

2

0

1

0

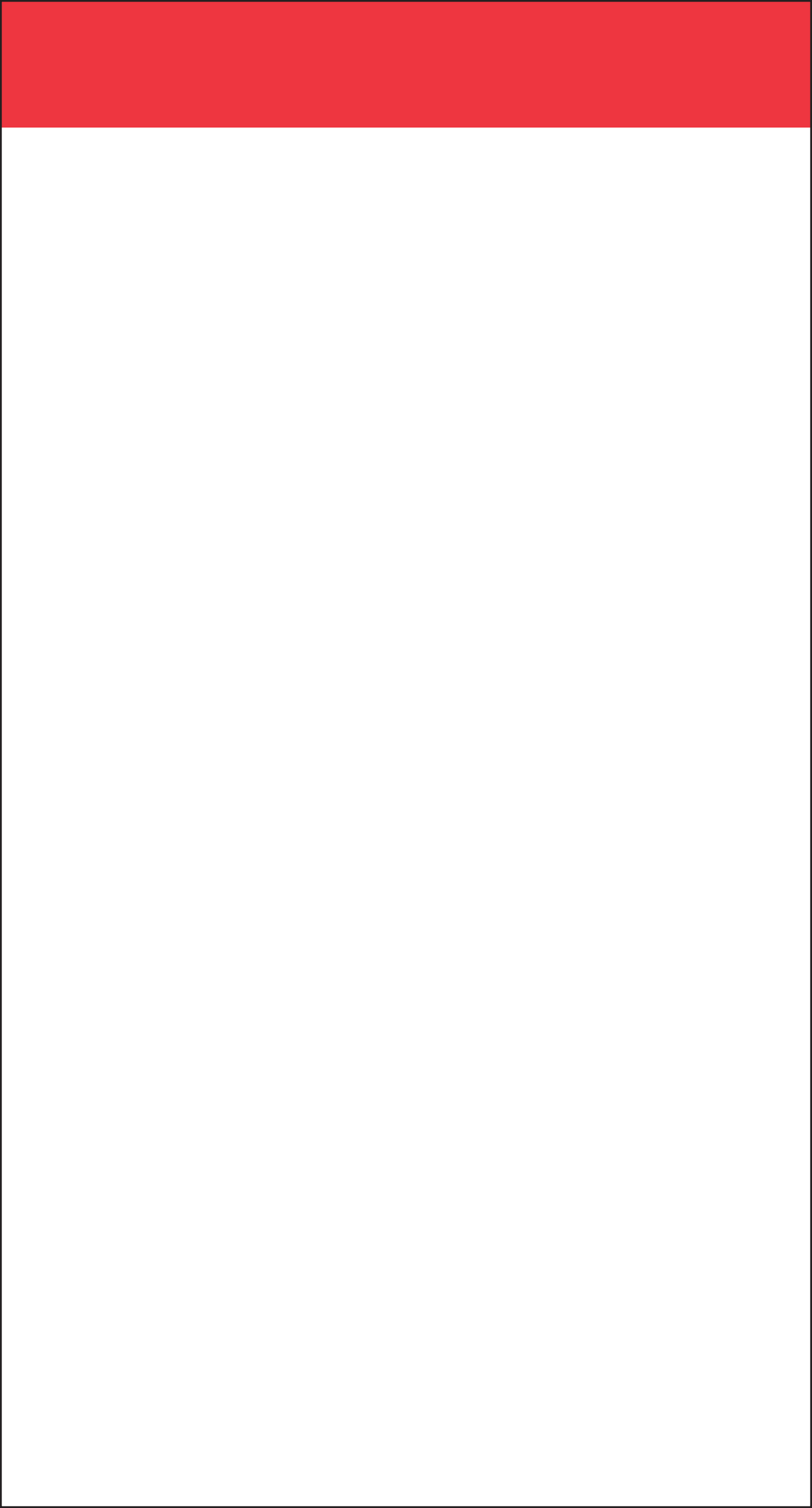
s e c o n d o

s e m e s t r e

LA TECHNOLOGY TRANSFER
PRESENTA:

**PROGRAMMA
DEI CORSI
DI FORMAZIONE
IN
ITALIANO**





**LA TECHNOLOGY TRANSFER
PRESENTA:**

**PROGRAMMA
DEI CORSI
DI FORMAZIONE
IN
ITALIANO**

Il programma è stato realizzato con la collaborazione di:

**Adriano Comai
Renato Comai
Stefano Rizzi
Roberto Tassan
TecnetDati**

CALENDARIO CORSI DI FORMAZIONE 2° SEMESTRE 2010

1. GESTIRE LE RISORSE UMANE DELL'IT

LE RETI E LE DINAMICHE DELLA COMUNICAZIONE NELL'AMBIENTE LAVORATIVO
LA LEADERSHIP ASSERTIVA: COME CAPIRE E MOTIVARE I PROPRI COLLABORATORI
PROBLEM SOLVING
IL MIGLIORAMENTO DELLE RELAZIONI INTERPERSONALI IN AZIENDA

2. ANALISI E MODELLI DI BUSINESS

L'ANALISI ORGANIZZATIVA NELLO SVILUPPO DEI SISTEMI
BUSINESS MODELING

3. ORGANIZZAZIONE E TECNICHE PER LO SVILUPPO

ANALISI DATI E PROGETTAZIONE LOGICA DI DATABASE
GESTIONE DI PROGETTI
STIMA DI TEMPI E COSTI NELLO SVILUPPO SOFTWARE
GESTIONE DEI REQUISITI CON I CASI D'USO
DOCUMENTARE LE ARCHITETTURE SOFTWARE
GOVERNARE L'OUTSOURCING DI PROGETTO
SVILUPPO DI APPLICAZIONI CON UML
GUIDA ALL'UTILIZZO PRATICO DEI FUNCTION POINTS
BEST PRACTICES PER LO SVILUPPO DI APPLICAZIONI J2EE
SERVICE LEVEL AGREEMENT
FUNCTION POINTS COSMIC 3.0
TESTING DI APPLICAZIONI
STATO DELL'ARTE SUL DATA MANAGEMENT
PRECISARE I REQUISITI
ANALISI DELLE FUNZIONI
SYSTEM ENGINEERING CON SYSML
ANALISI OBJECT-ORIENTED CON UML
DISEGNO OBJECT-ORIENTED CON UML
WORKSHOP DI ANALISI E DISEGNO CON UML
WORKSHOP DI DEFINIZIONE DEI REQUISITI E CASI D'USO

4. WEB, SOA, ARCHITETTURE E TECNOLOGIE

WEB 2.0: TECNOLOGIE E PROBLEMATICHE
SOA (SERVICE ORIENTED ARCHITECTURES): BEST PRACTICES
SOA PER L'INTEGRAZIONE DELLE APPLICAZIONI AZIENDALI
SVILUPPO DI APPLICAZIONI WEB CON AJAX
SVILUPPO DI APPLICAZIONI WEB CON XML E WEB SERVICES
IMPLEMENTARE I PROCESSI DI BUSINESS CON SOA E BPM

5. DATA WAREHOUSE

DATA WAREHOUSE: ARCHITETTURA E PRINCIPI
PROGETTAZIONE DEL DATA WAREHOUSE

6. CAPACITY MANAGEMENT

CAPACITY PLANNING
FONDAMENTI DI ITIL® V3
MISURARE E MANTENERE LE PRESTAZIONI DELLE APPLICAZIONI WEB
TEST DI CARICO PER IL COLLAUDO E PER IL TUNING PRESTAZIONALE

7. DB2

DB2-MVS: ARCHITETTURA E FUNZIONI
DB2-MVS: SVILUPPO DI APPLICAZIONI ORIENTATO ALLE PERFORMANCE

8. ORACLE

ORACLE 10G: ARCHITETTURA E FUNZIONI

SEDE	DATA	EURO	Pag.
ROMA	22-24 NOVEMBRE	1000 + IVA	4
ROMA	13-14 DICEMBRE	700 + IVA	6
SOLO IN HOUSE			7
SOLO IN HOUSE			8
ROMA	29-30 NOVEMBRE	700 + IVA	10
ROMA	1-3 DICEMBRE	1000 + IVA	12
ROMA	4-6 OTTOBRE	1000 + IVA	14
ROMA	20-22 OTTOBRE	1000 + IVA	16
ROMA	25-26 NOVEMBRE	700 + IVA	18
ROMA	13-15 DICEMBRE	1000 + IVA	20
ROMA	8-10 NOVEMBRE	1000 + IVA	22
ROMA	11-12 NOVEMBRE	700 + IVA	24
SOLO IN HOUSE			26
SOLO IN HOUSE			28
SOLO IN HOUSE			30
SOLO IN HOUSE			32
SOLO IN HOUSE			34
SOLO IN HOUSE			36
SOLO IN HOUSE			38
SOLO IN HOUSE			40
SOLO IN HOUSE			41
SOLO IN HOUSE			42
SOLO IN HOUSE			44
SOLO IN HOUSE			46
SOLO IN HOUSE			48
SOLO IN HOUSE			49
ROMA	13-15 OTTOBRE	1000 + IVA	50
ROMA	3-5 NOVEMBRE	1000 + IVA	52
ROMA	15-17 DICEMBRE	1000 + IVA	53
SOLO IN HOUSE			54
SOLO IN HOUSE			56
SOLO IN HOUSE			58
ROMA	27-29 OTTOBRE	1000 + IVA	60
ROMA	15-17 NOVEMBRE	1000 + IVA	62
ROMA	18-19 OTTOBRE	700 + IVA	64
ROMA	25-26 OTTOBRE	700 + IVA	66
ROMA	18-19 NOVEMBRE	700 + IVA	68
SOLO IN HOUSE			70
SOLO IN HOUSE			72
SOLO IN HOUSE			73
SOLO IN HOUSE			74

GESTIRE LE RISORSE UMANE

LE RETI E LE DINAMICHE DELLA COMUNICAZIONE NELL'AMBIENTE LAVORATIVO

DESCRIZIONE

Il corso si prefigge un obiettivo ambizioso: dare ai partecipanti – indipendentemente dalla mansione o dal ruolo svolto – alcuni strumenti utili per migliorare la comunicazione infra e inter-gruppo, allo scopo di aumentare la produttività dei singoli e dei gruppi di lavoro. Il corso introduce le moderne teorie sulla comunicazione e fornisce alcune metodologie di estrema efficacia per le più importanti attività aziendali basate sulla comunicazione.

PARTECIPANTI

Dato il carattere generale, il corso non richiede specifiche competenze e non è indirizzato a categorie particolari di pubblico ancorché costituisca la base minimale di conoscenze comportamentali per manager, quadri e capi progetto.

PREREQUISITI

Nessuno.

PROGRAMMA

Introduzione teorica

- la comunicazione secondo la scuola di Palo Alto
- i processi della comunicazione, il contenuto e la relazione

I canali e i codici della comunicazione

- la comunicazione verbale e non verbale (comunicazione analogica e numerica)
- la comunicazione assertiva. La comunicazione a una via e a due vie. Le reti di comunicazione
- fattori che ostacolano o favoriscono la comunicazione
- la “bussola” della comunicazione

La dinamica dei bisogni

- la scala di Maslow
- la matrice di rollo-may sull'autopercezione della conoscenza
- i legami tra bisogni, motivazioni, comportamenti

L'ottimizzazione delle riunioni

- le fasi: preparazione, pianificazione, introduzione, discussione, conclusioni
- pericoli
- difficoltà
- il brainstorming
- i 6 cappelli per pensare

Presentare un prodotto o un progetto

- gli elementi di base e lo scopo di una presentazione. La definizione degli obiettivi da perseguire
- il ruolo del presentatore; la tipologia dei destinatari
- la presentazione formale e informale e le regole della presentazione

- i tre fattori dell'efficacia nella presentazione: sapere, volere, potere
- il linguaggio verbale e non verbale nella presentazione
- le fasi della presentazione: preparazione, apertura, svolgimento, chiusura. Pericoli

Case Study. Role Play. Prove individuali e di gruppo

Bibliografia

Durata: 3 giorni

GESTIRE LE RISORSE UMANE

LA LEADERSHIP ASSERTIVA: COME CAPIRE E MOTIVARE I PROPRI COLLABORATORI

DESCRIZIONE

Il raggiungimento degli obiettivi aziendali e produttivi è un compito che talvolta mette in secondo piano la necessità, comune a ogni manager, di curare e migliorare le relazioni con i propri collaboratori e l'opportunità di acquisire le abilità di base per capire la loro personalità e le loro potenzialità. Il corso si propone, attraverso l'acquisizione di alcune metodologie di facile comprensione, di sviluppare adeguatamente queste abilità.

PARTECIPANTI

Il corso è indirizzato a coloro che gestiscono, a ogni livello, risorse umane.

PREREQUISITI

Nessuno.

