy. Aby uniknąć tego rodzaju nieprzyjemnych sytuacji, można skorzystać ze schematu organizacyjnego przedsiębiorstwa w celu odnalezienia wszystkich interesariuszy, na których będzie mieć wpływ nowy system.

Klienci przekazujący wymagania biznesowe czasami twierdzą, że wypowiadają się w imieniu rzeczywistych użytkowników, chociaż są zbyt oddaleni od ich pracy, aby mogli opisać ich faktyczne wymagania. W przypadku firmowych systemów informatycznych, zakontraktowanych zleceń oraz rozwijania aplikacji na zamówienie wymagania biznesowe powinny pochodzić od osoby, która ponosi ostateczną odpowiedzialność za wartość biznesową, jaka jest oczekiwana od produktu. Wymagania użytkowników powinny pochodzić od osób, które będą naciskać klawisze, dotykać ekranu albo korzystać z danych wyjściowych. Jeśli między nabywcą, który płaci za produkt, a końcowymi użytkownikami istnieje przepaść, można mieć pewność, że pojawią się kłopoty.

Inna sytuacja ma miejsce podczas prac nad oprogramowaniem komercyjnym, kiedy klient oraz użytkownik często są tą samą osobą. Osoby wcielające się w klientów, takie jak personel działu marketingu albo menedżer produktu, zwykle starają się przewidzieć, co będzie potrzebne klientowi. Jednak nawet w przypadku oprogramowania komercyjnego powinieneś dążyć do angażowania użytkowników w proces zbierania wymagań, co opisano w rozdziale 7., „Pozyskiwanie wymagań". Jeśli tego nie zrobisz, bądź gotowy na czytanie recenzji wytykających niedociągnięcia w produkcie, których można byłoby uniknąć, gdyby użytkownicy wnieśli swój wkład w jego kształtowanie.

Wśród interesariuszy projektu mogą pojawiać się konflikty. Wymagania biznesowe czasami odzwierciedlają strategie organizacyjne albo ograniczenia budżetowe, które nie są oczywiste dla użytkowników. Użytkownicy niepogodzeni z koniecznością korzystania z nowego systemu informatycznego — który został im narzucony siłą przez zarząd — mogą odmawiać współpracy z twórcami oprogramowania, których postrzegają jako zwiastun niechcianej przyszłości. Takie osoby często nazywane są „grupami przegranych" (Gause i Weinberg, 1989). Aby zapobiegać tego typu potencjalnym konfliktom, wypróbuj strategii komunikacyjnych dotyczących celów i ograniczeń projektu, które pomogą budować wzajemną zgodę oraz powstrzymywać spory i urazy.

Partnerstwo klient-twórca oprogramowania

Doskonałe oprogramowanie jest wynikiem dobrze opracowanego projektu bazującego na doskonałych wymaganiach. Doskonałe wymagania są wynikiem efektywnej współpracy między twórcami oprogramowania a klientami (w szczególności rzeczywistymi użytkownikami), innymi słowy — partnerstwa. Wspólny wysiłek może być skuteczny tylko wtedy, gdy wszystkie zaangażowane strony wiedzą, czego potrzebują, aby odnieść sukces, oraz rozumieją i szanują to, co jest potrzebne do odniesienia sukcesu przez ich partnerów. Gdy w trakcie realizacji projektu narasta presja, łatwo jest

zapomnieć, że wszyscy interesariusze mają wspólny cel. Jest nim wytworzenie produktu, który dla każdego z interesariuszy przedstawia odpowiednią wartość biznesową oraz go wynagradza. Osobą, która powinna podtrzymywać taką zbiorową współpracę, jest zazwyczaj analityk biznesowy.

Wymaganiowa karta praw klienta oprogramowania, zawarta w tabeli 2.1, wymienia 10 praw, z których mogą zasadnie korzystać klienci, dotyczących współpracy z analitykami biznesowymi oraz programistami podczas wykonywania czynności mających związek z opracowywaniem wymagań. Każde z tych praw implikuje odpowiadający mu obowiązek po stronie analityków biznesowych lub programistów. „Twoje" prawa i odpowiedzialności, wymienione na obu listach, dotyczą klienta oprogramowania, które jest tworzone w ramach projektu.

TABELA 2.1. *Wymaganiowa karta praw klienta oprogramowania*

Masz prawo
Oczekiwać, że analityk biznesowy będzie mówił Twoim językiem.
Oczekiwać, że analityk biznesowy zapozna się z Twoimi zadaniami oraz celami.
Oczekiwać, że analityk biznesowy w odpowiedniej formie zarejestruje Twoje wymagania.
Otrzymywać wyjaśnienia dotyczące praktyk związanych z pozyskiwaniem wymagań oraz na temat wymagań docelowych.
Zmieniać swoje wymagania.
Oczekiwać atmosfery wzajemnego poszanowania.
Poznawać nowe pomysły, a także alternatywne wymagania oraz ich rozwiązania.
Opisywać cechy, które sprawią, że produkt będzie prosty w użyciu.
Poznawać sposoby na takie dostosowanie wymagań, aby poprzez ich wielokrotne użycie przyspieszyć proces rozwoju oprogramowania.
Otrzymać system, który spełnia Twoje potrzeby funkcjonalne i oczekiwania jakościowe.

Ponieważ przeciwieństwem prawa jest obowiązek, w tabeli 2.2 wymieniono 10 obowiązków, które w ramach procesu opracowywania wymagań ma klient wobec analityków biznesowych i programistów. Jeśli wolisz, możesz uważać je za kartę praw twórców oprogramowania. Jeśli obu tych list nie da się w pełni zastosować w Twojej organizacji, możesz je zmodyfikować, aby lepiej pasowały do Twoich lokalnych realiów.

TABELA 2.2. *Wymaganiowa karta obowiązków klienta oprogramowania*

Masz obowiązek
Informować analityka biznesowego oraz programistów o swoich zadaniach.
Poświęcać czas na przekazywanie oraz wyjaśnianie swoich wymagań.
Konkretnie i szczegółowo przedstawiać informacje na temat wymagań.
Na prośbę analityka podejmować w porę decyzje.
Respektować oszacowania programistów dotyczące kosztu oraz możliwości realizacji wymagań.
We współpracy z programistami określać realne priorytety wymagań.
Przeglądać wymagania oraz oceniać prototypy.
Ustanawiać kryteria akceptacji.
Bezzwłocznie zgłaszać zmiany wymagań.
Uszanować proces opracowywania wymagań.

Powyższe prawa oraz obowiązki mają zastosowanie w stosunku do rzeczywistych klientów, gdy oprogramowanie jest tworzone na wewnętrzne potrzeby firmy, w ramach kontraktu lub dla określonej grupy głównych odbiorców. W przypadku produkcji oprogramowania na szeroki rynek powyższe prawa i obowiązki odnoszą się w większym stopniu do osób wcielających się w klientów, takich jak menedżer produktu.

W ramach planowania projektu główny klient oraz interesariusze powinni zapoznać się z tymi listami i wynegocjować poszczególne punkty w celu osiągnięcia obopólnej zgody. Należy upewnić się, że osoby uczestniczące w opracowywaniu wymagań rozumieją i akceptują swoje odpowiedzialności. Takie zrozumienie może ograniczyć późniejsze tarcia, gdy jedna ze stron będzie oczekiwać czegoś, czego nie chce lub nie może spełnić druga strona.

