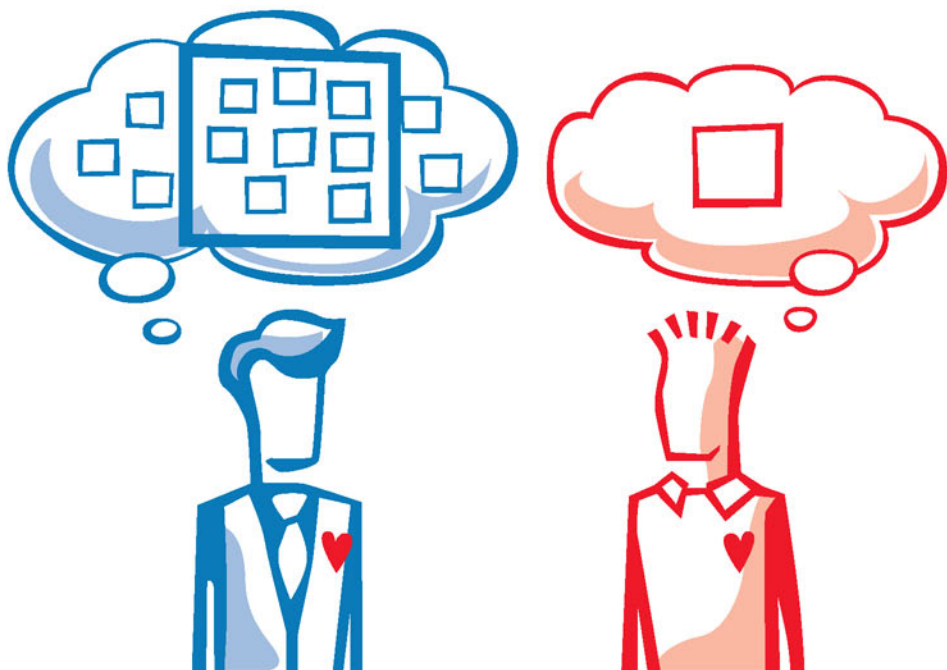


Michał Bartyzel

WYDANIE II  
ROZSZERZONE

# Oprogramowanie szyte na miarę



Jak rozmawiać z klientem,  
który nie wie, czego chce

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Magdalena Dragon-Philipczyk

Projekt okładki: ULABUKA

Rysunki: Maria Pankowska

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [onepress@onepress.pl](mailto:onepress@onepress.pl)

WWW: <http://onepress.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://onepress.pl/user/opinie/opszm2>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzje.

ISBN: 978-83-246-9826-4

Copyright © Michał Bartyzel 2015

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

# Spis treści

<b>Wstęp do wydania drugiego .....</b>	<b>9</b>
<b>Wstęp do wydania pierwszego .....</b>	<b>11</b>
<b>Rozdział 1. Między biznesem a IT .....</b>	<b>13</b>
Dla kogo jest ta książka? .....	14
Ile zarządzania jest w zarządzaniu wymaganiami? .....	17
Klient, który wie, czego chce .....	19
Podsumowanie .....	22
<b>Rozdział 2. Co to znaczy „myśleć biznesowo”? .....</b>	<b>23</b>
Nonszalancja programistów .....	23
Podsumowanie .....	29
<b>Rozdział 3. Wspólna wizja .....</b>	<b>31</b>
Czym jest wizja? .....	32
Wizja a zakres systemu .....	33
Formułowanie wizji .....	35
Nazwa jest ważna .....	36
Wizja jest współdzielona albo nie ma jej wcale .....	39
Kiedy wizja bywa niebezpieczna? .....	42
Podsumowanie .....	43
<b>Rozdział 4. Rozpoznanie procesu biznesowego .....</b>	<b>45</b>
Proces biznesowy, czyli co się dzieje u klienta .....	47
Podsumowanie .....	51

