

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Zarządzanie projektami informatycznymi. Eseje

Autor: Joe Marasco

Tłumaczenie: Agata Kliber, Paweł Kliber

ISBN: 83-246-0273-9

Tytuł oryginału: [The Software Development Edge: Essays on Managing Successful Projects](#)

Format: B5, stron: 352



Zarządzanie projektami informatycznymi różni się od zarządzania projektami innych typów. Jest to związane przede wszystkim ze specyfiką branży i produktu. Projekty IT oddawane nieterminowo, nie spełniające założeń funkcjonalnych i wielokrotnie przekraczające założone budżety przeszły już do legendy. Co jest przyczyną takich sytuacji? I co trzeba zrobić, by ich uniknąć?

Odpowiedzi na te pytania znajdziesz w książce „Zarządzanie projektami informatycznymi. Eseje”. Jej autor, Joe Marasco, legendarny menedżer z firmy Rational Software, przedstawia swój punkt widzenia na temat kierowania projektami IT. Czytając jego przemyślenia, dowiesz się, czy możliwe jest stworzenie realnego harmonogramu, rozwiązanie typowych problemów w niekonwencjonalny sposób i zbudowanie zgranego zespołu programistów. W każdym eseju znajdziesz ciekawe spostrzeżenia i sporo humoru.

- Problemy, przed jakimi staje każdy kierownik projektu
- Metodologie wytwarzania oprogramowania
- Modelowanie w UML i jego znaczenie w projekcie
- Tworzenie harmonogramów
- Budowanie zespołu i zarządzanie nim
- Sposoby rozwiązywania sytuacji kryzysowych
- Wypuszczanie produktu na rynek

Jeśli chcesz tworzyć wartościowe produkty, dostarczać je klientowi na czas i nie przekraczać zaplanowanego budżetu, ta książka jest przeznaczona właśnie dla Ciebie.



Spis treści

O autorze	11
Wstęp	13
Przedmowa	15
CZĘŚĆ I OGÓLNIC O ZARZĄDZANIU	19
Rozdział 1. Zaczniemy od początku	21
Dlaczego potrzebne jest dobre oprogramowanie?	22
Punkty stałe	23
Odbiorcy	24
Przyrostowe rozwiązywanie problemów na zegarze	25
Podsumowanie	28
Rozdział 2. Moja droga do informatyki	29
Wyzwanie	29
Odpowiedź	30
Jak to działało?	31
Dlaczego to pokolenie inżynierów było wyjątkowe?	33
Obliczenia	34
Szacowanie rzędu wyniku	35
Co zatem z komputerami?	37
Nasza spuścizna	38
Podsumowanie	39
Rozdział 3. Wspinaczka	41
Wspinaczka na wysoką górę	42
Najczęstsze przyczyny porażki	48
Składniki sukcesu	49
Czynnik ludzki	51
Podsumowanie	51

Rozdział 4. Zarządzanie	53
Kierowanie zespołem	54
Podsumowanie	60
CZĘŚĆ II RÓŻNICE W OPROGRAMOWANIU	63
Rozdział 5. Rzeczy najważniejsze	65
Programowanie przyrostowe	65
Roscoe Leroy	66
Rezygnujemy z modelu kaskadowego	67
Inna skrajność	69
Pierwszy rysunek Roscoe	69
Drugi rysunek Roscoe	70
Chwileczkę!	71
Skracanie wektorów	72
Zastosowanie do produkcji oprogramowania	72
Nauka stosowana i kierunek krótkich wektorów	73
Namierzanie ryzyka	74
Czy słyszałeś to już wcześniej?	75
Więcej o nauce praktycznej	76
Zastosowanie biznesowe	77
Efekt zatrudnienia	78
Na chłopski rozum	80
Podsumowanie	81
Rozdział 6. Modelowanie	83
Jak wyjaśnić, co to jest UML?	84
Co to jest UML i dlaczego jest ważny?	85
Drugi przykład, mniej trywialny	85
Trzeci przykład	87
A teraz jak to się ma do programowania...	89
Podnosimy poziom abstrakcji	90
Podsumowanie	90
Rozdział 7. Kodowanie	93
Jak menedżerowie mogą uczyć się nowych języków programowania?	94
Lepiej zdefiniowany problem	95
Co powinien zawierać „standardowy problem”?	95
„Zgadnij, jakie to zwierzę”	97
Czy „Zgadnij, jakie to zwierzę” spełnia właściwe kryteria?	98
A czy przechodzi przez test „No i co?”?	99
To jest Twoja gra	100
Podsumowanie	100
Rozdział 8. Wychodzimy z tym na świat	103
Zbuduj je, a oni przybędą	104
Na początku była skrzynia z piaskiem	105
A właściwie dlaczego tworzenie produktu ma być trudne?	106
A co z podejściem przyrostowym?	111
Podsumowanie	111