Pułapka. Nie zakładaj, że osoby biorące udział w projekcie będą instynktownie wiedzieć, jak współpracować ze sobą podczas opracowywania wymagań. Poświęć czas na wyjaśnienie, jak mogą one efektywnie pracować razem. Dobrym pomysłem będzie zapisanie decyzji dotyczących podejścia do rozwiązywania problemów związanych z wymaganiami, jakie mogą pojawić się podczas prac nad projektem. Taka informacja będzie cennym narzędziem ułatwiającym komunikację podczas wszystkich prac nad projektem.

Wymaganiowa karta praw klienta oprogramowania

Poniżej omówiono 10 praw, z których mogą korzystać klienci podczas prac nad wymaganiami.

Prawo nr 1. Oczekiwać, że analityk biznesowy będzie mówił Twoim językiem

Omawianie wymagań powinno być skoncentrowane na Twoich potrzebach biznesowych oraz zadaniach do wykonania, przy czym powinieneś posługiwać się słownictwem biznesowym. Zastanów się nad możliwością przekazania analitykowi biznesowemu swojej branżowej nomenklatury razem ze słownikiem terminów. Podczas rozmowy z analitykiem nie powinieneś stosować w nadmiarze żargonu technicznego.

Prawo nr 2. Oczekiwać, że analityk biznesowy zapozna się z Twoimi zadaniami oraz celami

Współpracując z Tobą podczas pozyskiwania wymagań, analityk biznesowy może lepiej zrozumieć Twoje zadania biznesowe oraz dowiedzieć się, jak wpasować nowy system w Twój świat. Dzięki temu programiści utworzą rozwiązanie, które spełni Twoje oczekiwania. Poproś analityków biznesowych oraz programistów, aby zobaczyli, czym zajmujecie się Ty i Twoi koledzy w pracy. Jeśli nowy system ma zastąpić stary, analityk powinien popracować na starym systemie, tak jak Ty to robisz. W ten sposób zobaczy on miejsce obecnego systemu w przepływie pracy oraz pozna obszary, w których można go ulepszyć. Nie oczekuj, że analityk biznesowy będzie wiedzieć wszystko o Twoich czynnościach biznesowych oraz znać całą potrzebną terminologię (patrz obowiązek nr 1).

Prawo nr 3. Oczekiwać, że analityk biznesowy w odpowiedniej formie zarejestruje Twoje wymagania

Analityk biznesowy zapozna się ze wszystkimi informacjami, jakich udzielą mu interesariusze, i zada dodatkowe pytania, które umożliwią mu oddzielenie wymagań użytkowników od ról biznesowych, wymagań funkcjonalnych, celów jakościowych i pozostałych elementów. Ostatecznym celem tej analizy jest uzyskanie dopracowanego zestawu wymagań zapisanych w odpowiedniej postaci, jaką

jest na przykład dokument specyfikacji wymagań oprogramowania albo narzędzie do zarządzania wymaganiami. Taki zbiór wymagań dokumentuje osiągnięcie porozumienia między interesariuszami, które dotyczy funkcji, jakości oraz ograniczeń przyszłego produktu. Wymagania powinny być zapisane oraz zorganizowane w formie, która ułatwi ich zrozumienie. Zweryfikowanie tej specyfikacji oraz pozostałych wymagań, a także przedstawienie ich w innej postaci, takiej jak graficzne modele analityczne, pomogą zagwarantować, że dokładnie odzwierciedlają one Twoje potrzeby.

Prawo nr 4. Otrzymywać wyjaśnienia dotyczące praktyk związanych z pozyskiwaniem wymagań oraz na temat wymagań docelowych

Rozmaite praktyki mogą przyczynić się do tego, że proces pozyskiwania wymagań oraz zarządzania nimi będzie zarówno skuteczny, jak i wydajny, a wiedzę na temat wymagań będzie można przedstawić w różnych postaciach. Analityk biznesowy powinien wytłumaczyć praktyki, które zaleca do stosowania, oraz wyjaśnić, jakie informacje znajdą się w końcowych wersjach poszczególnych dokumentów. Może on na przykład utworzyć graficzne schematy uzupełniające wymagania sformułowane pisemnie. Diagramy te mogą być dla Ciebie obce i dość skomplikowane, ale użyta w nich notacja nie powinna być trudna do zrozumienia. Analityk biznesowy powinien wyjaśnić cel każdego diagramu, znaczenie użytych w nim symboli oraz zweryfikować go pod względem poprawności. Jeżeli nie przedstawi on takich wyjaśnień, nie zawahaj się go o to poprosić.

Prawo nr 5. Zmieniać swoje wymagania

Analitycy biznesowi oraz programiści nie powinni oczekiwać, że z miejsca podasz wszystkie swoje wymagania, ani że wymagania te pozostaną niezmienne przez cały okres prac nad produktem. Masz prawo dokonywania zmian w wymaganiach w miarę rozwijania się działalności firmy, otrzymywania nowych informacji od interesariuszy albo gdy głębiej zastanowisz się nad swoimi potrzebami. Zmiana jednak zawsze ma swoją cenę. Czasami dodanie nowej funkcjonalności będzie wymagać zrezygnowania z innej funkcji albo zrewidowania budżetu lub harmonogramu prac. Do ważnych zadań analityka biznesowego należy ocena konsekwencji, jakie niosą ze sobą zmiany, zarządzanie nimi oraz informowanie o nich. Podejmij z nim współpracę w celu osiągnięcia porozumienia dotyczącego wypracowania prostego, ale jednocześnie skutecznego procesu obchodzenia się ze zmianami.

Prawo nr 6. Oczekiwać atmosfery wzajemnego poszanowania

Relacje między klientami a programistami czasami stają się napięte. Dyskusje na temat wymagań mogą prowadzić do frustracji, jeśli ich uczestnicy nie rozumieją się nawzajem. Wspólna praca może otworzyć oczy członków zespołu na problemy, z którymi boryka się każda z grup. Klienci biorący udział w opracowywaniu wymagań mają prawo oczekiwać, że analitycy biznesowi i programiści będą traktować ich z szacunkiem oraz docenią czas, jaki inwestują w sukces projektu. Podobnie klienci powinni okazywać szacunek członkom zespołu programistycznego, gdyż wszyscy razem współpracują po to, aby osiągnąć wspólny cel, jakim jest zakończony sukcesem projekt. Tutaj wszyscy znajdują się po tej samej stronie.

Prawo nr 7. Poznawać nowe pomysły, a także alternatywne wymagania oraz ich rozwiązania

Poinformuj analityka biznesowego o sytuacjach, w jakich Twój obecny system nie sprawdza się w procesie biznesowym, aby zyskać pewność, że nowy system nie zautomatyzuje mało efektywnych lub przestarzałych procesów. Nie chciałbyś także mieć do czynienia z doraźnymi rozwiązaniami. Analityk biznesowy może zasugerować wprowadzenie poprawek w Twoich procesach biznesowych. Kreatywny analityk dodaje także nowe wartości, proponując nowe funkcje, których użytkownicy nawet sobie nie wyobrażali.