<b>Rozdział 5. Sztuka zadawania pytań .....</b>	<b>53</b>
Kto prowadzi konwersację? .....	53
Co to znaczy patrzeć z perspektywy klienta? .....	54
Świadomość słownictwa w trakcie konwersacji .....	58
Podsumowanie .....	61
<b>Rozdział 6. Odkrywanie potrzeb .....</b>	<b>63</b>
Czym jest potrzeba? .....	64
Podsumowanie .....	89
<b>Rozdział 7. Sztuka konwersacji z klientem .....</b>	<b>91</b>
Podstawowa struktura konwersacji .....	92
Konkretyzowanie wymagań .....	97
Uogólnianie wymagań .....	132
Radzenie sobie z impasem w trakcie konwersacji .....	144
Proponowanie alternatywnych rozwiązań .....	149
Podsumowanie .....	152
<b>Rozdział 8. Pytania trafiające w samo sedno .....</b>	<b>153</b>
Bufor bezużytecznych odpowiedzi .....	153
Pytania zamknięte i otwarte .....	160
Wyrażanie oczekiwań poprzez zaprzeczenie .....	162
Ramy odniesienia i zmiana ram .....	166
Podsumowanie .....	176
<b>Rozdział 9. Czy między nami jest chemia? .....</b>	<b>177</b>
Mechanika a chemia konwersacji .....	177
Parafrazowanie .....	178
Technika pozytywnej intencji .....	184
Przejmowanie kierunku konwersacji .....	189
Chemia konwersacji w duchu Nonviolent Communication .....	193
Podsumowanie .....	208
<b>Rozdział 10. Ustalanie priorytetów wymagań .....</b>	<b>209</b>
Pytanie i eliminowanie .....	210
Ważność a pilność .....	215
Excelowe czary-mary .....	222
Podsumowanie .....	225

<b>Rozdział 11. Spotkania .....</b>	<b>227</b>
Spotkania efektywne i te, podczas których tylko tracisz czas .....	229
Przygotowanie spotkania .....	231
Prowadzenie spotkania .....	241
Zamykanie spotkania .....	247
Formuły spotkań na różne okazje .....	248
Podsumowanie .....	256
<b>Rozdział 12. Techniki zbierania wymagań w pigułce ....</b>	<b>257</b>
Wizja .....	258
Odkrywanie potrzeb .....	259
Konkretyzowanie .....	260
Technika skrzynki .....	262
Ekran użytkownika .....	263
Uogólnianie .....	263
Powiększanie przestrzeni możliwych rozwiązań .....	265
Proponowanie alternatywnych rozwiązań .....	265
Pytania trafiające w samo sedno .....	266
Zmiana ram odniesienia .....	267
Parafrazowanie .....	268
Technika pozytywnej intencji .....	269
Przejmowanie kierunku rozmowy .....	270
Komunikacja według modelu NVC .....	271
Ustalanie priorytetów za pomocą pytań .....	272
Spotkania .....	274
<b>Rozdział 13. Kiedy techniki przedstawione w tej książce NIE zadziałają? .....</b>	<b>277</b>
<b>Lektura uzupełniająca .....</b>	<b>279</b>



## Rozdział 3.

# Wspólna wizja



RYSUNEK 3.1. Wspólna wizja

**W**yobraź sobie okno z przyciskiem *Zamknij*.  
Jak wygląda Twoje wyobrażenie? Jaki kolor ma okno?  
Po której stronie umieszczony jest przycisk? Czy przycisk ma obrazek,  
czy nie? Czy napis *Zamknij* ma aktywny znak? Z pewnością Twoje  
wyobrażenie jest różne od wyobrażeń innych czytelników i czytel-  
ników. To naturalne, że każdy z nas ma indywidualne preferencje  
i wyobrażenia. Każdy z nas jest inny. Poszczególne elementy naszej

indywidualności powstały w wyniku rozwoju społecznego lub, jak chcą niektórzy, są w jakiejś części uwarunkowane genetycznie. Skądkolwiek pochodzą, stanowią ważny składnik naszej osobowości. Naprawdę zabawnie zaczyna się robić, gdy kilka indywidualności spotyka się przy jednym stole i ma za zadanie stworzyć COŚ (z uwagi na różne oblicza tego CZEGOŚ nazwijmy to *produktem programistycznym* lub w uproszczeniu: *aplikacją, oprogramowaniem* albo *systemem*).

