

Podstawy modelowania systemów w języku UML

Cele szkolenia

Celem szkolenia jest:

- praktyczne wykorzystanie języka UML w projektowaniu systemów.
- odczytywanie i interpretacja podstawowych, niezbędnych w dokumentacji projektowej diagramów UML.
- zapisanie poprawnej notacji UML (zgodnie ze standardami światowymi).
- tworzenie dokumentacji projektów informatycznych (również funkcjonalnej/reklamowej, użytkowej i wykonawczej/technicznej).
- wykorzystanie narzędzia wspomagającego użycie UML (na przykładzie programu Enterprise Architect).

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- wykonał wstępną analizę potrzeb klienta,
- modelował środowisko biznesowe oraz podstawową architekturę tworzonych systemów,
- tworzył algorytmy przebiegu procesów (biznesowych i informatycznych),
- budował procesy analityczno-projektowe i deweloperskie.

Profil uczestników

Szkolenie przeznaczone jest dla osób, które pracują w ramach zespołów tworzących systemy informatyczne i potrzebują języka usprawniającego współpracę (wewnątrz zespołu oraz z klientami).

Szkolenie dla analityków biznesowych i systemowych, programistów, testerów, dokumentalistów, wdrożeniowców oraz dla menedżerów w/w zespołów.

Przygotowanie uczestników

Wymagana wiedza uczestników: ogólna znajomość zagadnień związanych z procesami projektowania i tworzenia systemów.

Szczegółowy program szkolenia

1. Rozpoczęcie szkolenia – sprawy organizacyjne
2. Język UML
 - 2.1. Cel i zastosowanie
 - 2.2. Korzyści płynące z dobrej dokumentacji projektowej
 - 2.3. Potrzeba istnienia (sensownych) standardów
 - 2.4. Powstanie i wersje języka
 - 2.5. Do czego służy UML?

- 2.6. UML a projektowanie obiektowe
- 2.7. Typy diagramów
- 2.8. Cechy dobrego modelu w UML
- 2.9. Podstawowe problemy przy projektowaniu
- 2.10. Narzędzia wspomagające
3. Tworzenie modelu
 - 3.1. Budowa okna programu
 - 3.2. Tworzenie nowego modelu/projektu
 - 3.3. Podstawowe operacje na modelu
4. Class Diagram – słownik terminologii klienta i architektura systemu
 - 4.1. Podstawowe pojęcia związane z projektowaniem obiektowym
 - 4.2. Analiza językowa
 - 4.3. Tworzenie diagramu klas
 - 4.4. Ćwiczenia
 - 4.5. Informacje uściślające definicje klas
 - 4.5.1. Atrybuty klasy – właściwości obiektów systemu
 - 4.5.2. Operacje klasy – funkcje i działania obiektów systemu
 - 4.5.3. Dobra praktyka – typy danych dla atrybutów i argumentów operacji
 - 4.5.4. Dodatkowe informacje uściślające definicję klasy – cel i ograniczenia obiektów systemu
 - 4.5.5. Ćwiczenia
 - 4.6. Związki (ang. relationships) między klasami – struktura zależności obiektów systemu
 - 4.6.1. Złożenie lub agregacja
 - 4.6.2. Powiązanie (asocjacja) między klasami
 - 4.6.3. Role powiązanych klas
 - 4.6.4. Liczebność dla związku między klasami
 - 4.6.5. Ćwiczenia [40 min.]
 - 4.7. Stereotypy – rozszerzenia języka
 - 4.8. Kryteria poprawności diagramu UML
5. Object Diagram – specyficzne obiekty systemu
 - 5.1. Konwencje UML
 - 5.2. Tworzenie diagramu obiektów
6. Package Diagram
 - 6.1. Tworzenie diagramu pakietów
7. Use Case Diagram – procesy realizowane przez system
 - 7.1. Analiza funkcjonalna – cel i znaczenie
 - 7.2. Diagram przypadków użycia w analizie funkcjonalnej
 - 7.3. Podstawowe elementy
 - 7.3.1. Przypadek użycia (use case)
 - 7.3.2. Aktor (actor)
 - 7.3.3. System
 - 7.4. Konwencje UML
 - 7.5. Tworzenie diagramu przypadków użycia
 - 7.6. Ćwiczenia
 - 7.7. Scenariusze (scenarios) – czynności składowe procesów
 - 7.7.1. Struktura scenariusza
 - 7.7.2. Znaczenie scenariuszy w analizie
 - 7.7.3. Znaczenie scenariuszy w projekcie
 - 7.7.4. Dodawanie scenariuszy