CZĘŚĆ III Z PUNKTU WIDZENIA MENEDŻERA PROJEKTÓW	113
Rozdział 9. Coś za coś	115
Piramida projektu	116
Pięć, a nie cztery	118
Wchodzimy do piramidy	119
Nowa zmienna – wysokość	120
Objętość piramidy jest stała	120
Dygresja statystyczna	121
Dobry pomysł, zły rozkład	124
Znaczenie dla rzeczywistych projektów	126
Jak to się ma do rzutu monetą?	126
Więcej ufności	127
Ważne ograniczenia	128
A wszystko to przez ryzyko	133
Podsumowanie	134
Rozdział 10. Estymacja	137
A jeśli kierujemy się zdrowym rozsądkiem?	138
Czekolada a wanilia	138
Roscoe wyjaśnia	139
Roscoe kontynuuje	139
Kalendarz Roscoe	140
Obliczenia Roscoe	141
Roscoe zabiera się za oprogramowanie	141
Roscoe składa relację	141
No proszę, jednak zrobiliśmy coś dobrze	142
Roscoe podsumowuje	143
Roscoe podnosi rękawicę	143
No proszę, jednak zrobiliśmy coś dobrze – część druga	144
Roscoe przyjęty do bractwa kierowników projektów informatycznych	145
Podsumowanie	146
Rozdział 11. Harmonogramy	147
Roscoe porusza problem: jak bardzo się spóźnimy?	148
Joe ma okazję się odgryźć	149
Roscoe powraca	150
Kronika przestępców wg Roscoe	151
Wykres Roscoe	151
Ostatnie zastrzeżenie	152
Niemąla uwaga końcowa	153
Podsumowanie	154
Rozdział 12. Rytm	155
Postępy prac projektowych widziane okiem fizyka	156
Wtrąca się rzeczywistość	165
A co z podejściem przyrostowym?	166
Ostatni wykres	171
Podsumowanie	173

CZĘŚĆ IV CZYNNIK LUDZKI	175
Rozdział 13. Polityka	177
Kontekst	178
Definicja	179
Trzy możliwości	179
Polityka jest nieunikniona, ale...	181
Gdy w grę wchodzi polityka	182
Postrzeganie polityki przez inżynierów	185
Środowisko wysokiego zaufania	186
Inne rodzaje złej polityki	187
Podsumowanie	188
Rozdział 14. Negocjacje	191
Komunikacja jest wszystkim	191
Roscoe wykląda swoją teorię	192
Czy to już wszystko?	198
Podsumowanie	199
Rozdział 15. Werbowanie	201
Roscoe się sparzył...	202
...i natychmiast przechodzi do sedna sprawy	202
Wezuwiusz wybucha	202
Jak to się robi w Teksasie?	203
Znaczenie dla programowania	204
Pies zjadł moją pracę domową	204
Wojna o specyfikację?	205
Trzy najczęstsze wymówki	207
Jeszcze jedna sprawa	208
Pchnięcie, parada i riposta	209
Gra w kurczaka przy dużym projekcie	210
Koniec znanego nam sposobu tworzenia oprogramowania?	210
Rozwinięcie kontra konstrukcja	211
Trudna przyjaźń	212
Podsumowanie	212
Rozdział 16. Wynagrodzenia	215
Ruszyć z przepływem	216
Przepływ a produktywność programistów	217
Zastosowanie modelu przepływu do kwestii wynagrodzeń	219
Pieniądze to nie wszystko	228
Podsumowanie	229
CZĘŚĆ V MYŚLENIE NIEKONWENCJONALNE	233
Rozdział 17. Lekcja historii	235
Nie pozwólmy, aby król był architektem	236
Rzeczy nie zawsze są takie, jakimi się wydają	236