## Czym jest wizja?

Podobnie jak z wyobrażeniem sobie okna z przyciskiem, tak również w przypadku nowego systemu informatycznego opinie każdego z członków zespołu na temat tego, co należy zrobić, mogą być bardzo zróżnicowane. Tyle że w przypadku projektów informatycznych już nie chodzi o zabawę w wyobrażenia, lecz o bardzo wymierne korzyści lub straty liczone w twardej walucie. Z tego względu pierwszą rzeczą, którą należy zrobić podczas przygotowań do projektu, jest ustalenie ze wszystkimi zaangażowanymi osobami, do czego należy zmierzać. Innymi słowy: ustalenie wizji nowego systemu, którą będą współdziałać wszystkie osoby zaangażowane w projekt (rysunek 3.1).

## Kto jest odpowiedzialny za wizję?

Wizja oczywiście pochodzi od biznesu, gdyż to cele biznesowe system ma realizować. Jednak za ustalenie wizji, a następnie za jej realizowanie, odpowiedzialna będzie osoba, która otrzymała za zadanie zdefiniować, czego biznes potrzebuje, i na tej podstawie określić zakres prac dla IT. Najczęściej ta osoba nazywana jest klientem albo właścicielem produktu<sup>1</sup>. Jeśli więc pełnisz funkcję analityka, szczególnie uważnie przeczytaj ten rozdział.

---

<sup>1</sup> W języku angielskim *Product Owner*.



Ustalenie wizji systemu z biznesem to pierwszy krok w pracach nad nowym oprogramowaniem, jego kolejną wersją lub nad nowymi modułami.

Wizja jest po prostu „wyciągnięciem na wierzch” tego, co Ty, Twój klient i inne osoby zaangażowane w przedsięwzięcie myślicie, gdy zastanawiacie się nad pytaniem: *Jaki/czym będzie ten system?* Wyciągamy wszystkie odpowiedzi na powierzchnię, a potem ustalamy wspólną odpowiedź na wspomniane pytanie.

## Wizja a zakres systemu

Po pierwszym kontakcie z wizją może się nam wydawać, że to nie nieznaczące sformułowanie, które zniknie gdzieś w odmetach ogromnego dokumentu SRS<sup>2</sup>. Coś, co w dokumencie musi być, bo jest wymagane, ale nie warto przykładać do tego większej wagi. Nic bardziej mylnego! Wizja to niezwykle prosty i użyteczny sposób precyzowania zakresu systemu.

Wyobraź sobie następującą sytuację: ustaliłeś z klientem, że celem projektu będzie system do zarządzania kontaktami z klientami, w skrócie CRM<sup>3</sup>. Podczas jednego ze spotkań rozmowa przebiega następująco:

**Klient:** *Chciałbym, żeby doszła tu funkcjonalność raportowania sprzedaży kwartalnej.*

**Ty:** *Ale przecież nie mówiliśmy o tym wcześniej...*

**Klient:** *Oczywiście, że tak. Wspominałem o raportach sprzedaży.*

**Ty:** *Tak, ale półrocznych i rocznych. Kwartalnych nie.*

**Klient:** *No, raport to raport. Chyba nie będzie z tym kłopotu?*

**Ty:** *Hmmm...*

---

<sup>2</sup> W języku angielskim *Software Requirements Specification* — specyfikacja wymagań do oprogramowania.

<sup>3</sup> W języku angielskim *Customer Relationship Management*.

Przedstawiony dialog w ten czy inny sposób powtarza się w naszej pracy wyjątkowo często. Z grubsza rzecz biorąc, chodzi o to, **czy zgłaszana funkcjonalność mieści się w obszarze kontraktu i kto zapłaci za jej wykonanie**. Chodzi więc o **zakres** systemu, zakres prac, do wykonania których się zobowiązujesz<sup>4</sup>.