PROGRAMMA

- cos'è l'assertività e quali sono i suoi obiettivi. Gli ostacoli alla realizzazione
- le personalità di base (Estroverso/Introverso – Percettivo/Giudicante – Emotivo/Razionale – Assennato/Intuitivo)
- lo stress e come tenerlo sotto controllo
- la valutazione comportamentale di base: componenti cognitive, comportamentali, comunicative
- pregiudizi, miti, pensieri irrazionali
- attriti e conflitti. I segnali di attrito. I segnali di pericolo. Metodi di prevenzione e soluzione dei conflitti
- i nove approcci per evitare i conflitti e le quattro strategie (Esplorare – Pianificare – Attrezzarsi – Implementare)
- i comportamenti di difesa
- un sistema di autoanalisi

Case Study. Role Play. Prove individuali e di gruppo

Bibliografia

Durata: 2 giorni

***P*ROBLEM SOLVING**

DESCRIZIONE

Raggiungere un obiettivo aziendale è compito decisamente complesso perché richiede creatività e prontezza per risolvere le problematiche e le emergenze che via via si prospettano.

Alternando spunti teorici a simulazioni di casi reali, il corso si propone di rendere familiari e concretamente applicabili gli strumenti più frequentemente utilizzati per il Problem Solving.

PARTECIPANTI

Il corso non richiede specifiche competenze e può essere seguito da tutti.

PREREQUISITI

Nessuno.

PROGRAMMA

- Problem Finding, Problem Setting, Problem Solving, Decision Making, Decision Taking.
- cos'è un problema e una "preoccupazione" aziendale
- la "window analysis"
- le fasi del Problem Solving: Propedeutica – Esplorativa – Ideativa – Decisionale – Analisi del rischio
- la definizione di un problema, l'analisi delle cause, la ricerca della soluzione, la messa in atto della soluzione e il monitoraggio post-produzione
- le tecniche più usate: il Brainstorming, i "5 perché", i 6 cappelli per pensare, il diagramma di Ishikawa, l'awareness workshop
- le mappe mentali di Tony Buzan
- la tabella delle conseguenze
- l'analisi PMI (Più, Meno, Interessante)
- l'analisi ABC

Durata: 2 giorni

GESTIRE LE RISORSE UMANE

IL MIGLIORAMENTO DELLE RELAZIONI INTERPERSONALI IN AZIENDA

DESCRIZIONE

Nella moderna cultura aziendale è stato ormai acquisito un concetto basilare: l'efficienza e la produttività sono legate sia a fattori di tipo tecnico (formazione) sia a fattori collegati alla comunicazione e alle relazioni interpersonali tra le risorse aziendali che cooperano ai processi di produzione. È quindi un importante obiettivo strategico fornire a coloro che hanno mansioni di responsabilità nell'area IT alcuni strumenti di conoscenza utili al miglioramento dei rapporti interpersonali. L'analisi transazionale, nata come psicoterapia e poi applicata alle dinamiche aziendali, costituisce un ottimo metodo per raggiungere questi obiettivi.

PARTECIPANTI

Il corso è rivolto a risorse dell'area IT che gestiscono risorse umane (responsabili di struttura, capi progetto).

PREREQUISITI

Nessuno.

PROGRAMMA

La nascita dell'Analisi Transazionale

- il "corpus" teorico dell' A.T.
- scopi
- le prime applicazioni

Le "Basic Position"

- le posizioni esistenziali: gestazione e formazione
- come riconoscere le posizioni esistenziali dei collaboratori
- il loro utilizzo in azienda

Gli stati dell'io

- cosa sono e come riconoscerli

La transazione

- l'analisi della transazione e il disegno del diagramma relazionale
- il miglioramento delle relazioni interpersonali con l'utilizzo del diagramma

Stimoli e carezze psicologiche (*stroke*)

- gli stimoli comportamentali positivi e negativi
- le carezze "condizionate" e le "carezze incondizionate"
- come utilizzare efficacemente le carezze in azienda

Le teorie dell'A.T. utilizzabili in Azienda

- la teoria dei "giochi" ("manovre" messe in atto per ottenere un tornaconto psicologico)
- i "copioni" (comportamenti ripetitivi dettati da uno schema interno)
- la "simbiosi" (comportamenti iperprotettivi e sostitutivi, messi in atto da una figura genitoriale)

L'Azienda "OK"

- l'utilizzo dell'analisi transazionale per migliorare le relazioni in azienda
- le qualità del *Manager* che sa gestire efficacemente le risorse umane
- la leadership situazionale

Test di autodiagnosi transazionale

Test sulla leadership – Bibliografia

Durata: 2 giorni

2

ANALISI E MODELLI DI BUSINESS

L'ANALISI ORGANIZZATIVA NELLO SVILUPPO DEI SISTEMI

DESCRIZIONE

Il corso presenta i fondamenti dell'analisi organizzativa, nell'ottica di ottimizzare gli obiettivi di Business (qualità, tempi, costi) nell'ambito dei progetti.

Viene effettuata una ricognizione sulle tecniche di rilevazione e analisi organizzativa su processi e strutture e sui criteri relativi di revisione e miglioramento.

Il filo conduttore è la consapevolezza della priorità assoluta costituita dai clienti e dai processi di Business finalizzati alla soddisfazione dei clienti stessi.

Si tratta di una priorità non sempre chiara, ma è comunque basilare nell'attività di modellazione.

La metodologia di gestione del corso è interattiva anche nella parte più concettuale.

È fondato cioè su Case Study, su momenti di confronto, discussione e esercitazioni.

PARTECIPANTI

- capi progetto
- analisti di organizzazione
- analisti IT
- esperti di sviluppo delle risorse umane

PREREQUISITI

Nessuno.

PROGRAMMA

Terminologia e aspetti generali dell'analisi organizzativa

- funzioni e processi
- unità e posizioni
- strutture
- organigrammi e funzionigrammi
- mansionari
- teorie organizzative

L'analisi bottom-up (la scomposizione). Parte iniziale: rilevazione

- delle unità organizzative
- delle procedure
- dei processi
- dei tempi (la value stream map)
- dei carichi di lavoro (misurazione, stima, parametrizzazione)

L'analisi bottom-up. Il cambiamento

- miglioramento e innovazione
- il Job Design (funzione, analisi del valore, comunicazione, frazionamento dei processi, metodo, assegnazione del lavoro)
- il ruolo dell'IT nel cambiamento
- il ruolo della modellazione
- gli aspetti comportamentali

L'analisi top-down (approccio integrato)

- il Change Management
- il Business Process Reengineering
- il ruolo del benchmarking e del prato verde
- i criteri del BPR (Process Management, accorciamento, team di processo, disallineamento, sconfinamento)
- l'analisi della struttura (approccio tradizionale e approccio per processi)
- i modelli organizzativi e le strutture snelle

Durata: 2 giorni

2

ANALISI E MODELLI DI BUSINESS

BUSINESS MODELING

DESCRIZIONE

L'analisi e la rappresentazione di processi e sistemi organizzativi è essenziale quando:

- ruoli, responsabilità, processi organizzativi non siano già chiari per l'analista
- il prodotto o il servizio da creare (o da innovare) richieda non solo lo sviluppo e/o l'acquisizione di software, ma anche mutamenti nei processi delle organizzazioni coinvolte
- il confine di automazione (le attività che dovranno essere informatizzate) non sia predefinito
- si vogliano identificare servizi riusabili a livello di Business.

Il corso presenta la tecnica di modellazione standard BPMN (Business Process Modeling Notation) per l'analisi dei processi di Business, con l'uso integrato di UML (Unified Modeling Language) per la rappresentazione degli aspetti organizzativi di tipo strutturale.

I concetti teorici vengono acquisiti attraverso un ampio insieme di esempi concreti, e di esercitazioni pratiche.

PARTECIPANTI

- capi progetto
- analisti di organizzazione
- analisti IT
- esperti di processo coinvolti nelle attività di analisi

PREREQUISITI

Nessuno.

PROGRAMMA

Introduzione al Business Modeling

- cosa modellare
- livelli di modellazione

Individuare i processi

- sistemi e processi organizzativi
- definizione del contesto organizzativo
- tecniche per individuare i processi da analizzare:
 - * Casi d'Uso "Business"
 - * Approccio strutturato

Definire i processi: costrutti base BPMN

- caratteristiche generali
- processi, sottoprocessi, task
- flussi di sequenza e gateway
- eventi di inizio, intermedi, fine
- pool, lane, messaggi
- Data Object

Dettagliare i processi: costrutti avanzati BPMN

- tipologie di eventi
- tipologie di processi
- tipologie di gateway

Rappresentare strutture e ruoli: diagrammi strutturali UML

- panoramica sui diagrammi di UML e sul loro utilizzo per il Business modeling
- rappresentazione dei ruoli e delle risorse del Business
- gerarchie e relazioni non gerarchiche

Individuare servizi riusabili: diagrammi di interazione UML

- diagrammi di interazione
- rappresentazione di servizi riusabili
- gerarchie e relazioni non gerarchiche

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

ANALISI DATI E PROGETTAZIONE LOGICA DI DATABASE

DESCRIZIONE

La progettazione delle basi dati è un'attività essenziale nello sviluppo della maggior parte dei sistemi applicativi.

Il corso espone i due modelli teorici più usati per la definizione di schemi concettuali e logici dei dati, l'Entity Relationship (ER) e il relazionale, trattando in modo dettagliato i meccanismi di astrazione, di generalizzazione e di aggregazione.

In termini di rappresentazione vengono presentate le notazioni più diffuse: ER (nelle sue diverse varianti) ed il profilo di Data Modeling UML.

La progettazione logica dei dati ha basi scientifiche (teoria della normalizzazione), ma anche aspetti meno deterministici, per i quali è necessaria la capacità di analizzare i pro ed i contro delle diverse soluzioni possibili per uno stesso problema. Per questo motivo il corso prevede numerose esercitazioni, e studio di casi reali.

PARTECIPANTI

- esperti di dominio applicativo
- analisti
- progettisti software

PREREQUISITI

Nessuno.

PROGRAMMA

Concetti base

- natura, obiettivi e vincoli della modellazione dei dati
- architettura dei DBMS secondo il modello ANSI/SPARC
- i livelli concettuale, logico e fisico

Il modello Entity Relationship

- entità
- attributo
- tipi di dato base e user-defined
- identificatore
- associazione
- generalizzazione/specializzazione
- aggregazione

Notazioni

- Entity Relationship (Chen e varianti)
- profilo UML di Data Modeling

Il modello relazionale

- relazioni
- domini e Data Type
- vincoli di integrità dei dati
- operatori dell'algebra relazionale

PER LO SVILUPPO

Forme normali e normalizzazione

- dipendenza funzionale, dipendenza multivalente, dipendenza di join
- 1^a, 2^a, 3^a forma normale
- BCNF, 4^a e 5^a forma normale

Integrazione di schemi

- evoluzione di schemi dati nello sviluppo incrementale
- schemi di progetto e schemi aziendali

Progettazione logica di data base

- trasposizione di generalizzazioni/specializzazioni nel contesto relazionale
- denormalizzazioni e altre forme di ottimizzazione
- database Refactoring
- Data Warehouse
- reverse Engineering dei dati

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

GESTIONE DI PROGETTI

DESCRIZIONE

Il corso, frutto di un'esperienza maturata nella conduzione di progetti di ampio respiro, affronta il tema del Project Management considerando i principali aspetti implicati: validità di una strategia di "sviluppo per progetti"; individuazione dei momenti maggiormente critici di un progetto; definizione degli obiettivi di progetto; creazione di un piano realistico per il raggiungimento di tali obiettivi; assegnazione dei ruoli e delle responsabilità: attuazione di un'efficace azione di controllo e conduzione del progetto; gestione della qualità; gestione delle risorse umane (Peopleware).

Particolare peso viene dato all'esame degli aspetti riguardanti pianificazione, stima, controllo e convalida con una panoramica sulle nuove metodologie emergenti per la gestione dei progetti complessi.

Il Corso è conforme alle linee guida del PMBOK®.

Il Corso è certificato e dà diritto a 21 crediti formativi PMI (noti come PDU).

PARTECIPANTI

- responsabili dell'area sviluppo
- capi progetto
- analisti
- responsabili di funzioni aziendali

PREREQUISITI

Conoscenza delle problematiche di sviluppo applicativo.

PROGRAMMA

- la gestione degli obiettivi e dei risultati, bilanciamento tra costi, tempi, funzionalità e qualità
- il processo gestionale: avvio, pianificazione, esecuzione, monitoraggio, controllo e chiusura del progetto
- strutturazione del piano di progetto - obiettivi generali, traguardi intermedi, ruoli e responsabilità
- tecniche per l'organizzazione, la pianificazione e il controllo delle attività
- controllo di un progetto - cosa significa controllare un progetto, supervisione e controllo dell'avanzamento, azioni correttive
- costi di progetto - costi diretti e indiretti, conto economico previsionale, i costi come base per valutare l'avanzamento lavori
- rischi di progetto - margine di rischio e controllo del progetto, fattori di rischio, modelli per la valutazione del margine di rischio
- gestione di un progetto complesso - scomposizione in sottosistemi, coordinamento dell'avanzamento di sottoprogetti paralleli in fasi realizzative disomogenee; nuovi modelli di gestione (XPM, Metodi Agili, ecc.)

