

Kod szkolenia: **EA**

Tytuł szkolenia: **Enterprise Architect - narzędzie do modelowania**

Dni: **3**

Opis:

Adresaci szkolenia

Szkolenie adresowane jest do osób, które już potrafią modelować w UML jednakże mają potrzebę poznania narzędzia Enterprise Architect na poziomie zaawansowanym, wliczając w to zaawansowane opcje tworzenia dokumentacji, pracę zespołową, tworzenie stereotypów graficznych i wiele innych.

Cel szkolenia

Celem szkolenia jest poznanie i opanowanie obsługi zaawansowanych funkcjonalności Enterprise Architect. Po to aby można było łatwiej pracować w zespole, ujednoczyć dokumentację, czy pracować szybciej w czasie modelowania.

Wymagania:

Umiejętność modelowania w UML.

Parametry szkolenia

3*8 godzin (3*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Wielkość grupy: maks. 8-10 osób.

Program szkolenia:

1. UML - wprowadzenie
 - Podstawy obiektowości
 - I. Klasa (Class)
 - II. Atrybut (Attribute)
 - III. Metoda (Method)
 - IV. Obiekt (Object)
 - V. Dziedziczenie (Generalization)
 - VI. Polimorfizm (Polymorphism)
 - Czym jest UML i jego rozwój
 - Na czym polega modelowanie obiektowe



- Główne elementy UML
 - I. Główne kwalifikatory
 - i. Klasa (Class)
 - ii. Obiekt (Object)
 - iii. Interfejs (Interface)
 - iv. Komponent (Component)
 - v. Węzeł (Node)
 - vi. Aktor (Actor)
 - vii. Przypadek Użycia (Use Case)
 - II. Relacje
 - i. Zależność (Dependency)
 - ii. Asocjacja (Association)
 - iii. Generalizacja (Generalization)
 - iv. Realizacja (Realization)
 - III. Komentarze (Note)
 - IV. Rozszerzanie UML
 - i. Stereotypy (Stereotype)
 - ii. Ograniczenia (Constraints)
 - iii. Etykiety (Tagged Values)
- Różnica między diagramem a modelem
- Zastosowania
- 2. Wymagania i przypadki użycia (use case model)
 - Wymagania - czym są, trudności, po co analiza?
 - Analiza wymagań - proces
 - Przypadki użycia
 - I. Przypadek użycia (use case)
 - II. Analiza
 - III. Dokumentacja - opis i scenariusze
 - Diagram przypadków użycia (use case diagram)
 - I. Aktor (Actor)
 - II. Przypadek użycia (use case)
 - III. Granica kontekstu (boundary)
 - IV. Relacje
 - i. Asocjacja (association)
 - ii. Dziedziczenie (generalization)
 - iii. Zawieranie <<include>>
 - iv. Rozszerzanie <<extend>>
- 3. Modelowanie procesów biznesowych
 - Czym jest proces biznesowy i po co modelować
 - Różne języki opisu
 - Diagram aktywności (activity diagram)
 - I. Aktywność (activity)
 - II. Akcja (action)
 - III. Rozpoczynanie i kończenie
 - i. Początek (initial node)
 - ii. Koniec przepływu (flow final node)

