

**PORADNIK**

---

**DOBRYCH PRAKTYK  
ZARZĄDZANIA  
PROJEKTAMI  
IT**

**COMPUTERWORLD**

# 1.

## MITY I LEGENDY ZARZĄDZANIA PROJEKTAMI – JAK NIE DAĆ SIĘ IM OMAMIC

Zarządzanie, a jeszcze bardziej jego angielski odpowiednik management, to bardzo modne określenie. Wygląda to rozczulająco i czasem żałośnie, kiedy młodych ludzi rozmaite kursy i studia siłą uczą zarządzania, w tym zarządzania trudnymi przedsięwzięciami IT. Nie ma dziś trendu na porządne robienie tego, co się robi, natomiast bardzo popularne jest wyrażanie siebie, wywieranie wpływu i kreatywność. Jednocześnie, wielu wydaje się, że wykonywanie nie daje pola do rozwoju tych cech, natomiast rządzenie daje, co wzmacnia tendencję do kultu zarządzania.

Kiedy coś jest bardzo ważne, ale brak na ten temat rzetelnej wiedzy, rodzą się mity. Szczególną formą takich mitów są systemy certyfikacji, przypisujące sobie nieomyślność. Zresztą, istnienie dobrej wiedzy nie przeszkadza mitom, bo wiedza rozczarowuje: nie daje prostych recept na bolączki ani gwarancji sukcesu, więc legendy doskonale istnieją obok wiedzy, a nawet ją wypierają.



## JAK POWSTAJĄ BAŚNIE?

W swoich pionierskich pracach, zwieńczonych popularną książką „Logika katastrofy” [The Logic of Failure. Recognizing and Avoiding Error in Complex Situations], niemiecki psycholog Dietrich Dörner badał wzorce postępowania ludzi w różnych sytuacjach niepewności. W swoich pomysłowych eksperymentach, między innymi realizowanych z pomocą komputerowych gier i symulatorów, Dörner poddawał osoby badane różnym formom niepewności, takim jak brak informacji, wielka złożoność sytuacji, zmienność, opóźniona informacja zwrotna.

**Oto prosty przykład:** eksperymentator udostępnił osobom badanym dwa przełączniki elektryczne, a w pokoju, oprócz zwykłej lampy, były dwie żarówki, czerwona i zielona. Połączenie, – o czym badani nie wiedzieli – było bardzo proste, jeden przełącznik był od czerwonej żarówki, a drugi od zielonej. Tyle tylko, że działały z pięciosekundowym opóźnieniem. Postawmy się w sytuacji uczestnika badania: pstryka lewy przełącznik [ten, o czym badany nie wie, od zielonej żarówki], badany czeka chwilę, nic się nie dzieje, więc pstryka drugi przełącznik [od czerwonej lampki], zaraz zapala się zielona lampka. Wraca do pierwszego przełącznika, pstryk – i zapala się czerwona lampka. I już amen, gotowe, zupełnie fałszywa teoria jest ugruntowana i nie dała się usunąć.

Oczywiście, jeśli pozwolić osobie badanej na kontynuowanie eksperymentu, fakty szybko zaprzeczały wstępnej teorii i tutaj zaczynało się najciekawsze. Niektórzy badani upierali się przy swojej pierwszej teorii, wbrew faktom, wymyślając piętrowe wyjaśnienia, tłumaczące niezgodność. Niektórzy szli w kierunku szukania skomplikowanych serii, kombinacji lampek czerwonych i zielonych. Jeszcze inni szukali wyjaśnień zewnętrznych, przypisując kolor zapalającej się żarówki szybkości naciśnięcia przełącznika, użyciu lewej lub prawej dłoni, lub jeszcze bardziej fantastycznym teoriom.



**SKORO TAK NAS POGRĄŻA SYSTEM DWÓCH PRZEŁĄCZNIKÓW I DWÓCH ŻARÓWEK, CÓŻ MÓWIĆ O ZAWIŁYCH SYSTEMACH SPOŁECZNYCH I ORGANIZACYJNYCH?**



## WIARA W MODNE BZDURY

Jest wiele powodów, dla których lubimy wierzyć w bzdury i głosić bzdury. Niepewność, której pełen jest ten świat, a w nim także projekty IT, budzi lęk i niepokój, których próbujemy się pozbyć. Metody naukowe są czasochłonne, a poza tym często udzielają zgodnej z prawdą, ale rozczarowującej odpowiedzi, że czegoś nie wiadomo, lub że przyszłości nie da się przewidzieć, więc o ileż bardziej pocieszające jest myślenie magiczne.

Głoszenie bzdur, jeśli tylko inni w nie wierzą, zapewnia nam wyższe miejsce w stadnej hierarchii. Jak wszystkie ssaki naczelne, żyjemy w grupach o rozbudowanych relacjach społecznych, i wysokie miejsce w hierarchii zapewnia jednostkom wiele korzyści. Ten, któremu wierzą, którego naśladują i cytują, uzyskuje wszechstronną przewagę nad tymi, którzy nawet mają rację, ale nie zdołali się z nią przebić do innych.

Wierząc w bzdury, unikamy lęku przed nieznanym. Głosząc bzdury w taki sposób, aby brzmiały przekonująco i dawały pociechę i komfort innym, zapewniamy sobie liczne społeczne korzyści. Wszystko to może dziać się zupełnie nieświadomie, więc w bzdury wierzymy równie chętnie dziś, jak dziesięć tysięcy lat temu.