- 7.7.5. Ćwiczenia
- 7.7.6. Optymalizacja tworzenia scenariuszy
- 7.7.7. Kryteria wydzielenia procesów w trakcie pisania scenariuszy
- 7.8. Związki na diagramach przypadków użycia – zależności między procesami w systemie
 - 7.8.1. Asocjacja aktora z przypadkiem użycia
 - 7.8.2. Przypadek zawierany (include)
 - 7.8.3. Przypadek rozszerzający (extend)
 - 7.8.4. Punkt rozszerzenia przypadku użycia (extension point)
 - 7.8.5. Ćwiczenia
- 8. Activity Diagram – algorytmy przebiegu procesów
 - 8.1. Podstawowe pojęcia
 - 8.2. Konwencje UML
 - 8.3. Tworzenie diagramu czynności
 - 8.3.1. Punkt początkowy (initial point)
 - 8.3.2. Czynność (activity)
 - 8.3.3. Punkt końcowy (final point)
 - 8.3.4. Przepływ sterowania (control flow)
 - 8.3.5. Czynność uszczegółowiona
 - 8.4. Przepływy decyzyjne – uwarunkowane rozgałęzienie procesu
 - 8.5. Przepływy współbieżne – przetwarzanie równoległe
 - 8.5.1. Rozwidlenie (fork node) lub scalenie (join node)
 - 8.5.2. Specyfikacja scalenia (join specification)
 - 8.6. Tory (swimlanes) i partycje (partitions) – oznaczanie odpowiedzialności lub miejsca realizacji
 - 8.7. Wagi (weights) – uruchomienie czynności po przekroczeniu wartości granicznej
 - 8.8. Sygnały (signals) – uruchomienie czynności zdalne lub w/po określonym czasie
 - 8.9. Obszary rozszerzenia (expansion regions) – przetwarzanie strumieni danych lub kolejek obiektów
 - 8.10. Obszary przerywania (interruptible activity regions) – zdalne, czasowe lub awaryjne przerywanie czynności/procesu
 - 8.11. Ćwiczenia
- 9. Communication Diagram – szkic komunikacji obiektów w procesie
 - 9.1. Podstawowe elementy
 - 9.1.1. Klasyfikator (classifier)
 - 9.1.2. Asocjacja (association)
 - 9.1.3. Komunikat (communicate)
 - 9.2. Konwencje UML
 - 9.3. Tworzenie diagramu komunikacji
 - 9.4. Ćwiczenia
- 10. Requirements Diagram – potrzeby klienta/użytkowników systemu
 - 10.1. Wymagania funkcjonalne i нефункционалне
 - 10.2. Dobra specyfikacja wymagań
 - 10.3. Zbieranie wymagań
 - 10.4. Związki między wymaganiami
 - 10.5. Konwencje UML
 - 10.6. Tworzenie diagramu wymagań
 - 10.6.1. Automatyczna numeracja wymagań
 - 10.6.2. Kolory oznaczające status wymagań
 - 10.6.3. Status wymagania
 - 10.7. Sprawdzanie kompletności modelu [10 min.]

- 10.8. [opcja] Ćwiczenia [30 min.]
11. Wdrażanie diagramów UML w projektach
12. Przydatne mechanizmy programu EA
13. [opcja] Szkolenie a egzamin OCUP
14. Zakończenie szkolenia (test + omówienie wyników, ocena szkolenia)

Metoda realizacji szkolenia

Wykład + ćwiczenia w ilości proporcjonalnej do wykładu + indywidualne sprawdzenie i omówienie ćwiczeń

Liczba dni, liczba godzin szkoleniowych

3 dni, 24 godzin szkoleniowe

Ścieżka rozwoju po szkoleniu

- Szkolenia z zakresu używania pozostałych typów diagramów UML.
- Szkolenie „*Modelowanie procesów biznesowych w BPMN 2.0*” – dla analityków biznesowych i systemowych (przygotowujących i wykorzystujących opisy procesów biznesowych jako podstawę do tworzenia systemów informatycznych).