FAZA STRATEGICZNA

Faza strategiczna jest wykonywana **zanim** podejmowana jest **decyzja o realizacji** przedsięwzięcia. Jej zadaniem jest określenie celów tworzonego systemu oraz wymagań odnośnie szczegółów jego funkcjonowania.

Podstawowe hasło: **Nie skupiać się na szczegółach!** (na razie)

Czynności w fazie strategicznej:

1. Dokonanie serii **rozmów** z przedstawicielami klienta
2. Określenie **celów** przedsięwzięcia z punktu widzenia klienta
3. Określenie **zakresu** oraz **kontekstu** przedsięwzięcia
4. Ogólne określenie **wymagań**
5. **Propozycja** kilku możliwych **rozwiązań** (sposobów realizacji systemu)
6. Oszacowanie **kosztów** oprogramowania
7. Analiza i ocena rozwiązań
8. **Prezentacja** wyników fazy strategicznej przedstawicielom klienta oraz ewentualna korekta wyników
9. Określenie wstępnego **harmonogramu** przedsięwzięcia
10. Określenie **standardów**, zgodnie z którymi realizowane będzie przedsięwzięcie

Decyzje strategiczne:

- Wybór modelu, zgodnie z którym będzie realizowane przedsięwzięcie
- Wybór technik stosowanych w fazach analizy i projektowania
- Wybór środowiska implementacji
- Wybór narzędzia CASE
- Ocena możliwości wykorzystania gotowych komponentów
- Podjęcie decyzji o współpracy z podwykonawcami lub o zatrudnieniu ekspertów

Po stronie klienta warto osobno wyróżnić **zleceniodawcę** i przyszłych **użytkowników**. Określenie celów przedsięwzięcia z punktu widzenia klienta nie zawsze jest oczywiste. To powoduje częste nieporozumienia pomiędzy klientem i wykonawcą.

OCENA ROZWIĄZAŃ

W fazie strategicznej zazwyczaj rozważa się kilka alternatywnych rozwiązań. Stosowane są różne kryteria oceny i różne jednostki miary. Na przykład:

- koszt [tys. zł]
- czas realizacji [miesiące]
- niezawodność [ilość błędów/miesiąc]
- możliwość ponownego użycia [%]
- przenośność na inne platformy [%]
- wydajność [ilość operacji/godzinę]

Zebrane informacje o różnych wariantach są prezentowane i porównywane w postaci tabelarycznej. Najpierw usuwamy rozwiązania zdominowane (tzn. gorsze według wszystkich stosowanych kryteriów)

Aby umożliwić porównanie pozostałych wprowadzana jest normalizacja wartości dla poszczególnych kryteriów oraz przypisanie wag.

Normalizację przeprowadzamy poprzez wyznaczenie wartości skrajnych a następnie wliczenie nowych wartości ocen według formuły:

$$x_{znormalizowane} = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})}$$

Przykład:

		wprost					znormalizowana		
N	**Rozwiązanie**	**A**	**B**	**C**	**min**	**max**	**A**	**B**	**C**
1	Koszt	120	80	175	80	175	0.42	0	1
2	Czas	33	30	36	30	36	0.5	0	1
3	Zawodność	5	9	13	5	13	0	0.5	1
4	Ponowne wykorz.	40	40	30	30	40	1	1	0
5	Przenośność	90	75	30	30	90	1	0.75	0
6	Wydajność	0.35	0.75	1	0.35	1	0	62	1

Oceny końcowe wyznaczamy jako sumę ważoną ocen częściowych. Problemem jest dobór odpowiednich wag (często „na wyczucie”).

SZACOWANIE KOSZTU OPROGRAMOWANIA

Na koszt oprogramowania składa się:

- koszt sprzętu będącego częścią tworzonego systemu
- koszt zakupu wykorzystywanych narzędzi
- koszt szkoleń, wyjazdów
- nakład pracy !!!