To, że w inżynierii oprogramowania dość bezkarnie szerzą się rozmaite modne bzdury, nie oznacza przecież, że nic innego tam nie ma.



**PRZECIWNIE – JEST TAM MNÓSTWO TRAFNYCH, SŁUSZNYCH METOD, TECHNIK, TECHNOLOGII, TWÓRCZYCH POMYSŁÓW, KTÓRYCH PRZEMYSŁ IT NIE WYKORZYSTUJE NAWET W POŁOWIE.**

Jedną z przyczyn jest właśnie popularność modnych bzdur, które szumem informacyjnym przykrywają metody sprawdzone i skuteczne. Jeśli nauczymy się je wykrywać i odwracać do nich plecami, będziemy pracować sprawniej i efektywniej.



## OBRONA PRZED MITOLOGIA

Wąska i ciernista jest ścieżka wiodąca do praktycznej mądrości w zarządzaniu projektami IT. Z jednej strony, kuszą mas niezliczone metodyki i teorie zarządzania o chwytliwych nazwach, często nierzetelne. Z drugiej strony, sceptycyzm i praktyka mówi, żeby zamiast na ulotnych zasadach zarządzania, skoncentrować się na tym, co rozumiemy najlepiej, czyli samej technologii, a z trzeciej strony, skoro sama technologia nie wystarczy, a teorie zarządzania budzą słuszną podejrzliwość, mamy ochotę w ogóle zrezygnować z próby nauczenia się jakichkolwiek zasad i poprzestać na chaotycznych, wyrwanych z kontekstu przykładach i tak zwanych najlepszych praktykach. Unikanie tych trzech pokus wymaga trudu, a ich rozpoznanie – solidnej interdyscyplinarnej wiedzy, szerszej, niż zakres jednego certyfikatu, choćby miały go mającego setki tysięcy osób.

# 2.

CZYM RÓŻNI SIĘ IT?  
MUSIMY TO BRAĆ POD UWAGĘ

---

IT jest dziedziną szczególną, różniącą się od wielu innych gałęzi przemysłu. Stąd nie zawsze zasady zarządzania projektami, których nauczyło nas doświadczenie tysięcy lat, obowiązują i sprawdzają się w IT.



## BRAK OBCIĄŻEŃ TRADYCJI W IT

Gałęzie przemysłu takie jak konstruowanie okrętów czy budownictwo istnieją od tysięcy lat, metalurgia i mechanika od setek lat, a nawet dziedziny nowsze – przemysł motoryzacyjny, lotniczy, elektronika – mają swoje korzenie w tych starszych. IT jest o wiele młodsza i zupełnie do innych przemysłów niepodobna, ponadto swój złoty wiek – okres szczególnie burzliwego rozwoju – przeżywała w czasach hippisowskiej kontr-kultury, więc łatwiej w niej – psychologicznie – o niestandardowe rozwiązania.



## ELASTYCZNOŚĆ OPROGRAMOWANIA

Technologia IT, zbudowana wokół architektury von Neumanna, pozwala na znacznie większą elastyczność procesu wytwarzania, a koszt modyfikacji jest niższy. Architektura von Neumanna oznacza, że zarówno instrukcje, jak i dane komputera znajdują się w jego pamięci i mogą być modyfikowane. Kamieniarz, który musi zmienić wykute na granitowym pomniku nazwisko, solidnie się napracuje, natomiast programista zmienia ciąg znaków, czy inne dane, w mgnieniu oka. A więc, zalecane przez metody lean – odkładanie jak najdłużej podejmowania nieodwracalnych, krytycznych decyzji, jest możliwe znacznie dłużej przy wytwarzaniu oprogramowania, niż przy wytwarzaniu produktów z kamienia.



## BRAK PRODUKCJI TAŚMOWEJ W IT

IT nigdy na dobrą sprawę nie dorobiło się odpowiednika efektywnej produkcji taśmowej. Ma to pewne uzasadnienie, bo w IT większość przedsięwzięć ma charakter pionierski, tworzy odmienne od poprzednich prototypy, więc proces produkcyjny, taki jak przy tworzeniu długich serii jednakowych egzemplarzy pralek albo samochodów, nie ma w IT racji bytu.

Z drugiej strony, swoista hackerska kultura branży, idealizująca twórczy aspekt budowania oprogramowania oraz debugowanie powoduje, że nawet w bardzo hierarchicznych i planowanych działaniach dominuje poprawianie (debugowanie) nad zapobieganiem błędom, czyli kontrola jakości jest powszechniejsza od zapewnienia jakości.



**W TEJ SYTUACJI, METODY ZWINNE – ŁOWIECKIE BARDZO NIEWIELE TRACĄ W PORÓWNIANIU Z HIERARCHICZNYMI – PLANOWANYMI, JEŚLI CHODZI O JAKOŚĆ I STARANNOŚĆ, KTÓRE I TAK SIĘ ZANIEDBUJE, A ZYSKUJĄ WIELE POD WZGLĘDEM ELASTYCZNOŚCI.**



## ŁOWIECKI CHARAKTER PRZEDSIĘWZIĘĆ IT

Wiele produktów IT jest właśnie tego typu, do których wytwarzania formuła łowiecka pasuje lepiej, niż hieratyczne metody konstruowania piramid. Istnieją wprawdzie systemy IT nawet bardziej złożone i o zawilszej architekturze nawet niż transatlantyk czy Airbus A380 – na przykład systemy sygnalizacyjne, systemy kontroli lotów, systemy nadzoru produkcji – ale większość jest stosunkowo prostych, niewymagających bardzo dokładnego planowania.