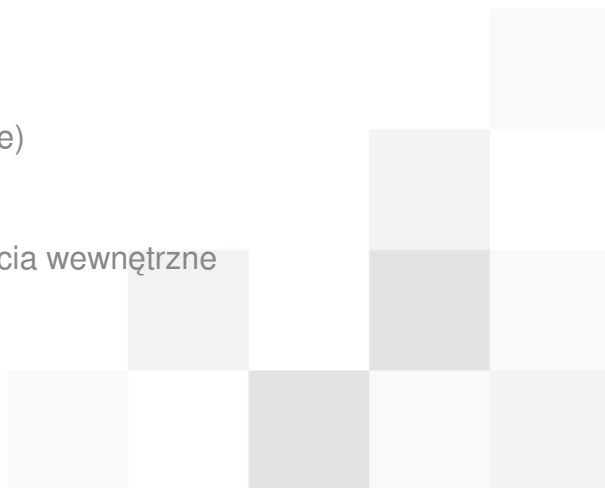


- iii. Koniec aktywności (activity final node)
 - IV. Warunki (guard)
 - V. Zrównoleglanie (fork, join)
 - VI. Redefiniowanie złączenia (join specification)
 - VII. Decyzje i złączenia (decision, merge)
 - VIII. Przepływy sterowania (control flow)
 - IX. Waga (weight)
 - X. Przepływy obiektów (object flow)
 - i. Notacje obiektu (obiekt, pin)
 - ii. Obiekt w danym stanie
 - XI. Zaawansowane notacje pinów
 - i. Wejściowy i wyjściowy
 - ii. O charakterze wyjątku
 - iii. W przepływie strumieniowym
 - iv. Anonimowa notacja pinów
 - XII. Stereotypy dla obiektów
 - i. Bufor centralny (central buffer)
 - ii. Składnica danych (datastore)
 - XIII. Partycje (activity partition)
 - XIV. Wyjątki (exception)
 - XV. Sygnały
 - i. Wysłanie (send signal action)
 - ii. Odbiór (accept event action)
 - iii. Sygnał czasowy (accept time event)
 - XVI. Obszar przerywalny (interruptible region)
 - i. Sygnały w obszarze przerywalnym
 - XVII. Wyjątki
 - XVIII. Regiony rozszerzenia (expansion region)
 - i. Iterowanie (iterative)
 - ii. Zrównoleglanie (parallel)
 - iii. Przetwarzanie strumieniowe (stream)
4. Projekt statyczny
- o Porównanie modelu projektowego z analitycznym
 - o Diagram klas (class diagram)
 - I. Klasa (class)
 - i. Atrybuty i metody
 - ii. Elementy statyczne
 - II. Widoczność (visibility)
 - III. Dziedziczenie (generalization)
 - i. Zbiory dziedziczenia (generalization set)
 - ii. Ograniczenia dla dziedziczenia
 - A. disjoint
 - B. overlapping
 - C. complete
 - D. incomplete
 - IV. Abstrakcje i interfejsy (abstract class, interface)



- V. Realizacja (realization)
 - VI. Zależności i stereotypy
 - i. Tworzenie <<instantiate>>
 - ii. Wysłanie <<send>>
 - iii. Wywołanie <<call>>
 - iv. Udoskonalenie <<refine>>
 - v. Ślad <<trace>>
 - vi. Dostarczenie <<derive>>
 - VII. Asocjacja (association)
 - i. Elementy opisu asocjacji
 - A. Nazwa i role
 - B. Liczności
 - C. Poziomy dostępu
 - D. Nawigowalność/Kierunkowość
 - ii. Asocjacje wielokrotne
 - iii. Asocjacje zwrotne
 - iv. Agregacja i kompozycja
 - v. Asocjacja kwalifikowana (qualified association)
 - VIII. Klasa asocjacji (association class)
 - IX. Jak dokonać identyfikacji klas
 - X. Dane, czyli model domeny
5. Projekt dynamiczny
- o Analiza CRC - opis i zagrożenia
 - o Diagram sekwencji (sequence diagram)
 - I. Obiekt (object)
 - II. Linia życia (life line)
 - III. Rodzaje komunikatów (messages)
 - i. Synchroniczny (synchronous message)
 - ii. Asynchroniczny (asynchronous message)
 - iii. Zwrotny (return message)
 - iv. Zgubiony (lost message)
 - v. Odnaleziony (found message)
 - vi. Tworzenie obiektu <<create>>
 - vii. Niszczanie obiektu <<destroy>> i destruction event
 - IV. Ośrodek sterowania (execution specification)
 - V. Bloki złożone (combined fragment)
 - i. Pętla (loop) – loop
 - ii. Alternatywa (alternative) – alt
 - iii. Przerwanie (break) - break
 - iv. Współbieżność (parallel) – par
 - v. Blok krytyczny (critical) - critical
 - vi. Ścisłe uporządkowanie (strict sequencing) - stricte
 - vii. Odwołanie (interaction use) - ref
 - VI. Bloki złożone a notacja w UML 1.x
 - o Sekwencje w analizie - stereotypy analityczne
 - I. Granica <<boundary>>

- II. Logika <<control>>
- III. Dane <<entity>>
- Jak modelować analitycznie
- 6. Inne diagramy (przegląd)
 - Komponentów (component diagram)
 - I. Komponent (component)
 - II. Interfejs (interface)
 - i. Wymagany (required interface)
 - ii. Dostarczany (provided interface)
 - III. Złączenie interfejsów (assembly)
 - Diagram wdrożenia (deployment diagram)
 - I. Węzeł (node)
 - II. Łącze komunikacji (communication path)
 - Pakietów (package diagram)
 - I. Pakiet
 - II. Podstawowe zależności
 - i. <<import>>
 - ii. <<access>>
 - Obiektów (object diagram)
 - I. Obiekty nazwane i anonimowe
 - II. Instancja/Obiekt (instance specification)
 - Struktur połączonych (composite structure diagram)
 - I. Część (part)
 - i. Port
 - ii. Interfejsy i złączenie
 - II. Połączenie
 - III. Współpraca (collaboration)
 - i. Rola i jej przypisanie
 - ii. Element współpracujący
 - IV. Wystąpienie współpracy (collaboration use)
 - Maszyna stanowa (state machine diagram)
 - I. Stan (state)
 - II. Przejście (transition)
 - i. Proste
 - ii. Zwrotne
 - iii. Wewnętrzne
 - iv. Automatyczne
 - III. Sygnatura przejścia
 - IV. Pseudo-stany
 - i. Początek (initial)
 - ii. Koniec (final)
 - iii. Zniszczenie (terminate)
 - V. Czynności w stanie
 - i. entry, exit i do
 - ii. Własne akcje - przejścia wewnętrzne
 - VI. Zdarzenia



