\*\*Zarządzanie projektem informatycznym\*\*

1. \*\*Klasyfikacja metodyk zarządzania projektami IT: jakie grupy metodyk można wyróżnić i czym się charakteryzują.\*\*

Metodyki zarządzania projektami IT dzielą się głównie na dwie grupy: tradycyjne i zwinne. Metodyki tradycyjne, takie jak kaskadowa (Waterfall), opierają się na sekwencyjnym podejściu, gdzie fazy projektu są ściśle określone i realizowane jedna po drugiej. Z kolei zwinne metodyki, takie jak Scrum i Kanban, charakteryzują się iteracyjnością i elastycznością, gdzie projekt jest realizowany w krótkich cyklach, a zmiany są łatwo wprowadzane na bieżąco. Wybór metodyki zależy od specyfiki projektu, złożoności i potrzeb klienta.

2. \*\*Wymienić nazwy znanych metodyk wytwórczych.\*\*

Do znanych metodyk wytwórczych należą: Scrum, Kanban, XP (Extreme Programming), PRINCE2, oraz Lean. Scrum opiera się na iteracyjnych sprintach, które pozwalają zespołom programistycznym regularnie dostarczać nowe funkcje. Kanban, z kolei, koncentruje się na wizualizacji zadań i ograniczaniu ilości prac w toku (Work In Progress - WIP). XP jest metodyką skoncentrowaną na wysokiej jakości kodu i częstym wydawaniu nowych wersji oprogramowania.

3. \*\*Co to jest RUP (objaśnić nazwę); jaki model procesu wytwórczego realizuje?\*\*

RUP to skrót od Rational Unified Process. Jest to metodyka inżynierii oprogramowania, która opiera się na iteracyjnym podejściu do rozwoju oprogramowania. Proces RUP składa się z czterech głównych faz: fazy początkowej (Inception), fazy elaboracji (Elaboration), fazy budowy (Construction) i fazy przekazania (Transition). Każda z tych faz ma jasno określone cele i zadania, co pozwala na stopniowe rozwijanie i weryfikację systemu.

4. \*\*Co oznacza nazwa MSF i jaki model procesu wytwórczego odwzorowuje?\*\*

MSF oznacza Microsoft Solutions Framework. Jest to zestaw metodyk i najlepszych praktyk opracowanych przez firmę Microsoft, które pomagają w zarządzaniu projektami IT i wdrażaniu rozwiązań technologicznych. MSF nie narzuca jednego konkretnego modelu procesu wytwórczego, ale wspiera zarówno podejścia sekwencyjne, jak i iteracyjne, w zależności od potrzeb projektu i zespołu.

5. \*\*Wymienić nazwy grup procesów wg PMBoK/PMI; jak są ze sobą powiązane?\*\*

W PMBoK (Project Management Body of Knowledge) wyróżnia się pięć głównych grup procesów: inicjowanie, planowanie, wykonanie, monitorowanie i kontrola oraz zamknięcie. Procesy te są powiązane ze sobą w sposób iteracyjny i cykliczny. Inicjowanie projektu prowadzi do planowania, które określa szczegóły wykonania. W trakcie realizacji monitoruje się postępy i kontroluje odchylenia od planu, a ostatecznie projekt jest zamykany po spełnieniu jego celów.

\*\*Systemy informacyjne\*\*

1. \*\*Czym jest ścieżka krytyczna w harmonogramie działań?\*\*

Ścieżka krytyczna to najdłuższa sekwencja zadań w projekcie, która wyznacza minimalny czas potrzebny na jego realizację. Oznacza to, że każde opóźnienie w realizacji zadań znajdujących się na ścieżce krytycznej spowoduje opóźnienie całego projektu. Identyfikacja ścieżki krytycznej jest kluczowa dla skutecznego zarządzania czasem, ponieważ pozwala skupić zasoby na najbardziej newralgicznych zadaniach.

2. \*\*Co jest wielkością sterującą - danymi wejściowymi - dla algorytmu planowania zapotrzebowań na materiały (MRP)?\*\*

Głównymi danymi wejściowymi dla algorytmu MRP są prognozy popytu na produkt końcowy, listy materiałowe (BOM), dostępne stany magazynowe oraz czas dostaw materiałów i komponentów. Algorytm MRP analizuje te dane w celu ustalenia, ile materiałów i komponentów należy zamówić, w jakich ilościach i w jakim czasie, aby zapewnić ciągłość produkcji i minimalizować zapasy.