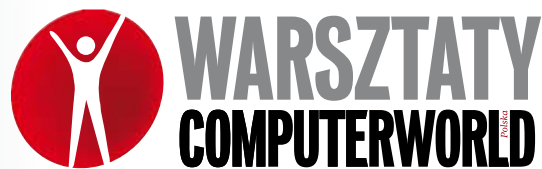
**PRZY ICH TWORZENIU KORZYSTNIEJSZA BIZNESOWO JEST ELASTYCZNOŚĆ, ZDOLNOŚĆ DO ZMIAN I SZYBKOŚĆ DOSTAWY PIERWSZEJ WERSJI, NIŻ DOKŁADNOŚĆ, TRAFNOŚĆ PRZEWIDYWAŃ I MOŻLIWOŚĆ DOKŁADNEGO PLANOWANIA Z WYPRZEDZENIEM.**



## PODSUMOWANIE

Zrozumienie, w jaki sposób systemy IT różnią się od piramid, domów i okrętów, pozwala lepiej zrozumieć, jak zasady zarządzania budowaniem oprogramowania różni się od zasad kierowania budową innych, bardziej tradycyjnych produktów. Tak, istnieje wiele zasad wspólnych dla zarządzania w ogóle i dla zarządzania projektami, bardzo różnymi projektami, ale projekty IT mają swoją specyfikę, którą trzeba brać pod uwagę, chcąc nimi dobrze zarządzać.

**START:** 11 października 2014, Warszawa



# AKADEMIA MANAGERÓW PROJEKTÓW IT

**AKADEMIA MANAGERÓW PROJEKTÓW IT** to system zaawansowanych warsztatów podnoszących praktyczne oraz teoretyczne umiejętności kierowników projektów IT w najbardziej **strategicznych obszarach zarządzania projektami**.

## I WARSZTAT **Klient a biznes - gdzie w tym jest IT? Analiza i realizacja celów projektowych**

Cele biznesowe a cele projektowe • Pozyskiwanie i modelowanie wymagań • Kryteria formalne i testy akceptacji

## II WARSZTAT **Opłacalność projektów – koszty a zasoby**

Strategie kosztowe projektu IT • Szacowanie kosztów zapewnienia jakości • Koszty stosowania modelu iteracyjnego a sekwencyjnego

## III WARSZTAT **Jak obliczyć ryzyko i podjąć właściwe decyzje?**

Systematyczny proces zarządzania ryzykiem • Jak skutecznie szacować konsekwencje ryzyka • Analiza różnych obszarów biznesowych pod kątem podejmowania decyzji

## IV WARSZTAT **Zmiana: procesy i ludzie**

Psychologiczne i społeczne aspekty zmian • Podejście ADP - automatyczne zapobieganie błędom • Idee zmian: Kaizen, Six Sigma, TQM

## V WARSZTAT **Agile i Lean w IT – jak dostarczać więcej wartości biznesowej w krótszym czasie?**

Model tradycyjny, zwinny, a szczupły • Skalowanie Agile i Lean w organizacji • Jak zbudować tablicę Kanban?

PROWADZĄCY WARSZTATY:



**Monika Perendyk**  
Analityk  
biznesowo-systemowy



**Jarosław Żeliński**  
Analityk biznesowy &  
Projektant systemów



**Jacek Mamot**  
Ekspert w zakresie  
zarządzania ryzykiem



**Bogdan Bereza**  
Ekspert ds.  
jakości oprogramowania



**Daniel Skowroński**  
Trener i coach  
metodyk zwinnych

[akademaiit.computerworld.pl](http://akademaiit.computerworld.pl)



# 3.

## JEST MODA NA SZCZUPŁYCH

Grubi nie są dziś popularni, co widać nawet w nazewnictwie metodyk IT. Pewne popularne zasady zarządzania nazwano „lean” (szczupłe). Wprawdzie oryginalnie dotyczyły one procesów produkcyjnych, ale pewne idee z nich przeformułowano tak, aby pasowały również do projektów. Na przykład, zalecany w oryginalnej wersji produkcji „lean”, pochodzącej z Toyoty, brak zbędnych zapasów w procesie produkcyjnym, bo trzymanie zapasów kosztuje, przetworzono na projektową zasadę unikania wąskich gardeł.

No i oczywiście „agile” – cały świat metodyk, znanych, jako zwinne.

Te metodyki nie są złe, ale trzeba zachować ostrożność: ich atrakcyjne nazwy, mogą wprowadzić w błąd. Obsesja szczupłości i zwinności, tej cielesnej, nie projektowej, jest dziś tak potężna, że potrafi stłumić zdrowy rozsądek. Liczne miarodajne sondaże pokazują, że dziś depresja jest znacznie powszechniejsza niż 50 lat temu, oraz że cierpi na nią znacznie więcej kobiet, niż mężczyzn. Wiele wskazuje na to, że przyczyną tego stanu rzeczy jest obsesja zachowania szczupłej sylwetki.

Nie przesadzajmy z naciąganiem tej powierzchownej analogii do konkretów świata IT, ale niech będzie ona dla nas inspiracją do zachowania ostrożności wobec modnych metodyk. Metodyki „agile” oraz zasady „lean” są często lepsze niż inne metodyki z powodu swoich konkretnych zasad, które w określonej sytuacji są korzystne. Przykładowo, agile jest lepsze niż nie-agile wówczas, gdy korzystne jest:

- ❖ iteracyjne budowanie,
- ❖ gdy mają sens dostawy przyrostowe,
- ❖ gdy korzyści z komunikacji bezpośredniej są większe, niż koszty braku obszernej dokumentacji.

