

Ryzyko w projektach informatycznych

Andrzej Kiesz

Gdańsk, 26 kwietnia 2007 r.



Teoretyczne wprowadzenie do zarządzania ryzykiem na bazie metodyki PMI

Omówienie zagrożeń występujących w poszczególnych obszarach zarządzania projektami informatycznymi:

- budowanie zespołu
- specyfikowanie wymagań
- harmonogramowanie
- problemy z zespołem projektowym
- różnorodne środowiska uruchamiania aplikacji
- problemy z partnerami zewnętrznymi
- kwestie prawne
- wdrożenie

**Zarządzaj projektem, zarządzając ryzykiem,
jakie się z nim wiąże.**

Tom de Marco „Zdażyć przed terminem”

Czym jest ryzyko?

„Ryzyko jest możliwością poniesienia straty”

„Ryzyko to niepewne wydarzenie, które - jeśli zajdzie - może mieć **negatywny** albo **pozytywny** wpływ na projekt”

„Ryzyko to niepewność wyniku (rezultatu)”

Wartość ryzyka w projekcie można określić jako:

Prawdopodobieństwo zdarzenia x Wpływ na projekt

- Planowanie zarządzania ryzykiem
- Identyfikacja ryzyka
- Jakościowa analiza ryzyka
- Ilościowa analiza ryzyka
- Planowanie reakcji na ryzyko
- Monitorowanie i kontrola ryzyka

Określamy jak podchodzimy do zarządzania ryzykiem w projekcie i jak planujemy działać w tym zakresie

Plan zarządzania ryzykiem obejmuje:

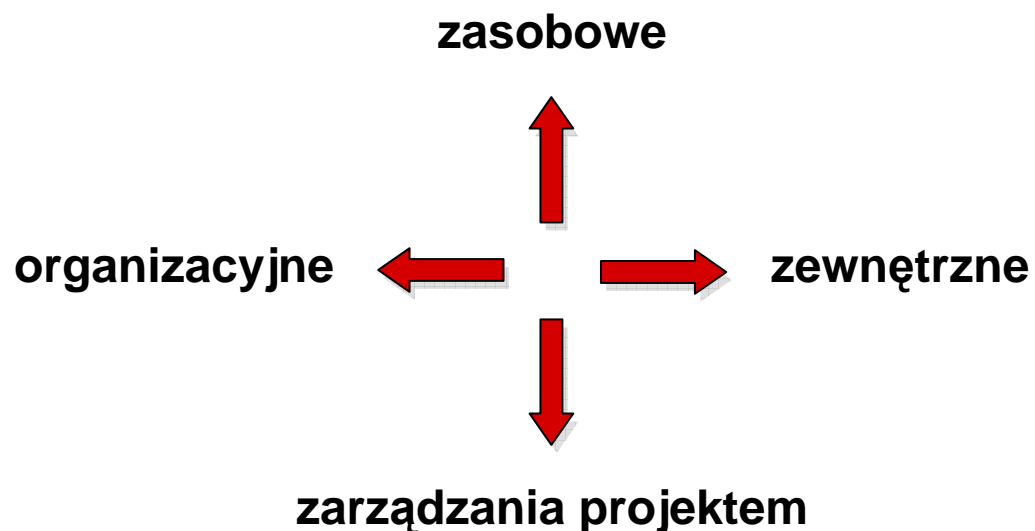
- metodyka
- przydział ról i odpowiedzialności
- budżet na zarządzanie ryzykiem
- realizacja procesów zarządzania ryzykiem w czasie projektu
- wyniki i ich interpretacja
- kryteria progowe
- raportowanie
- śledzenie/kontrolowanie

Wynik:

- plan zarządzania ryzykiem

Określenie jakie ryzyka mogą wpływać na projekt oraz dokumentowanie ich charakteru

Kategorie ryzyka



Ryzyko może dotyczyć...