Sprawdzanie planów	236
Znaj swoją niewiedzę	237
Kontynuacja przywództwa	237
Jak zwykle w pośpiechu	237
Skupiamy się na rzeczach nieistotnych	237
Gdy plan jest zły	237
Testowanie	238
Prototyp kontra produkt	238
Śledztwo	238
Podsumowanie	238
Rozdział 18. Błędne analogie	239
Houston, mamy problem	239
Prawa Newtona	241
Wszystko jest względne	243
Kwantowy nonsens	246
Śmierć cieplna	249
Inne przykłady	251
Dobra nauka	252
Podsumowanie	252
Rozdział 19. Problem aktualizacji	253
Odświeżanie oprogramowania wbudowanego	254
Obecna sytuacja	255
Gry w aktualizację oprogramowania	256
Skromna propozycja	256
Jeszcze raz — aktualizacja oprogramowania	257
Dodatkowe korzyści	258
Dlaczego będzie to działać?	259
Dalsze możliwości	260
Co z piractwem komputerowym?	261
Zanim przejmie to słońce	262
Podsumowanie	262
Rozdział 20. Nie tak bardzo losowe liczby	265
Roscoe zaczyna opowiadanie	266
Symulacja uderzenia	266
Pierwsze kroki	268
Kolejne kroki	270
Otrzymanie większej liczby prawdopodobieństw	271
Oczywiście, dawno już zapomnieliśmy o baseballu	273
Rzeczywistość jest paskudna	273
Rozwiązanie Poniedziałka	274
Lekcja nauczona	279
Podsumowanie	280

CZĘŚĆ VI SPRAWY ZAAWANSOWANE	281
Rozdział 21. Kryzys	283
Pięć dni ryby	284
Rynek ryb	284
Dzień pierwszy – nieświadomość	284
Dzień drugi – nie poruszamy tematu	284
Dzień trzeci – wkracza „Złota Rączka”	285
Dzień czwarty – punkt zwrotny	286
Dzień piąty – dwie ścieżki krytyczne	286
Morał	287
Podsumowanie	287
Rozdział 22. Wzrost	289
Kwestia wzrostu	289
Model naiwny	291
Konsekwencje modelu	294
Przykład	299
Nieliniowość	301
Wezwanie do działania	302
Wnioski	304
Nomogram	305
Arkusze kalkulacyjny	307
Podsumowanie	307
Rozdział 23. Kultura	309
Czym jest kultura?	310
Słabe i silne kultury	311
Określanie wartości, jakimi kieruje się przedsiębiorstwo	312
A przy tworzeniu oprogramowania oznacza to...	315
Tworzenie silnej kultury	317
Gdy szukamy pracy	321
Ostatnie słowo	322
Podsumowanie	322
Rozdział 24. Zbieramy wszystko razem	323
Chłopiec na posyłki	324
Macher	326
Mistrz	329
Więcej o mistrzu	330
Rozkład populacji	331
Jeszcze kilka słów o modelu	333
Podsumowanie	333
Podziękowania	337
Skorowidz	341

ROZDZIAŁ 11.

Harmonogramy

Tak naprawdę to dopiero na tym poziomie zaczyna się problem. Przeciętny menedżer projektów traktuje harmonogram jako dokument, nad którym się stale pracuje, natomiast jego przełożony — jako kontrakt. Ta „subtelna” różnica jest przyczyną wielu nieporozumień.

Cała sytuacja prowadzi do tego, że zwykle powstają dwa harmonogramy: jeden, tzw. „harmonogram wewnętrzny”, jest dość napięty, tak aby programiści zmieścili się z pracą w czasie krótszym niż wymagany; drugi, tzw. „zewnętrzny”, przeznaczony jest dla reszty firmy i stanowi modyfikację harmonogramu wewnętrznego o pewną granicę bezpieczeństwa. Szczerze mówiąc, nigdy nie byłem zwolennikiem dwóch harmonogramów. Trudno jest utrzymać je w ukryciu, a kiedy sytuacja wyjdzie na jaw, każdy z nich traci wiarygodność, bez względu na to, jak bardzo staramy się wyjaśnić, że jeden jest „harmonogramem prac programistycznych”, a drugi: „harmonogramem biznesowym”. Radziłbym trzymać się jednego harmonogramu i starać się, aby był on jak najbardziej spójny.