Stosowanie tych nazw pozwala dostrzegać faktyczne zalety i wady metodyk.



### HISTORYCZNE POMYŁKI

Historia IT obfituje w przykłady, kiedy pewne autentycznie dobre metody, mające nośną nazwę, stały się modne poza zakres swojego faktycznego stosowania.

Niektórzy z czytelników wiedzą, a może nawet pamiętają, że w językach programowania w latach 50-ych i 60-ych dominowały straszne konstrukcje z użyciem instrukcji GOTO, a ich efektem był tak zwany kod „spaghetti”: trudny albo i niemożliwy do zrozumienia, do przetestowania, do modyfikowania.

#### **Słynny holenderski informatyk Edsger Dijkstra wynalazł dramatycznie lepszy sposób:**

języki programowania pozbawione GOTO, a dysponujące instrukcjami takimi jak IF, CASE, WHILE, FOR. Dzięki temu wynalazkowi nastąpił prawdziwie kwantowy skok wydajności pracy programistów: zamiast tracić czas i energię na walkę z potworem GOTO, mogli skoncentrować się na sprawnej implementacji algorytmów. **Testowanie kodu stało się możliwe!**

Takie nowe języki Dijkstra nazwał strukturalnymi. Odtąd w latach 70-ych zaczęła się wielka kariera tego słowa, masowo nadużywanego, sprowadzonego przez złodziei idei do nic nieznaczącego, ogólnikowego przymiotnika zastępującego słowo „dobry”. Tej modzie na słówko „strukturalny” ulegali zarówno cwaniacy, szmuglujący przy jego pomocy modne bzdury, jak i rzetelni fachowcy, zmuszeni z powodów marketingowych popularyzować swoje dobre pomysły przy użyciu słowa klucza STRUKTURALNY. Znani - słusznie - do dziś twórcy systematycznych zasad analizy i dekompozycji wymagań DeMarco, Yourdon i Constantine nazwali swoje podejście ANALIZĄ STRUKTURALNĄ. Strukturalne programowanie i strukturalna analiza nie mają za sobą nic wspólnego, oprócz tego modnego słówka, pewnie wtedy koniecznego, aby zostać w ogóle zauważonym.

# 4.

## ○ FOLWARCZNYM STYLU ZARZĄDZANIA

---

Mówiąc inaczej, chodzi tutaj o niepartycypacyjny, niedemokratyczny styl zarządzania, ale nie tylko. Słowo „folwarczny” oznacza więcej, niż jedynie przepływ rozkazów z góry na dół, sugeruje też pewne prostactwo, brak wyrafinowania, oraz stwarzanie pozorów nierówności między zarządzającym, a zarządzanym.



## PRZYKŁADY, JAK PROJEKT IT ZMIENIAŁ SIĘ W FOLWARK

Wielokrotnie byłem świadkiem sytuacji, gdy zleceniodawca, albo sponsor projektu, albo właściciel firmy dość brutalnie – nawet odwołując się do groźby zwolnienia – narzucał kierownikowi nieprze-myślany harmonogram i termin ostateczny projektu, a kierownik w ten sam sposób traktował uczestników projektu.

Nie zawsze takie działanie wynika ze złej woli lub może być wyrazem autentycznej troski, że informatycy, zwłaszcza dział IT własnej firmy, będą nadmiernie przedłużać projekt, realizując w nim zbędne wymagania lub stosując bez potrzeby, tylko z ciekawości, najnowsze technologie, architektury, języki programowania.

Inny, często spotykany przykład, to przeświadczenie, że zasady procesu dobre są dla maluczkich, ale nie dla szefa. W pewnej firmie, usiłującej wdrożyć dyscyplinę Agile Scrum, nagminnym problemem było nieustanne wyrywanie przez prezesów uczestników zespołów scrumowych do niezaplanowanych zadań. Podobnie, łamana bywa zasada niezmienności wymagań podczas trwania przebiegu [sprint] Agile Scrum, przez tak zwane „wrzutki”.



### JAK SOBIE RADZIĆ Z PODOBNYMI SYTUACJAMI, JAK SAMEMU UNIKAĆ FOLWARCZNEGO STYLU ZARZĄDZANIA?

Tutaj nie wystarczą dobre chęci. Tak zwany dobry kierownik, unikający takich metod, w organizacji o folwarcznej kulturze będzie przez innych, w tym swoich podwładnych, lekceważony i oszukiwany. To wymaga diagnozy kultury organizacji i podjęcia wysiłku jej zmiany na wielu płaszczyznach.



## ODEJŚCIE OD KULTURY FOLWARCZNEJ

Aby zmienić kulturę, którą – za Jackiem Santorskim – nazwałem „folwarczną”, trzeba zacząć od diagnozy aktualnej kultury, i wskazania, do jakiej kultury dążymy. Wystarczy do tego zwykła ankieta, jeśli tylko wypełniamy ją anonimowo, i jeśli mamy odwagę zmierzyć się z jej wynikami.

Jest wiele sposobów klasyfikacji i opisów kultur organizacyjnych. Jeden z nich, według mnie szczególnie przydatny w praktyce, opisuje kulturę przy pomocy dwóch zmiennych: orientacji na zewnątrz – do wewnątrz, oraz kontroli – elastyczności.