3. \*\*Jak należy rozumieć Punkt Oddzielenia Zlecenia Klienta (Customer Order Decoupling Point)?\*\*

Punkt Oddzielenia Zlecenia Klienta (Customer Order Decoupling Point, CODP) to moment w procesie produkcji lub dostawy, w którym zamówienie klienta staje się częścią procesu. Do tego punktu procesy mogą być realizowane w sposób prognozowany, a po tym punkcie są dostosowywane do specyficznych wymagań klienta. CODP jest kluczowy w zarządzaniu łańcuchem dostaw, gdyż wpływa na czas realizacji zamówienia i elastyczność produkcji.

4. \*\*Podać znane sposoby opisu funkcji systemu: informacyjnego i informatycznego: nazwy i różnice.\*\*

Do znanych sposobów opisu funkcji systemu informacyjnego i informatycznego należą: diagramy przepływu danych (DFD), diagramy przypadków użycia (Use Case), oraz modele ERD (Entity-Relationship Diagram). Diagramy przepływu danych przedstawiają, w jaki sposób dane są przetwarzane i przepływają przez system. Diagramy przypadków użycia koncentrują się na interakcji użytkowników z systemem, opisując funkcjonalności z perspektywy użytkownika. Modele ERD opisują relacje między danymi w systemie.

5. \*\*Wymagania niefunkcjonalne przy analizie i projektowaniu systemu – wymienić i krótko scharakteryzować (3) główne grupy.\*\*

Wymagania niefunkcjonalne można podzielić na trzy główne grupy: wydajność, bezpieczeństwo i użyteczność. Wymagania dotyczące wydajności określają, jak szybko i efektywnie system ma działać w określonych warunkach. Wymagania bezpieczeństwa odnoszą się do ochrony danych i zapewnienia, że system jest odporny na ataki. Wymagania użyteczności dotyczą łatwości obsługi systemu, zarówno dla użytkowników końcowych, jak i administratorów.

6. \*\*Jakie zależności mogą wystąpić pomiędzy przypadkami użycia na diagramach Use Case a jakie pomiędzy aktorami?\*\*

Na diagramach przypadków użycia (Use Case) zależności między przypadkami mogą obejmować relacje "extends" i "includes". Relacja "includes" oznacza, że jeden przypadek użycia włącza inny jako część swojego działania, natomiast "extends" oznacza, że przypadek może być rozszerzony o dodatkową funkcjonalność w określonych warunkach. Aktorzy mogą być powiązani relacją dziedziczenia, co oznacza, że jeden aktor dziedziczy rolę i funkcje innego aktora.

\*\*Podstawy teoretyczne informatyki\*\*

1. \*\*Jednym z prostych sposobów zapisu algorytmu jest schemat blokowy. Proszę podać zasady tworzenia schematu blokowego.\*\*

Schemat blokowy to graficzne przedstawienie algorytmu za pomocą standardowych symboli. Każdy symbol ma przypisane określone funkcje, np. prostokąt reprezentuje operacje przetwarzania danych, a romb symbolizuje decyzje. Zasady tworzenia schematów blokowych obejmują przede wszystkim czytelność i logikę przepływu informacji, czyli strzałki, które wyznaczają kierunek realizacji poszczególnych kroków. Ważne jest, aby każdy schemat blokowy miał jasno określony początek (Start) i koniec (Stop).

2. \*\*Proszę omówić znaczenie pojęć: składnia i semantyka języka programowania.\*\*

Składnia języka programowania to zbiór reguł, które definiują poprawną strukturę instrukcji w danym języku. Oznacza to, że składnia określa, jak mają być zapisywane elementy języka (np. zmienne, operatory, wyrażenia). Z kolei semantyka to znaczenie tych instrukcji, czyli co one faktycznie robią. Na przykład w języku C składnia instrukcji warunkowej "if" mówi, jak ją zapisać, ale semantyka definiuje, że sprawdzany warunek decyduje o wykonaniu określonego bloku kodu.

3. \*\*Jednym z podstawowych pojęć w informatyce jest pojęcie zmiennej. Proszę podać cechy charakterystyczne zmiennej oraz omówić przynajmniej dwie klasy zmiennych.\*\*

Zmienna to miejsce w pamięci komputera, które przechowuje dane i którym przypisane jest unikalne imię lub identyfikator. Kluczowymi cechami zmiennej są jej typ (np. liczbowy, tekstowy), nazwa oraz wartość, którą może przechowywać. Dwie podstawowe klasy zmiennych to zmienne lokalne i zmienne globalne. Zmienne lokalne są dostępne tylko w ramach określonej funkcji lub bloku kodu, natomiast zmienne globalne są widoczne w całym programie.