Ostatecznie trudno jest dyrektorowi naczelnemu dyskutować z kierownikiem produkcji oprogramowania o jego harmonogramie. Na podobne trudności napotyka kierownik produkcji oprogramowania, próbując omawiać harmonogram stworzenia komponentu czy podsystemu z kierownikiem technicznym, odpowiedzialnym za tę część. Przyczyna jest prosta — trudno tu o cokolwiek się targować. Oczywiście można się spierać, że coś mogłoby zostać wykonane szybciej, ale ostatecznie jest to kwestia subiektywnego osądu i naprawdę bardzo rzadko się zdarza, żeby czas wykonania jakiejś części był bardzo źle oszacowany. W rzeczy samej, trudności w szacunkach dotyczących prac przy projektach informatycznych mają dwie przyczyny. Pierwsza z nich to zależności,

które na początku projektu nie są znane. Skutkiem jest to, że jeśli jedna część się opóźnia, to opóźnia się też inna, której projektanci czekają na krytyczny komponent, dotychczas nieznan. Dopiero jeśli okaże się, że jeszcze jakaś część się opóźnia, zależność zostaje wykryta.

Dobrym rozwiązaniem problemów związanych z opóźnianiem się harmonogramów jest podejście przyrostowe. We wczesnych iteracjach następuje bowiem złożenie dużych części w celu przetestowania, jak będzie z grubsza działał cały system. To złożenie powinno pozwolić na wykrycie „ukrytych” zależności. Następnie można spróbować zastosować rozmaite strategie: wymusić rozłączenie związanych ze sobą elementów, skupić więcej uwagi na punkcie krytycznym i inne. Innym sposobem jest zmuszenie poszczególnych zespołów do dopracowania zewnętrznych interfejsów podsystemowych, zanim jeszcze zostaną stworzone interfejsy wewnętrzne. W ten sposób, dzięki zaistnieniu publicznych interfejsów, inne grupy będą mogły kontynuować pracę. O ile interfejsy zewnętrzne pozostaną stabilne, wewnątrz podsystemu może się zmieniać bez szkody dla całości.

Druga, bardziej zdradziecka przyczyna opóźnień w projektach informatycznych wynika z tego, że opóźnienia następują stopniowo i przyrostowo. Fred Brooks wykazał już wiele lat temu, że projekty, za każdym razem opóźniając się o dzień, opóźniają się o rok. Jeśli osiągnięcie pierwszego kamienia milowego opóźni się o tydzień czy dwa, to prawdopodobnie prac nie da się już przyspieszyć i pozostałe kamienie milowe zostaną osiągnięte z takim samym lub większym opóźnieniem. Tak przysłowiowa kropla draży kamień. Nawet posiadając najbardziej gorliwego menedżera, trudno jest tego uniknąć. Z drugiej strony menedżer, który nie ma oczu i uszu otwartych, powinien mieć się na baczności!

Roscoe porusza problem: jak bardzo się spóźnimy?

Już wcześniej poznaliśmy Roscoe Leroya¹, niecierpliwego, emerytowanego inżyniera górnictwa. To był ponury, deszczowy dzień, gdy Roscoe zaparkował swojego pikapa przed moim domem i, walcząc z wiatrem, dobiegł do drzwi. „Nadchodzi słońce”, pomyślałem.

Jak zapewne pamiętacie, Roscoe jest kumplem mojego taty, człowiekiem z bogatym doświadczeniem życiowym. Przyczyną, dla której słucham Roscoe, jest to, że wnosi on powiew świeżości do wszystkiego, czego się tknie, a jego sposób myślenia nie jest obciążony mądrością konwencjonalną, powszechnie

¹ Czytelnik, który czyta tę książkę nie po kolei i jest to jego pierwsze spotkanie z Roscoe, powinien zajrzeć do rozdziału piątego: „Rzeczy najważniejsze” i dziesiątego: „Estymacja”.

uznanymi doktrynami czy teoretycznymi dywagacjami. Moje źródła donoszą, że uratował więcej niż jedną inżynierską skórę w trakcie swojej „karier”².

„No więc”, zacząłem, „jak ci idzie kariera menedżera projektów informatycznych?”

„Oglądacie skutek braku porozumienia”, rozpoczął cytatem z filmu *Nieugięty Luke*³. Niemalże ujrzałem przed oczami błysk ciemnych okularów kapitana i mogłem tylko mieć nadzieję, że zakończenie będzie mniej tragiczne niż w filmie.