CZTERY SZCZEGÓLNIIE WYRAŻNE FORMY TEJ KULTURY NAZYWANE SĄ KULTURĄ KLANOWĄ, ADHOKRACJĄ, KULTURĄ RYNKOWĄ (ZORIENTOWANĄ NA RYNEK) ORAZ KULTURĄ HIERARCHICZNĄ:

Rysunek 1.  
Przykładowa klasyfikacja kultur organizacyjnych



Określenie kultury własnej organizacji i zadanie sobie pytania, czy obecna kultura jest odpowiednia do zadań, jakie przed nami stoją, to dopiero początek. Kultura niedopasowana do celów organizacji rodzi wprawdzie napięcia i frustracje, których wyrazem mogą być „folwarczne” relacje, ale ich usunięcie nie gwarantuje jeszcze dobrych relacji – istnieje w każdym z nas wiele napięć wewnętrznych i toksycznych emocji, ale jak sobie z nimi radzić – to temat, na co najmniej dwudniowy warsztat, nie zmieści się już w tym artykule.

# 5.

## WSZYSTKO, BYLE NIE WYMAGANIA

W pewnym piśmie rzucił mi się w oczy artykuł „Zarządzanie projektami: jak definiować oczekiwane rezultaty i nimi zarządzać?”. Artykuł traktuje o inżynierii wymagań, cały czas skutecznie omijając właściwe nazwy i udając, że dotyczy zarządzania projektami.

„Wczesne zaangażowanie” opowiada, że projekt trzeba rozpocząć od określenia wymagań, bez tego nici – ale są tam jakby rozważania na temat sprzedaży (?): ani słowa o ani o wizji biznesowej, ani o wymaganiach.

„Zaangażowanie wszystkich stron, zwłaszcza IT” mówi na temat kluczowego procesu określenia granic i kontekstu systemu, o konieczności uwzględnienia wszystkich interesariuszy... nie wymieniając tych nazw, za to cytując jakiegoś Richarda Bextona z Namu Trabel Group. Kto to jest?

„Czytelne i wspólnie uzgodnione priorytety” – owszem, istotne. Priorytety wymagań. Nawet na ten temat jest certyfikat, i szkolenie: IREB CPRE „Pozyskiwanie i konsolidacja wymagań”. O tym w artykule ani słycho, ani widu – artykuł za to udaje, że odkrywa Amerykę, albo zgoła Księżyc, głosząc swoje banały.



## JAK KIEROWNIK PROJEKTU RADZI SOBIE Z WYMAGANIAMI?

Jeśli z wymaganiami jest bardzo źle, kłopoty spadają na cały projekt, ale próby poradzenia sobie z nimi spadają przede wszystkim na kierownika projektu. Jeśli projekt IT nie uda się z powodu braku, niejasności czy zbyt dużej zmienności wymagań, czy też z braku jasnej wizji biznesowej celu projektu, winę przypisuje się kierownikowi projektu, toteż bardzo często, to właśnie kierownicy biorą na siebie odpowiedzialność za ich określenie.

Zarządzanie projektem IT oznacza planowanie, monitorowanie i nadzór nad wysiłkami, służącymi do osiągnięcia konkretnego celu. Ponieważ ten cel określają i opisują właśnie wymagania, są one osią, wokół której zarządzanie projektami się obraca. Nic dziwnego, że elementy inżynierii wymagań wchodzą w skład wzorców zarządzania. Popularne standardy zarządcze, takie jak PMI, IPMA i PRINCE 2, w znacznej części o niej właśnie mówią, co trafnie odzwierciedla fakt, że w projektach, gdzie inżynierii wymagań, jako osobnej aktywności pozornie nie ma, zwykle uprawniają ją właśnie kierownicy projektów.

Co w tym złego? Takie rozwiązanie, jak i inne przeze mnie tutaj opisane, do pewnego stopnia sprawdza się przecież. Zgoda – w niewielkich projektach łącznie wielu ról przez jedną osobę jest pożądane, nawet konieczne: programista jest zarazem architektem i testerem, a kierownik projektu wykonuje obowiązki analityka systemowego oraz inżyniera wymagań. Jednak to bardzo ryzykowne rozwiązanie. Jeśli inżynieria wymagań – w istocie osobna dziedzina – kryje się pod innymi rolami, często prowadzi do jej zaniedbywania, do licznych błędów, niełatwych do wykrycia. Czasem takie łączenie ról kierownika projektu oraz inżyniera wymagań jest sposobem na pozorne zaoszczędzenie czasu i pracy, co oczywiście skutkuje błędami i koniecznością późniejszych, bardzo kosztownych przeróbek. Wiele z typowych problemów w projektach IT, na przykład brak dostatecznej znajomości wymagań przez programistów i zwłaszcza testerów, bierze się z tego PM-centricznego sposobu uprawiania inżynierii wymagań.



## CO NA TO ZARADZIĆ?

Co na to zaradzić? W szczegółach, będzie to zależeć od modelu realizacji projektu, na przykład inaczej wyglądając w agile, a inaczej w projekcie sekwencyjnym, ale ogólna zasada będzie ta sama: wyodrębnić inżynierię wymagań, traktować ją, jako osobną dziedzinę i przyjąć zasadę, że szczegóły zarządzania wynikają z wymagań, a nie odwrotnie.

Agile jest gotowe na taki sposób działania: tam osią, wokół której kręci się projekt, jest [w Agile Scrum] rejestr wymagań produktu [product backlog], a rola klasycznego kierownika jest zastąpiona przez właściciela wymagań [product owner]. W dużym, realizowanym sekwencyjnie projekcie, rozwiązaniem najczęściej jest stworzenie roli inżyniera wymagań, zarządzającego produktem. W projektach, gdzie dużą rolę dogrywa klient, rozszerzenie i uprawnień, i obowiązków popularnego „analityka biznesowego” bywa dobrą metodą, a w małym, hackerskim projekcie rolę tę mogą pełnić sami programiści – byle w uporządkowany sposób.