- harmonogramu
- kosztów
- jakości
- zakresu prac
- zasobów
- satysfakcji klienta

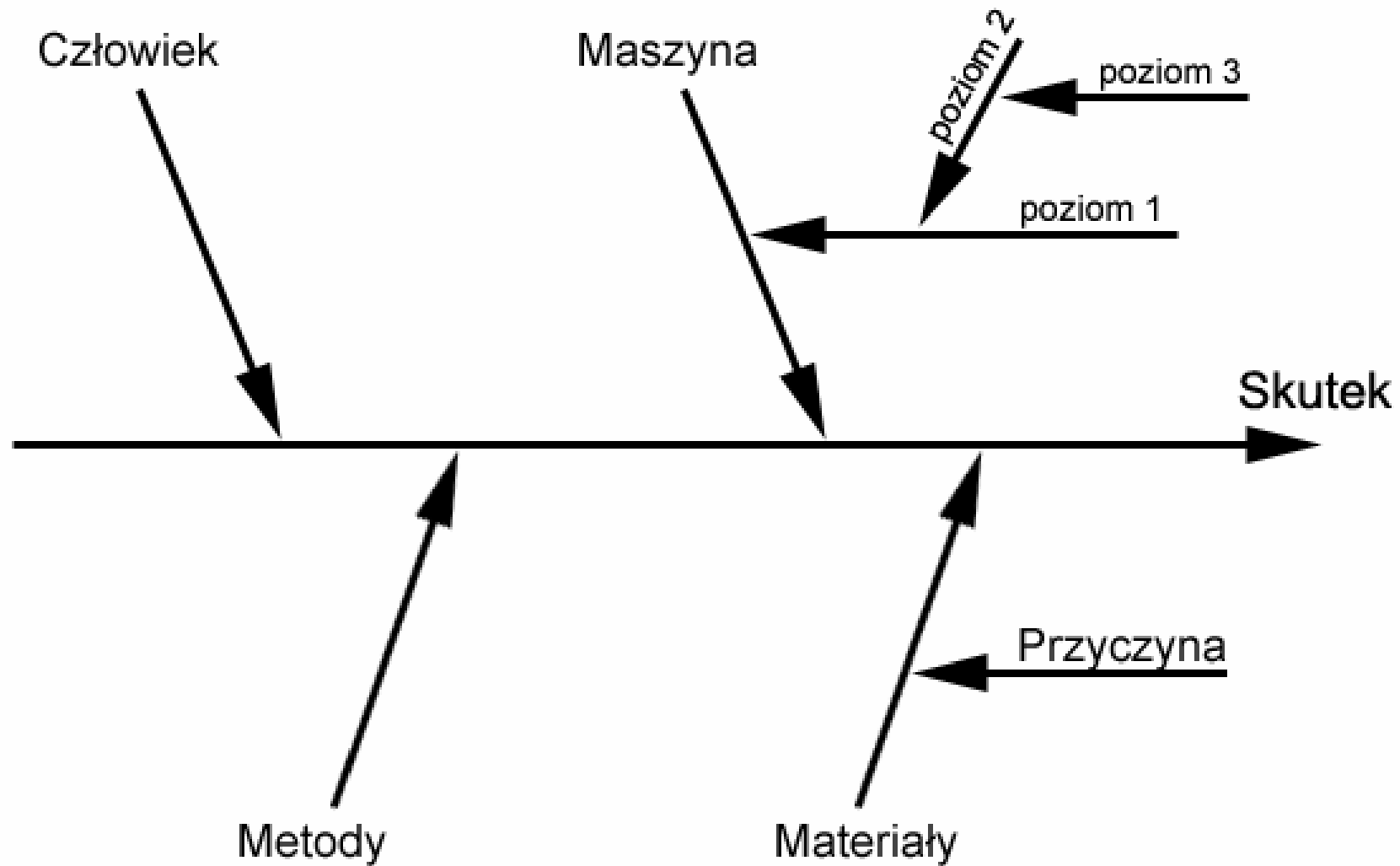
Techniki zbierania informacji:

- Burza mózgów (dwa etapy: **zielony** i **czerwony**)
- Diagram Ishikawy
- Listy kontrolne
- SWOT (ang. *strengths, weaknesses, opportunities, threats*)

Wyniki:

- Lista ryzyk
- symptomy ryzyka

Identyfikacja ryzyka ▶ diagram Ishikawy (3/3) WIRTUALNA POLSKA



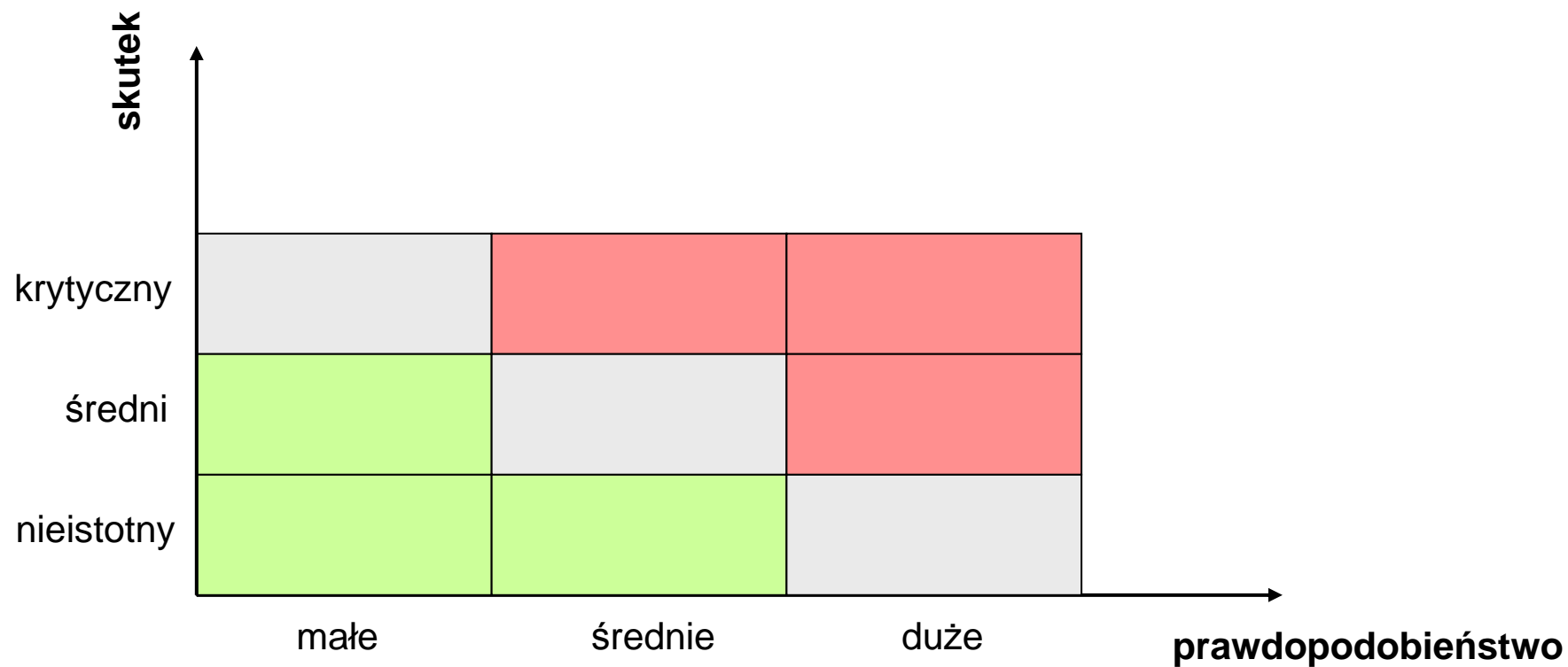
Subiektywna analiza prawdopodobieństwa i skutku w kategoriach intuicyjnych typu mały-duży

Technika:

- ustalenie kilkustopniowej skali określającej poziomy prawdopodobieństwa i skutku
- wyznaczenie granic tolerancji ryzyka
- przygotowanie macierzy oceny ryzyka

Wyniki:

- lista ryzyk uszeregowana wg priorytetów
- lista ryzyk wymagających dalszej analizy oraz tych o mniejszym znaczeniu



Liczbowa analiza prawdopodobieństwa i konsekwencji.