Prawo nr 8. Opisywać cechy, które sprawią, że produkt będzie prosty w użyciu

Możesz spodziewać się, że analityk biznesowy zada Ci pytania o cechy oprogramowania, które wykraczają poza Twoje potrzeby funkcjonalne. Cechy te, innymi słowy, atrybuty jakościowe, sprawiają, że oprogramowanie jest łatwiejsze lub przyjemniejsze w użyciu i umożliwia użytkownikom efektywniejsze realizowanie ich celów. Użytkownicy czasami domagają się, żeby produkt był **przyjazny** albo **odporny**, chociaż tego typu określenia są zbyt subiektywne, aby mogli się nimi kierować programiści. Analityk biznesowy powinien zapytać Cię o to, co rozumiesz przez pojęcia „przyjazny" albo „odporny". Opowiedz mu, które aspekty Twojego obecnego systemu wydają Ci się „przyjazne", a które nie. Jeżeli nie omówisz tego aspektu z analitykiem, będziesz mieć duże szczęście, jeśli gotowy produkt spełni Twoje oczekiwania.

Prawo nr 9. Poznawać sposoby na takie dostosowanie wymagań, aby poprzez ich wielokrotne użycie przyspieszyć proces rozwoju oprogramowania

Wymagania czasami mogą być w pewnym stopniu elastyczne. Analityk biznesowy może znać istniejące składniki oprogramowania albo inne wymagania, które w znacznej mierze pokrywają się z opisanymi przez Ciebie potrzebami. W takich przypadkach powinien on zasugerować Ci wprowadzenie zmian w wymaganiach lub zrezygnowanie z nadmiernego dostosowywania systemu do Twoich potrzeb, dzięki czemu programiści będą mogli skorzystać z tych składników. Korekta wymagań, gdy nadarza się praktyczna okazja do skorzystania z gotowych rozwiązań, pozwala zaoszczędzić czas i pieniądze. Czasami elastyczne podejście do wymagań jest niezbędne, jeśli chcesz zintegrować ze swoim systemem komercyjne produkty z półki, gdyż rzadko zdarza się, aby miały one dokładnie takie charakterystyki, jakie są Ci potrzebne.

Prawo nr 10. Otrzymać system, który spełnia Twoje potrzeby funkcjonalne i oczekiwania jakościowe

Jest to najważniejsze prawo użytkownika, **ale** może ono zostać zrealizowane tylko wtedy, gdy wyraźnie przekażesz wszystkie informacje umożliwiające programistom zbudowanie właściwego produktu, gdy programiści poinformują Cię o dostępnych opcjach i ograniczeniach oraz gdy wszystkie strony osiągną porozumienie. Nie zapomnij o wyrażeniu wszystkich swoich oczekiwań; w przeciwnym razie programiści nie będą się mogli do nich prawidłowo odnieść. Klienci czasami nie zgłaszają potrzeb, co do których są przekonani, że są oczywiste. Potwierdzenie wspólnego punktu widzenia w zespole pracującym nad projektem jest jednak równie ważne, jak zaproponowanie czegoś nowego.

Wymaganiowa karta obowiązków klienta oprogramowania

Ponieważ przeciwieństwem prawa jest obowiązek, poniżej wymieniono 10 obowiązków, które mają przedstawiciele klienta podczas definiowania wymagań i zarządzania nimi w ramach projektu.

Obowiązek nr 1. Informować analityka biznesowego oraz programistów o swoich zadaniach

W kwestii zdobycia informacji na temat pojęć związanych z Twoimi zadaniami oraz poznania branżowego słownictwa zespół programistyczny jest zdany na Ciebie. Twoim celem nie będzie przekształcenie analityków biznesowych w ekspertów w Twojej dziedzinie, ale udzielenie im pomocy w zrozumieniu Twoich problemów i celów. Z dużym prawdopodobieństwem analitycy nie dysponują wiedzą, którą Ty i Twoi koledzy uważacie za oczywistą.

Obowiązek nr 2. Poświęcać czas na przekazywanie oraz wyjaśnianie swoich wymagań

Klienci nie mają zbyt dużo czasu do dyspozycji. Ci z nich, którzy są zaangażowani w pracę nad wymaganiami, często należą do najbardziej zajętych osób w firmie. Niemniej Twoim obowiązkiem jest poświęcenie czasu na warsztaty, rozmowy i inne aktywności związane z pozyskiwaniem i walidacją wymagań. Czasami analityk biznesowy może sądzić, że zrozumiał Twoje wyjaśnienia, ale później dojdzie do wniosku, że potrzebuje dalszych informacji. Należy okazać mu cierpliwość, gdyż w taki właśnie iteracyjny sposób zbiera on i precyzuje wymagania — na tym polega złożona natura komunikacji międzyludzkiej oraz taki jest klucz do osiągnięcia sukcesu w tworzeniu oprogramowania. Jeśli wysiłek związany z wyjaśnianiem wymagań zostanie skupiony w kilku nieprzerwanych godzinach, praca nad wymaganiami zajmie mniej czasu, niż gdyby ten sam wysiłek był rozproszony na przestrzeni paru tygodni.

Obowiązek nr 3. Konkretnie i szczegółowo przedstawiać informacje na temat wymagań

Kuszące jest pozostawienie wymagań w postaci mało szczegółowej i niejasnej, gdyż precyzowanie wymagań jest żmudne i wymaga czasu (lub gdy ktoś chce uniknąć odpowiedzialności związanej z podejmowaniem decyzji). W końcu jednak ktoś będzie musiał rozwiać wątpliwości i wyjaśnić niejednoznaczności. Najbardziej kompetentną osobą do podejmowania takich decyzji jesteś Ty. W przeciwnym wypadku będziesz musiał zaufać, że analityk biznesowy oraz programiści trafnie odgadną Twoje wymagania. Nie ma przeszkód, aby tymczasowo oznaczać wymagania jako **do wyjaśnienia** (**TBD**) w celu wskazania, że potrzebne są dodatkowe ustalenia albo informacje. Czasami adnotacja TBD jest używana, ponieważ określone wymaganie jest trudne do określenia i nikt nie chce się nim zająć. Postaraj się wyjaśnić cel istnienia każdego z wymagań, dzięki czemu analityk biznesowy będzie mógł je poprawnie sformułować. Jest to najlepszy sposób na zagwarantowanie, że produkt spełni Twoje oczekiwania.

Obowiązek nr 4. Na prośbę analityka podejmować w porę decyzje

Podobnie jak wykonawca budujący Twój wymarzony dom, analityk biznesowy będzie Cię prosić o podjęcie wielu decyzji, wśród których można wymienić rozwiewanie wątpliwości wynikających ze sprzecznych próśb otrzymanych od różnych klientów, dokonywanie wyborów spośród niezgodnych ze sobą atrybutów jakościowych oraz ocenianie dokładności informacji. Klienci upoważnieni do podejmowania takich decyzji powinni bezzwłocznie rozstrzygać wątpliwości analityka, gdy zostaną o to poproszeni. Programiści nie będą mogli sprawnie kontynuować pracy, jeśli nie podejmiesz stosownych decyzji, a czas spędzony przez nich na czekaniu na Twoją odpowiedź może opóźnić rozwój produktu. Jeśli prośby o Twój czas staną się uciążliwe, pamiętaj, że system jest tworzony dla Ciebie. Analitycy biznesowi często potrafią pomagać innym osobom w podejmowaniu decyzji, jeśli więc utkniesz w miejscu, będziesz mógł poprosić ich o pomoc.