PER LO SVILUPPO

- gestione dei requisiti - definizione, ordinamento in funzione delle priorità, verifica, governo delle modifiche
- stime - problematiche e approcci metodologici
- creazione, sviluppo e mantenimento di un team efficace - gestione dei rapporti interpersonali, fattori sociologici e loro impatto sulla produttività, leadership formale e informale, ruolo del capo progetto
- la gestione della qualità all'interno del progetto

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

STIMA DI TEMPI E COSTI NELLO SVILUPPO SOFTWARE

DESCRIZIONE

Stimare in modo efficace i tempi e i costi di un'attività di sviluppo software, che si tratti di un progetto articolato o di un semplice intervento di manutenzione, è una pratica con cui si cimentano costantemente i Project Manager ma anche, in generale, tutti gli addetti alla realizzazione del software.

La stima basata sull'esperienza personale, che ad oggi rimane ancora la più consueta, pur potendo portare a buoni risultati presta, sempre e comunque, il fianco a critiche.

Nasce quindi l'esigenza di disporre di tecniche di stima più oggettive e formali, basate su criteri più scientifici, raffrontabili in ambienti eterogenei.

Il corso, dopo aver ripreso le problematiche più comuni nella realizzazione delle stime, illustra alcune tecniche di stima largamente utilizzate e le inquadra nell'ambito di un processo che ne guida l'utilizzo.

Nel corso sono previsti ampi ed esaurienti riferimenti alla teoria dei Function Points, oggi strumento cardine per il dimensionamento del software, e al loro utilizzo nell'ambito delle stime.

Al termine del corso, i partecipanti:

- avranno appreso i principi base dei Function Points e le problematiche legate alla loro applicazione (per poter iniziare ad applicare la tecnica a casi pratici)
- avranno chiara idea dei fattori di cui tenere conto nelle stime dei tempi e dei costi dello sviluppo software
- conosceranno le più diffuse tecniche di stima (applicabili in funzione delle dimensioni del progetto) e le modalità con cui calare le stime in un processo di sviluppo del software
- avranno conoscenza dei principi fondamentali di un processo di gestione delle stime

PARTECIPANTI

- capi progetto, team leaders
- analisti funzionali
- specialisti nel campo delle metriche del software

PREREQUISITI

Conoscenza delle problematiche legate al dimensionamento del software e alla stima di tempi e costi dello sviluppo del software.

PROGRAMMA

Le problematiche legate alle stime

Il problema del dimensionamento del software

- le metriche del software
- i Function Points: elementi teorici di base per conteggio e stima

Stima dell'effort

- la stima per analogia
- la stima bottom-up
- la stima top-down basata sui workproduct
- la stima basata sui Function Points
- la stima basata su modelli parametrici

PER LO SVILUPPO

Il processo di stima calato nel processo di sviluppo

- quando e come stimare (i momenti di stima e le tecniche più idonee)
- la gestione delle stime in ambito di progetto e in ambito aziendale

Gli strumenti per la stima

- Templates, Repository, ecc.

Caso Studio 1

- stima dell'effort di un piccolo intervento di manutenzione evolutiva

Caso Studio 2

- stima di tempi e costi di un progetto in "fase alta"

Caso Studio 3

- stima finale di un progetto

Durata: 2 giorni

3

ORGANIZZAZIONE E TECNICHE

GESTIONE DEI REQUISITI CON I CASI D'USO

DESCRIZIONE

Il corso fornisce il bagaglio di conoscenze necessario a definire, in accordo con i committenti, i requisiti di un sistema software, ed a gestire il cambiamento di tali requisiti durante le diverse fasi di un progetto.

Vengono affrontati in particolare: i ruoli coinvolti nella gestione dei requisiti; il “ciclo di lavorazione” dei requisiti, dalla scoperta all’analisi, alla negoziazione, all’accordo, alle verifiche in corso d’opera; il conflitto di requisiti e quello tra requisiti, costi e tempi; la definizione dei criteri di accettazione per la verifica di corrispondenza tra requisiti e sistema.

Per l’individuazione e la specifica dei requisiti vengono presentate tecniche complementari, con una particolare attenzione alla definizione dei Casi d’Uso ed al loro utilizzo concreto, sia per la modellazione dei processi del Business che per l’approfondimento e la specifica degli scenari di operatività degli utilizzatori nei confronti del sistema informatico.

Il corso propone ai partecipanti una serie di esercitazioni sulle tecniche affrontate in sede di esposizione teorica, con attenzione particolare alla definizione dei modelli di Casi d’Uso ed alla loro descrizione testuale.

PARTECIPANTI

- management sistemi informativi
- capi progetto
- analisti
- esperti di dominio coinvolti nella definizione dei requisiti
- sviluppatori

PREREQUISITI

È opportuna una conoscenza generale delle problematiche relative allo sviluppo e all’evoluzione dei sistemi software.

PROGRAMMA

I requisiti nello sviluppo dei sistemi

- ruoli coinvolti nella gestione dei requisiti, e le loro relazioni
- ciclo di lavorazione dei requisiti e dei sistemi software nei diversi approcci metodologici (a cascata, incrementale, Unified Process, processi “agili”)
- gestione dei requisiti in ISO 9000 e nel Capability Maturity Model (CMMI)
- gestione della tracciabilità dei requisiti, cioè dei legami tra requisiti, componenti del sistema, test
- gestione del cambiamento dei requisiti in corso d’opera

Tecniche per la scoperta dei requisiti

- identificazione degli interlocutori da coinvolgere
- interviste e workshop
- recupero e analisi di documentazione da sistemi esistenti
- descrizione degli scenari di utilizzo
- prototipi degli scenari di interazione

PER LO SVILUPPO

Analizzare e specificare i requisiti

- classificazione dei requisiti per tipologia
- importanza e priorità relativa dei singoli requisiti
- eliminazione delle ambiguità dai requisiti, e modalità di verifica
- forme di documentazione e template
- strumenti per la definizione e la gestione dei requisiti

Casi d'Uso

- teoria dei Casi d'Uso, e ambito della loro applicazione
- individuazione degli attori e definizione del contesto del sistema
- relazioni tra attori e Casi d'Uso
- modello dei Casi d'Uso "Business"
- modello dei Casi d'Uso per il sistema software
- differenze tra Casi d'Uso e altre tecniche di definizione dei requisiti

Specificare i Casi d'Uso

- descrizione testuale degli scenari (scenario base, scenari alternativi, varianti)
- stili utilizzabili per la descrizione, e linee guida di scrittura
- esercitazioni pratiche di scrittura dei Casi d'Uso, con verifica in aula
- ruolo ed utilizzo dei diagrammi UML dei Casi d'Uso
- relazioni tra Casi d'Uso in UML (specializzazione, extend, include), e ambito della loro applicazione

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

DOCUMENTARE LE ARCHITETTURE SOFTWARE

DESCRIZIONE

La documentazione dell'architettura di un sistema software agevola la comunicazione tra i ruoli coinvolti in un progetto, particolarmente quando le persone si trovano in luoghi diversi, appartengono a società o unità organizzative diverse, o devono intervenire sul sistema in momenti successivi.

La documentazione architeturale deve specificare l'organizzazione interna del sistema in "parti" distinte (sottosistemi, componenti) e le modalità con cui tali "parti" interagiscono tra loro per fornire le funzionalità complessive. Deve anche indicare le tecnologie usate per l'implementazione e l'esecuzione.

Per la documentazione delle architetture software è da alcuni anni disponibile uno standard - UML, Unified Modeling Language. Il corso si focalizza esclusivamente sugli elementi di UML utili per documentazioni delle architetture di sistema e software (per una trattazione completa di UML, fare riferimento al corso "*Sviluppo di applicazioni con UML*"). Per documentare l'integrazione del software nell'ambito di un sistema più complesso, composto sia da hardware che da software, viene usato un profilo specifico di UML - SysML.

Durante il corso viene proposta ai partecipanti una serie di esercitazioni di complessità graduale.

PARTECIPANTI

- architetti di sistema
- architetti software
- analisti
- progettisti e sviluppatori software

PREREQUISITI

Nessuno.

PROGRAMMA

Cos'è l'architettura, e quali forze la determinano

- architettura del sistema, architettura del software
- requisiti (funzionali e non-funzionali) e architettura
- pattern architeturali

Architettura e organizzazione del progetto

- l'organizzazione si adatta all'architettura? O l'architettura all'organizzazione?
- ruolo dell'architetto software
- altri ruoli coinvolti nelle scelte architeturali
- l'architettura nei processi di sviluppo iterativi ed agili

Standard per la rappresentazione architeturale

- viste architeturali
- panoramica su UML (Unified Modeling Language)
- panoramica su SysML (System Modeling Language)
- altre notazioni

PER LO SVILUPPO

Rappresentazione del contesto del sistema

Scomposizione in sottosistemi e componenti

Aspetti strutturali: caratteristiche e livello di dettaglio

Aspetti dinamici e comportamentali

Rappresentazione dei servizi e delle orchestrazioni

Il deployment

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

GOVERNARE L'OUTSOURCING DI PROGETTO

DESCRIZIONE

I motivi per cui si decide di collaborare con terze parti per realizzare progetti di sviluppo del software possono essere diversi. Tattici, oppure strategici. Tecnici, o economici. Anche i possibili livelli di esternalizzazione sono diversi: possono essere delegate l'intera gestione del progetto dal punto di vista tecnico, oppure una o più attività progettuali, come l'analisi, la progettazione, lo sviluppo, il test.

In ogni caso, chi si affida in tutto o in parte per lo sviluppo software a fornitori esterni non può permettersi di sottovalutare e trascurare la gestione del rapporto di Outsourcing: il committente deve comunque raggiungere gli obiettivi del progetto, soddisfare gli stakeholder, tenere conto dei vincoli economici e temporali, evitare i rischi di insuccesso.

Il corso affianca agli aspetti teorici, tratti da standard internazionali, esercitazioni derivate da casi reali.

PARTECIPANTI

- committenti (informatici e non) che collaborano con fornitori esterni nello sviluppo di sistemi
- capi progetto
- analisti
- architetti software

PREREQUISITI

Nessuno.

PROGRAMMA

Introduzione

- statistiche internazionali sull'andamento dei progetti di sviluppo, e sulle principali cause di successo e di insuccesso
- modelli di processo (a cascata, iterativo, agile) e relative ripercussioni sulle relazioni tra le parti
- gestione dei rapporti con i fornitori nel Capability Maturity Model Integration (CMMI) e nel Project Management Book of Knowledge (PMBOK)

Ruoli e responsabilità

- committente
- utenti e altri stakeholder
- project manager
- analista
- architetto software
- fornitore

Gestione delle relazioni tra le parti

- definizione e negoziazione di risultati attesi, tempi, costi e qualità
- tipologie di contratto
- esternalizzazione di fasi e/o di discipline
- gestione delle variazioni in corso d'opera
- gestione dei conflitti

PER LO SVILUPPO

Pratiche essenziali

- determinare il tipo di acquisizione opportuno
- selezionare i fornitori
- stabilire e mantenere gli accordi con i fornitori
- rendere operativi gli accordi
- controllare il modo di lavorare dei fornitori
- valutare i prodotti realizzati dai fornitori
- prendere in carico i prodotti acquisiti

Gestione dei rischi di progetto

- tipologie di rischio
- definizione dei rischi e delle contromisure
- influenza dei rischi sulla pianificazione di progetto

Fattori critici di successo

- stime
- pianificazione e controllo avanzamento
- requisiti
- test

Durata: 2 giorni

3

ORGANIZZAZIONE E TECNICHE

SVILUPPO DI APPLICAZIONI CON UML

DESCRIZIONE

UML (Unified Modeling Language) è il linguaggio di rappresentazione standard dei sistemi software. Con le versioni 2.x, ha raggiunto un livello di maturità sufficiente per essere usato in modo efficace e produttivo nell'analisi e nella progettazione, oltre che nella documentazione di sistemi preesistenti. Il corso offre una panoramica completa sui concetti, sugli elementi e sui diagrammi di UML, con particolare attenzione alle modalità concrete del loro uso nelle diverse situazioni progettuali, fornendo linee guida minimali e pragmatiche che distinguono tra ciò che è indispensabile fare in ogni caso e ciò che può invece risultare utile solo in ambiti particolari. Nel corso viene effettuata una esercitazione/Caso Studio completa, con utilizzo di tutti i diagrammi UML.

PARTECIPANTI

- management sistemi informativi
- capi progetto
- analisti
- architetti software
- sviluppatori

PREREQUISITI

È opportuna una conoscenza generale delle problematiche relative allo sviluppo e all'evoluzione dei sistemi software.

PROGRAMMA

Il processo di sviluppo (e manutenzione)

- i ruoli coinvolti nello sviluppo, le loro relazioni, le loro modalità di comunicazione
- sviluppo a cascata, Unified Process, processi "agili" (Extreme Programming, Scrum, ecc.)