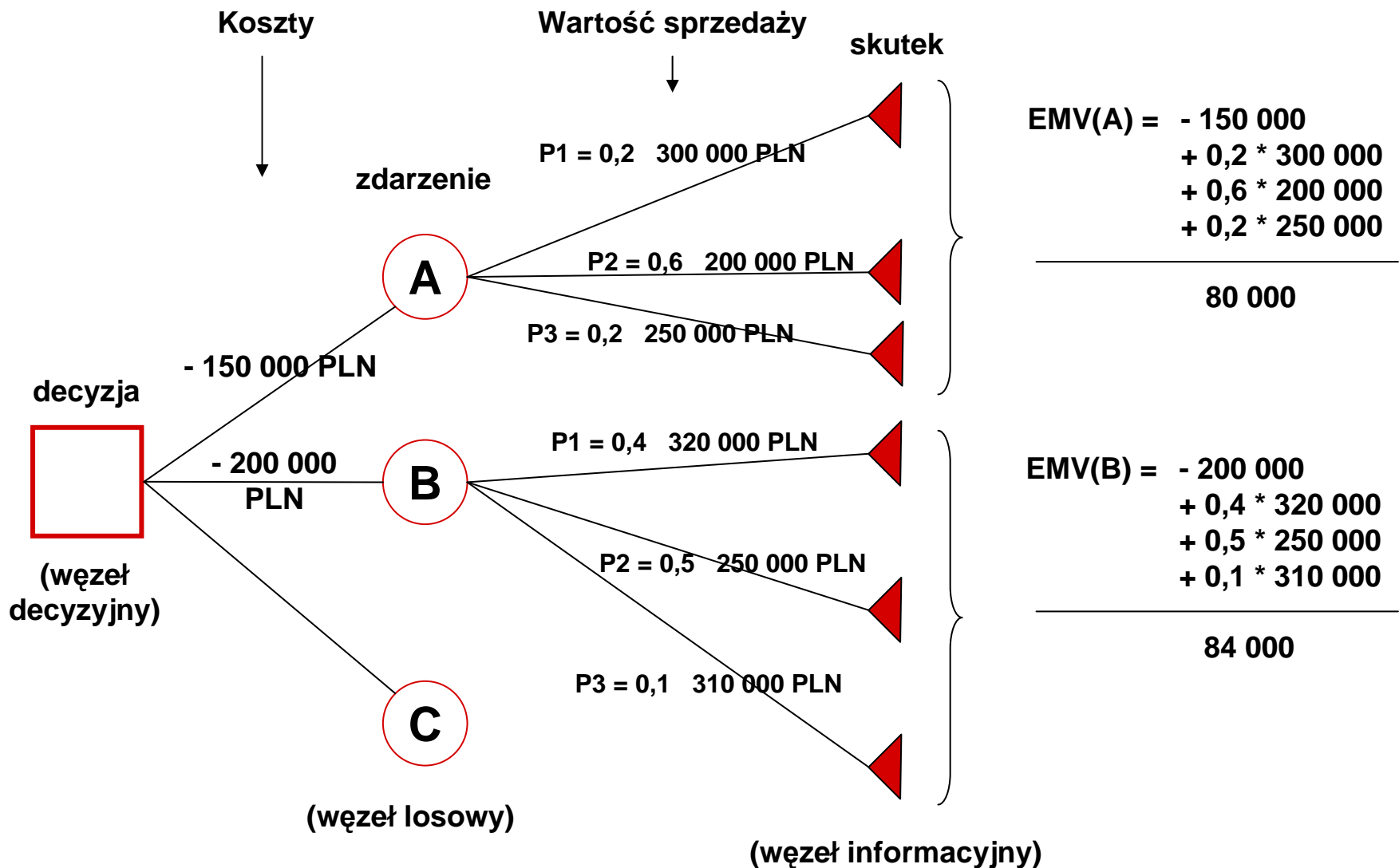
Techniki analizy ryzyka:

- Expected Monetary Value
- drzewo decyzyjne
- symulacja Monte Carlo

Wyniki:

- lista ryzyk wg priorytetów (w ujęciu liczbowym)
- wyznaczenie optymalnych ścieżek dla projektu
- prognoza przewidywanych kosztów i harmonogramu projektu
- lista ryzyk o mniejszym znaczeniu

Ilościowa analiza ryzyka ▶ EMV w węźle decyzyjnym (2/2) WIRTUALNA POLSKA



Źródło: Jacek Jamroz

Określenie sposobu zmniejszenia lub eliminacji ryzyka.

Sposoby reakcji na ryzyko:

- unikanie – zmiany w planie projektu
- transfer – przesunięcie skutków ryzyka na innych (np. na firmy ubezpieczeniowe)
- łagodzenie ryzyka – ograniczenie prawdopodobieństwa wystąpienia negatywnego zdarzenia oraz jego wpływu
- akceptacja ryzyka – pozostawienie planu projektu bez zmiany, aktywna akceptacja ryzyka, np. przez przygotowanie środków na pokrycie strat

- ciągła obserwacja zdarzeń w kontekście ryzyka
- weryfikacja planu reakcji na ryzyko
- rozpoznanie nowych ryzyk i tworzenie dla nich nowych planów reakcji

Wyniki:

- zmodyfikowany plan reakcji na ryzyko
- zmiany w projekcie

- Próba identyfikacji ryzyka bez dostatecznej wiedzy na temat projektu
- Ogólna i szybka identyfikacja ryzyka, przy wykorzystaniu jednej metody
- Wyodrębnione ryzyka są zbyt ogólnie określone. Przeoczenie ukrytych zagrożeń
- Negatywne zdarzenie, które zajdzie na 100% to fakt, a nie ryzyko
- Pominięcie całych grup ryzyka, np. czynnik ludzki
- Brak monitoringu ryzyka podczas realizacji projektu
- Kluczowe ustalenia (np. dotyczące harmonogramu) zapadają zazwyczaj przed analizą ryzyka

