Object-oriented analysis and design (OOA/D)

OOD: progettazione guidata dalle responsabilità: quali sono gli oggetti/classi? Che caratteristiche hanno? Come interagiscono tra loro?

Gli oggetti rappresentano il dominio del problema

OOA ⇒ OOD ⇒ OOP

1. Definizione dei casi d’uso
2. Definizione del dominio
3. Assegnamento responsabilità e collaborazioni
4. Definizione dei diagrammi delle classi

Unified Process (UP): processo iterativo orientato ad agli oggetti, utilizza UML. Può essere applicato usando un approccio agile come XP o scrum

Unified Modeling Languguage: linguaggio visuale per la costruzione di documentazione di un sistema software. Può essere usato come abbozzo (anche informale, nella modellazione agile l’uso di UML come abbozzo è enfatizzato) per esplorare e comunicare.

* Livello concettuale: visualizzazione dei concetti del mondo reale
* Livello software: visualizzazione delle classi ed elementi software

Unified process

Processo iterativo ed evolutivo (incrementale) orientato agli oggetti. Le iterazioni iniziali sono guidate dal rischio, dal cliente e dall’architettura.

4 fasi:

1. Ideazione: visione approssimativa, studio e stime dei costi e dei tempi. Non si entra nel merito dei requisiti bensì si indaga sulla fattibilità del progetto.
2. Elaborazione: raffinamento visione, implementazione iterativa del nucleo dell’archiettura, identificazione dei maggiori requisiti, della portata, risoluzione problemi maggiori
3. Costruzione: implementazione iterativa degli elementi rimaneneti, più facili e meno rischiosi
4. Transizione: beta test, rilascio

La fine di ogni fase, sequenziale, è detta milestone.

Disciplina: insieme di attività e dei relativi elaborati (prodotto del lavoro, dell’attività) appartenenti ad una determinata area, e.g. le attività di analisi dei requisiti

Discipline ingegneristiche di UP:

* Modellazione del business: modello del dominio e dell’ambito del problema
* Requisiti: raccolta dei requisiti del sistema
* Progettazione: analisi e design dei requisiti e dell’architettura
* Implementazione: progetto, codifica e test dei componenti
* Test: controllo qualità, test di integrazione e di sistema
* Rilascio: consegna e messa in funzione
* Gestione delle configurazioni e del cambiamento: manutenzione
* Gestione progetto: attività di pianificazione e governo del progetto
* Infrastruttura: attività a supporto del team, riguardo ai progessi e strumenti usati

Le discipline non sono sequenziali, bensì si eseguono ad ogni iterazione.

Requisiti evolutivi

Capacità o requisito al quale il sistema dev’essere conforme

* Requisiti funzionali: detti “comportamentali”, descrivono il comprotamento, le funzioni del sistema
* Requisiti non funzionali: l’insieme delle proprietà del sistema, includendo sicurezza, prestazioni, scalabilità, usabilità…