Introduzione a UML

- principi Object Oriented (classe e oggetto, incapsulamento, ereditarietà, polimorfismo)
- caratteristiche generali di UML
- panoramica sui diagrammi UML
- versioni di UML (1.x e 2.x)
- SysML per la rappresentazione dei sistemi non software
- Model Driven Architecture (MDA)
- UML e processi di sviluppo (a cascata, Unified Process, Extreme Programming, Scrum)
- costrutti generali (package, stereotipi, meccanismi di estensione, profili)
- cenni sugli strumenti di modellazione

I Casi d'Uso

- attori, Casi d'Uso, sistema di riferimento (subject)
- diagrammi dei Casi d'Uso
- cenni sulla specifica dei Casi d'Uso

PER LO SVILUPPO

Modellare gli aspetti strutturali del sistema

- Diagramma dei package
 - * Package per organizzare i modelli e la documentazione
 - * Package per organizzare il software
- Diagramma delle classi
 - * strategie per l'individuazione e la definizione delle classi
 - * forme di rappresentazione delle classi e degli oggetti
 - * attributi
 - * operazioni
 - * relazioni tra classi
 - * rappresentazione delle strutture dati
- Diagramma dei componenti
 - * componenti
 - * interfacce
 - * dipendenze
 - * relazioni tra componenti e classi
- Diagramma delle strutture composite
 - * rappresentazione interna di un componente
 - * rappresentazione interna di una classe
- Diagramma di deployment
 - * nodi
 - * connessioni
 - * ambienti di esecuzione
 - * artifact e componenti

Modellare gli aspetti dinamici del sistema

- Diagrammi di interazione
 - * messaggi e responsabilità
 - * diagramma di sequenza
 - * diagramma di comunicazione
 - * diagramma di overview dell'interazione
 - * diagramma di timing
- Diagramma di stato
 - * stati e transizioni
 - * stati composti
 - * transizioni composite e pseudostati
- Diagrammi di attività
 - * attività e azioni
 - * flussi di controllo
 - * parallelismi e sincronizzazioni
 - * flussi di input e output

Durata: 4 giorni

3

ORGANIZZAZIONE E TECNICHE

GUIDA ALL'UTILIZZO PRATICO DEI FUNCTION POINTS

DESCRIZIONE

Il calcolo dei Function Points è attualmente la tecnica più efficace per dimensionare i progetti software in modo comprensibile da parte dell'utente, in quanto permette di correlare lo sforzo necessario per produrre il software ai prodotti effettivamente rilasciati. La tecnica ha ormai raggiunto un alto grado di standardizzazione. È sempre più utilizzata, inoltre, come metrica oggettiva per il dimensionamento del software, nella contrattazione tra aziende e fornitori. L'applicazione dei Function Points, sebbene guidata da precise regole codificate (standard IFPUG 4.2), può tuttavia lasciare spazio a dubbi e interpretazioni soggettive.

Il corso, articolato in forma di workshop, intende fornire un aiuto pratico per il conteggio dei Function Points, attraverso la presentazione e la soluzione congiunta delle situazioni applicative più ricorrenti nell'ambito dello sviluppo dei sistemi.

Vengono inoltre affrontati aspetti inerenti l'utilizzo dei Function Points nel governo dei progetti e nella stesura di contratti per la realizzazione del software.

PARTECIPANTI

- capi progetto
- analisti
- specialisti nel campo delle metriche

PREREQUISITI

Conoscenza delle problematiche di dimensionamento del software e dello sviluppo applicativo.

PROGRAMMA

Il metodo IFPUG 4.2

- criteri di utilizzo, ambiti applicativi, elementi e regole di conteggio, cambiamenti rispetto alla versione 4.1

Problematiche applicative

- stima dei Function Points (FP) nelle fasi "alte" di progetto, rilevazione del portafoglio applicativo esistente, dimensionamento dei progetti di manutenzione evolutiva in base all'effettivo sforzo richiesto, riutilizzo di componenti software già esistenti

Applicazione dei Function Points ai nuovi paradigmi di sviluppo

- Object Oriented, applicazioni Web, Data Warehouse

Cenni all'utilizzo dei Function Points nei contratti

Indicatori per il governo di progetto tramite i FP

Impiego dei Function Points per la determinazione del Service Level Agreement

PER LO SVILUPPO

Caso Studio 1

- calcolo dei FP per un progetto di nuovo sviluppo

Caso Studio 2

- calcolo dei FP per un progetto di manutenzione evolutiva

Caso Studio 3

- calcolo dei FP nell'ambito di un'applicazione sviluppata ad oggetti

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

BEST PRACTICES PER LO SVILUPPO DI APPLICAZIONI J2EE

DESCRIZIONE

La piattaforma J2EE è oggi uno standard “de facto” per la realizzazione di applicazioni Enterprise.

I Pattern di progettazione sono un “nocciolo duro” di Best Practices e soluzioni standard a problemi ricorrenti cui attingere abbondantemente nella fase di disegno del software. Rappresentano un modo semplice ed efficace di riutilizzare le migliori tecniche ed esperienze nella realizzazione di applicazioni robuste, modulari, scalabili.

Il corso, partendo da una introduzione sugli “stili architetturali” standard nelle architetture J2EE e dalle nuove tipologie di componenti (EJB 3.0, POJO, JSF), con relativi pro e contro, fornisce la visione dei concetti base relativi ai Pattern per passare poi a illustrare nel dettaglio come sfruttarli per intervenire su diversi problemi e situazioni architetturali non ideali introducendo un miglioramento oggettivo.

La descrizione avviene sia da un punto di vista teorico sia con esempi tratti da situazioni reali di progetti realizzati, che aiutano a comprendere pienamente l'utilizzo pratico.

Il contesto di riferimento in cui viene calato il corso prevede l'utilizzo della notazione UML 2.0 e un processo di realizzazione derivato direttamente da RUP – Rational Unified Process.

PARTECIPANTI

- architetti di sistema
- analisti programmatori
- specialisti di metodi e standard

PREREQUISITI

Corso di “*Analisi Object-Oriented con UML*” o conoscenze equivalenti.

È molto utile l'esperienza di progettazione di applicazioni a oggetti e programmazione con linguaggi OO.

PROGRAMMA

J2EE

- tiers, containers, elementi base dell' architettura, ecc.
- stili architetturali standard

Pattern: concetti base

- cos'è un “Pattern”, Pattern vs Framework
- come si descrive e si classifica un Pattern: l'approccio secondo Eric Gamma

Ottimizzare la progettazione con i Pattern

- “Bad Practices” comuni e relativi inconvenienti introdotti

• Presentation Tier Patterns

- Composite View
- View Helper
- Service to Worker
- Dispatcher View
- Intercepting Filter

PER LO SVILUPPO

- Integration Tier Patterns
 - Data Access Object
 - Service Activator
 - Domain Store
 - Web Service Broker
- Business Tier Patterns
 - Business Delegate
 - Service Locator
 - Session Facade
 - Application Service
 - Business Object
 - Transfer Object Assembler
- Architectural Patterns
 - MVC: il Pattern architetturale di riferimento
 - Layer: altro Pattern architetturale importante
- Design Patterns
 - Template (o Algorithm)
 - Visitor
 - Iterator
 - Proxy
 - Strategy
 - Adapter
 - Decorator
 - Singleton
 - State
 - Abstract Factory
 - Publish-Subscribe
 - GRASP
- Altri Patterns: cenni
- Usare i Patterns nel progetto: scegliere, definire e modellare i propri Patterns

Durata: 2 giorni

3

ORGANIZZAZIONE E TECNICHE

SERVICE LEVEL AGREEMENT

DESCRIZIONE

La forte evoluzione delle tecnologie informatiche e il volume sempre crescente del patrimonio software da gestire, costituiscono due delle principali criticità dei settori IT aziendali. L'impegno necessario per la manutenzione del software applicativo esistente, infatti, condiziona e limita pesantemente lo sviluppo delle nuove "soluzioni", richieste dai reparti operativi aziendali e in linea con i trend evolutivi IT.

In questo contesto molte aziende si orientano verso la concessione in Outsourcing di parte o tutto il servizio di Application Management, con il duplice obiettivo di definire in modo chiaro i costi relativi e liberare risorse per l'evoluzione dei Sistemi Informatici in modo coerente con il proprio core Business. La definizione dei livelli di servizio tra tutti i soggetti coinvolti nell'erogazione del servizio stesso, Cliente-Fornitore-Subfornitori, è una condizione fondamentale per il raggiungimento degli obiettivi di qualità ed efficienza richiesti. Il corso costituisce una guida alla definizione di servizi di Application Management e all'impostazione/redazione di un documento di Service Level Agreement. Il corso introduce all'utilizzo dei Function Points quale strumento per prima esprimere e poi monitorare in modo oggettivo i livelli di servizio

PARTECIPANTI

- account manager
- service manager
- sales manager
- application manager

PREREQUISITI

Conoscenza delle problematiche di manutenzione e assistenza di applicazioni software di grandi dimensioni.

PROGRAMMA

Il servizio di Application Management e le sue componenti

- analisi delle finalità e dei contenuti del servizio, mirata a facilitare l'individuazione del "perimetro" e la suddivisione nelle singole componenti
- assistenza
- manutenzione: la suddivisione tecnica e quella contrattuale
- definizione delle priorità di intervento

L'utilizzo dei Key Performance Indicators per la misurazione dei livelli di servizio

- utilizzo delle metriche di misurazione del software (Function Points) nei servizi di Application Management
- i KPI più utilizzati

Il Sistema di Misurazione delle Performances e il Reporting

- la definizione di un "accordo" sui Livelli di Servizio implica un corrispondente adeguamento operativo
- il Sistema di Misurazione e la conseguente organizzazione operativa
- la gestione del Reporting

PER LO SVILUPPO

I contenuti di un documento di Service Level Agreement

- analisi delle singole componenti di uno SLA

L'utilizzo del documento di SLA come strumento operativo per l'erogazione di un servizio qualitativamente valido

Il periodo di avviamento del servizio

- gli obiettivi e gli aspetti organizzativi

Case Study

Rilettura di uno SLA, evidenziando gli aspetti salienti in relazione a quanto trattato nell'ambito del corso

Durata: 2 giorni

3

ORGANIZZAZIONE E TECNICHE

FUNCTION POINT COSMIC 3.0

DESCRIZIONE

Disporre di una metrica funzionale per il dimensionamento del software è ormai un'esigenza primaria in campo informatico.

Nonostante la diffusione storica della FPA (Function Point Analysis) secondo le regole IFPUG, emergono alcuni limiti nella sua applicazione in contesti e paradigmi di sviluppo moderni.

Per questo dal 1998 si propone un nuovo metodo di misurazione funzionale chiamato 'COSMIC', dal nome dell'organismo internazionale che lo ha realizzato: the Common Software Measurement International Consortium.

Il metodo COSMIC si va costantemente diffondendo; l'ISO/IEC lo ha riconosciuto, al pari del metodo IFPUG Unadjusted, come metrica funzionale rispondente ai propri standard (rif. ISO/IEC 19761:2003).

È chiaro che i Function Points COSMIC, pur appartenendo alla famiglia delle metriche funzionali, non sono identici ai Function Points IFPUG (pur essendo possibile una conversione dagli uni agli altri): i Function Points COSMIC permettono la misurazione di applicazioni "tradizionali", con risultati analoghi a quelli IFPUG ma, in virtù di alcune caratteristiche particolarmente interessanti, si applicano con altrettanto successo anche a componenti software "particolari", per esempio di tipo GIS (cartografico), SOA (Service Oriented Architecture), Web Services, Middleware evoluto ecc, e a processi software complessi, come nel caso dei batch.

Il corso fornisce gli elementi di base per l'apprendimento del metodo COSMIC Function Points 3.0 (2007) e per la sua applicazione pratica e le regole complete per il conteggio e la stima di applicazioni, progetti di sviluppo e progetti di manutenzione evolutiva del software.

Il corso prevede inoltre l'analisi e la discussione di un Caso Studio mutuato da esperienze pratiche, la discussione di casi proposti dai partecipanti, un confronto tra le metriche IFPUG e COSMIC e un'analisi delle modalità di conversione tra i due metodi.

PARTECIPANTI

- capi progetto
- analisti
- specialisti nel campo delle metriche

PREREQUISITI

Nessuno.