# 6.

## UCIECZKA PRZED RYZYKIEM

---

Częstym błędem, zaburzającym zarządzanie projektami IT, jest nieuwzględnianie analizy ryzyka przy szacowaniu przewidywanej pracochłonności projektu.



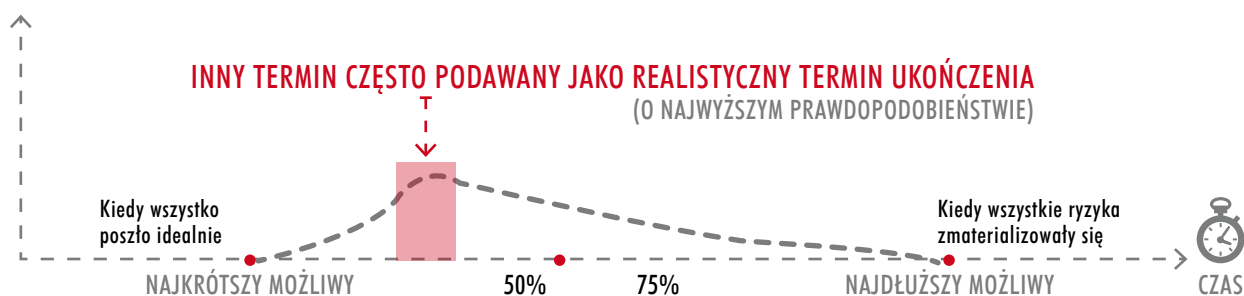


**CZĘSTYM BŁĘDEM, ZABURZAJĄCYM ZARZĄDZANIE PROJEKTAMI IT, JEST NIEUWZGLĘDNIANIE ANALIZY RYZYKA PRZY SZACOWANIU PRZEWIDYWANEJ PRACOCHOŃNOŚCI PROJEKTU.**

Niezależnie od stosowanych metoda oszacowania – a jest ich wiele – każde oszacowanie obciążone jest pewną niepewnością. Mogą nastąpić niekorzystne wydarzenia, które spowodują większą pracochłonność. Aby to uwzględnić, pracochłonność, i wynikający z niej harmonogram, należałoby określać statystycznie, czego z reguły się nie robi.

Ilustracja poniżej pokazuje te zależności.

**Rysunek 2.**  
Metody szacowania pracochłonności i harmonogramu projektu



# 7.

## SPRAWDZAJCIE, ALE NIE ZNAJDUJCIE BŁĘDÓW

---

Testowanie, choć w ciągu ostatnich dwudziestu lat z pomijanego pariasa IT przeistoczyło się w dziedzinę bardziej docenianą i lepiej realizowaną, nadal bywa niepoprawnie traktowane podczas planowania, monitorowania i nadzorowania projektów.

Testowanie, choć w ciągu ostatnich dwudziestu lat z pomijanego pariasa IT przeistoczyło się w dziedzinę bardziej docenianą i lepiej realizowaną, nadal bywa niepoprawnie traktowane podczas planowania, monitorowania i nadzorowania projektów.

Oszacowania pracochłonności z reguły nie uwzględniają dostatecznie trzech czynników: wagi różnych atrybutów jakości, wymaganego poziomu jakości w ramach istotnych atrybutów, oraz wymaganej trafności i precyzji pomiaru tych atrybutów.

To, czy dla danego systemu IT ważniejsze jest doświadczenie użytkownika, czy funkcjonalność, czy niezawodność, czy może wydajność, ma kluczowe znaczenie dla oszacowania, które z tych atrybutów należy testować przede wszystkim, a które mniej. Poziom jakości, wymagany dla danego atrybutu, wpływa na czasochłonność i wytwarzania, i testów. Te czynniki trzeba nauczyć się brać pod uwagę i uwzględniać w oszacowaniach. W przeciwnym razie mają miejsce tak absurdalne działania jak szacowanie pracochłonności projektu według wzoru:



$$\text{CZAS PROJEKTU} = \text{CZAS TWORZENIA} * 120\%$$

Wzór zakłada, że testowanie to stały odsetek – na przykład 20% – kosztów tworzenia. Jest to oczywiście zupełnie błędne założenie.

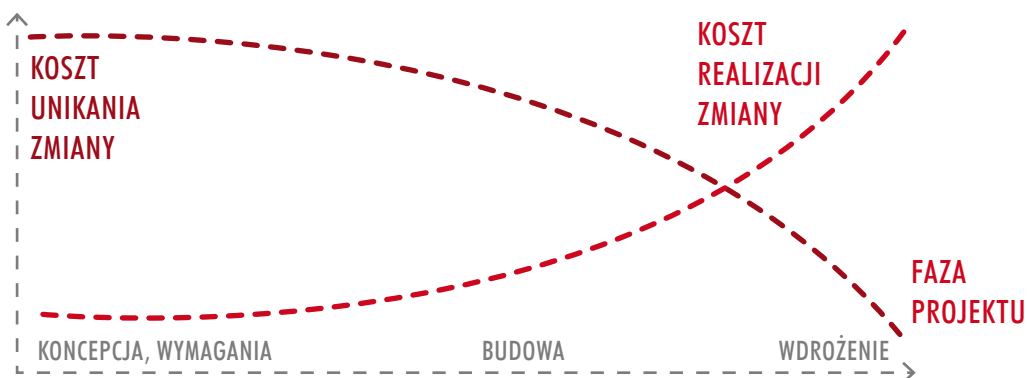
Tradycyjne wręcz błędy popełniane przy zarządzaniu testowaniem, to:

- ❖ pomijanie w planowaniu konieczności naprawiania znajdowanych podczas testowania defektów, oraz potrzebę wykonywania na kolejnych wersjach testów regresji;
- ❖ przy próbach wdrożenia automatyzacji testów, chronicznie zaniżana jest czasochłonność zarówno samego wdrożenia, jak i późniejszego utrzymania i modyfikacji programów testowych.