Najtrudniej oszacować pracę. Stąd najczęściej szacowanie kosztów oprogramowania jest utożsamiane z oszacowaniem nakładu pracy

Metody szacowania nakładu pracy:

- **Ocena przez eksperta** (na wyczucie),
- **Przez analogię** do poprzednio realizowanych przedsięwzięć,
- **Wycena dla wygranej** – oszacowanie na podstawie kosztu oczekiwanego przez klienta i na podstawie kosztów podawanych przez konkurencję.
- **Przez dekompozycję** – przedsięwzięcie dzieli się na mniejsze zadania, następnie sumuje się koszt poszczególnych zadań.
- **Metody algorytmicznie** np. model COCOMO lub Function Point Analysis. Polega na opisaniu przedsięwzięcia przez wiele atrybutów liczbowych a następnie odpowiedni algorytm lub formuła matematyczna wyznacza wynik.

Metoda **COCOMO (COst COnstruction MOdel)** jest oparta na kilku formułach pozwalających oszacować całkowity koszt przedsięwzięcia na podstawie oszacowanej liczby linii kodu.

Metoda **COCOMO** proponuje proste formuły dla oceny ilości osobo-miesięcy pracy oraz czasu potrzebnego na całość projektu:

$$n = a * x^b$$

gdzie:

n – **nakład pracy** [*w osobomiesiącach*]
x – długość ostatecznie dostarczonego kodu źródłowego [*w tysiącach instrukcji*]
a oraz ***b*** – współczynniki wyznaczone eksperymentalnie:

$$2.4 \le \boldsymbol{a} \le 3.6$$
$$1.05 \le \boldsymbol{b} \le 1.20$$

wyższe współczynniki stosowane są w przypadku występowania niepewności.

$$t = c * n^d$$

gdzie:

t – optymalny czas wykonania [*w miesiącach*],
n – nakład pracy [*w osobo-miesiącach*],
c oraz ***d*** – współczynniki wyznaczone eksperymentalnie:

$$\boldsymbol{c} = 1.5$$
$$0.32 \le \boldsymbol{d} \le 0.38$$

im bardziej dziedzina problemu jest nieznana, tym współczynnik d jest wyższy.

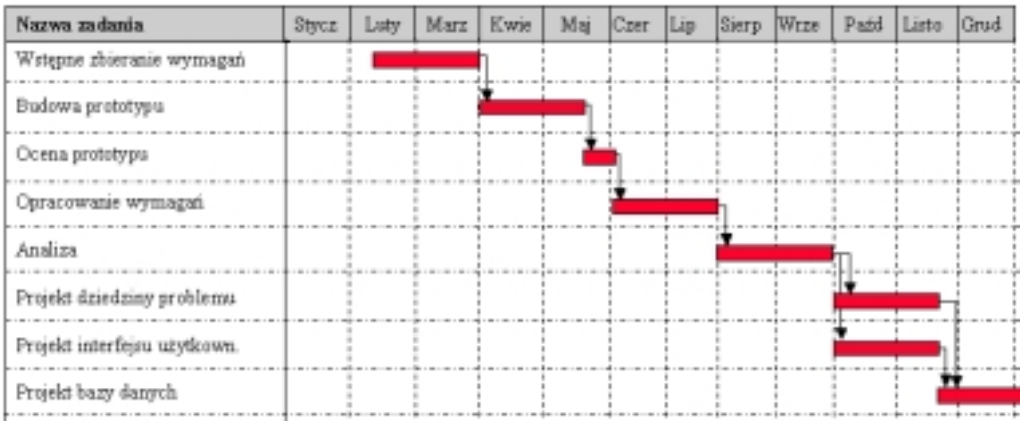
Wzory i współczynniki stosowane w tej metodzie są empiryczne.

Metoda **Function Point Analysis (FPA)**

Metoda „punktów funkcyjnych” szacuje koszt projektu na podstawie **funkcji użytkowych**, które system ma realizować. Metoda jest oparta na zliczaniu ilości wejść i wyjść systemu, miejsc przechowywania danych i innych kryteriów. Te dane są następnie mnożone przez zadane z góry wagi i sumowane. Rezultatem jest **liczba „punktów funkcyjnych”**.