„Po pierwsze, projekty informatyczne zawsze mają opóźnienia. *Zawsze*. A to wcale nie jest dobrze. Poza tym podając jakiegokolwiek oszacowania, uwzględnia się zwykle błąd: »plus minus coś tam«. Co do projektów informatycznych — zdaje się, że zgubiliście gdzieś ten »minus«. Ten, kto dokonuje szacowania, mógłby równie dobrze powiedzieć: zrobimy, co w naszej mocy”.

Nie mogłem się z nim nie zgodzić. Obserwacje Roscoe nie były pozbawione dozy słuszności. Roscoe skrytykował mnie za tę „dozę”.

„Pokaż mi choć jeden projekt, który udało się zrobić przed czasem!”, zażądał. Skrzywiłem się. Owszem, przypominam sobie, że raz osiągnęliśmy kamień milowy przed czasem, ale cały projekt?

„Według mnie problem z oszacowaniem długości prac polega na tym, że należy obliczyć, jak bardzo się opóźnimy”, uśmiechnął się Roscoe.

Joe ma okazję się odgryźć

Na chwilę odebrało mi mowę. Pomyślałem, że mógłbym ukazać w nieco innym, pozytywnym świetle moje lata opóźnień przy projektach. Pokażę temu przemądrzalcowi, że nie zjadł wszystkich rozumów.

„W zasadzie, Roscoe”, rozpocząłem spokojnie, „w projektach nie robimy *jednego* oszacowania. Na początku dokonujemy oszacowań wstępnych. Następnie, kiedy się już trochę wdrożymy, dokonujemy kolejnej estymacji. W zasadzie oszacowań czasu zakończenia dokonuje się przez cały czas trwania projektu. Jeśli więc dokonujemy predykcji czasu zakończenia, możemy jedynie powiedzieć, jak bardzo się spóźnimy, pod warunkiem że wykonamy ostatecznie oszacowanie, gdyż tak naprawdę to właśnie ostateczna estymacja zawiera wszystkie informacje, jakie mamy aż do danego momentu”.

² Roscoe uśmiełby się, słysząc słowo „kariera” w odniesieniu do jego pracy.

„Po prostu staram się zrobić to, co do mnie należy, synu”, odpowiada, gdy zapyta go o którekolwiek z jego osiągnięć czy klęsk.

³ *Nieugięty Luke*, USA (1967), reż. S. Rosenberg (cytat „What we’ve got here is failure to communicate” pochodzący z filmu zajął 11 miejsce w rankingu 100 najsłynniejszych cytatów filmowych, przeprowadzonym w 2005 r. przez Amerykański Instytut Filmowy — *przyp. tłum.*).

„Zgadza się, synu”, powiedział Roscoe. „I powiem ci coś jeszcze. Lepiej, żeby twoje oszacowania w miarę upływu czasu były coraz dokładniejsze. Są dwa powody. Po pierwsze, w miarę jak prace się posuwają, stajesz się mądrzejszy i bardziej doświadczony. Po drugie, jest coraz mniej do zrobienia. Na początku masz dużo niewiadomych i dużo problemów do rozwiązania. Z drugiej strony — masz więcej czasu na naprawę poprzednich błędów. Niech się zastanowie...”

Przyjąłem, że ta runda zakończyła się remisem. Roscoe dokończył swoją kawę (czarna, bez cukru) i, z pewną złością, zapalił swoje cygaro. Mógłbym powiedzieć, że dałem mu trochę do myślenia. Było jasne, że starał się rozgryźć, dlaczego oszacowanie końca prac w projektach informatycznych było o wiele trudniejsze niż w przypadku innych projektów, którymi kierował wcześniej.

Roscoe powraca

Nie minęło dużo czasu, gdy Roscoe powrócił. Nastawiłem się na ponowne wyrócenie mojego małego wszechświata do góry nogami.

Roscoe był tym razem dużo spokojniejszy. „Jak wykazałem poprzednim razem, najprawdopodobniej każdy projekt informatyczny będzie miał opóźnienie. Pozostaje pytanie: jak bardzo? No więc myślę, że mam pewien pomysł”.

„Jak zwykle rozwiązanie wymaga policzenia pierwiastka kwadratowego, a jednostką pomiaru jest — ta dam! — tydzień. Uważam, że dobrze zarządzany projekt informatyczny będzie miał opóźnienie, mierzone w tygodniach, równe pierwiastkowi z liczby tygodni pozostałych do końca projektu. Jeśli na przykład powiesz mi, że skończysz za 16 tygodni, to uznam, że będziesz gotów gdzieś między 16 a 20 tygodniem”.