Mimo zmian na lepsze, testowanie nadal bywa po prostu niedoceniane, traktowane jak zło konieczne i zbędny koszt.

Samo testowanie również ma kłopoty z planowaniem, jaki tryb testowania jest najkorzystniejszy. Zagadnienie pierwsze, to względne korzyści i koszty wczesnego testowania:

Rysunek 3.  
Krzywa Boehm'a  
oraz krzywa Rybera



Dobrze nam znana krzywa Boehma [czerwona na ilustracji powyżej] mówi, że testowanie należy zacząć jak najwcześniej, gdyż, we wczesnych fazach projektu, zmiany – w tym poprawianie bugów – są tańsze. Krzywa Rybera [zielona linia] przypomina, że z kolei przygotowanie i wykonywanie testów staje się często tym tańsze, im później w projekcie jest wykonywane.

# 8.

## DZIAŁ IT A GOSPODARKA SOCJALISTYCZNA

**Szczególna forma złej organizacji powstaje niekiedy na styku własnego działu IT firmy z jej działami biznesowymi: marketingiem, sprzedażą, produkcją. Chronicznym problemem, o którym często słyszymy w praktyce, jest z jednej strony przeciążenie działu IT, a z drugiej strony narzekania pozostałych działów na dramatycznie zbyt długi czas potrzebny IT do realizacji nawet małych zmian. Prawdziwą przyczyną tej sytuacji nie jest technika, a finanse!**

Działy IT z reguły finansowane są nie za wykonywane zadania, lecz w formie stałego, rocznego budżetu, natomiast jego usługi są dla pozostałych działów dostępne za darmo. Powstaje klasyczny efekt gospodarki socjalistycznej: zapotrzebowanie na darmowe usługi jest nieskończone, motywacja zleceńodawcy do precyzyjnego określenia swoich potrzeb i wymagań – zerowa, a motywacja wykonawcy do usprawnienia swojej pracy, minimalna.

# 9.

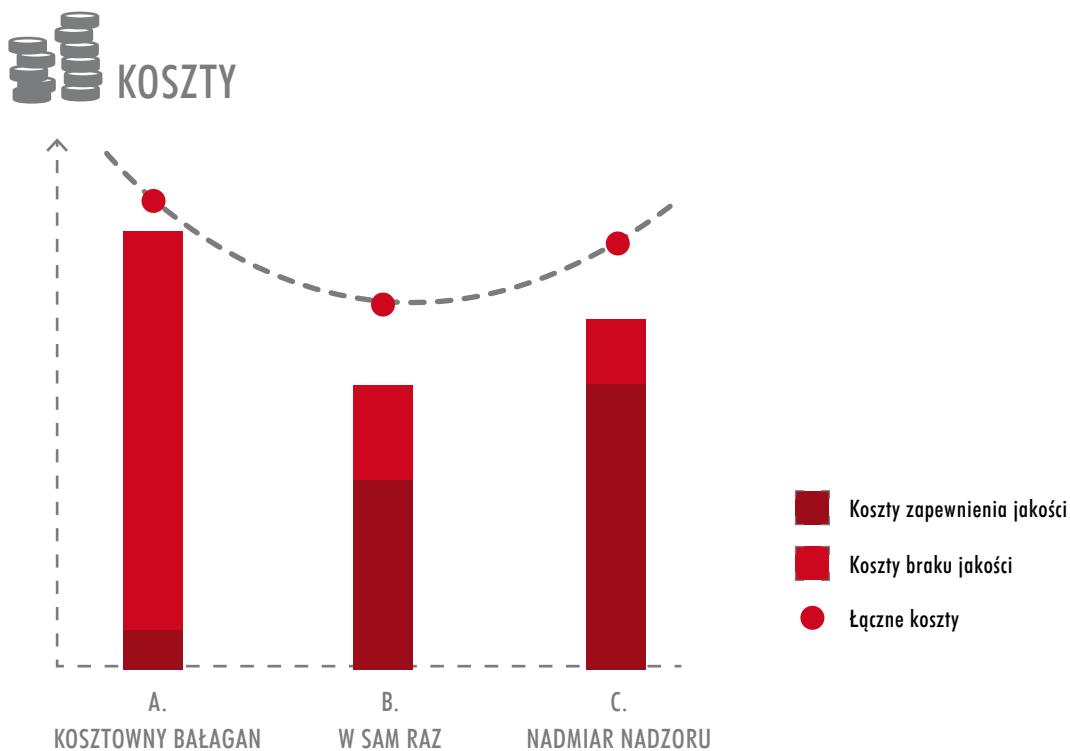
## OWSZEM, JAKOŚĆ JEST ZA DARMO

---

Książka „Jakość nic nie kosztuje” („Quality Is Free”, nie tłumaczona na polski) Philipa B. Crosby’ego wyszła 35 lat temu, ale jej tezy nadal nie dotarły do wielu osób, zarządzających projektami IT.