Zagrożenia:

- zatrudnienie niewłaściwego kandydata podczas rekrutacji:
 - **osoba niedostatecznie bystra** – dobre rozwiązania są lepsze od średnich
 - **brak kompetencji**
 - **brak zorientowana na cel** – nie ma sensu realizować zadań, które nie mają końca, określonego celu, ani praktycznego zastosowania
 - **nieodpowiednie cechy osobowe (np. konfliktowość)** – rewelacyjny programista, który nie potrafi współpracować z resztą zespołu przyniesie więcej szkody niż pożytku
- utrata bardzo dobrego kandydata podczas rekrutacji
- utrata pracownika niedługo po zatrudnieniu
- konflikty w zespole

Zapobieganie:

- rekrutacja:
 - pytania pozwalające zbadać bystrość kandydata podczas rozmowy rekrutacyjnej
 - zlecenie rozwiązywania zadań testowych w ramach rekrutacji
 - analiza roli i zachowania kandydata podczas realizacji poprzednich projektów
 - wywiad środowiskowy
 - prowadzenie rozmów rekrutacyjnych przez kilka osób
 - dopasowanie kompetencji osoby przeprowadzającej rozmowę rekrutacyjną i osoby kandydującej
 - szybki kontakt z kandydatami, wobec których jesteśmy pewni
 - szczerą rozmowa – upewnienie się, że nowy pracownik nie poszukuje w dalszym ciągu pracy, zapewnienie satysfakcjonujących obie strony warunków

Zapobieganie:

- budowanie zespołu z osób już zatrudnionych:
 - nie warto niszczyć zespołów, które są zgrane
 - należy dobrać osoby (w miarę możliwości) o odpowiednich stanowiskach i cechach osobowościowych

„Lepiej odrzucić dobrego kandydata niż przyjąć kiepskiego”

Zagrożenia:

- zbyt ogólne i niejasne sformułowanie zadań
- różna interpretacja tych samych zapisów przez klienta i wykonawcę
- specyfikacja nie obejmuje wszystkich funkcjonalności
- ustalenia nie są rejestrowane
- ciągłe zmiany, brak ostatecznej wersji

Zapobieganie:

- rozmowy z klientem przed opisaniem konkretnych funkcjonalności
- eliminacja wszystkich niezrozumiałych dla wykonawcy fragmentów specyfikacji
- tworzenie klikalnych/graficznych prototypów i włączanie ich do specyfikacji
- zatwierdzenie (pisemne, bądź mailowe) ostatecznej wersji specyfikacji
- oddzielenie funkcjonalności technicznej od funkcjonalnej

„Niejasna specyfikacja świadczy o nierozwiązanym konflikcie między stronami zainteresowanymi budową systemu”

**„To, co zostało zapisane to jest.
To, co nie zostało zapisane, tego nie ma”**

Zagrożenia:

- daty narzucone z góry (system został sprzedany przed analizą)
- strach lub ambicja prowadzące do deklaracji nierealnych terminów
- zbyt ogólne sformułowane zadania
- brak konsultacji terminów z podwykonawcami
- brak dostatecznej wiedzy na temat systemu, harmonogramowanie przez nieodpowiednie osoby
- błędne oszacowanie liczby dni pracujących

Zapobieganie:

- szacowanie poszczególnych zadań powinno być powierzone osobom, które faktycznie będą realizować te zadania
- konsultacje harmonogramu ze wszystkimi „zainteresowanymi”
- analiza planów urlopowych członków zespołów, weryfikacja dni ustawowo wolnych od pracy w harmonogramowanym okresie
- warto poświęcić więcej czasu na dokładne projektowanie, co pozwoli zredukować czas potrzebny na usuwanie błędów
- duże zadania należy dzielić na kilka mniejszych
- rezerwacja bufora czasowego

„Optymizm jest u informatyków chorobą nieuleczalną”

„Pamiętaj, że każdy dzień stracony na początku projektu szkodzi tak samo, jak dzień stracony na jego końcu”

**„Jest nieskończenie wiele sposobów na to, by stracić dzień...
ale nie ma żadnego, by go odzyskać”**

„Dziewięć kobiet nie urodzi dziecka w 1 miesiąc”

Zagrożenia:

- odejście z firmy cennych ludzi
- brak motywacji i identyfikacji z zadaniami
- konflikty personalne
- groźby jako metoda motywowania ludzi

Zapobieganie:

- rozmowy okresowe
- przydział odpowiednich zadań do poszczególnych osób
- podział długofalowych prac na podprojekty – cel powinien być osiągalny
- zapewnienie poczucia bezpieczeństwa zespołowi
- pozwolenie na proponowanie i realizację (w odpowiednim zakresie) pomysłów swoich ludzi
- pozostawienie swobody członkom zespołu (wzmocnienie odpowiedzialności)
- warto najpierw wysłuchać, a potem mówić
- czujność i szybka reakcja na konflikty
- należy zrozumieć indywidualnie każdą osobę, jej oczekiwania i potrzeby, aby porwać za sobą cały zespół

**„Pamiętaj: jesteśmy obaj po tej samej stronie;
to problem jest po drugiej stronie”**

„Ludzie pod presją wcale nie myślą szybciej”

„Złość jest efektem lęku”

Zagrożenia:

- błędne działanie systemów w różnych środowiskach (brak zapewnienia jakości):
 - niejednorodne warunki sieciowe
 - różne systemy operacyjne
 - różne wersje przeglądarek
 - zróżnicowanie innych środowisk uruchamiania aplikacji, np. wirtualna maszyna Javy
- konieczność usuwania błędów (niedotrzymanie harmonogramu)

Zapobieganie:

- zebranie doświadczenia od ekspertów
- wygospodarowanie czasu na:
 - zbadanie zachowania się aplikacji w różnych środowiskach (testy)
 - usuwanie błędów

Zagrożenia:

- partner nie jest dostatecznie kompetentny
- partner nie dotrzymuje terminów
- problemy w komunikacji z partnerem
- realizacja prac bez podpisanej umowy

Zapobieganie:

- weryfikacja partnera przed podpisaniem umowy
- umowa powinna obejmować co najmniej (np. w formie załączników):
 - harmonogram
 - zapisy na temat SLA (ang. *service level agreement*)
 - zdefiniowane sposoby reakcji na błędy i awarie
 - specyfikację (funkcjonalną i/lub techniczną)
 - listę osób i odpowiadające im zakresy odpowiedzialności
- stałe monitorowanie postępu prac podczas realizacji projektu

Zagrożenia:

- regulamin usługi może zmienić kształt produktu
- zmiana obowiązującego prawa podczas realizacji projektu
- problemy z licencjami do wykorzystywanych modułów lub elementów graficznych
- pominięcie kosztów i brak analizy problemów związanych z rejestracją znaków towarowych
- konstrukcja umowy o dzieło w kontekście autorskich praw majątkowych

Zapobieganie:

- odpowiednio wczesne konsultacje z prawnikami
- analiza wymagań licencyjnych

Zagrożenia:

- brak oficjalnej akceptacji na wdrożenie produktu w określonym kształcie
- problem z dostępnością decydentów (nieobecność, brak czasu), którzy muszą wyrazić zgodę na wdrożenie (mogą je też zablokować)
- kłopot z dostępem do ludzi o niezbędnych kompetencjach
- problemy techniczne

Zapobieganie:

- wczesne przedkładanie systemów do akceptacji
- akceptacje tylko na piśmie lub via email – nic na słowo
- rezerwacja czasu niezbędnych specjalistów oraz ustalanie z nimi terminów wdrożenia
- stworzenie planu wdrożenia *krok po kroku*
- symulacja wdrożenia (weryfikacja mechanizmów)
- przygotowanie planu awaryjnego

„Nie uda się nawet wtedy, gdy właściwie nie powinno się nie udać. Wszystko wali się naraz.”

„Wszystko zabiera znacznie więcej czasu, niż by się wydawało.”

O czym warto pamiętać? (1/2)

- archiwizacja emaili, okresowy backup dysku
- pomysł „piszemy wszystko od nowa” to bardzo zły pomysł
- żadne zmiany nie przyniosą nagłego podniesienia wydajności, usprawnienia procesów do inwestycje długofalowe
- negatywne skutki układów politycznych w firmie

**„Najbardziej zabójcze dla Ciebie nie jest to,
czego nie wiesz, ale to, co wydaje Ci się, że wiesz,
choć w istocie nie jest prawdą”**

**"Jeżeli wszystko idzie dobrze,
to jest to symptom, że będzie źle"**

Dziękuję za uwagę

akiesz@wp-sa.pl