PROGRAMMA

- introduzione
- standard e riferimenti normativi
- regole di misurazione Function Point COSMIC 3.0
 - modello del software generale e del contesto
 - fase di strategia
 - * scopo della misurazione
 - * ambito della misurazione (strati, componenti)
 - * utenti funzionali
 - * livelli di granularità (standard e superiori)

PER LO SVILUPPO

- fase di mappatura
 - * processi funzionali
 - * oggetti di interesse e gruppi di dati
- fase di misurazione
 - * movimenti di dati (Entry, Exit, Read, Write)
 - * funzione di misurazione
 - * aggregazione dei risultati e formule finali
 - * misurazione delle modifiche (manutenzioni evolutive)
- esempi (applicazioni tradizionali, Middleware, Web Services ecc.) e esercizi
- confronto e convertibilità misure COSMIC-IFPUG e viceversa
- temi avanzati
 - livelli di scomposizione (ambito della misurazione)
 - benchmarking
 - stima anticipata
 - modalità di reporting
- Casi Studio
 - proposti dal docente (porzioni)
 - suggeriti dai partecipanti (porzioni)

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

TESTING DI APPLICAZIONI

DESCRIZIONE

Il Testing ha come obiettivo la verifica del funzionamento di un prodotto software rispetto a criteri predefiniti, per esempio il comportamento presunto del sistema. È un elemento chiave per garantire un elevato standard di qualità del software e richiede, per essere efficace, una corretta impostazione organizzativa e metodologica.

Il corso illustra in modo esauriente principi, strategie e tecniche su cui si fonda il Testing: approccio di tipo sia funzionale ("black box" Testing), sia strutturale ("white box" Testing); strategia basata su più livelli e tipologie di test; esistenza di un ciclo di vita del test con conseguente definizione di attività, responsabilità e prodotti. Partendo da questa base di conoscenza, viene affrontato il tema della "customizzazione" del Testing in funzione delle caratteristiche di progetto, per esempio le specificità delle applicazioni Web e dei nuovi paradigmi di sviluppo, per esempio approccio al test basato sui Casi d'Uso, uso dei diagrammi UML come ausilio per il test di applicazioni Object-Oriented, Testing di applicazioni sviluppate per componenti, organizzazione del test nell'ambito di un processo di sviluppo e manutenzione iterativo e incrementale.

Il corso fornisce inoltre gli elementi per l'inquadramento del Testing nel più ampio contesto della Quality Assurance, alla luce dello standard ISO9000: 2000.

PARTECIPANTI

- responsabili progetti software
- personale dell'assicurazione e del controllo di qualità
- analisti e progettisti di applicazioni software

PREREQUISITI

Conoscenza delle problematiche di sviluppo applicativo.

PROGRAMMA

- introduzione - obiettivi del Testing. Il Testing nell'ambito del sistema qualità, fattori di qualità del software, Le norme ISO/IEC 9126
- livelli di Testing (Unit, Integration e System Testing, Collaudo e Accettazione, Load, Stress, Capacity Testing ecc.)
- tipologie di Testing: funzionali, di affidabilità, di usabilità ecc.
- finalità di Testing: verifica e validazione, "bug fixing", non regressione
- tecniche di Testing "white box" e "black box"
- il processo di Testing - Pianificazione (definizione della Strategia di test in funzione dei requisiti di qualità e degli obiettivi concordati con l'utente, livello di copertura funzionale e topologico, piano dei test), Progettazione (checklist, casi di test, classi di equivalenza, specifiche di test, unità di lancio), Esecuzione (aspetti organizzativi, scheda esito test, rapporto anomalie), Gestione delle anomalie

PER LO SVILUPPO

- adattare il Testing in funzione delle caratteristiche del progetto - Testing di applicazioni Web: peculiarità, principi, tecniche, standard in uso (linee guida W3C ecc.), Testing di applicazioni realizzate secondo i nuovi paradigmi (Casi d'Uso, Object-Oriented e UML, processo iterativo e incrementale ecc.)
- strumenti di supporto al Testing: tipologie, criteri per la selezione e l'utilizzo
- le metriche di prodotto e di processo - indice di difetto del prodotto, valutazione dell'efficacia dei test, determinazione dell'impegno mediante Function Point
- Casi Studio ed esercizi

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

STATO DELL'ARTE SUL DATA MANAGEMENT

DESCRIZIONE

Il corso si prefigge di dare un'ampia panoramica delle nuove problematiche e delle soluzioni che i progettisti di basi di dati odierne sono chiamati a fornire.

Il modello di database definito nello standard SQL2 ha avuto nel tempo estensioni che raccolgono sia le idee del modello SQL3 (Object Relational), sebbene mai ratificato dai produttori, sia dalle nuove richieste applicative: dai Data Warehouse, all'XML, ai nuovi paradigmi multimediali.

La proliferazione di basi di dati in azienda su piattaforme differenti (fra poco persino sui dispositivi mobili) ha fatto accettare il concetto che non esista più un solo DBMS in azienda ed un DBMS integrato ma un insieme di database tematici più o meno interconnessi fra loro, con versioni dello stesso dato che convivono fra loro. Questo fenomeno deve essere gestito da data analyst e database administrator e deve essere conosciuto da chi si occupa di sviluppo e di auditing. Cresce quindi la sensibilità al problema dei Metadati e al ciclo di vita dei dati come unico modo per unificare la conoscenza sui dati e sul loro utilizzo, che va al di là dei confini aziendali (dalle applicazioni Web sino ai servizi SOA pubblicati).

I dati non sono più solo strutturati ed è importante dare un quadro di riferimento di standardizzazione e di sicurezza complessiva anche per i dati non strutturati: multimedia e documenti che contengono dati e informazioni una volta memorizzati nei database aziendali e che oggi rischiano di circolare anche all'esterno.

E proprio perché si apre il sistema informativo aziendale all'esterno, un tema di crescente interesse è quello della sicurezza che verrà affrontato secondo una visione delle problematiche e delle tendenze tecnologiche che partono dai database Firewall e dalla crittografia fino al tema del Digital Rights Management (DRM) di forte impatto organizzativo.

Il corso ha quindi l'obiettivo di fornire un aggiornamento puntuale, contestualizzato e sistematizzato sullo stato dell'arte del Data Management esteso.

PARTECIPANTI

- capi progetto
- data administrator
- database administrator
- analisti
- progettisti

PREREQUISITI

Conoscenza delle problematiche legate alla gestione dei dati.

PROGRAMMA

- le idee di SQL3 e di ODMG 93 ed il loro impatto negli RDBMS attuali e nel processing
- dati e Metadati
- ciclo di vita di un dato
 - organizzazione e gestione
 - tecnologie

PER LO SVILUPPO

- XML e DBMS, convivenza o sovrapposizione?
- XML e DBMS, problematiche per chi sviluppa (performance, sicurezza anche in ambito SOA)
- Data Warehouse e DBMS, situazione stabile
- panoramica dell'offerta dei fornitori: IBM, Microsoft, Oracle e MySQL alla luce di quanto visto precedentemente
- CMIS, un modello standard basato su SQL e SOA per la gestione documentale
- sicurezza nei DBMS
 - Access Control e granularità delle utenze: basta il comando GRANT?
 - Crittografia, interna ed esterna, pro e contro
 - DB Firewall, black e white list
 - DB Firewall, Behaviour Analysis
 - Auditing e Logging
 - Basi di dati e normative Privacy, PCI e SOX (EuroSOX)
- Digital Rights Management
- l'impatto della virtualizzazione
- il DBA, una professione in forte evoluzione

Durata: 2 giorni

3

ORGANIZZAZIONE E TECNICHE

PRECISARE I REQUISITI

DESCRIZIONE

Precisare i requisiti serve a fare chiarezza, evidenziando differenze di interpretazione tra punti di vista diversi che altrimenti rimarrebbero nascoste. Serve a tutti i ruoli coinvolti nei progetti per valutare meglio le priorità e individuare soluzioni accettabili (trade-off) tra esigenze potenzialmente in contrasto. Serve per decidere le scelte architeturali. Serve per lo sviluppo: spesso i requisiti da cui si parte sono generici, ma il sistema da realizzare deve essere specifico.

Che si tratti di requisiti booleani (verificabili in termini di vero/falso, come i requisiti funzionali) o scalari (verificabili con misurazioni, come la maggior parte dei requisiti non funzionali), la precisazione dei requisiti consente di risparmiare tempi e budget e di aumentare la qualità del sistema realizzato.

Nel corso vengono insegnate le tecniche più efficaci per precisare i requisiti, con esercitazioni pratiche per consentirne un apprendimento effettivo.

PARTECIPANTI

- analisti
- capi progetto
- esperti di dominio coinvolti nella definizione dei requisiti
- architetti, progettisti e sviluppatori software

PREREQUISITI

Questo è un corso di approfondimento, ed è quindi opportuna la partecipazione preventiva al corso “*Gestione dei Requisiti con i Casi d’Uso*”, o il possesso di conoscenze equivalenti.

PROGRAMMA

Planguage, un linguaggio per precisare i requisiti

- elementi base per la specifica
- precisare i requisiti funzionali
- precisare i requisiti non funzionali
- scomposizione di requisiti complessi in requisiti elementari
- scale e misurazioni

Verifiche sui requisiti documentati

- verifiche strutturate della documentazione
- walkthrough di scenari e Casi d’Uso

Derivare i test dai requisiti

- la relazione tra requisiti e test
- test per precisare i requisiti
- derivare casi di test dai Casi d’Uso
- specifiche eseguibili: FIT, Behaviour Driven Development

Durata: 2 giorni

PER LO SVILUPPO

ANALISI DELLE FUNZIONI

DESCRIZIONE

Il corso presenta un insieme di tecniche e di modelli di analisi (Scomposizione funzionale, Dependency Diagram, Data Flow Diagram, State Transition Diagram, Action Diagram) per l'individuazione e la definizione delle esigenze funzionali di un sistema applicativo.

Sul piano teorico il corso si sofferma in particolare sulla tecnica di modellazione basata sui "Data Flow Diagram" (DFD), presentata in un quadro di riferimento ispirato ai principi dell'Analisi Strutturata e dell'Information Engineering.

Il corso prevede una serie di esercitazioni pratiche di difficoltà crescente, per permettere un'assimilazione effettiva dei concetti e delle tecniche proposti in sede teorica.

PARTECIPANTI

- capi progetto
- analisti
- data administrator
- database administrator

PREREQUISITI

Conoscenza generale delle problematiche relative allo sviluppo.

PROGRAMMA

Introduzione

- obiettivo dell'analisi
- collocazione nel ciclo di vita dei sistemi
- interazione con l'analisi dati

Scomposizione funzionale

Dependency Diagram

Data Flow Diagram (DFD): concetti base

Il diagramma di contesto

Individuazione degli eventi

Criteri per la scomposizione di un sistema

Definizione della logica di un processo elementare

Rappresentazione dei processi di regolazione e controllo di un sistema

Diagrammi di transizione di stato

Preparazione delle fasi successive di disegno tecnico e realizzazione

Case Study ed esercitazioni

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

SYSTEM ENGINEERING CON SysML

DESCRIZIONE

Il Systems Modeling Language (SysML) è lo standard OMG per l'ingegneria dei sistemi. Deriva da UML (Unified Modeling Language), lo standard per la rappresentazione del software, e ne costituisce un profilo compatibile. Rispetto a UML, SysML ha introdotto estensioni utili per la rappresentazione di sistemi complessi in cui il software, se presente, costituisce solo una delle componenti.

Il corso presenta tutti i diagrammi SysML, che coprono dalla specifica dei requisiti alla definizione degli aspetti strutturali e comportamentali, evidenziando le relazioni che legano tra loro gli elementi per consentire la loro tracciabilità.

L'esposizione dei concetti viene accompagnata da una nutrita serie di esercitazioni.

DESTINATARI

- capi progetto
- ingegneri di sistemi
- analisti ed architetti IT

PREREQUISITI

Dato che SysML estende UML è opportuna la preventiva partecipazione al corso "Sviluppo di applicazioni con UML", o il possesso di conoscenze equivalenti.

PROGRAMMA

Introduzione a SysML

- ambito di applicazione di SysML
- relazioni con UML
- panoramica sui diagrammi

Rappresentazione dei requisiti

- Requirement Diagram
- caratteristiche dei requisiti
- tracciabilità dei requisiti con gli elementi di implementazione

Aspetti strutturali

- Block Definition Diagram
- Internal Block Diagram
- Parametric Diagram
- Package Diagram

Aspetti comportamentali

- Use Case Diagram
- Sequence Diagram
- State Machine Diagram
- Activity Diagram

PER LO SVILUPPO

Costrutti ulteriori

- Allocazioni
- Modelli, View, Viewpoint

Durata: 2 giorni

3

ORGANIZZAZIONE E TECNICHE

ANALISI OBJECT-ORIENTED CON UML

DESCRIZIONE

Per sfruttare pienamente le potenzialità delle nuove tecnologie e piattaforme di sviluppo del software (sviluppo per componenti, interoperabilità dei sistemi, riusabilità del codice) è necessario “pensare” in termini Object -Oriented fin dalle prime fasi del ciclo di vita delle applicazioni.

Il corso presenta, con un’alternanza di lezioni teoriche ed esercitazioni, un procedimento generale di Analisi OO basato sulle più recenti esperienze pratiche e proposte metodologiche, e in particolare su UML: l’Unified Modeling Language, oggi giunto alla versione 2.0, di cui adotta notazioni e semantiche. Vengono esaminate le principali tecniche e Best Practices per progettare, fin dalla fase di analisi, applicazioni robuste ed estensibili.