Założmy, że wizja systemu została zdefiniowana następująco: *CRM sprzedażowy to oprogramowanie webowe dla przedstawicieli handlowych, które pozwoli na bieżące wyznaczanie zadań sprzedażowych i uprości generowanie raportów na zamknięcie roku obrotowego.*

Podczas wspomnianego spotkania rozmowa mogłaby przebiegać następująco:

**Klient:** *Chciałbym, żeby doszła tu funkcjonalność raportowania sprzedaży kwartalnej.*

**Ty:** *Ale przecież nie mówiliśmy o tym wcześniej...*

**Klient:** *Oczywiście, że tak. Wspominałem o raportach sprzedaży.*

**Ty:** *Tak, lecz czy to mieści się w ustalonym zakresie prac, w którym CRM „upraszcza generowanie raportów na zamknięcie roku obrotowego”?*

**Klient:** *Hmmm...*

Posługując się wizją, mogłeś bez problemu ustalić, czy nowe wymaganie należy do zakresu prac, czy nie. Podkreślmy wyraźnie, że nie chodzi tu o wyprowadzenie klienta na manowce. Chodzi przede wszystkim o szybkie, jednoznaczne i nienaruszające relacji interpersonalnych określenie, czy nowa funkcjonalność mieści się w zakresie prac, czy nie (rysunek 3.2). Chodzi o jasną odpowiedź na pytanie: *Kto ostatecznie zapłaci za Twoją pracę?*

Nawet najciekawsze technicznie systemy tworzone są z założeniem, że „zarobią na sobie”. Każda aktywność podjęta w projekcie jest

---

<sup>4</sup> Dla uproszczenia toku myślowego pomijam rozróżnienie różnych sposobów kontraktowania pracy: *fixed price* oraz *time & material*. Nie jest to szczególnie istotne dla omawianego tematu.



KLIENT

WYKONAWCA

RYSUNEK 3.2. Które wymagania wchodzą do zakresu prac?

w konsekwencji i tak przeliczana na pieniądze. Pieniądze, które musisz wydać Ty lub Twój klient. To, że wizja pozwala w prosty sposób bez zbędnych konfliktów zaliczyć zlecone prace na Twoje konto lub na konto Twojego klienta, czyni z niej praktyczne i potężne narzędzie współpracy z biznesem.

## Formułowanie wizji

Zgodnie z naszą definicją wizja jest wspólną odpowiedzią wszystkich zaangażowanych osób z biznesu na pytanie: *Jaki/czym będzie ten system?* W tej definicji istotne jest słowo *wspólna*. Wizja powinna być wypracowana *wspólnie* z klientem (z kluczowymi decydentami). Może się to odbyć w trakcie moderowanego przez Ciebie spotkania. Wtedy powstaje wizja, cel, do którego wszyscy będą od tej chwili zmierzać.

Kolejnym krokiem jest **komunikowanie wizji** wśród reszty zespołu i decydentów, aby każda osoba zaangażowana w projekt miała to samo rozumienie końcowego efektu.

## Wizja jako krótki opis

Najprostszym sformułowaniem wizji jest krótki opis w postaci zdania:

*System <NAZWA> jest to <PRZEZNACZENIE,  
GŁÓWNE FUNKCJONALNOŚCI>.*

System CRM sprzedażowy to narzędzie CRM dla sprzedawców, które ujednolica sposób przechowywania informacji na temat klientów.

Powyższy schemat można nieco uszczegółowić, tak aby przekazywał precyzyjniejszą informację, nie tracąc nic ze swej zwięzłości oraz lekkości:

*System <NAZWA> jest to <WIĘKSZA KATEGORIA>  
w <CZYJA? GDZIE?> dla <GŁÓWNI UŻYTKOWNICY>  
odpowiedzialna za <GŁÓWNE USŁUGI,  
FUNKCJONALNOŚCI>.*

ABCMobi to aplikacja mobilna firmy ABX Taxi w Łodzi dla pasażerów komunikacji miejskiej odpowiedzialna za: wyszukiwanie najkrótszych lub najszybszych połączeń tramwajowych, zbieranie informacji o zapytaniach wpisywanych przez użytkowników i proponowanie atrakcyjniejszych niż dane wyszukiwanie przejazdów taksówką.