- i. Zdarzenia czasowe (at, after)
 - ii. Zdarzenie zmiany stanu (when)
 - iii. Sygnał (signal)
 - o Przegląd interakcji (interaction overview diagram)
 - I. Przepływ sterowania
 - II. Początek i koniec
 - III. Fragment i wystąpienie interakcji
 - IV. Decyzja
 - o Harmonogramowanie (timing diagram)
 - I. Ograniczenia i komunikaty
 - II. Dwie notacje harmonogramowania
 - o Komunikacja (communication diagram)
 - I. Obiekt i rodzaje komunikatów
 - II. Kolejność komunikatów (notacja zagnieżdżona)
- 7. Przejrzysty model w narzędziach UML
 - o Perspektywy 4+1
 - o Podział na pakiety z przykładami
 - o Podpinanie diagramów pod inne elementy
- 8. Enterprise Architect a praca w zespole
 - o Import/Export (XMI)
 - o Wewnętrzna korespondencja (Team Review)
 - o Punkty porównania zmian (Baseline)
 - o Wersjonowanie na bazie danych
 - o Wersjonowanie z systemami wersjonowania (CVS, SVN...)
 - o Najlepsze praktyki
- 9. Zaawansowane aspekty UML
 - o Meta Object Facility (MOF)
 - o Pakiet CORE
 - o Profile
 - I. Profil (profile)
 - II. Meta-Klasa, stereotyp i rozszerzenie
 - III. Użycie profilu (profile application)
 - o OCL - język ograniczeń
 - I. Zastosowanie i składnia
 - II. Kontekst wyrażenia
 - o Model Driven Architecture (MDA)
 - I. Modele MDA
 - i. Computation independend model (CIM)
 - ii. Platform independend model (PIM)
 - iii. Platform specific model (PSM)
 - iv. Platform model
 - II. Transformacje PIM
- 10. Ćwiczenia obejmują poniższe zagadnienia z obsługi Enterprise Translator
 - o Wytwarzanie dokumentacji
 - I. Do pliku i do HTML
 - II. Dla szablonów gotowych i własnych (tworzenie)

- III. Słownik terminów dla dokumentacji w innym języku
- IV. Dokumenty wewnętrzne (Linked Document)
- V. Grupy dokumentów
- VI. Tworzenie dokumentu z kilku innych dokumentów
- Definiowanie słownika projektowego
- Praca zespołowa z Enterprise Architect
 - I. Import/Export
 - II. Wersjonowanie
 - III. Unikanie konfliktów przy pracy zespołowej
- Zarządzanie treścią na poziomie zaawansowanym
 - I. Śledzenie zależności między elementami
 - II. Macierze zależności
 - III. Elementy złożone (composite)
 - IV. Jak ułatwić pracę na elementach zagnieżdżonych
 - V. Definiowanie list wyboru (statusy, rodzaje elementów itp.)
 - VI. Ustrukturalizowane scenariusze (Use Case)
 - i. Szybkie dodawanie pojęć słownikowych
 - ii. Automatyczne generowanie diagramów ze scenariuszy
 - iii. Automatyczne tworzenie relacji <<include>> i <<extend>>
 - VII. Powiadamianie o zmianach w modelu
 - VIII. Ulubione (Favorites)
- Automatyczne generowanie kodu
 - I. Domyślne typy kolekcji
 - II. Określanie get'terów i set'terów
 - III. Wybór silnika bazy danych
 - IV. Typy generyczne
 - V. Szablony kodu
 - VI. Reverse engeneering
- Elementy graficzne na diagramach
 - I. Diagram "rysowany ręcznie" w Enterprise Architect
 - II. Kontrola wyświetlania szczegółów na diagramie i dla elementu
 - III. Dopinanie elementów graficznych
 - i. Pod element
 - ii. Definiowanie bibliotek obrazków
 - iii. Własne stereotypy graficzne
 - iv. Tworzenie i używanie nowych profili
 - IV. Problemy prezentacji na diagramach
 - i. Scalanie błędnych podziałów aktywności
 - ii. Wyłączenie automatycznego odświeżania
 - iii. Logika w konstruktorze
- Skrótory czyli szybkie wytwarzanie
 - I. Kopiowanie elementu
 - II. Stworzenie ostatnio dodanego elementu
 - III. Stworzenie ostatnio dodanej referencji
 - IV. Wiązanie relacjami przez zagnieżdżanie
 - V. Szybkie edytowanie



- i. Dodanie/usunięcie pod-elementów
 - ii. Zmiana nazwy
- VI. Szybkie generowanie dokumentacji
- I wiele innych...