Il corso fornisce inoltre un workflow di riferimento per le attività da svolgere e una serie di indicazioni pratiche per la produzione dei deliverables di ogni singola attività.

A conclusione vengono illustrate le principali opzioni tecnologiche oggi a disposizione e le linee guida per le successive fasi di disegno e implementazione.

PARTECIPANTI

- data e database administrator
- capi progetto
- analisti
- analisti programmatori
- specialisti di metodi e standard

PREREQUISITI

Conoscenza generale delle problematiche relative allo sviluppo e nozioni di modellazione E-R.

PROGRAMMA

Modelli per l’analisi

- l’analisi nel processo di sviluppo
- l’approccio Object-Oriented

Concetti base dell’Object-Oriented

- oggetti e classi, istanziazione, incapsulazione
- messaggi e metodi
- polimorfismo e ereditarietà
- abstract data type
- benefici dell’approccio OO

Unified Modeling Language

- UML come linguaggio universale di modellazione
- diagrammi UML e fasi del processo di sviluppo

PER LO SVILUPPO

Casi d'Uso e analisi dei requisiti

- dai requisiti ai Casi d'Uso
- attori e Casi d'Uso
- delimitazione del contesto dell'analisi
- scenari di un Caso d'Uso
- relazioni tra Casi d'Uso
- partizionamento del sistema in sottosistemi

Realizzazione dei Casi d'Uso (dai Casi d'Uso ai componenti)

- oggetti nel dominio dell'analisi
- allocazione delle funzionalità del Caso d'Uso agli oggetti
- ruoli, responsabilità e collaborazioni degli oggetti

Analisi Object-Oriented: modello statico

- classi e tipi
- identificatori, attributi e operazioni
- relazioni fra classi (associazioni, gerarchie di specializzazione, gerarchie di aggregazione)
- diagrammi delle classi e degli oggetti
- vincoli di integrità

Analisi Object-Oriented: modello dinamico

- interazioni tra gli oggetti: diagrammi di sequenza e collaborazione
- ciclo di vita degli oggetti: diagrammi di transizione di stato
- diagrammi di attività

Dall'analisi al disegno

- le opzioni tecnologiche
- lo sviluppo per componenti
- riusabilità e interoperabilità
- DBMS Object-Oriented vs RDBMS estesi
- mapping del modello delle classi nel relazionale

Cenni ai Patterns utilizzabili nella fase di analisi

Case Study

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

DISEGNO OBJECT-ORIENTED CON UML

DESCRIZIONE

Il disegno è un momento cruciale dello sviluppo di un'applicazione, in cui si porta a pieno compimento la definizione della sua architettura.

Nelle pratiche abituali, tuttavia, troppo spesso si passa dalle fasi "alte" dell'analisi direttamente all'implementazione, avendo definito in modo sommario l'architettura e non affrontando con la dovuta attenzione e disciplina la fase di progettazione. Si ha così un impatto fortemente negativo su qualità essenziali del software quali le prestazioni, la manutenibilità, l'efficienza, la portabilità. Inoltre non viene favorito il riutilizzo, non solo del codice, ma in senso lato.

Il corso illustra inizialmente come si cala la fase di Disegno nei vari tipi di processo di sviluppo, definendone presupposti e risultati; vengono quindi esaminate le tecniche di Disegno e i principi che guidano la realizzazione delle migliori architetture, attingendo abbondantemente alle "Best Practices" e ponendo particolare enfasi sui Patterns di disegno.

In tale ottica va anche visto l'utilizzo della notazione UML (Io Unified Modeling Language: qui viene fatto riferimento alla versione 2.0) e il riferimento all'approccio MDA, Model Driven Architecture, di cui UML è un fondamento.

Il corso è completato da una serie di cenni all'implementazione e ai fondamenti di progettazione dell'interfaccia utente

PARTECIPANTI

- architetti di sistema
- analisti applicativi
- analisti programmatori
- specialisti di metodi e standard

PREREQUISITI

Corso di "Analisi Object-Oriented con UML" o conoscenze equivalenti. È utile l'esperienza di programmazione con linguaggi OO.

PROGRAMMA

UML e processi software

- UML: convergenza dei metodi, convergenza sul metodo
- UML e processi software
- i principali artefatti di UML
- un linguaggio senza processo? Linee guida per uno sviluppo incrementale, iterativo, basato sull'architettura

Dall'analisi al disegno

- ruolo dell'analisi e del disegno nel Processo software
- il disegno in un processo di tipo incrementale, iterativo, basato sull'architettura e sui Casi d'Uso
- prodotti "ereditati" dall'Analisi: specifiche funzionali, requisiti non funzionali, Diagramma dei Casi d'Uso, Diagramma delle Classi, Diagrammi di Stato e di Attività, Diagrammi di Sequenza; caratteristiche richieste a questi prodotti
- estensione e affinamento in disegno dei Diagrammi UML "ereditati"
- documentare l'architettura applicativa: concetti base dell'approccio Model Driven Architecture (PIM, PSM, ecc.)

PER LO SVILUPPO

Principi per il disegno Object-Oriented

- coesione e coupling delle classi
- ruoli, responsabilità e collaborazioni: contratti e deleghe, distribuzione delle responsabilità
- framework e Pattern: concetti fondamentali; come usarli per ottenere economie in fase di progettazione e per favorire il riuso non solo del codice
- Patterns fondamentali: MVC, Façade, Publish-Subscribe, Proxy, Abstract Factory, GRASP, ecc. Quali usare, quando e come

Disegno dell'applicazione

- Disegno dell'architettura applicativa: partizionamento in sottosistemi, stratificazione dei sottosistemi, tecniche di interfacciamento e disaccoppiamento
- Disegno delle componenti: individuazione delle componenti, definizione delle loro responsabilità e collaborazioni, realizzazione dei diagrammi di disegno (diagrammi di sequenza, delle classi, delle azioni, delle componenti)

Linee guida alla implementazione

- ergonomia: impostazione di uno standard di progettazione delle interfacce
- problemi applicativi: validazione dell'input, exception handling, accesso a fonti di dati eterogenee
- mapping a un DBMS relazionale: classi vs tabelle, riferimenti all'uso di chiavi artificiali, trigger e stored procedure
- cenni all'Aspect Oriented Programming, nuovo paradigma di programmazione

Case Study

Durata: 3 giorni

3

ORGANIZZAZIONE E TECNICHE

WORKSHOP DI ANALISI E DISEGNO CON UML

DESCRIZIONE

UML si sta imponendo all'attenzione di quanti intendono estendere il rigore della notazione formale e dei meta modelli alle fasi "alte" del ciclo di sviluppo, senza sacrificare l'efficienza del processo di sviluppo.

Il workshop rappresenta un'opportunità per favorire l'adozione e l'utilizzo di UML nelle aziende, in quanto permette di applicare i concetti e le tecniche esposte nei corsi teorici in un Caso Studio di adeguata complessità, mutuato da un'esperienza reale.

In tal senso il workshop è il giusto complemento ai corsi di Analisi e Disegno. I corsi teorici, infatti, sono pensati per trasferire ai partecipanti le conoscenze necessarie per comprendere le potenzialità e le modalità di utilizzo dei metodi UML; in tale ambito le esercitazioni proposte sono volte ad aumentare il livello di comprensione dei partecipanti. Il workshop, viceversa, ha l'obiettivo di permettere ai partecipanti di sperimentare le tecniche di analisi e disegno con UML apprese nei corsi nell'ambito di un progetto pilota.

Il corso si articola in un'alternanza di sessioni operative e momenti di consolidamento e verifica del lavoro svolto, che consentono di valutare assieme al mentore i risultati prodotti e fornire le linee guida per lo svolgimento delle attività.

Al fine di agevolare la definizione dei modelli UML, il workshop sarà svolto con l'ausilio di IBM Rational Rose.

PARTECIPANTI

- capi progetto
- analisti
- analisti programmatori

PREREQUISITI

La partecipazione ai corsi "Analisi Object-Oriented con UML" e "Disegno Object-Oriented con UML" o il possesso di conoscenze equivalenti.

PROGRAMMA

Presentazione del Caso Studio

Elaborazione dei risultati dell'analisi

- individuazione e descrizione dei Casi d'Uso, Modello dei Casi d'Uso
- diagramma delle Classi
- diagrammi di Stato
- diagrammi di Sequenza

Estensione e affinamento dei modelli dell'analisi e definizione dei modelli del disegno con utilizzo dei Patterns di progettazione

- definizione dell'architettura logico-fisica dell'applicazione
- diagramma delle Classi
- diagrammi di Stato
- diagrammi di Sequenza e di Collaborazione

Definizione dell'architettura fisica

- package
- diagrammi delle Componenti e di Allocazione delle Componenti (cenni)

Aspetti implementativi (cenni)

Durata: 3 giorni

PER LO SVILUPPO

WORKSHOP DI DEFINIZIONE DEI REQUISITI E CASI D'USO

DESCRIZIONE

Il workshop ha l'obiettivo di guidare i partecipanti nell'impostazione e nella definizione dei requisiti e del modello dei Casi d'Uso di un progetto.

PARTECIPANTI

- personale di progetto (analisti, progettisti)
- committenti e altre parti interessate (stakeholders)

PREREQUISITI

È opportuno che i partecipanti abbiano seguito il corso "*Gestione dei requisiti con i Casi d'Uso*" o posseggano conoscenze equivalenti.

Impostazione Preliminare

- definizione degli obiettivi specifici dell'intervento
- acquisizione da parte del conduttore della documentazione progettuale, in modo da permettere una preparazione preventiva per la conduzione della sessione

Sessione (due giorni consecutivi, a distanza di almeno una settimana dall'acquisizione della documentazione progettuale)

I contenuti specifici della sessione variano in funzione degli obiettivi concordati. Possono comprendere (tipicamente):

- elicitazione e analisi dei requisiti
- definizione modello Casi d'Uso
- definizione modalità di validazione e verifica

Durata: 2 giorni

4

WEB, SOA, ARCHITETTURE E TE

WEB 2.0: TECNOLOGIE E PROBLEMATICHE

DESCRIZIONE

Le applicazioni Web tradizionali sono poco interattive e offrono interfacce utente a volte poco ergonomiche e lente da utilizzare. Nel corso degli anni gli sviluppatori hanno adottato numerosi trucchi per ovviare a questi problemi, ma le soluzioni trovate non sono mai state divulgate in modo corretto.

Con l'avvento del Web 2.0 e della crossmedialità, gli utenti richiedono applicazioni con caratteristiche di usabilità e interattività che le tecnologie di sviluppo standard (ASP, JSP, ecc.) non sono in grado di garantire. Le soluzioni tecnologiche esistono da molti anni, ma solo oggi cominciano a uscire i primi standard. Le tecnologie basate su XML e Javascript (AJAX) si stanno imponendo come soluzione principe per l'implementazione di applicazioni per il Web 2.0.

Il corso fornisce un apparato teorico e pratico per la progettazione di applicazioni Web adeguate alle nuove caratteristiche del Web 2.0 e della crossmedialità. Il corso intende fornire inoltre indicazioni di base per la conoscenza degli strumenti di authoring e delle diverse modalità di interazione possibili tra l'utente e il sistema e le linee guida per il publishing di pagine multimediali.

PARTECIPANTI

Tutti coloro che sono interessati all'utilizzo delle nuove tecnologie per l'implementazione di applicazioni Web complesse.

PREREQUISITI

Nessuno.

PROGRAMMA

L'evoluzione del Web: dai primordi al futuro Web 2.0

- nuovi paradigmi della comunicazione
- cambiamenti nella fruizione dei servizi Web
- cambiamento del ruolo dell'utente: da passivo ad attivo
- confronto fra le applicazioni Web standard e le applicazioni Web 2.0

Architettura delle applicazioni Web 2.0

- Client/Server su Web
- User Interface dinamica
- Back-end a servizi
- esempi di architetture
- la comunicazione fra il Client e il Server

Tecnologie per l'implementazione delle applicazioni Web 2.0

- esecuzione lato Client o lato Server?
- AJAX: un nome tante soluzioni
- Microsoft Web Form ed estensioni 2.0
- SUN Java Server Face e relativi cloni
- tecnologie Open Source per l'implementazione di interfacce dinamiche
- confronto fra le tecnologie
- Casi Studio ed esempi

CNOLOGIE

Utilizzo di componenti multimediali

- integrazione di audio e video nelle applicazioni Web
- integrazione di contenuti proprietari

Portal Server: evoluzione dei Portali

- nuova generazione dei Portali Web
- Portlet e componenti riusabili

Criticità delle applicazioni Web 2.0

- ottimizzazione delle prestazioni
- sicurezza delle transazioni
- gestione dello stato del sistema

Durata: 3 giorni

4

WEB, SOA, ARCHITETTURE E TE

SOA (SERVICE ORIENTED ARCHITECTURE): BEST PRACTICES

DESCRIZIONE

Al fine di ottimizzare i costi e di migliorare la gestione dell'IT, le aziende guardano sempre con maggiore interesse le problematiche di governance dei processi aziendali. Fra i vari modelli di riferimento per l'implementazione e la gestione dei Business Process, SOA (Service Oriented Architecture) è quello più promettente e apprezzato. Il censimento dei processi aziendali e la loro implementazione in una SOA consente di sviluppare e mantenere le applicazioni con minor sforzo consentendo il riutilizzo del codice e ottimizzando le comunicazioni fra le varie aree di competenza presenti all'interno di un'azienda. Le attuali tecnologie dei Web Services e i sistemi di BPM (Business Process Management) si pongono al vertice degli strumenti attualmente utilizzati per implementare una SOA, ma le tecnologie distribuite meno recenti (CORBA, DCOM, MQ, ecc.) stanno rivivendo una seconda giovinezza anche a causa della presenza di nuovi framework di sviluppo e alle prestazioni di livello superiore.