## **Arkusz wizji**

Szczegółową metodę konsolidowania wizji zaproponował Karl Wiegers w książce *Software Requirements, Second Edition* w postaci siedmiopunktowego arkusza wizji (tabela 3.1).

## **Nazwa jest ważna**

Jeden z moich klientów nazwał kiedyś system, nad którym wspólnie pracowaliśmy, *zintegrowanym systemem informatycznym (ZSI)*. Ta z pozoru niewinna nazwa spowodowała sporo zamieszania. Byłem niemal zmuszony do godzenia się na wszystkie kolejne funkcjonalności.

TABELA 3.1. Arkusz wizji

Element wizji	Przykład
Jak się nazywa produkt?	<i>System „CRM sprzedażowy”</i>
Jakiej kategorii/klasę jest to produkt?	<i>jest aplikacją webową klasy CRM</i>
Dla kogo jest przeznaczony?	<i>przeznaczoną dla sprzedawców i marketingowców,</i>
Jakie potrzeby zaspokaja? Jakie możliwości wykorzystuje?	<i>k którzy potrzebują wsparcia procesu sprzedaży usług szkoleniowych.</i>
Jakie przynosi korzyści, za które klient zechce zapłacić?	<i>CRM sprzedażowy przeprowadza sprzedawcę i marketingowca przez pełen proces sprzedaży, w którym centrum procesu jest klient, dzięki czemu oszczędzimy użytkownikom konieczności pamiętania o terminach i etapach procesu.</i>
Jakie są alternatywy?	<i>W przeciwieństwie do sytuacji obecnej, w której każdy z naszych sprzedawców ma własną metodę działania, co powoduje sporo zamieszania,</i>
Co odróżnia produkt od konkurencyjnych alternatyw?	<i>CRM sprzedażowy ujednocila cały proces sprzedaży. Pozwala również zintegrować się z systemem organizacji szkoleń.</i>

– Jakże mogłoby ich zabraknąć w zintegrowanym systemie informatycznym? – argumentował klient.

– Ale przecież to się nie uda! – próbowałem oponować.

Nic z tego. ZSI to ZSI i musi posiadać wszystkie funkcjonalności, których klient od niego oczekiwał.

Zrozumiałem potem, jaki błąd popełniłem. Gdy nadajesz czemuś nazwę, to nazwa zaczyna budować Twoje oczekiwania w stosunku do nazwanej rzeczy. Gdy nazwiesz buty *butami sportowymi*, to będziesz oczekiwał, że będą wygodne podczas biegania. Gdy nazwiesz buty *butami wizytowymi*, to będziesz oczekiwał, że będą się dobrze prezentować w zestawieniu z garniturem. Jeśli zatem nieopatrznie określisz tworzony system nieadekwatną nazwą, to klienci będą oczekiwać od systemu innych funkcjonalności niż powinni.

W opisanym przykładzie *zintegrowanego systemu informatycznego* **nazwa była zbyt ogólna**, w związku z czym niemal każda funkcjonalność pasowała do zakresu tak nazwanego systemu.

Jeśli nadamy nazwę konkretną, np.: *system obiegu dokumentów*, *system zarządzania produkcją rowerów*, *system katalogowania produktów w hurtowni*, to **nazwa ta przekazuje w miarę jednoznacznie informację o tym, które funkcjonalności mieszczą się w zakresie prac nad systemem**, a które nie. Odpowiednia nazwa powinna skupiać oczekiwania klientów na tym, czym w istocie jest tworzony system, oraz pomagać odrzucać zbędne funkcjonalności, zwane potocznie *wodotryskami*. Jest przecież oczywiste lub łatwe do wykazania, że *odtworzenie plików mp3* nie jest naturalną funkcjonalnością *systemu księgowego*. Trudno jednak będzie przeprowadzić podobne rozumowanie w przypadku *zintegrowanego systemu informatycznego* — skoro *zintegrowany*, to dlaczego nie z *mp3*?