Obowiązek nr 5. Respektować oszacowania programistów dotyczące kosztu oraz możliwości realizacji wymagań

Wszystkie funkcje oprogramowania mają swój koszt. Najbardziej kompetentnymi osobami do szacowania tego kosztu są programiści. Implementacja niektórych funkcji może być niewykonalna technicznie lub zaskakująco kosztowna. Niektóre z wymagań mogą wymagać nieosiągalnej w danym środowisku wydajności albo korzystania z danych, które nie są dostępne w systemie. Programista może odgrywać rolę posłańca złych wieści na temat wykonalności albo kosztów. Powinieneś szanować jego ocenę, nawet jeśli oznacza ona, że nie otrzymasz tego, o co prosiłeś w dokładnie takiej postaci, jaką sobie wyobrażałeś. Czasami będziesz mógł zmienić swoje wymagania w taki sposób, aby ich realizacja stała się możliwa albo tańsza. Na przykład spełnienie prośby, aby jakaś reakcja następowała „natychmiastowo", może być niewykonalne, ale precyzyjniejsze określenie ram czasowych („do 50 milisekund") może sprawić, że osiągnięcie założonego celu będzie możliwe.

Obowiązek nr 6. We współpracy z programistami określać realne priorytety wymagań

Tylko w przypadku nielicznych projektów można wygospodarować wystarczające zasoby oraz czas, aby zaimplementować wszystkie funkcjonalności, które chcą mieć do dyspozycji klienci. Określenie, które funkcje są niezbędne, które przydatne, a bez których klienci mogą się obyć, stanowi ważną część analizy wymagań. W określaniu priorytetów wymagań to Ty odgrywasz wiodącą rolę. Programiści mogą udostępniać informacje o kosztach i ryzyku związanym z każdym wymaganiem lub opowieści użytkowników pomagające ustalić ostateczne priorytety. Jeśli priorytety, jakie ustanowisz, będą realne, pomożesz programistom wytworzyć maksymalną wartość przy najniższym koszcie i we właściwym czasie. Wspólne ustalanie priorytetów jest kluczem do sukcesu w projektach zwinnych, dzięki czemu programiści mogą zacząć oddawać użyteczne oprogramowanie tak szybko, jak to jest tylko możliwe.

Powinieneś uszanować oszacowanie zespołu programistycznego określające, jaka część żądanej funkcjonalności może zostać oddana w zadanym czasie i przy istniejących ograniczeniach zasobów. Jeśli wszystko, co jest Ci potrzebne, nie mieści się w zakresie projektu, osoby podejmujące decyzje będą musiały w oparciu o priorytety ograniczyć zakres projektu, wydłużyć harmonogram albo wygospodarować dodatkowe fundusze lub ludzi. Nadawanie każdemu wymaganiu wysokiego priorytetu nie jest rozsądne ani nie świadczy o dobrej współpracy.

Obowiązek nr 7. Przeglądać wymagania oraz oceniać prototypy

Jak przekonasz się w rozdziale 17., „Walidacja wymagań", przeglądy koleżeńskie należą do najwydajniejszych aktywności związanych z produkcją oprogramowania, jakie są dostępne. Dopuszczenie klientów do uczestnictwa w przeglądach stanowi klucz do stwierdzenia, czy wymagania wykazują pożądane charakterystyki dotyczące ich kompletności, poprawności oraz wymagalności. Przegląd daje także okazję przedstawicielom klienta do oceny, w jakim stopniu praca wykonywana przez analityka biznesowego zaspokaja potrzeby projektu. Zapracowani klienci często niechętnie poświęcają czas na weryfikowanie wymagań, ale ich zaangażowanie zwróci się z nawiązką. Analityk biznesowy powinien Ci przekazywać wymagania do weryfikacji w ciągu całego procesu ich pozyskiwania w możliwych do ogarnięcia fragmentach. Nie powinien rzucić na Twoje biurko grubego tomu z dokumentacją, gdy gromadzenie wymagań zostanie już przez niego „zakończone".

Trudno wyobrazić sobie, jak będzie działać oprogramowanie wyłącznie na podstawie wymagań sformułowanych na piśmie. Aby lepiej zrozumieć Twoje potrzeby i poznać sposoby na ich zaspokojenie, analityk biznesowy albo programiści budują czasami prototypy przyszłego produktu. Twoja opinia na temat tych wstępnych, częściowych albo próbnych implementacji stanowi dla programistów bardzo cenną informację.

Obowiązek nr 8. Ustanawiać kryteria akceptacji

Skąd programiści wiedzą, że skończyli? Skąd wiedzą, że oprogramowanie, które utworzyli, spełnia wymagania różnych klientów? Ponieważ jesteś klientem, jednym z Twoich zadań będzie określanie kryteriów akceptacji, czyli wstępnie zdefiniowanych warunków, które powinien spełnić produkt, aby mógł zostać uznany za gotowy do odbioru. Do kryteriów tych należą testy akceptacji, weryfikujące, czy produkt pozwala użytkownikom prawidłowo wykonywać pewne ważne czynności biznesowe. Inne kryteria akceptacji mogą dotyczyć nieprzekroczenia szacowanego poziomu defektów, osiągnięcia założonej wydajności przeprowadzania określonych operacji w środowisku roboczym albo możliwości spełnienia wymagań stawianych przez certyfikaty zewnętrzne. W celu wyłonienia szczegółów zawartych w opowieściach użytkowników projekty zwinne w znacznie większym stopniu bazują na testach akceptacji niż na wymaganiach podanych w formie pisemnej. Testerzy są w stanie ocenić, czy dane wymaganie zostało prawidłowo zaimplementowane, ale nie zawsze będą dokładnie wiedzieć, co **Twoim** zdaniem jest możliwym do przyjęcia efektem pracy oprogramowania.

Obowiązek nr 9. Bezzwłocznie zgłaszać zmiany wymagań

Nieustannie zmieniające się wymagania w poważnym stopniu zagrażają terminowemu oddaniu przez zespół programistyczny wysokiej jakości produktu. Zmian nie da się uniknąć i często są one cenne, ale im później w procesie tworzenia oprogramowania nastąpi ich wdrożenie, tym większe będzie ich oddziaływanie na projekt. Gdy tylko zdasz sobie sprawę z tego, że musisz zmienić wymaganie, poinformuj o tym analityka biznesowego. Aby ograniczyć negatywny wpływ zmian, kieruj się zdefiniowanym dla projektu procesem kontrolowania zmian. Dzięki temu zagwarantujesz, że zmiany nie zostaną pominięte, wpływ każdej z nich zostanie przeanalizowany, a wszystkie proponowane zmiany będą konsekwentnie rozpatrywane. W rezultacie interesariusze będą mogli podejmować słuszne decyzje dotyczące wprowadzania określonych zmian na właściwych etapach prac nad projektem.