Di SOA si parla molto ma: quali sono i tempi e i costi necessari per l'implementazione di una SOA? Quali sono le piattaforme e le tecnologie più promettenti per l'implementazione di una SOA? Quali sono le architetture e i Pattern più utilizzati per progettare una SOA? Quali sono gli impatti organizzativi che derivano dall'implementazione di una SOA? Quali vantaggi derivano e quali rischi si corrono nell'implementazione di una SOA?

Il corso, partendo dal confronto delle architetture di sviluppo tradizionali, introduce i concetti fondamentali su cui si basa una SOA descrivendone le funzionalità e le peculiarità tramite l'utilizzo di Case Study rappresentanti problemi reali. Vengono poi affrontate le problematiche relative alla progettazione e all'implementazione di una SOA descrivendone gli aspetti metodologici, tecnologici e organizzativi.

PARTECIPANTI

- capi progetto
- analisti
- professionisti IT
- chiunque voglia approfondire gli aspetti legati alla SOA

PREREQUISITI

Concetti base di progettazione e sviluppo applicazioni Web.

PROGRAMMA

- che cos'è una SOA: descrizione, concetti fondamentali e confronto con le architetture tradizionali
- requisiti, tempi e costi di una SOA
- progettare oggetti di Business orientati ai servizi
- strumenti per il BPM (BPEL, BPEL4WS, ecc.)
- vantaggi e rischi di una SOA
- architettura di una SOA e Pattern di progettazione e sviluppo
- tecnologie per l'implementazione di una SOA
- impatti organizzativi, strategici, economici e tecnologici

Durata: 3 giorni

SOA PER L'INTEGRAZIONE DELLE APPLICAZIONI AZIENDALI

DESCRIZIONE

Lo sviluppo della nuova generazione di applicazioni Web e la creazione di architetture orientate ai servizi riporta in primo piano il problema dell'integrazione delle applicazioni esistenti, siano esse Web, Client/Server o su piattaforme host con le nuove infrastrutture tecnologiche.

In tale ambito, infatti, non basta focalizzare la propria attenzione sull'interoperabilità dei protocolli per l'implementazione dei servizi (XML e SOAP), ma è necessario individuare meccanismi e tecniche sia per l'integrazione tra la rete e il sistema informativo aziendale che per il riuso di componenti applicative già esistenti. A tale scopo gli Enterprise Service Bus, gli integration Server e le suite per il Business Process Management vengono in aiuto del progettista e dello sviluppatore.

Il corso fornisce un quadro di riferimento per progettare in modo adeguato l'interconnessione tra le diverse componenti tecnologiche e applicative distribuite, utilizzando le principali soluzioni che le infrastrutture SOA rendono disponibili sul mercato.

PARTECIPANTI

- capi progetto
- analisti
- analisti programmatori
- programmatori

PREREQUISITI

La partecipazione al corso "SOA: Best Practices" o il possesso di conoscenze equivalenti.

PROGRAMMA

L'evoluzione delle applicazioni in azienda: dal mainframe alle SOA

Il problema dell'integrazione: obiettivi, scenari, implicazioni

Integrare le applicazioni aziendali tramite SOA

- tipi di applicazioni
- gli strumenti per l'integrazione: Integration Server e Enterprise Service Bus
- i connettori di interfacciamento: tipologie e funzionamento
- architetture applicative per l'integrazione tramite SOA
- Casi Studio ed esempi

Integrare SOA con i DBMS aziendali

- database e XML
- database federati
- architetture di integrazione
- Casi Studio ed esempi

Integrazione dei processi aziendali

- gli strumenti per il Business Process Management
- Orchestration e Choreography dei servizi e dei Business Process
- architetture per la gestione dei Business Process
- Casi Studio ed esempi

Analisi comparata fra gli strumenti e le tecnologie di integrazione

- strumenti commerciali
- strumenti Open Source

Cenni sulle problematiche di sicurezza di una SOA

Durata: 3 giorni

4

WEB, SOA, ARCHITETTURE E TE SVILUPPO DI APPLICAZIONI WEB CON AJAX

DESCRIZIONE

Una nuova generazione di applicazioni Web si sta affacciando sul mercato; queste applicazioni hanno come principale caratteristica una elevata interattività, che si sposa con un aspetto gradevole e una elevata velocità di collegamento con il Server, tanto da sembrare quasi applicazioni tradizionali.

Queste applicazioni sono basate sulla tecnologia AJAX (Asynchronous JavaScript and XML), una tecnologia giovane che sta dando luogo a sviluppi interessanti, sia sul fronte delle applicazioni che sul fronte degli strumenti di sviluppo.

La tecnologia AJAX è anche un passo importante verso la realizzazione di applicazioni basate sul concetto di Service Oriented Architecture (SOA), ottenute integrando a livello Client diversi servizi indipendenti basati sul concetto di Web Service.

Gli sviluppatori di sistemi Web saranno chiamati, nel futuro prossimo, a introdurre la tecnologia AJAX sempre più spesso nelle loro applicazioni. Devono perciò avere il panorama della tecnologia, partendo dai concetti di base per realizzare applicazioni con AJAX fino a comprendere come usare i vari framework di sviluppo integrato.

Il corso ha l'obiettivo di introdurre tutto questo. Si parte dai fondamenti, introducendo le caratteristiche essenziali di JavaScript, le nozioni fondamentali di XML, la rappresentazione DOM. Quindi, si passa a vedere il funzionamento del livello fondamentale di AJAX, la comunicazione asincrona con l'oggetto XMLHttpRequest. Si procede usando il framework Yui (Yahoo User Interface) che consente di realizzare facilmente interfacce grafiche evolute con AJAX.

Infine, si vedono i vari framework che consentono di sviluppare applicazioni Client/Server fortemente integrate, quali: GWT (Google Web Toolkit), ASP.NET AJAX, Sun JSF (Java Server Faces). Durante il corso verranno svolti esercizi pratici di sviluppo di interfacce AJAX.

PARTECIPANTI

Tutti coloro che sono interessati all'utilizzo delle nuove tecnologie per l'implementazione di applicazioni Web complesse.

PREREQUISITI

Conoscenze generali sulle piattaforme Web e conoscenze di base di programmazione.

È necessario essere muniti di notebook, per svolgere gli esercizi di programmazione AJAX.

PROGRAMMA

La visione d'insieme

- applicazioni Web Dinamiche
- i Web Services
- SOA: Service Oriented Architecture

I concetti fondamentali

- il linguaggio JavaScript
- XML: caratteristiche fondamentali
- DOM (Document Object Model): la rappresentazione dei documenti XML
- alcuni esercizi pratici con il DOM in JavaScript

AJAX base

- l'oggetto XMLHttpRequest
- il lato Server
- gestire la risposta
- JSON: AJAX senza XML
- alcuni esercizi pratici

Le interfacce grafiche

- i framework grafici per la realizzazione di interfacce grafiche
- il framework YUI (Yahoo User Interface)
- AJAX e YUI
- alcuni esercizi pratici di realizzazione di interfacce grafiche con YUI

I framework integrati

- i framework integrati per la realizzazione di applicazioni Client/Server
- GWT (Google Web Toolkit)
- ASP.NET AJAX
- JSF (Java Server Faces)
- piccoli esercizi pratici

Durata: 3 giorni

4

WEB, SOA, ARCHITETTURE E TE

SVILUPPO DI APPLICAZIONI WEB CON XML E WEB SERVICES

DESCRIZIONE

Lo sviluppo di applicazioni Web enfatizza ulteriormente gli aspetti che riguardano l'indipendenza dell'applicazione dal dispositivo di visualizzazione, la comunicazione fra componenti realizzate con tecnologie eterogenee, spesso in concorrenza tra loro (Microsoft, Java, ecc.), e la creazione di componenti facilmente riusabili. Le tecnologie Web standard non consentono di ottenere un buon livello di flessibilità nello sviluppo e una semplicità di manutenzione in grado di soddisfare le esigenze più complesse. Inoltre, con l'avvento del Web 2.0 le tecnologie basate su XML si stanno imponendo come standard di fatto per lo sviluppo di applicazioni con interfacce Web complesse. Il corso, partendo dall'esame delle tecnologie basate su XML e dei nuovi paradigmi dello sviluppo (Web Services), intende fornire ai partecipanti un quadro teorico-pratico per la progettazione di applicazioni indipendenti dal formato iniziale dei dati, dai dispositivi di visualizzazione e dalla tecnologia utilizzata per lo sviluppo delle componenti software.

PARTECIPANTI

- capi progetto
- analisti
- analisti programmatori
- programmatori

PREREQUISITI

Nessuno.

PROGRAMMA

Introduzione allo sviluppo con XML

- il linguaggio XML
- DTD e XML Schema
- Document Object Model
- i fogli di stile XSL e l'indipendenza dal dispositivo di output
- l'utilizzo di XML nei database (XSQL, ecc.)
- la creazione di nuovi linguaggi di markup tramite XML
- XHTML
- estensioni al linguaggio (XLINK, VXML, ecc.)

Lo sviluppo di applicazioni XML

- l'architettura di un'applicazione XML Based
- le principali api per lo sviluppo con XML (SAX, ecc.)
- lo sviluppo di applicazioni XML con tecnologia Microsoft
- lo sviluppo di applicazioni XML con tecnologia SUN

Interoperabilità con SOAP

- introduzione a SOAP
- SOAP e XML, meccanismo di funzionamento
- architettura di un'applicazione basata su SOAP
- creazione di applicazioni SOAP con tecnologia Microsoft e SUN

CNOLOGIE

Creazione di componenti riutilizzabili tramite i Web Services

- introduzione ai Web Services
- architettura di un sistema basato sui Web Services
- la proposta Microsoft
- la proposta Sun
- utilizzo congiunto di XML, SOAP e Web Services

Esempi e confronti fra le tecnologie

Durata: 3 giorni

4

WEB, SOA, ARCHITETTURE E TE

IMPLEMENTARE I PROCESSI DI BUSINESS CON SOA E BPM

DESCRIZIONE

SOA (Service-Orient e Architecture) e BPM (Business Process Management) consentono alle aziende di creare nuovo valore di Business a partire dagli investimenti esistenti, di riutilizzare le componenti software e di Business realizzate nei vari progetti e ottenere nuovi livelli di agilità, grazie a una maggiore flessibilità e a strutture di minor costo. Spesso SOA e BPM vengono confusi e descritti come intercambiabili, probabilmente perché molti dei vantaggi che li caratterizzano sono simili. Tuttavia si tratta di iniziative molto diverse...

Il BPM è un complemento naturale della SOA, rappresenta un meccanismo attraverso il quale un'organizzazione può applicare la SOA a importanti e avanzate sfide aziendali. Il BPM rappresenta un nuovo modo per analizzare e misurare il Business in termini di processi aziendali che vanno oltre i tradizionali confini dell'organizzazione e dei sistemi. Per l'IT, BPM e SOA rappresentano un nuovo modo per implementare le soluzioni aziendali, sottolineando l'importanza di ricorrere a meno programmazione, promuovere il riutilizzo dei componenti e ottenere un maggiore coinvolgimento dell'azienda.

Sia SOA che BPM possono essere adottati singolarmente, ma è l'unione dei due modelli architetturali che esprime il massimo delle potenzialità: su scala aziendale, l'applicazione del BPM su una solida base SOA lo rende più agile e flessibile, lo diffonde più facilmente nelle varie realtà aziendali e strutture collaborative ed è più resistente alle modifiche dei sistemi IT sottostanti.

Il corso, partendo da una breve introduzione a SOA, descrive i principali concetti relativi al BPM, analizza le problematiche di progettazione e implementazione dei Business Process tramite utilizzo di esempi e Casi Studio reali.

PARTECIPANTI

- capi progetto
- analisti
- professionisti IT
- chiunque voglia approfondire gli aspetti legati al Business Process Modelling

PREREQUISITI

Concetti base di progettazione e sviluppo applicazioni Web, la partecipazione al corso "SOA: Best Practices" o conoscenze equivalenti.