Nowemu produktowi nadawaj konkretne nazwy, związane z rzeczywistością (domeną) informatyzowanego procesu.

Dobra nazwa powinna odwoływać się do tego, co klienci już znają — do rzeczywistości (dziedziny) informatyzowanego procesu. Jeśli stworzysz oprogramowanie wspierające dział HR, to oczywiście można mu nadać nazwę *Szybki Lopez*, ale czy to komukolwiek cokolwiek mówi? Klienci będą musieli nauczyć się odpowiedniego rozumienia sformułowania *Szybki Lopez*, a to zabierze nieco czasu. Lepiej odwołać się do doświadczenia zaangażowanych osób i nazwać oprogramowanie *kadry i płace*.

Nic nie stoi na przeszkodzie, aby stosować również nazwy łączone, np. *Szybki Lopez — kadry i płace*. W ten sposób wszyscy zaangażowani w projekt zaczną dość szybko posługiwać się krótką nazwą *Szybki Lopez*, lecz przypiszą tej nazwie to samo znaczenie co nazwie *kadry i płace*, a o to przecież chodziło.

## Wizja jest współdzielona albo nie ma jej wcale

Jeśli chcesz się przekonać, czy wizja spełnia swoją funkcję, zapytaj dowolną osobę zaangażowaną w prace nad oprogramowaniem o to, jaka jest wizja tego produktu. Jeśli nie potrafi o niej opowiedzieć, wizja nie działa.

Co to znaczy, że wizja nie działa? Oznacza to, że oprócz terminów i przydzielonych zadań nie istnieje nic, co nadaje kierunek pracom. Nie istnieje jasny dla wszystkich zaangażowanych osób cel pracy.

### Wizja pochodzi od klienta

Przeanalizuj fragment konwersacji na temat wizji produktu, zamieszczonej w tabeli 3.2.

TABELA 3.2. Wizja, która NIE pochodzi od klienta

Klient	Specjalista
<i>Jak wspominałem wcześniej, chciałbym, aby ten program pomagał mi w rozliczaniu czasu pracy kierowców. Przede wszystkim chodzi o wykrywanie naruszeń związanych z czasem pracy oraz o opisywanie tarczek.</i>	—
—	<i>Aha, czyli możemy napisać: „system do rozliczania czasu pracy kierowców pracujących w firmach pana klientów, dla pana pracowników, odpowiedzialny za rozliczanie czasu pracy kierowców i opisywanie tarczek”. Zgadza się?</i>
<i>Hm, sam nie wiem...</i>	

W wizji nie chodzi wyłącznie o wypełnienie pozycji *Vision Statement* w raporcie czy o sformułowanie jej, bo tak trzeba. Chodzi

o zdefiniowanie punktu odniesienia, który w każdym momencie trwania prac pomoże odpowiedzieć na pytanie: *Czy wciąż robimy to, co na początku zamierzaliśmy robić?*

W przykładzie z tabeli 3.2 specjalista przedstawił wizję produktu, ale to była jego — specjalisty — wizja, nie klienta. Wizja sformułowana w podobny sposób pozwoli zachować kompletność dokumentacji, ale nie będzie działać.

Zobacz w tabeli 3.3, jak może wyglądać skuteczniejsze formułowanie wizji. Techniki użyte w konwersacji poznasz w kolejnych rozdziałach książki.

Zauważ, że do sformułowania wizji absolutnie **niezbędna jest konwersacja**. Szablony, formularze, schematy postępowania mają charakter pomocniczy, ich wypełnienie nie jest celem samym w sobie. Dlatego tak istotna jest umiejętność świadomego prowadzenia konwersacji z biznesem, co stanowi zasadniczy temat tej książki.