Obowiązek nr 10. Uszanować proces opracowywania wymagań

Pozyskiwanie oraz określanie wymagań należą do największych wyzwań związanych z tworzeniem oprogramowania. Podejście analityka biznesowego do prac związanych z wymaganiami ma swoje racjonalne uzasadnienie. Chociaż możesz przy tym odczuwać frustrację, czas, jaki poświęcisz na zrozumienie wymagań, jest doskonałą inwestycją. Cały proces stanie się mniej uciążliwy, jeżeli uszanujesz techniki stosowane przez analityka biznesowego. Nie wahaj się prosić go o wyjaśnienie, do czego potrzebne mu są określone informacje, albo dlaczego chce, abyś brał udział w czynnościach związanych z pozyskiwaniem wymagań. Wzajemne zrozumienie oraz szacunek dla sposobu pracy oraz potrzeb drugiej osoby w znacznym stopniu pomagają w ustanowieniu skutecznej współpracy, być może nawet dającej satysfakcję.

Tworzenie kultury poszanowania wymagań

Szefowa firmowego działu odpowiedzialnego za wymagania zwierzyła się kiedyś z następującego problemu:

— Mam kłopot z uzyskaniem od niektórych z naszych programistów zgody na ich uczestnictwo w pracach nad wymaganiami. W jaki sposób mogę wytłumaczyć im, jaką wartość ma ich wkład?

W innej organizacji analityk biznesowy był świadkiem konfliktu między programistami zbierającymi szczegółowe informacje na temat danych wprowadzanych do systemu księgowego a szefem działu IT, który w kwestii wymagań chciał przeprowadzić burzę mózgów bez uciekania się do jakiejkolwiek techniki pozyskiwania wymagań. Analityk ten zapytał mnie, czy czytelnicy mojej książki są narażeni na podobny konflikt kulturowy.

Problemy te egzemplifikują wyzwania, z jakimi można się spotkać podczas prób zaangażowania analityków biznesowych, programistów oraz klientów do wspólnej pracy nad wymaganiami. Być może sądzisz, że dla użytkownika oczywisty jest fakt, iż z większym prawdopodobieństwem otrzyma to, czego potrzebuje, jeśli przedstawi swoje wymagania. Programiści powinni zdawać sobie sprawę z tego, że ich udział w procesie pozyskiwania wymagań ułatwi im życie i że nie będą później dostawać po głowach nadlatującymi ze wszystkich stron dokumentami zawierającymi wymagania. Oczywiście nie wszyscy są aż tak entuzjastycznie nastawieni do pracy nad wymaganiami, jak Ty. Gdyby tak było, prawdopodobnie każdy z nich zostałby analitykiem biznesowym!

Konflikty kulturowe często pojawiają się wtedy, gdy zespoły programistyczne pracują nad wymaganiami. Istnieją osoby, które znają ryzyko związane z próbami tworzenia oprogramowania w oparciu o minimalne lub komunikowane telepatycznie wymagania. Istnieją też osoby uważające, że wymagania są zbędne. Nawiązanie współpracy przy takich przedsięwzięciach, jak wymiana istniejącego systemu, może być trudne, jeśli użytkownicy są przekonani, że dane przedsięwzięcie nie ma związku z ich obowiązkami służbowymi i nie warto poświęcać na nie czasu. Zrozumienie,

dlaczego ludzie nie chcą uczestniczyć w opracowywaniu wymagań, stanowi pierwszy krok w kierunku rozwiązania tego problemu.

Być może niechętnym osobom nie przedstawiono konkretnych praktyk związanych z określaniem wymagań, a może wyniosły one złe doświadczenia ze źle przeprowadzonego procesu implementacji wymagań, uczestnicząc w działaniach, których efektem była obszerna, niekompletna i pomijająca rzeczywiste wymagania specyfikacja. Takie wspomnienia u każdego wywoływałyby niesmak. Prawdopodobnie osoby te nie rozumieją i nie doceniają wartości prawidłowo realizowanych praktyk. Mogą też nie zdawać sobie sprawy z ceny, jaką musiały płacić, pracując w przeszłości w prowizorycznym i źle ustrukturyzowanym środowisku. Ceną tą jest najczęściej konieczność wprowadzania poprawek skutkujących opóźnieniami w oddawaniu finalnych wersji produktu oraz słabej jakości oprogramowanie. Tego typu poprawki są dokonywane przez osoby uczestniczące w projekcie w ramach ich codziennych zadań, w związku z czym nikt nie dostrzega w nich przejawu nieefektywnej pracy.

Jeśli próbujesz zaangażować programistów, menedżerów oraz klientów, upewnij się, że wszyscy z nich zrozumieją, iż niedogodności, z jakimi wcześniej miała do czynienia organizacja oraz jej klienci, wynikały z niewłaściwie zdefiniowanych wymagań. Osobom, które osobiście nie doświadczyły żadnych problemów, pokaż przykłady niekorzystnego wpływu, jaki wywierają niewłaściwe wymagania. Wyraź ich koszt w jednostkach, które będą miały sens w danej organizacji; mogą to być pieniądze, czas, niezadowolenie klienta albo utracone korzyści biznesowe. Menedżerowie ds. rozwoju zwykle nie zdają sobie sprawy, jak bardzo niedociągnięcia w opracowaniu wymagań ograniczają produktywność ich zespołów. Pokaż im zatem, w jakim stopniu niewłaściwe wymagania spowalniają prace nad projektami oraz prowadzą do obszernych i drogich korekt kursu.

Programiści pełnią w projekcie rolę interesariuszy, ale czasami ich wkład nie jest uwzględniany i stają się oni „ofiarami" wymagań, które im narzucono. Programiści powinni zatem odnieść korzyści z udzielania informacji, które sprawią, że dokumentacja wymagań będzie przydatna i zrozumiała w możliwie najwyższym stopniu. Osobiście wolę weryfikować wymagania programistów w miarę ich ewoluowania. Dzięki temu programiści będą wiedzieć, z czym mają do czynienia i zauważą kwestie wymagające dodatkowych wyjaśnień. Wkład programisty jest również potrzebny podczas specyfikowania wewnętrznych atrybutów jakościowych, które nie są widoczne dla użytkowników. Programiści mogą zgłaszać sugestie, o których nikt wcześniej nie pomyślał — prostsze sposoby na realizowanie pewnych czynności, funkcjonalności zbyt czasochłonne do zaimplementowania, niepotrzebnie narzucone ograniczenia, brakujące wymagania (takie jak metody obsługiwania wyjątków), a także okazje do skorzystania z przewagi, jaką dają nowe technologie.

Zespół kontroli jakości oraz testerzy także wnoszą cenny wkład w opracowywanie doskonałych wymagań. Zamiast czekać na ukończenie produktu, zaangażuj te spostrzegawcze osoby na wczesnym etapie iteracyjnych przeglądów wymagań. Z dużym prawdopodobieństwem podczas opracowywania przypadków użycia i scenariuszy znajdą one wiele niejednoznaczności, konfliktów oraz wątpliwości związanych z wymaganiami. Testerzy mogą także udzielić informacji na temat specyfikowania weryfikowalnych wymagań dotyczących atrybutów jakościowych.

Opór wobec zmian w procesie albo kulturze może wskazywać na obawy, brak pewności lub niedostateczną wiedzę. Jeśli zdołasz określić przyczyny tego oporu, będziesz mógł go załagodzić za pomocą wsparcia, wyjaśnień i edukacji. Pokaż tym osobom, że ich wkład nie leży wyłącznie w ich własnym interesie, ale także przyczynia się do lepszych rezultatów osiąganych przez cały zespół.

