

Dr. Andrea Baruzzo

Consulente Indipendente

Via Circonvallazione 26, 33055 Muzzana del Turgnano, (UD)

PI: 02567160300 CF: BRZNDR73D15I403M

e-mail: abaruzzo@computer.org

Mobile: (+39)3393955753

Formazione

Corso di Formazione: Progettazione orientata agli oggetti

Descrizione del corso

Il corso è strutturato in due parti ed affronta alcune tematiche relative alla progettazione del software, con particolare riferimento all'approccio orientato agli oggetti.

La parte A (I Fondamenti) descrive il paradigma ad oggetti come evoluzione di quello procedurale. Quindi vengono descritti alcuni principi, tecniche ed euristiche che caratterizzano il "progettare ad oggetti". Viene inoltre descritta la metodologia del Design-by-Contract. Chiude il corso una discussione critica sui design pattern (esempi ricorrenti di buone soluzioni di design) e sugli anti-pattern (esempi ricorrenti di cattive soluzioni di design). L'insieme di linee guida proposte nasce dall'intento di fornire strumenti utili sia per *valutare* la qualità interna di un design, sia per analizzare *design alternativi* prima di prendere una specifica decisione progettuale. La prospettiva con la quale vengono discussi gli argomenti è *pragmatica*, bilanciando contenuti teorici ed esempi, esercitazioni e piccoli casi di studio (workshop) scelti di comune accordo con l'azienda presso cui il corso viene tenuto. L'obiettivo finale è di trasferire concetti e strumenti per progettare meglio "in pratica", in un'ottica di qualità intesa come soddisfacimento dei requisiti funzionali e non funzionali (vincoli) del sistema da sviluppare. Vengono tenuti in particolare considerazione aspetti quali la comunicazione interna al gruppo di lavoro e la *sostenibilità economica* dello sviluppo complessivo (sviluppo iniziale, manutenzione nel breve periodo e manutenzione/estensione nel lungo periodo).

La parte B (La metodologia Domain-Driven Design) propone un approccio concreto alla progettazione orientata agli oggetti basato sulla modellazione del software. Il Domain-Driven Design (DDD) è

particolarmente efficace nelle seguenti situazioni:

- per progettare sistemi la cui complessità risiede principalmente nella componente di dominio (*complessità essenziale*), anziché nella specifica tecnologia utilizzata (*complessità accidentale*);
- per separare la componente di dominio, tipicamente riutilizzabile, dalla componente tecnologica, solitamente più fragile e soggetta a continue modifiche;
- per guidare l'implementazione mediante un modello utile sia per mantenere l'integrità concettuale del design *durante* la codifica, sia per distillare precocemente concetti, relazioni e assunzioni (anche latenti e/o tacite) che possono avere un forte impatto sulla struttura della soluzione.

Il corso introduce la metodologia DDD, i pattern fondamentali, alcune tecniche di refactoring del modello ed alcuni pattern di "design strategico".

A chi è rivolto il corso

Architetti del software, analisti e programmatori.

Livello di difficoltà

Parte A: Beginner–intermediate.

Parte B: Intermediate.

Prerequisiti

Parte A: Esperienza con la programmazione imperativa di tipo procedurale (C, Fortran, Cobol, ecc.); familiarità con gli elementi di base dei linguaggi di programmazione orientati agli oggetti (ad es., C++, Java, C#, ecc.).

Parte B: Conoscenza di base di UML e della modellazione orientata agli oggetti. Buona conoscenza del paradigma orientato agli oggetti e dei principi di design OO.

Personalizzazione del corso

I contenuti del corso possono essere personalizzati su richiesta in base alle specifiche necessità dell'azienda, con riferimento particolare alla composizione del gruppo di lavoro e/o del tipo di progetti. Anche i casi di studio che affiancano le ore di teoria possono venire suggeriti dall'azienda stessa. Ogni caso di studio viene tenuto in modo fortemente interattivo. Ogni partecipante al corso ha quindi la possibilità di sperimentare direttamente i concetti e le tecniche apprese sotto la guida del docente. Per maggiori informazioni sui contenuti e per concordare durata e programmazione, contattatemi.

Esempio di programmazione del corso

Parte A: 40–56 ore, normalmente suddivise in 10–14 lezioni da 4 ore ciascuna.

Parte B: 56–60 ore, normalmente suddivise in 14–15 lezioni da 4 ore ciascuna.

Caso di studio da definire a parte, o come integrazione nelle rispettive parti (A e/o B)

Il docente

Il Dr. Andrea Baruzzo si è laureato nel 2001 in Scienze dell'Informazione presso l'Università degli Studi di Udine dove, nel 2008, ha conseguito il dottorato di ricerca in informatica. Si occupa di ricerca e formazione sia in ambito accademico, sia in ambito industriale. Ha svolto attività di consulenza e trasferimento tecnologico in svariati domini applicativi: in avionica (sistemi UAV velivoli teleguidati senza pilota), in progetti di tipo gestionale orientati sia ai dati, sia ai task/servizi (sistemi informativi enterprise, sistemi supply-chain, sistemi di simulazione finanziaria), in sistemi GIS (Geographic Information System) e grafica 3D.

Le sue principali aree di interesse sono: l'analisi, la progettazione e lo sviluppo di sistemi software ad oggetti (OOA/OOD/OOP), le tecniche di test e la modellazione del software. È inoltre membro di IEEE Computer Society.