Widać było, że Roscoe głęboko to przemyślał. Znów pierwiastki kwadratowe? Jakim cudem? Czy Roscoe ma jeszcze coś w zanadrzu?

„W zasadzie można wyróżnić pięć osobnych przypadków”, kontynuował. „Rozpracowałem wszystkie”.

Im dalej Roscoe zagłębiał się w swoje teorie, tym uważniej go słuchałem.

„Po pierwsze, wyróżniłem projekty dobrze zarządzane, kierowane przez faceta, który wie, co robi. To on właśnie zakończy projekt gdzieś między swoim oszacowaniem a pierwiastkiem z tygodni pozostałych do końca projektu”.

„Od czasu do czasu zdarzają się tacy menedżerowie, którzy są dobrymi prognostykami, a także wspaniale sprawdzają się w roli *słomianego szefa*⁴.”

⁴ Dla naszych przyjaciół zza oceanu: *słomianym szefem* nazywamy faceta, który zarządza nami z dnia na dzień. Nazywamy go też czasem: „kopiącym w tyłek”.

„Moc”⁵ zstąpi na niego i tak dalej i uda mu się wcześniej skończyć. Może nawet z dokładnością do jednego pierwiastka. To też może się zdarzyć”. Przypomniałem sobie moją konsternację, kiedy ostatnio nie mogłem podać takiego przykładu.

„To są te pozytywne przypadki”, dodał Roscoe.

Kronika przestępców wg Roscoe

„Kolejne 75% stanowią pozostałe trzy przypadki. Będą to przede wszystkim ci, którzy kończą szybciej niż jeden pierwiastek kwadratowy. Tych denerwujących typów nazywamy *szczwanymi lisami*. Są niebezpieczni, dlatego że generują wysokie koszty. Nie ze względu na spóźnienia, ale na przeszacowanie. Jedynym sposobem na to, aby skończyć szybciej niż jeden pierwiastek, jest błędnie oszacować czas zakończenia. A właściwie osobą, którą powinieneś wyłać, jest szef, który kupił to oszacowanie. Albo nie, jak się lepiej zastanowię, powinieneś zwolnić obu”.

Zacząłem utwierdzać się w przekonaniu, że pan Leroy zrewolucjonizuje sposób, w jaki postrzegamy problem.

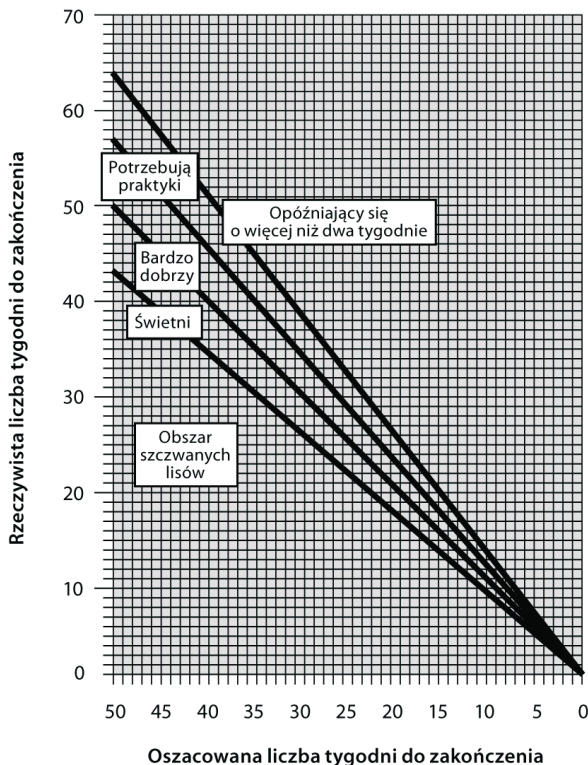
„Potem mamy jeszcze tych, którzy kończą między jednym a dwoma pierwiastkami. Z nich może jeszcze będą ludzie. Prawdopodobnie dopiero zaczynają. Muszą się jeszcze wiele nauczyć w sprawach estymacji i zarządzania. Oni też generują koszty, ale jeśli trochę nad nimi popracujemy, uda się nam przenieść ich do przedziału jednego pierwiastka. Jeśli mają choć trochę oleju w głowie, znajdą się ostatecznie w odpowiednim miejscu”.