Książka „Jakość nic nie kosztuje” („Quality Is Free”, nie tłumaczona na polski) Philipa B. Crosby’ego wyszła 35 lat temu, ale jej tezy nadal nie dotarły do wielu osób, zarządzających projektami IT. Wciąż pokutuje przesąd, że jeśli chcemy ograniczyć koszty wytwarzania, to powinniśmy pracować na łapu-capu, pośpiesznie, bo staranne procedury są kosztowne. Tymczasem, w rzeczywistości, do pewnego poziomu, staranność jest w sumie tańsza, niż praca byle jaka:

**Rysunek 4.**  
W jakim sensie, jakość może być za darmo?



A→B

obniżenie łącznych kosztów dzięki temu, że poniesienie pewnych kosztów zapewnienia porządku zmniejsza koszty braku porządku o większą sumę („jakość jest za darmo”)

B→C

tutaj już przesadziliśmy, dalsze zwiększenie nadzoru przestaje się opłacać, bo klienci nie są gotowi dużo płacić np. za grę komputerową o niezawodności Rolls-Royce’a

# 10.

## TO NIE JEST SKANDYNAWSKI KRYMINAŁ

---

Skandynawskie kryminały są dziś bardzo modne. Akcja jest tam chaotyczna i skomplikowana, motywy bohaterów niejasne, czytelnik ma uczucie, że ugrzązł w jakimś sennym, ponurym i bardzo długotrwałym koszmarze. W rzeczywistości jednak, zwykle działania biznesowe i organizacyjne Skandynawów cechuje oszczędność, skuteczność i prostota, zupełnie inne niż nastrój ich kryminałów.

Skandynawskie kryminały są dziś bardzo modne. Akcja jest tam chaotyczna i skomplikowana, motywy bohaterów niejasne, czytelnik ma uczucie, że ugrzązł w jakimś sennym, ponurym i bardzo długotrwałym koszmarze. W rzeczywistości jednak, zwykle działania biznesowe i organizacyjne Skandynawów cechuje oszczędność, skuteczność i prostota, zupełnie inne niż nastrój ich kryminałów.

Podobnie jest z zarządzaniem projektami. Lektura PMBOK, czy rozbudowanych sylabusów PRINCE2 wzbudza wrażenie, że zanurzamy się w rzeczywistości ogromnie skomplikowanej, pełnej tajemniczych nazw i nie do końca zrozumiałych rytuałów, których opanowanie w teorii wymaga wielu lat, a stosowanie w praktyce, drugie tyle.



**W RZECZYWISTOŚCI, SKUTECZNE ZARZĄDZANIE PROJEKTAMI MUSI BYĆ PROSTE,  
W PRZECIWNYM RAZIE JEST NIESKUTECZNE.**

Trudną sztuką jest osiągnięcie tej prostoty w zawilej, chaotycznej rzeczywistości, pełnej pogmatwanych szczegółów. Ludzie, uczestnicy projektu, kiedy są zestresowani i zaniepokojeni, chętnie chronią się w gąszczu szczegółów i wymaga wielkich psychologicznych umiejętności, żeby skłonić ich do pracy w trybie przejrzystej prostoty.

Chcielibyśmy, aby warsztaty Computerworld w ramach „Akademii Managerów Projektów IT”, pragmatyczne, ale osadzone w solidnej wiedzy teoretycznej prowadzących, pomogły uczestnikom w osiągnięciu tych cennych, niełatwych umiejętności przejrzystego, prostego, inspirującego innych zarządzania projektami.





**Bogdan Bereza**  
Ekspert ds.  
jakości oprogramowania,  
trener

Ma ponad dwudziestoletnie doświadczenie w dziedzinie inżynierii oprogramowania. Współpracował lub prowadził szkolenia w tym zakresie z między innymi z takich firmami jak Ericsson, Nohau, BorgWarner, Agora, Skania, Poczta Polska i oraz wieloma innymi w Polsce, w Szwecji, w Czechach. Na krajowych i międzynarodowych konferencjach wygłasza wykłady na temat inżynierii wymagań (ostatnio m.in. CzechTest 2011 i 2013, HUSTEF 2013) oraz testowania. W tym roku prowadzi całodniowy tutorial na 22 międzynarodowej konferencji inżynierii wymagań IEEE w Karlskrona ([webhotel.bth.se/re14](http://webhotel.bth.se/re14)). Jest skutecznym trenerem, autorem kilku książek z i dziesiątków artykułów z dziedziny inżynierii oprogramowania, inżynierii wymagań i testowania. Jest ceniony za umiejętność szerokiego, kompleksowego traktowania jakości oraz łączenia w całość różnorodnych obszarów, takich jak programowanie, projektowanie systemów, inżynierii wymagań, zarządzanie projektami oraz psychologiczne aspektów projektów IT. Wykształcenie w tych dziedzinach zdobył na Uniwersytecie w Lund (Szwecja), Uniwersytecie Warszawskim oraz Uniwersytecie Londyńskim. Uczestniczył w tworzeniu wielu międzynarodowych i polskich organizacji, m.in. SAST, ISTQB, SSTB, SJSI. Obecnie współdziała aktywnie z IREB (International Requirements Engineering Board) oraz jest inicjatorem, współzałożycielem i wiceprezesem Stowarzyszenia Inżynierii Wymagań ([wymagania.org.pl](http://wymagania.org.pl)), inicjatorem i organizatorem Ogólnopolskiego Turnieju Inżynierii Wymagań ([re-challenge.pl](http://re-challenge.pl)).