Kierownictwo organizacji musi zrozumieć potrzebę prowadzenia skutecznej analizy biznesowej oraz inżynierii wymagań w ramach strategicznych działań firmy. Chociaż ukierunkowane na dany projekt działania oraz oddolne inicjatywy odgrywają ważną rolę, żadne z osiągniętych ulepszeń ani korzyści prawdopodobnie nie przetrwają zakończenia prac nad projektem ani reorganizacji firmy, jeśli kierownictwo także nie będzie zaangażowane w pracę nad wymaganiami.

Identyfikowanie osób decyzyjnych

W przypadku projektów programistycznych mogą być podejmowane setki decyzji. Bardzo często mają one krytyczne znaczenie dla możliwości kontynuowania dalszych prac. Może pojawić się potrzeba rozwiązania jakiegoś konfliktu, przyjęcia (albo odrzucenia) zaproponowanej zmiany lub zatwierdzenia zestawu wymagań odnoszących się do określonego wydania produktu. Już na początku prac nad projektem określ, kto będzie podejmować decyzje i w jaki sposób będzie to robić. Mój znajomy, Chris, który jest doświadczonym menedżerem projektu, powiedział mi kiedyś: „Stwierdziłem, że zazwyczaj w projekcie jest jedna osoba podejmująca decyzje. Może to być główny sponsor w ramach danej organizacji. Nie poddaję się, dopóki nie zidentyfikuję tej osoby, po czym pilnuję, aby zawsze była ona informowana o postępach prac nad projektem". Nie ma definitywnej odpowiedzi na pytanie, kto powinien podejmować kluczowe decyzje. Zwykle najlepiej sprawdza się niewielka grupa odpowiedzialna za główne obszary systemu — mogą to być zarząd, klienci, analitycy biznesowi, programiści albo dział marketingu. W rozdziale 28., „Zmiany się zdarzają", opisano radę kontroli zmian, która podejmuje decyzje dotyczące wcielania w życie proponowanych zmian w wymaganiach.

Grupa odpowiedzialna za podejmowanie decyzji powinna wybrać **lidera ds. decyzji** oraz określić **regułę decyzyjną**, opisującą sposób podejmowania decyzji. Istnieje wiele różnych reguł decyzyjnych, z których można wybierać. Należą do nich między innymi następujące reguły (Gottesdiener, 2001):

- Wybory podejmuje lider ds. decyzji i omawia je (lub nie) z pozostałymi osobami w grupie.
- Grupa głosuje i podejmowana jest decyzja większości.
- Grupa głosuje, ale do zatwierdzenia decyzji potrzebna jest jednomyślność.
- Grupa prowadzi dyskusję i negocjuje w celu osiągnięcia porozumienia. Wszyscy muszą zaakceptować decyzję i popierać ją.
- Lider ds. decyzji może przekazać odpowiedzialność za podjęcie decyzji określonej osobie.
- Decyzję podejmuje grupa, ale istnieje możliwość zgłoszenia weta do tej decyzji.

Nie istnieje uniwersalnie słuszna ani najlepsza reguła decyzyjna. Dana reguła nie sprawdzi się w każdej sytuacji, w związku z czym grupa powinna przyjąć wytyczne, dzięki czemu będzie wiadomo, kiedy głosować, kiedy dążyć do porozumienia, kiedy przekazywać odpowiedzialność itd. Osoby, które będą podejmować decyzje związane z wymaganiami, powinny wybrać regułę decyzyjną, zanim jeszcze skonfrontują się ze swoją pierwszą poważną decyzją.

Osiąganie porozumienia co do wymagań

Osiągnięcie porozumienia co do wymagań produktu lub jego fragmentu, jaki ma zostać utworzony, leży u podstaw partnerstwa klient-twórca oprogramowania. W osiągnięcie tej zgody zaangażowanych jest wiele stron:

- Klienci zgadzają się, że wymagania zaspokajają ich potrzeby.
- Programiści zgadzają się, że zrozumieli wymagania i że są one wykonalne.
- Testerzy zgadzają się, że wymagania są weryfikowalne.
- Zarząd zgadza się, że wymagania pozwolą osiągnąć jego cele biznesowe.

W wielu organizacjach stosowane jest kwitowanie wymagań na znak ich akceptacji przez interesariusza. Wszyscy uczestnicy procesu zatwierdzania wymagań powinni zdawać sobie sprawę z tego, co takie kwitowanie oznacza i jakie problemy mogą z tego wyniknąć. Jednym z takich zagrożeń jest traktowanie przez przedstawiciela klientów lub menedżera projektu podpisywania wymagań jako pozbawionego znaczenia rytuału: „Dostałem jakąś kartkę papieru z moim nazwiskiem, więc ją podpisałem, bo inaczej programiści nie mogliby zacząć pisać kodu". Takie podejście może prowadzić do wystąpienia w przyszłości problemów, gdy osoba, która podpisała dokument, jest zaskoczona tym, co zostało wyprodukowane: „No pewnie, że pokwitowałem wymagania, ale nie miałem czasu, żeby je wszystkie czytać. Zaufałem wam, chłopaki, a wy mnie zawiedliście".

Równie problematyczne jest traktowanie przez menedżera ds. rozwoju kwitowania jako sposobu na zamrożenie wymagań. Gdy tylko nastąpi zmiana wymagań, zaprotestuje on: „Przecież pokwitowaliście wymagania, tak więc właśnie to będziemy budować. Skoro chcieliście czegoś innego, trzeba było to powiedzieć".

Obie te postawy ignorują fakt, że nie da się poznać wszystkich wymagań na wczesnym etapie projektu i że wymagania bez wątpienia ulegną zmianie wraz z upływem czasu. Zatwierdzanie zestawu wymagań jest słusznym działaniem, które zamyka pewien etap prac nad wymaganiami. Wszyscy uczestnicy projektu powinni jednak dokładnie zdawać sobie sprawę, co oznacza ich podpis złożony na dokumencie.

Ważne. Nie stosuj kwitowania jako broni. Uważaj je za kamień milowy; za jasne wyrażenie wspólnego porozumienia co do czynności, które doprowadziły do złożenia podpisów na dokumencie, oraz skutków, jakie będą one wywierać na przyszłe zmiany. Jeśli osoby decyzyjne nie muszą czytać każdego słowa w specyfikacji wymagań, wybierz jakąś technikę komunikacji, taką jak pokaz slajdów, która pozwoli podsumować najważniejsze elementy i ułatwi szybkie osiągnięcie porozumienia.