4. \*\*Stos i sterta. Proszę omówić te pojęcia i na przykładzie w języku ANSI C pokazać kiedy i w jaki sposób korzystamy ze stosu, a kiedy ze sterty.\*\*

Stos i sterta to dwa różne obszary pamięci używane w programowaniu. Stos (stack) to struktura LIFO (Last In, First Out), w której dane są dodawane i usuwane w ustalonej kolejności. Jest on często używany do przechowywania zmiennych lokalnych i parametrów funkcji. Z kolei sterta (heap) to dynamiczny obszar pamięci, z którego można przydzielać pamięć w czasie działania programu. W ANSI C, zmienne lokalne są umieszczane na stosie, a dynamicznie alokowana pamięć (np. za pomocą funkcji `malloc`) pochodzi ze sterty.

5. \*\*Proszę omówić metody przekazywania argumentów do funkcji i pokazać je na przykładach w języku ANSI C.\*\*

W języku C istnieją dwie główne metody przekazywania argumentów do funkcji: przekazywanie przez wartość i przekazywanie przez wskaźnik. W przekazywaniu przez wartość, do funkcji przekazywana jest kopia argumentu, co oznacza, że zmiany dokonane w funkcji nie wpływają na oryginalną wartość. W przekazywaniu przez wskaźnik, do funkcji przekazywany jest adres zmiennej, co pozwala na modyfikowanie oryginalnej wartości. Na przykład:  
 void zmienPrzezWartosc(int x) {  
 x = 10;  
 }  
 void zmienPrzezWskaznik(int \*x) {  
 \*x = 10;  
 }

6. \*\*Rekurencja. Proszę omówić na czym polega i pokazać prosty przykład – można posłużyć się metajęzykiem.\*\*

Rekurencja to technika programowania, w której funkcja wywołuje samą siebie, aby rozwiązać mniejsze instancje problemu. Każda rekurencyjna funkcja musi posiadać warunek stopu, który zakończy wywołania, aby uniknąć nieskończonej pętli. Przykładem rekurencji jest obliczanie silni liczby:

int silnia(int n) {

if (n <= 1) return 1; // warunek stopu

return n \* silnia(n - 1); // wywołanie rekurencyjne

}

\*\*Metody probabilistyczne i statystyka\*\*

1. \*\*Miary tendencji centralnej i miary rozrzutu.\*\*

Miary tendencji centralnej to statystyki, które opisują centralny punkt rozkładu danych. Najczęściej używanymi miarami są średnia arytmetyczna, mediana i moda. Średnia to suma wszystkich wartości podzielona przez ich liczbę, mediana to wartość środkowa po uporządkowaniu danych, a moda to wartość najczęściej występująca. Miary rozrzutu, takie jak odchylenie standardowe i wariancja, opisują, jak rozproszone są dane wokół wartości centralnej.

2. \*\*Omówić problem estymacji przedziałowej.\*\*

Estymacja przedziałowa to metoda statystyczna polegająca na określeniu przedziału wartości, w którym z określoną pewnością znajduje się prawdziwa wartość estymowanego parametru (np. średniej populacji). Przedział ufności jest najczęściej używanym narzędziem estymacji przedziałowej. Na przykład, możemy obliczyć, że średnia populacji z 95% pewnością leży w przedziale od 10 do 15. Wartości skrajne przedziału są obliczane na podstawie rozkładu próbkowego oraz poziomu ufności.

3. \*\*Scharakteryzować podstawowe pojęcia i omówić krótko procedurę weryfikacji hipotezy statystycznej.\*\*

Weryfikacja hipotezy statystycznej polega na ocenie, czy założenie o populacji, na podstawie próby, jest prawdziwe. Procedura obejmuje sformułowanie hipotezy zerowej (H0), która zakłada brak efektu lub różnicy, oraz hipotezy alternatywnej (H1), która zakłada istnienie efektu lub różnicy. Następnie, na podstawie danych z próby, oblicza się statystykę testową i porównuje ją z wartością krytyczną dla danego poziomu istotności. Jeśli statystyka przekroczy wartość krytyczną, odrzuca się hipotezę zerową.

4. \*\*Statystyczne miary ilości informacji (entropia Shannona i miary pochodne).\*\*

Entropia Shannona to miara ilości niepewności związanej z losowym procesem lub zmienną. Wyraża, ile informacji potrzeba, aby dokładnie opisać rozkład prawdopodobieństwa danego procesu. W teorii informacji, entropia jest używana do mierzenia efektywności kodowania informacji. Im większa entropia, tym bardziej "nieprzewidywalny" jest sygnał. Miary pochodne, takie jak nadmiar informacyjny, określają, ile dodatkowej informacji można pozyskać w procesie kompresji danych.