„No i na końcu są jeszcze ci, którzy opóźniają się bardziej niż o dwa pierwiastki. Cóż, albo wymknęli się oni spod kontroli, albo nie mają zielonego pojęcia o estymacji. Sam musisz zdecydować, czy są oni beznadziejnymi prognostami, czy też w ogóle nie potrafią zarządzać. W zasadzie to, jaka jest odpowiedź, nie ma znaczenia. Jedynym rozwiązaniem jest wyrzucenie tych ludzi, gdyż pracując z nimi, szybko możesz pójść z torbami”.

Wykres Roscoe

Wtedy Roscoe z dumą wyciągnął swój wykres (rysunek 11.1) w celu zilustrowania przedstawionej teorii. Był tak dumny z tego, że mógł użyć swojej nowej zabawki — Microsoft Excel, że omal nie pękł. „Tych dobrych menedżerów oznaczyłem jako „bardzo dobrzy” i „świetni”. Tych, którzy kończą między jednym a dwoma pierwiastkami, oznaczyłem jako „potrzebują praktyki”. Dwa przypadki patologiczne też są odpowiednio oznaczone”.

⁵ Domyślam się, że Roscoe odwołuje się tu do boskiej interwencji.



Rysunek 11.1. Szacowanie końca prac

„No to masz wszystko. Granica między „bardzo dobrym” a „świetnym” to: „na czas”. Wszystko, czego potrzebujesz, na jednej kartce papieru”.

Tu Roscoe zakończył, moszcząc się wygodniej na krześle.

Ostatnie zastrzeżenie

„No dobrze, Roscoe”, odpowiedziałem. „Jak zwykle zrobiłeś coś z sensem. I naprawdę, jestem pod wrażeniem, że chcesz *policzyć*, jak bardzo się spóźnisz. Ale jest jeden słaby punkt twojej teorii”.

Roscoe natychmiast się ożywił. Zawsze lubił wyzwania.

„Problem leży w tym, że nie wiem, do której kategorii należy menedżer, z którym pracuję. Jeśli wiem, że pracuję z asem, to nie ma problemu, ale co, jeśli nigdy wcześniej nie pracowałem z tym człowiekiem?”.

„Ha!”, odpowiedział Roscoe. „To nic innego, jak problem kalibracji. Zaraz ci pokażę, jak to zrobić”.

„Przede wszystkim, za każdym razem, kiedy proszę kogoś o zrobienie czegoś, proszę o też o oszacowanie czasu zakończenia pracy. A następnie — *i teraz uważaj* — zapisuję to”.

W tym momencie nieopatrznie rzuciłem ciętą uwagę o jego krótkim ołówku.

„Słuchaj, Panie Mądraliński. Najkrótszy ołówek jest dłuższy niż najdłuższa pamięć. I nie zapominaj, że pan Tiger Woods⁶, kiedy gdzieś tam walczy o swoje miliony, też zapisuje swoje wyniki przy pomocy takiego krótkiego ołówka”.

Po takiej krytyce, zdecydowałem się zamknąć usta i zamienić się w słuch.

„A więc, wracając do oprogramowania, spróbowałbym skłonić moich kandydatów do zaangażowania się w rozmaite podprojekty i zadania o różnym czasie trwania. Naturalnie, powinieneś uzyskać od nich oszacowania na początku każdej iteracji w projekcie przyrostowym. Zapisujemy każdą prognozę i sprawdzamy, jak nasi kandydaci sobie poradzili”.

„Otrzymujemy masę punktów na moim wykresie dotyczącym czasu szacowanego i rzeczywistego. Niektórzy stale plasują się w jednej strefie. Świetnie. Zostali ustawieni”.

„Niektórzy będą »rozrzućni« po całej mapie. Na twoim miejscu usiadłbym razem z nimi i spróbował wspólnie to zrozumieć. Ale wcześniej czy później i oni staną się przewidywalni, albo ty sam skierujesz ich na odpowiednią drogę. I nawet ty już wiesz, co zrobić z tymi, którzy znajdują się w strefie »szczwanych lisów« lub »więcej niż dwa pierwiastki«. Uważaj tylko, żeby drzwi nie uderzyły ich za mocno w plecy, jak będą wychodzić”.