Baza dla wymagań

O wiele ważniejsze od rytuału kwitowania jest ustanowienie **bazy odniesienia** dla porozumienia dotyczącego wymagań, tj. obrazu chwili, który reprezentuje obecne porozumienie (Wiegers, 2006). Baza odniesienia jest zweryfikowanym i zatwierdzonym zbiorem wymagań, które służą jako podstawa do dalszego rozwijania projektu. Obojętne, czy Twój zespół stosuje formalny proces kwitowania, czy też jakiś inny sposób na osiąganie porozumienia w kwestii wymagań, tekst takiego porozumienia powinien brzmieć mniej więcej następująco:

> *Potwierdzam, że powyższy zbiór wymagań przedstawia nasze rozumienie wymagań, które dotyczą kolejnego etapu prac nad projektem, oraz że opisane tu rozwiązania spełnią nasze potrzeby, jak je w obecnej chwili rozumiemy. Wyrażam zgodę na wprowadzanie w niniejszej bazie przyszłych zmian dotyczących wymagań, zgodnie ze zdefiniowanym dla projektu procesem zmian. Zdaję sobie sprawę z faktu, że zmiany mogą wymagać renegocjowania zobowiązań dotyczących kosztów, zasobów oraz harmonogramu.*

W niektórych organizacjach podobny do powyższego zapis jest umieszczany na stronie z podpisami, dzięki czemu osoby parafujące wymagania będą dokładnie wiedzieć, co kwitowanie oznacza w ich sytuacji.

Porozumienie, które jest osiągnięte w podobny sposób, pomaga zredukować konflikty, które mogą się pojawić, gdy zostaną ujawnione przeoczone wymagania lub kiedy w czasie prac nad projektem zmienią się wymagania rynkowe albo biznesowe. Dobrze zdefiniowany proces tworzenia bazy odniesienia pozwala zachować wszystkim głównym interesariuszom pewność co do następujących sytuacji:

- Dział kontaktu z klientem albo dział marketingu jest pewny, że zakres projektu nie wymknie się spod kontroli, ponieważ klienci weryfikują decyzje dotyczące zmiany zakresu projektu.
- Przedstawiciele użytkowników są pewni, że zespół programistyczny będzie z nimi współpracować w celu znalezienia właściwego rozwiązania, nawet jeśli w chwili rozpoczęcia prac nie wzięto pod uwagę wszystkich wymagań.
- Kierownictwo zespołu programistycznego może być spokojne, gdyż zespół programistyczny ma partnera biznesowego, który dopilnuje, aby praca nad projektem była skupiona na osiągnięciu założonych celów, i który współpracuje z zespołem w celu zrównoważenia harmonogramu, kosztów, funkcjonalności oraz jakości.
- Analitycy biznesowi oraz menedżerowie projektu są pewni, że są w stanie zapanować nad zmianami w sposób, który ograniczy do minimum chaos.
- Zespół kontroli jakości oraz testerzy mogą bez obaw opracowywać skrypty testowe i przygotowywać się do wykonywania swoich zadań w ramach projektu.

Gdy osoby decyzyjne zdefiniują już bazę odniesienia, analityk biznesowy powinien zacząć kontrolować zmiany, jakim są poddawane wymagania. Dzięki temu zespół będzie mógł w razie konieczności zmienić zakres projektu w kontrolowany sposób, który uwzględnia badanie wpływu proponowanych zmian na harmonogram prac oraz bierze pod uwagę pozostałe czynniki mające wpływ na osiągnięcie sukcesu. Potwierdzenie jednoznaczną umową wstępnych czynności związanych z opracowywaniem wymagań pozwala kształtować współpracę klient-twórca oprogramowania na drodze zmierzającej do osiągnięcia sukcesu w projekcie.

Co zrobić, jeśli nie osiągnięto porozumienia?

Zdobycie pokwitowań od wszystkich istotnych interesariuszy może być trudne. Przeszkody mogą wynikać z przyczyn logistycznych, napiętych terminarzy, a także z niechęci niektórych osób do zaciągania zobowiązań, z których będą później rozliczani. Jeśli interesariusze obawiają się, że po zatwierdzeniu wymagań nie będą mogli ich zmieniać, mogą zwlekać z ich pokwitowaniem. Taka sytuacja może spowodować wpadnięcie w straszną pułapkę paraliżu analitycznego. Wiele zespołów próbowało wysyłać wiadomości e-mail o następującej treści: „Jeśli do najbliższego piątku nie zadeklarujesz się co do swoich zmian albo ich nie pokwitujesz, uznamy, że akceptujesz wymagania w ich obecnej postaci". Jest to jedna z dostępnych opcji, ale tak naprawdę **nie** jest ona równoznaczna z osiągnięciem porozumienia. Opcja ta wiąże się także z ryzykiem wytworzenia napiętych relacji z tymi interesariuszami, co do których założyłeś ich milczącą zgodę. Postaraj się zrozumieć, dlaczego nie czują się oni komfortowo, musząc kwitować wymagania, i postępuj stosownie do sytuacji.

W takich przypadkach postąpisz lepiej, ruszając ostrożnie do przodu przy jednoczesnym założeniu, że nie uzyskałeś potwierdzeń od wszystkich niesubordynowanych interesariuszy. Odnotuj na swojej liście zagrożeń fakt, że pewni interesariusze nie pokwitowali wymagań, łącznie z przewidywanym skutkiem, jaki brak części wymagań lub ich nieprawidłowe określenie wywrze na projekt. Skontaktuj się z tymi osobami w ramach czynności związanych z zarządzaniem ryzykiem. W pozytywny sposób powiedz im, że zdajesz sobie sprawę, iż nie zatwierdzili oni jeszcze wymagań, ale że prace nad projektem posuwają się do przodu z obecnymi wymaganiami stanowiącymi bazę odniesienia po to, aby nie opóźniać projektu. Poinformuj ich, że jeśli chcą wprowadzić zmiany, istnieje służący do tego celu proces. Powinieneś postępować w taki sposób, jakby interesariusz istotnie zatwierdził wymagania, a Ty zajmujesz się tylko nawiązaniem z nim kontaktu.

Zgoda co do wymagań w projektach zwinnych

W projektach zwinnych nie ma miejsca na formalne pokwitowania. W tego typu projektach zarządzanie wymaganiami przyjmuje zwykle formę opowieści użytkowników w rejestrze wymagań. Właściciel produktu oraz zespół osiągają podczas sesji planistycznych porozumienie co do opowieści, które zostaną zaimplementowane w kolejnej iteracji. Zestaw opowieści jest dobierany na podstawie ich priorytetów oraz sprawności (wydajności) zespołu. Po wybraniu i zatwierdzeniu zestawu opowieści w nim zawarte zostają zamrożone. Zgłoszone zmiany w wymaganiach zostaną rozpatrzone w przyszłych iteracjach. W projektach zwinnych nie próbuje się z góry uzyskać zatwierdzenia pełnego zakresu wymagań przez interesariusza. W takich projektach pełny zbiór funkcjonalności wyłania się wraz z upływem czasu, chociaż wizja oraz pozostałe wymagania biznesowe powinny zostać zdefiniowane już na początku. W rozdziale 20., „Projekty zwinne", omówiono, jak obsługiwane są wymagania w programowaniu zwinnym.

Pracowałem kiedyś z klientem, który domagał się kwitowania wymagań, chociaż projekt był tworzony zgodnie z metodyką zwinną. Nasz zespół musiał wykazać się kreatywnością, aby dawać sobie radę w kontekście, w którym zwykle nie stosuje się kwitowania. Zespół analityków biznesowych współpracował blisko z użytkownikami w celu pozyskiwania wymagań w postaci opowieści użytkowników oraz innych modeli, takich jak przepływy procesów oraz tabele stanów. Prosiliśmy użytkowników, aby „kwitowali" te dokumenty w odpowiednich momentach, gdy nie było żadnych brakujących wymagań, **o których mogliby oni wiedzieć**, a także gdy nie było żadnych zastrzeżeń co do spisanych przez nas wymagań, **które mogliby oni zgłosić**. Ponieważ użytkownicy brali udział w czynnościach związanych z określaniem wymagań, programiści nie pracowali nad rozwiązaniem, które daleko odbiegałoby od przyjętych założeń. Taka definicja „kwitowania" daje również użytkownikom prawo do późniejszego stwierdzenia, że potrzebne im jest nowe rozwiązanie albo że coś zdefiniowali źle.