## Komunikowanie wizji

Raz — na początku projektu — przeprowadzone spotkanie ustalające wizję niestety nie zapewni, że wizja będzie działać. Zazwyczaj prace nad oprogramowaniem przebiegają bardzo dynamicznie. Bieżące działania szybko przesłaniają to, co zostało ustalone miesiące temu. Aby wizja była ciągle żywa w świadomości osób pracujących nad produktem, musi stać się elementem codzienności, zarówno wizualnie, jak i w postaci codziennego języka, a zatem:

- umieścić sformułowanie wizji w formalnych dokumentach, mailach, na stronie WWW;
- przypominać wizję na spotkaniach formalnych i nieformalnych;
- rozmawiać o wizji z interesariuszami oraz zespołem.



TABELA 3.3. Wizja, która pochodzi od klienta

Klient	Specjalista
<i>Jak wspominałem wcześniej, chciałbym, aby ten program pomagał mi w rozliczaniu czasu pracy kierowców. Przede wszystkim chodzi o wykrywanie naruszeń związanych z czasem pracy oraz o opisywanie tarczek.</i>	—
—	<i>Co oznacza „opisywać tarczki”?</i>
<i>Chodzi o to, że są dwa rodzaje tachografów: cyfrowe i analogowe. Cyfrowe zapisują czas pracy na karcie chipowej, a analogowe na papierowych okrągłych tarczках. W przypadku naruszeń czasu pracy należy odszyfrować zapis tachografu na tarcze i opisać każdą przyczynę każdego naruszenia, co może uchronić przed mandatem.</i>	—
—	<i>Czy dobrze zrozumiałem, że właściwie chodzi o to, aby program wykrywał i opisywał naruszenia związane z czasem pracy?</i>
<i>Cóż, w zasadzie tak.</i>	—
—	<i>Kto przede wszystkim będzie z niego korzystał?</i>
<i>Moi pracownicy.</i>	—
—	<i>Jak się nazywa ich stanowisko?</i>
<i>Specjalista do spraw rozliczania czasu pracy kierowców — tak to wymyśliłem.</i>	—
—	<i>Będzie to więc oprogramowanie „dla specjalistów do spraw rozliczania czasu pracy kierowców, odpowiedzialne za wykrywanie i umożliwienie opisywania naruszeń czasu pracy kierowców”?</i>
<i>Dokładnie tak.</i>	—

## Kiedy wizja bywa niebezpieczna?

Czy po wszystkich zaletach wizji, które wymieniłem, można zastanawiać się nad tym, czy jest ona niebezpieczna? Oczywiście, że można. Każde narzędzie czy metoda ma ograniczony zakres stosowalności. Wizja również.

- Wizja staje się niebezpieczna, kiedy służy do wyciągania wniosków na temat systemu, który opisuje.

Powiedzmy, że tworzysz system, którego wizja została określona następująco:

- *SuperTest jest systemem do przeprowadzania badań satysfakcji klienta banku.*

Wyobraź sobie, że w ramach Twojej organizacji powstała koncepcja innego systemu określonego wizją:

- *T-Expert jest systemem do przeprowadzania badań potrzeb szkoleniowych wśród programistów.*

Wśród klientów, zwłaszcza tych, którzy są daleko od zagadnień technicznych, mogą pojawić się wnioski oparte na następującym rozumowaniu:

- Skoro *SuperTest* służy do przeprowadzania badań i *T-Expert* będzie służył do przeprowadzania badań, zatem do budowy systemu *T-Expert* można użyć już istniejących modułów systemu *SuperTest*. Tworzenie *T-Experta* będzie więc trwało krócej i kosztowało mniej.

Z pewnością widzisz ogrom zagrożenia powstałego z powodu nieprawidłowego wnioskowania opartego na sformułowaniach wizji. Było to możliwe dlatego, że wizja została użyta niezgodnie z jej przeznaczeniem. Zamiast wskazywać cel, do którego zmierzamy, stała się podstawą wnioskowania. Raczej nie powstrzymasz klientów przed wyciągnięciem tego typu ogólnych wniosków. Jedyne, co możesz zrobić, to być na nie wyczulonym i w porę interweniować.