5. \*\*Kompresja, kodowanie i szyfrowanie w informatyce.\*\*

Kompresja to proces redukcji rozmiaru danych przy zachowaniu ich treści. Kodowanie odnosi się do konwersji danych z jednej formy do innej, zazwyczaj w celu ich efektywnej transmisji lub przechowywania (np. kodowanie Huffmana). Szyfrowanie polega na przekształceniu danych w taki sposób, aby były one nieczytelne dla osób nieuprawnionych, poprzez zastosowanie algorytmów kryptograficznych. Szyfrowanie gwarantuje poufność danych, podczas gdy kompresja optymalizuje ich przechowywanie i transmisję.

\*\*Języki formalne i kompilatory\*\*

1. \*\*Kompilacja i interpretacja, języki skryptowe i kompilowane, język Java.\*\*

Kompilacja to proces, w którym cały kod źródłowy programu jest przetwarzany na kod maszynowy przed jego uruchomieniem. Kompilowane języki, takie jak C czy C++, wymagają wcześniejszego przetworzenia kodu źródłowego na plik wykonywalny. Interpretacja natomiast oznacza, że program jest przetwarzany i uruchamiany w trakcie jego działania przez interpreter, jak ma to miejsce w językach skryptowych, takich jak Python czy JavaScript. Java łączy cechy obu podejść: kod źródłowy jest najpierw kompilowany do kodu pośredniego (bytecode), który następnie jest interpretowany przez maszynę wirtualną JVM (Java Virtual Machine).

2. \*\*Podstawowe moduły funkcjonalne kompilatora, rola kodu pośredniego przy kompilacji.\*\*

Kompilator składa się z kilku podstawowych modułów funkcjonalnych: analizatora leksykalnego, analizatora syntaktycznego, analizatora semantycznego, generatora kodu pośredniego oraz optymalizatora i generatora kodu maszynowego. Kod pośredni, generowany po analizie syntaktycznej i semantycznej, stanowi abstrakcyjną formę programu, która może być optymalizowana i przekształcana na kod maszynowy odpowiedni dla różnych architektur sprzętowych. Użycie kodu pośredniego pozwala na większą przenośność i efektywność kompilacji.

3. \*\*Wyjaśnić pojęcia: analiza leksykalna, wyrażenie regularne, automat skończony.\*\*

Analiza leksykalna to proces, w którym kod źródłowy jest przetwarzany na mniejsze elementy zwane leksemami (tokeny). Analizator leksykalny grupuje ciągi znaków w logiczne jednostki, takie jak słowa kluczowe, identyfikatory, operatory czy literały. Wyrażenie regularne to formalny sposób opisywania wzorców w ciągach znaków i są one często wykorzystywane w analizie leksykalnej do identyfikowania leksemów. Automat skończony to model matematyczny służący do reprezentacji procesów o skończonej liczbie stanów, wykorzystywany w analizatorach leksykalnych do przetwarzania wejściowego ciągu znaków.

4. \*\*Wyjaśnić pojęcia: analiza syntaktyczna, gramatyka, automat ze stosem.\*\*

Analiza syntaktyczna (parsing) to etap przetwarzania kodu źródłowego, w którym analizator syntaktyczny sprawdza, czy ciąg tokenów generowany przez analizator leksykalny spełnia zasady gramatyki języka programowania. Gramatyka to zbiór reguł definiujących poprawną strukturę wyrażeń w języku. Automat ze stosem to zaawansowany model automatu, który korzysta ze stosu do przechowywania dodatkowych informacji o stanie analizy, co pozwala mu na skuteczne przetwarzanie bardziej złożonych struktur językowych, takich jak zagnieżdżone nawiasy.

5. \*\*Przykładowe techniki optymalizacji kodu pośredniego.\*\*

Optymalizacja kodu pośredniego ma na celu poprawę efektywności kodu wynikowego poprzez redukcję nadmiarowych operacji lub wykorzystanie bardziej efektywnych instrukcji. Przykłady technik optymalizacji obejmują eliminację martwego kodu (usuwanie instrukcji, które nie wpływają na wynik), minimalizację liczby przejść do innych bloków kodu (redukowanie skoków) oraz zastępowanie złożonych wyrażeń prostszymi, które mają taką samą funkcjonalność (optymalizacja siłowa). Inne techniki to propagacja stałych, gdzie wartości stałe są wstawiane bezpośrednio do kodu zamiast zmiennych, oraz unikanie powtarzających się obliczeń.