Harmonogram przedsięwzięcia

Ustalenie planu czasowego dla poszczególnych faz i zadań.
Na przykład w postaci diagramu Gantta:



Oszacowanie niepewności i ryzyka

Najczęściej za pomocą „Drzewa ryzyka”, w którym:

- wierzchołki drzewa odpowiadają sytuacjom - zdarzeniom.
- krawędzie oznaczają przejścia do nowych sytuacji.
- krawędziom są przypisane prawdopodobieństwa.
- każde zakończenie scenariusza (liść w drzewie) jest opisane kosztem

Na podstawie takiego drzewa można oszacować wartość oczekiwaną zysku/kosztu.

Podstawowe rezultaty fazy strategicznej

<u>W postaci raportu dla klienta zawierającego:</u>

- definicję celów przedsięwzięcia
- opis zakresu przedsięwzięcia
- opis systemów zewnętrznych, z którymi system będzie współpracować
- szkicowy opis wymagań
- szkicowy model systemu
- opis proponowanego rozwiązania
- oszacowanie kosztów
- wstępny harmonogram prac

FAZA OKREŚLANIA WYMAGAŃ (SPECYFIKACJI I ANALIZY)

Celem fazy określenia wymagań jest skonstruowanie zbioru wymagań klienta wobec tworzonego systemu. Dokonywana jest zamiana celów klienta na konkretne wymagania zapewniające osiągnięcie tych celów. Błędy popełnione w tej fazie są bardzo kosztowne!

- Klient z reguły nie wie dokładnie w jaki sposób osiągnąć założone cele.
- Cele klienta mogą być osiągnięte na wiele sposobów.
- Duże systemy są wykorzystywane przez wielu użytkowników o niezgodnych (czasem sprzecznych) celach i posługujących się różną terminologią.
- Zleceniodawca nie musi być ostatecznym użytkownikiem i może źle rozumieć rzeczywiste potrzeby.

Opis wymagań powinien:

- być **kompletny** i **niesprzeczny**,
- opisywać zewnętrzne **zachowanie systemu** a nie sposób realizacji,
- opisywać **ograniczenia**, w jakich ma pracować system,
- opisywać zachowanie systemu, gdy te ograniczenia nie są spełnione,
- brać pod uwagę możliwość zmiany wymagań w przyszłości.

Dokument określający wymagania powinien zawierać:

- cele, zakres i kontekst. Czyli wyniki fazy strategicznej.
- wymagania funkcjonalne. Opis funkcji i operacji systemu.
- wymagania niefunkcjonalne. Opis ograniczeń sposobu działania.
- Model systemu.
- Słownik. Wyjaśnienie terminów niejasnych dla którejś ze stron lub niejednoznacznych.

Do wszystkich ustaleń należy podawać powody,
bo ułatwia to późniejsze wprowadzanie zmian.

Metody rozpoznania wymagań

- **Wywiady** – przygotowane w postaci listy pytań. Przeprowadzone na reprezentatywnej grupie użytkowników.
- **Studia** na istniejącym oprogramowaniem (zwłaszcza w przypadku, gdy nowe oprogramowanie zastępuje stare)
- **Budowanie prototypów** systemu działających w zmniejszonej skali lub z uproszczonymi interfejsami.

Metody specyfikacji wymagań:

- **Język naturalny** - najczęściej stosowany. Opis taki jest rezultatem wstępnych rozmów z klientem.
  Wady: niejednoznaczność, utrudnia wykrycie powiązanych wymagań, powoduje trudności w wykryciu sprzeczności.
- **Formalizm matematyczny** (rzadko stosowany).
- **Język naturalny strukturalny**. Język naturalny z ograniczonym słownictwem i składnią.
- **Tablice, formularze**. Wyspecyfikowanie wymagań w postaci tablic, kojarzących różne aspekty.
- **Diagramy blokowe**: forma graficzna pokazująca cykl przetwarzania.
- **Diagramy kontekstowe**: ukazują system w postaci jednego bloku oraz jego powiązania z otoczeniem, wejściem i wyjściem.
- **Diagramy przypadków użycia**: poglądowy sposób przedstawienia aktorów i funkcji systemu.