## Podsumowanie

W tym rozdziale dowiedziałeś się, czym jest wizja systemu albo precyzyjniej: produktu programistycznego. Dowiedziałeś się też, że celem wizji jest wyznaczenie kierunku prac, a jej praktycznym zastosowaniem — wstępne ocenianie, czy pomysły na nowe funkcjonalności mieszczą się w ustalonym zakresie, czy nie.

Poznałeś również metody na formułowanie oraz komunikowanie wizji. Jednym z ważniejszych spostrzeżeń w tym rozdziale było to, że aby wizja działała, musi pochodzić od klienta, być współdzielona przez wszystkich interesariuszy oraz przez zespół, który dostarcza kolejnych porcji funkcjonalności.



# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA



**Helion SA**

Przy tworzeniu projektów informatycznych ludzie muszą umieć się dogadać. Brak wspólnej wizji, świadomości istnienia innych perspektyw czy zrozumienia potrzeb współpracujących stron zawsze prowadzi do porażki. Tym bardziej w środowisku Agile, gdzie ludzie ze świata biznesu i deweloperzy pracują razem, zdolność do skutecznej komunikacji jest niezwykle ważna. Można nawet powiedzieć, że umiejętności miękkie mają tam większe znaczenie niż umiejętności techniczne, które łatwiej zdobyć.

Skuteczna komunikacja to taka sama umiejętność jak jazda samochodem czy programowanie. Michał Bartyzel doskonale to pokazuje, rozkładając cały proces rozmowy na proste elementy. Dzięki temu czytelnik może nauczyć się rozpoznawać wzorce i dopasowywać odpowiednie techniki do kontekstu rozmowy. Autor postępuje się prostym i zrozumiałym językiem, uważnie wybiera najistotniejsze aspekty tematów z pogranicza psychologii i coachingu. Narzędzia, które czytelnik dostaje do ręki, można od razu wykorzystać w pracy.

„Oprogramowanie szyte na miarę” wyróżnia się spośród pozycji o podobnej tematyce skupieniem na ludzkiej stronie tematu — zamiast proponowania samych szablonów i metod. Nie jest to kolejne tłumaczenie z języka angielskiego ani próba zastosowania amerykańskich technik w polskich realiach. Liczne przykłady zaczerpnięte prosto z życia pokazują, jakie błędy popełniamy w codziennych rozmowach i jak można ich uniknąć. Gwarantuję, że niejedną raz uśmiechniesz się podczas czytania zapisów rozmów.

Jest to książka promująca pierwszy punkt Manifestu Agile: ludzie i ich interakcje ponad procesy i narzędzia. Polecam ją gorąco wszystkim osobom zaangażowanym w pracę nad projektami informatycznymi wytwarzanymi w relacji klient - dostawca albo na wewnętrzne potrzeby organizacji.

Krystian Kaczor, Agile Coach, trener, autor książki *Scrum i nie tylko*

**Michał Bartyzel** — konsultant i trener w firmie szkoleniowo-doradczej BNS IT. Zajmuje się doskonaleniem programistów i zespołów programistycznych, wdrażaniem metodyki pracy oraz rozwijaniem kompetencji pracowników branży IT. Prowadzi szkolenia oraz konsultacje z zakresu inżynierii oprogramowania, zwiększania efektywności zespołów projektowych i zarządzania projektami programistycznymi.

sięgnij po WIĘCEJ



KOD KORZYŚCI

**Helion**

**one**  
PROS

34937 numer katalogowy

księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

Informatyka w najlepszym wydaniu

Sprawdź najnowsze promocje:  
● <http://helion.pl/promocje>  
Książki najchętniej czytane:  
● <http://helion.pl/bestsellery>  
Zamów informacje o nowościach:  
● <http://helion.pl/nowości>

Helion SA  
ul. Kościuszki 1c, 44-100 Gliwice  
tel.: 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
<http://helion.pl>

ISBN 978-83-246-9826-4



9 788324 698264

cena: 49,00 zł