W odróżnieniu od historycznego znaczenia kwitowania jako „zatwierdzania i zamrażania wszystkich wymagań na wstępie" tego typu podejście nie spycha nikogo w róg, gdzie można się poczuć, jakby się całym swoim życiem poświadczało ogromny dokument, który z trudem można zrozumieć. Klienci także nie są zmuszani do potwierdzenia, że wymagania są bliskie ideałowi i że wszystkie kwestie zostały uwzględnione już na samym początku. Taki sposób kwitowania pozwala zachować istotę metod zwinnych. Podobnie jak to jest w przypadku opisanego wcześniej procesu kwitowania, jego istotą jest osiągnięcie porozumienia co do pewnego zestawu wymagań — bazy odniesienia — który zostanie zaimplementowany w następnym cyklu wytwórczym; porozumienia dającego jasny obraz oraz wspólne rozumienie tego, co tak naprawdę ono oznacza.

Zazwyczaj w projektach zwinnych to właściciel produktu jawnie zatwierdza lub odrzuca wymagania do wdrożenia w danej iteracji. Wymagania te składają się ze zbioru opowieści wraz z towarzyszącymi im kryteriami akceptacji oraz testami akceptacyjnymi. Ostatecznym „pokwitowaniem" jest przyjęcie działającego i przetestowanego oprogramowania, które zostało otrzymane w danej iteracji.

Jak sformułowała to konsultantka Nanette Brown: „Nawet w środowisku zwinnym koncepcja kwitowania może pełnić ważną funkcję. Proces zwinny nakazuje nam »objąć zmianę«, ale pojęcie zmiany funkcjonuje wyłącznie z punktem odniesienia. Nawet w zespołach, w których istnieje dobra komunikacja, poszczególne osoby mogą różnie interpretować bieżące plany oraz aktualny status projektu. Dla jednej osoby »zmiana« może oznaczać to, co inna osoba uważa za coś już zatwierdzonego. Nie ma jednak przeszkód, abyś uważał kwitowanie za błahą czynność, potwierdzającą, że »jesteśmy tutaj«. Dzisiejsze »jesteśmy tutaj« nie oznacza, że jutro nie możemy być gdzie indziej, ale przynajmniej dzięki temu wiemy, że osiągnęliśmy porozumienie oraz wspólny punkt odniesienia".

Następne kroki

- Zidentyfikuj klientów, w tym użytkowników końcowych, którzy są odpowiedzialni za definiowanie w Twoim projekcie wymagań biznesowych oraz wymagań użytkowników. Które z zapisów zawartych w karcie praw oraz karcie obowiązków stosują oni w praktyce? Których nie stosują?
- Przedyskutuj kartę praw ze swoimi głównymi klientami, aby dowiedzieć się, czy ich zdaniem korzystają oni ze swoich praw. Omów kartę obowiązków, aby osiągnąć porozumienie co do obowiązków, których mogą oni przestrzegać. Zmień kartę praw oraz kartę obowiązków w taki sposób, aby wszystkie strony zgodziły się co do zasad opisujących Waszą współpracę. Sprawdź, czy wszyscy interesariusze zachowują równowagę między przysługującymi im prawami a obowiązkami.
- Jeśli jesteś klientem biorącym udział w projekcie programistycznym i uważasz, że Twoje prawa nie są przestrzegane w zadowalającym stopniu, omów kartę praw z menedżerem projektu albo analitykiem biznesowym. Ze swojej strony zadeklaruj chęć przestrzegania karty obowiązków w dążeniu do relacji opartej w większym stopniu na współpracy.
- Jeśli w Twojej organizacji istnieje formalny proces kwitowania, zastanów się, jaką rolę odgrywa on obecnie. Podejmij współpracę z menedżerem programistów albo szefem działu kontaktu z klientem (albo marketingu), aby osiągnąć porozumienie dotyczące faktycznego znaczenia kwitowania w ramach procesu zatwierdzania wymagań.
- W bieżącym projekcie lub w jednym z poprzednich projektów poszukaj przykładu na niedostateczne zaangażowanie klienta. Zastanów się, jaki wpływ miało to na projekt. Sprawdź, czy możesz oszacować ryzyko pod względem liczby późniejszych zmian w wymaganiach, czasu przeznaczonego na wprowadzanie poprawek w produkcie już po jego oddaniu lub straconych możliwości biznesowych. Na podstawie zdobytego doświadczenia wyciągnij wnioski na przyszłość, a także przekonaj inne osoby, że zaangażowanie klienta odgrywa ważną rolę.

Skorowidz

E

F

G

H

I

J

K

L

Ł

M

N

O

P

R

S

Ś

T

U

W

Z

Ź

Lektura obowiązkowa każdego analityka i osób odpowiedzialnych za wymagania!

Zebranie i opracowanie wymagań dotyczących tworzonego oprogramowania to jeden z fundamentów udanego projektu. Znajomość zakresu prac jest kluczową informacją dla wszystkich osób prowadzących projekt oraz bezcennym źródłem wiedzy dla deweloperów tworzących kod. Brzmi prosto, ale wcale tak nie jest! Identyfikacja interesariuszy, dokumentacja wymagań, określanie ich wartości biznesowej — to tylko niektóre z wyzwań stojących przed analitykami i ich zespołami!

Sięgnij po tę książkę, by uniknąć typowych problemów i pułapek. W kolejnych rozdziałach znajdziesz kluczowe informacje na temat wymagań dotyczących oprogramowania, roli analityka biznesowego oraz dobrych praktyk w inżynierii wymagań. Część II tej książki została poświęcona opracowywaniu wymagań. Dowiedz się, jak określać wymagania biznesowe, rozmawiać z użytkownikami oraz dokumentować i walidować wymagania. W prawdziwym świecie spotkasz się z różnymi typami projektów. W zależności od ich charakteru trzeba będzie na bieżąco dostosowywać poznane techniki. Projekty zwinne, projekty systemów wbudowanych, automatyzacja procesów biznesowych to tylko część z omawianych obszarów. Książka ta jest klasycznym podręcznikiem, obowiązkową lekturą każdego analityka oraz osób odpowiedzialnych za wymagania.

Dzięki tej książce:

- nauczysz się identyfikować interesariuszy oraz rozmawiać z klientami
- poznasz dobre praktyki w inżynierii wymagań
- zrozumiesz zadania analityka biznesowego
- ograniczysz ryzyko dzięki prototypowaniu
- poznasz projekty różnego typu
- zrozumiesz proces zarządzania wymaganiami

helion.pl
księgarnia internetowa

Nr katalogowy: 25286

Księgarnia internetowa
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

Microsoft Press

Helion

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Informatyka w najlepszym wydaniu

sięgnij po WIĘCEJ
KOD KORZYŚCI

ISBN 978-83-246-9166-1

cena: 99,00 zł