Typowym błędem występującym w tej fazie jest koncentrowanie się na sytuacjach typowych i pomijanie wyjątków oraz przypadków granicznych.

WYMAGANIA FUNKCJONALNE

Opisują funkcje (czynności, operacje) wykonywane przez system. Funkcje te mogą być również wykonywane przy użyciu systemów zewnętrznych.

Formularz wymagań funkcjonalnych - zapis jest podzielony na konkretne pola, co pozwala na łatwe stwierdzenie kompletności opisu oraz na jednoznaczną interpretację

Przykład:

Nazwa funkcji	***Edycja dochodów pracownika***
Opis	Funkcja pozwala edytować łączne dochody podatnika uzyskane w danym roku.
Dane wejściowe	Informacje o dochodach pracowników uzyskane uzyskanych z różnych źródeł: kwoty przychodów, koszty uzyskania, zapłacone zaliczki
Źródło danych wejściowych	Dokumenty oraz informacje dostarczone przez podatnika.
Warunek wstępny	Kwoty przychodów i kwoty kosztów są liczbami dodatnimi
Warunek końcowy	Kwota dochodu = kwota przychodu - kwota kosztów
Efekty uboczne	Uaktualnienie podstawy opodatkowania

Zazwyczaj operacji jest bardzo dużo i trzeba je jakoś uporządkować. Standardowym rozwiązaniem jest wprowadzenie hierarchii wymagań funkcjonalnych poprzez rozbicie funkcji na podfunkcje. Taka hierarchia może być reprezentowana w formacie tekstowym lub graficznym

Przykład:

WYMAGANIA NIEFUNKCJONALNE

Opisują inne ograniczenia, przy których system ma realizować swoje funkcje. Na przykład: wymagania dotyczące formy produktu (rodzaj interfejsu), dotyczące procesu (np. spełnianie norm czy standardów):

Cecha	**Ograniczenia**
Wydajność	Liczba transakcji na sekundę Czas odpowiedzi
Użycie zasobów	Wymagana pamięć operacyjna lub dyskowa
łatwość używania	Czas przeszkolenia pracowników Rodzaj i wielkość dokumentacji użytkownika
Niezawodność	Częstotliwość błędnego wykonania Procent czasu niedostępności systemu
Odporność	Czas restartu po awarii Prawdopodobieństwo zniszczenia danych w wyniku awarii
Przenośność	Procent kodu zależnego od platformy Liczba platform, na których działa Koszt przeniesienia na nową platformę

DOKUMENT SPECYFIKACJI WYMAGAŃ

Wymagania powinny być zebrane w dokumencie - opisie wymagań. Oprócz cech wymienionych na str. 6. ten dokument powinien:

- być podstawą szczegółowego kontraktu między klientem a producentem oprogramowania,
- pozwalać na weryfikację stwierdzającą, czy wykonany system rzeczywiście spełnia postawione wymagania,
- być zrozumiały dla obydwu stron.

Przykładowa zawartość dokumentu specyfikacji wymagań – Norma ANSI/IEEE „Recommended Practice for Software Requirements Specifications”

1. Wstęp
   1.1. Cel
   1.2. Zakres
   1.3. Definicje, akronimy i skróty
   1.4. Referencje, odsyłacze do innych dokumentów
   1.5. Krótki przegląd
2. Ogólny opis
   2.1. Walory użytkowe i przydatność projektowanego systemu
   2.2. Ogólne możliwości projektowanego systemu
   2.3. Ogólne ograniczenia
   2.4. Charakterystyka użytkowników
   2.5. Środowisko operacyjne
   2.6. Założenia i zależności
3. Specyficzne wymagania
   3.1. Wymagania funkcjonalne (funkcje systemu)
   3.2. Wymagania niefunkcjonalne (ograniczenia).