„To narzędzie jest naprawdę przydatne dla tych, którzy znajdują się w strefie »potrzebują doświadczenia«. Musimy pokazać im, co powinni zrobić, aby pomóc nam w zdobywaniu pieniędzy, a nie traceniu ich. Muszą postarać się pracować tak, aby trafić do strefy »mniej niż jeden pierwiastek«, niezależnie od ich oszacowań”.

„Widzisz, Joe”, uśmiechnął się Roscoe, siadając ponownie, „dopóki rzecz jest przewidywalna, to się nie martwię. Mogę przeżyć opóźnienie wielkości jednego pierwiastka, jeśli wiem, że taki będzie rozmiar strat. Mogę to zaplanować”.

Zrobić coś z niczego. Nigdy o tym nie pomyślałem.

Niemila uwaga końcowa

„Nawiasem mówiąc”, zakończył Roscoe, „czy zauważyłeś, co się dzieje w moim modelu, kiedy zbliżamy się do końca projektu?”. Nie zauważyłem, ale teraz mnie to uderzyło jako interesujące przejście graniczne.

„Kiedy dochodzimy do prognozy: »skończymy za tydzień«, pierwiastek z 1 daje 1. Koniec z precyzją. W ostatnim tygodniu zawsze możemy powiedzieć, że potrzebujemy jeszcze tygodnia. Bądźmy szczerzy — zawsze znajdzie się jakaś

⁶ Tiger Woods (ur. 1975r., Cypress, Kalifornia) — amerykański gracz w golfa, jeden z największych przedstawicieli tej dyscypliny — *przyp. tłum.*

rzecz do zrobienia natychmiast po tym, jak poprzednią wykreśli się z listy. Więc, nawet teoretycznie, projekty mogą trwać w nieskończoność, wydłużając się z tygodnia na tydzień”.

„Dlatego nazywamy to »dążeniem do najkrótszego strzału«⁷. W ostatnim tygodniu podejmujesz decyzje o innym charakterze. Szacowanie, jak wiele masz jeszcze zrobić, mija się z celem. Mogę cię doprowadzić do ostatniego tygodnia, ale wykończenie to już twoja sprawa”.

Mam szczerą nadzieję, że Roscoe będzie rozwijał swoją karierę. Tak wiele musimy się jeszcze nauczyć.

Podsumowanie

Kiedy mówimy o harmonogramach, zawsze pojawiają się dwie kwestie dotyczące ludzi. Pierwsza z nich to *przewidywalność*. Menedżerowie ciągle się skarżą, że nie mogą żyć w tej niepewności. Jak wykazał Roscoe, winne są wyniki, które pojawiają się niezależnie od planu, tak jak im się podoba. Gdyby ludzie stale się opóźniali, można by z tym jakoś żyć.

To prowadzi nas do drugiej kwestii: pomysłu *kalibracji*. Aby poprawić przewidywalność, należy ustalić dla każdego programisty, co zapisy historyczne mówią o jego zdolnościach do estymacji. Jeśli wiemy, kto jest zwykle zbyt optymistyczny, a kto widzi wszystko na czarno, jest to cenna informacja.

Wydaje mi się, że dobrzy menedżerowie robią to wszystko w ramach analizy jakościowej. Roscoe pokazuje, że powinniśmy zbierać dane w trakcie posuwania się prac projektowych, kalibrując i ponawiając kalibrację oszacowań naszych pracowników *przez cały czas*. Wiedząc, do jakiego stopnia można ufać ich przewidywaniom, możemy uzyskać lepszą jakość szacunków zawartych w naszych harmonogramach.

Interesujące, jak zwięzył jest ten rozdział. Zawdzięczam to Roscoe, którego pomysły są tak treściwe. Jeszcze raz udowodnił on, że nie trzeba być rozwlekłym, żeby przedstawić dobry pomysł.

W następnym rozdziale przeanalizuję dziwne zjawisko. Wszystkie udane projekty wydają się mieć jakiś wewnętrzny rytm, który rządzi ich postępowaniem. Skąd się on bierze? Przedstawiam model, który może wyjaśnić ten fenomen.

⁷ Jedno z nielicznych nawiązań Roscoe do golfa. Aby zdobyć punkt, należy wbić piłkę do dołka. Są to zazwyczaj tzw. „krótkie strzały”. Dążenie do krótkiego strzału oznacza dokonanie ostatnich poprawek, aby zakończyć pracę.