PROGRAMMA

Breve introduzione a SOA

- cos'è SOA
- vantaggi e svantaggi
- perchè introdurre SOA in Azienda
- i layer di una SOA

I Processi

- definizione di processo
- l'azienda e i processi organizzativi
- BAM (Business Activity Modelling)
- elementi costitutivi dei processi
- il Business Process Reengineering, dalla situazione AS IS al TO BE

Introduzione al BPM

- cos'è e a che cosa serve il BPM
- i tre livelli di integrazione
- Workflow Management
- Business Process Management
- Enterprise Management
- Business Process e Business Model
- pro e contro

Composizione di processi

- Workflow e BPM
- Orchestration e Choreography
- Pattern di Workflow
- Business Process Design
 - * identificazione dei Business Process
 - * Business Process Modelling Notation (BPMN)
 - * BPMN vs UML/RUP
 - * Tool per la progettazione
 - * Test
 - * Monitor e Tracciamento

Infrastruttura tecnologica

- motori di Workflow
- le suite per il BPM
- esempi di architettura
- i linguaggi per la descrizione dei processi (BPEL/WS-BPEL, WS-CDL)

Esempi e Casi Studio

Durata: 3 giorni

5

DATA WAREHOUSE

DATA WAREHOUSE: ARCHITETTURA E PRINCIPI

DESCRIZIONE

Il Data Warehouse si trova oggi al centro dell'interesse di numerose aziende che intendono supportare in modo adeguato i processi decisionali.

I Data Warehouses poggiano le basi sui sistemi gestionali, ma utilizzano anche dati di Sistemi Informativi esterni. L'organizzazione delle informazioni e lo stile d'interazione tra utente e sistema sono evidentemente differenti rispetto ai Sistemi Informativi tradizionali.

Il corso non si limita a illustrare i possibili benefici che si possono prospettare al Management, ma esamina in toto il ciclo di vita del Data Warehouse, le tecnologie disponibili e le risposte che può dare a seconda della tipologia utente.

Obiettivo del corso è, infine, definire una chiara strategia di sviluppo e di utilizzo anche alla luce del nuovo paradigma di sviluppo applicativo Internet/Intranet.

PARTECIPANTI

- responsabili dello sviluppo
- analisti
- analisti programmatori

PREREQUISITI

Conoscenza di base su Qualità, Modello Dati, Teoria Relazionale.

PROGRAMMA

Ragioni aziendali tecnologiche ed economiche del Data Warehouse e differenze fra i sistemi gestionali e i Data Warehouses.

Data Warehouse Framework

- esame delle componenti di base di un ambiente di Data Warehousing

Problematiche e modalità di gestione/pianificazione di un progetto in ambiente di Data Warehousing

Integrazione e Qualità

- problematiche e modalità di approccio

Meta Dati

- ruolo all'interno dell'ambiente di Data Warehousing
- repository dei Meta Dati

Architettura di un Data Warehouse

- confronto fra le modalità di approccio (Global Data Warehouse, Interconnected Data Mart)
- confronto fra i modelli usabili: Relational, Star Schema e Snowflake Schema

Area di ETL

- analisi delle componenti (Capture, Transform, Apply)
- problematiche e tecniche di approccio

Le applicazioni tipiche che operano su un Data Warehouse

- Query/Reporting
- DSS/EIS
- OLAP
- Data Mining
- distribuzione dei risultati

EAI e Business Intelligence Portal (Closed Loop)

Durata: 3 giorni

5

DATA WAREHOUSE

PROGETTAZIONE DEL DATA WAREHOUSE

DESCRIZIONE

Larga parte delle organizzazioni mancano della necessaria esperienza e capacità per affrontare con successo le sfide implicite nei progetti di Data Warehousing. In particolare, uno dei fattori che a tutt'oggi maggiormente minaccia la riuscita di questi progetti è la mancata adozione di un approccio metodologico che, basandosi sulle Best Practices del settore, sia in grado di minimizzare i rischi di insuccesso.

Obiettivo del corso è individuare un ampio repertorio di tecniche di progettazione di Data Warehouse, integrandole in un quadro metodologico organico e completo che funga da riferimento costante per il progettista. Punto di forza della metodologia proposta è l'utilizzo di una fase di progettazione concettuale, con l'obiettivo di minimizzare i rischi di progetto.

Il corso affronta gli argomenti di progettazione dal punto di vista pratico ma con solide fondamenta teoriche. L'esposizione è avvalorata da numerosi esempi applicativi, da esercitazioni di progetto e da un Caso di Studio completo.

PARTECIPANTI

- capi progetto
- analisti e analisti programmatori
- amministratori di database
- specialisti di Data Warehousing

PREREQUISITI

La partecipazione al corso "*Data Warehouse: architettura e principi*" o il possesso di conoscenze equivalenti; conoscenza del modello relazionale e delle tecniche di progettazione delle basi di dati.

PROGRAMMA

Introduzione al Data Warehousing

- concetti di base e terminologia
- il modello multidimensionale
- analisi dei dati: reportistica, OLAP, mining, analisi what-if
- architetture a 1, 2 e 3 livelli
- architetture evolute: BAM e BPM

Metodologie di progettazione del Data Warehouse

- il ciclo di sviluppo
- fasi della progettazione
- approccio supply-driven vs approccio demand-driven

Analisi e riconciliazione delle sorgenti operazionali

- ricognizione, normalizzazione, integrazione

Analisi dei requisiti utente

- interviste e glossari

Progettazione concettuale

- modelli concettuali per il Data Warehouse
- Pattern avanzati di modellazione: attributi descrittivi e cross-dimensionali; convergenze; gerarchie condivise, incomplete e ricorsive; archi multipli e opzionali; dinamicità; additività.
- gestione del tempo: gli aggiornamenti retrospettivi

Progettazione logica

- schemi logici per il Data Warehouse: star schema, snowflake schema, constellation schema
- scenari temporali e slowly-changing dimension
- traduzione dei Pattern di modellazione
- carico di lavoro e volume dati
- tecniche di ottimizzazione logica: viste e frammentazione

Progettazione dell'alimentazione

- analisi delle problematiche legate all'estrazione, caricamento e pulizia dei dati

Progettazione fisica

- gli indici per il Data Warehouse
- tecniche di ottimizzazione fisica

La documentazione di progetto

- il livello del Data Warehouse
- il livello dei Data Mart
- il livello dei fatti

Esercitazione, Case Studies ed esempi applicativi

Durata: 3 giorni

6

CAPACITY MANAGEMENT

CAPACITY PLANNING

DESCRIZIONE

Se, da un lato, un sistema informatico deve soddisfare dei ben precisi requisiti di prestazioni nei confronti degli utenti del sistema, d'altro canto l'azienda deve contenere i costi dell'infrastruttura informatica. Per questa ragione deve essere effettuata una attività periodica di Capacity Planning che garantisca un corretto dimensionamento dei sistemi al variare degli scenari di funzionamento dell'applicazione.

L'obiettivo del corso è quello di fornire una metodologia efficace e di mostrare gli strumenti più idonei per effettuare periodicamente il Capacity Planning.

Il processo di Capacity Planning parte da una sistematica misurazione delle prestazioni e dei volumi di carico, passa per la parametrizzazione di un modello matematico di facile "intuizione" e si conclude con un'attività altrettanto sistematica di what-if-analysis, che principalmente consente di:

- valutare la capacità massima del sistema; ad esempio, stimare qual è il numero massimo di utenti che il sistema è in grado di gestire prima che determinati indicatori di qualità non vengano più soddisfatti
- confrontare la capacità massima del sistema con le curve di trend del carico per prevedere se e sino a quando il sistema sarà in grado di funzionare entro i limiti operativi prefissati
- valutare l'impatto sulle prestazioni in seguito al fail-over
- identificare le risorse e i sistemi che costituiscono un collo di bottiglia per le prestazioni in modo da pianificare eventuali upgrade
- valutare gli effetti sulle prestazioni di buffer applicativi quali, ad esempio, la dimensione del connection-pool verso un database Server

PARTECIPANTI

- responsabili dei sistemi informativi
- responsabili delle applicazioni
- responsabili dei servizi
- addetti alla funzione di gestione e controllo della qualità
- persone coinvolte nello sviluppo del software (capi progetto, analisti, ecc.)

PREREQUISITI

La partecipazione al corso *"Misurare e mantenere le prestazioni delle applicazioni Web"*. Conoscenza generale delle problematiche relative alla progettazione, manutenzione o assistenza di applicazioni e sistemi. Il corso non richiede particolari background matematici.

PROGRAMMA

Richiami a concetti di base

- prestazioni di un sistema informatico
- tempo di risposta e throughput di un'applicazione
- cosa è il Capacity Planning: modello del sistema e modello del carico

Teoria delle reti di code

- risorse e richieste
- classificazione delle risorse (a coda, a ritardo, multiservente, passive)
- modello del carico e visite
- calcolo dei parametri

Tecniche risolutive

- sistemi aperti e chiusi
- tecniche approssimate e tecniche esatte
- individuazione dei bottleneck

Il processo di Capacity Planning

- definizione degli obiettivi
- analisi dell'architettura e dell'applicazione
- costruzione del modello del sistema e del modello del carico
- attivazione degli agenti di raccolta
- calcolo dei parametri del modello
- risoluzione del modello

What-if analysis

- capacità massima e residua di un sistema
- bottleneck removal
- analisi di un sistema in caso di failure o di upgrade

Case Studies (applicazioni Web e Intranet)

Durata: 2 giorni

6

CAPACITY MANAGEMENT

FONDAMENTI DI ITIL® V3

DESCRIZIONE

ITIL®, è ad oggi il modello più diffuso e apprezzato per la gestione dei Servizi IT. La versione 3 di ITIL® si compone di cinque discipline fondamentali: Service Strategy, Service Design, Service Transition, Service Operation, Continual Service Improvement. Le Best Practice di ITIL®, basate sul ciclo di vita del servizio, hanno soddisfatto le aspettative di molteplici aziende permettendo una riduzione dei costi e un allineamento dei servizi IT agli obiettivi di Business. La riorganizzazione dei processi di gestione dei servizi IT richiede il coinvolgimento dell'intera azienda ed è quindi fondamentale che il personale disponga di competenze adeguate per essere in grado di offrire un concreto supporto. Il corso fornisce ai partecipanti la visione dei concetti base necessaria per comprendere i processi, le relazioni tra di essi e i benefici apportati dall'approccio orientato al ciclo di vita del servizio.

PARTECIPANTI

Tutti i professionisti IT che operano nella progettazione, implementazione, erogazione e supporto al Servizio.

PREREQUISITI

Esperienza delle problematiche IT.

PROGRAMMA

Framework ITIL®

- introduzione e obiettivi del Service Management
- concetti chiave di IT Library
- differenze con ITIL® V2

Service Lifecycle

- i processi ITIL® nel Ciclo di Vita del Servizio

Service Strategy

- trasformare il Service Management in un asset strategico
- utility e warranty
- value creation
- i processi di riferimento
- i ruoli e le responsabilità

Service Design

- una guida per la progettazione e lo sviluppo dei servizi e dei processi di gestione dei servizi value to Business
- il concetto di Architettura
- i processi di riferimento
- i ruoli e le responsabilità

Service Transition

- lo sviluppo e il miglioramento delle capacità necessarie per la transizione dei servizi verso la produzione
- i processi di governo dei cambiamenti
- il V-Model
- il Service Knowledge Management System
- i ruoli e le responsabilità

Service Operation

- efficacia ed efficienza nell'erogazione e nella gestione del servizio in linea
- i processi e le funzioni di riferimento
- i ruoli e le responsabilità

Continual Service Improvement

- il miglioramento continuo alla ricerca del "cost effectiveness"
- il CSI Model
- il concetto di misura
- il ciclo di Deming
- i processi di miglioramento
- i ruoli e le responsabilità

Durata: 2 giorni

6

CAPACITY MANAGEMENT

MISURARE E MANTENERE LE PRESTAZIONI DELLE APPLICAZIONI WEB

DESCRIZIONE

La qualità di un'applicazione Web dipende non solo dal contenuto e dalle funzionalità applicative, ma anche dal modo in cui le informazioni sono presentate all'utente e dalle prestazioni che egli percepisce nell'utilizzo dell'applicazione stessa. Le statistiche indicano che vi è una correlazione stretta tra prestazioni di un'applicazione Web e tasso di abbandono degli utenti.

È pertanto importante disporre, in ogni fase del processo che porta dalla realizzazione alla gestione delle applicazioni Web, di metodologie e strumenti che permettano di prevedere e misurare l'affidabilità, le prestazioni e il livello di servizio erogato sulla base di diverse condizioni di funzionamento.

Obiettivo del corso è di fornire competenze e metodologie per affrontare le diverse fasi del processo di progettazione e gestione di applicazioni Web con un occhio di riguardo ai livelli di servizio delle applicazioni, sia in termini di prestazioni che in termini di presentazione delle informazioni.

Il corso, partendo da nozioni elementari di teoria delle code, affronta inoltre gli aspetti connessi con il Capacity Planning dei Sistemi Informatici.

PARTECIPANTI

- responsabili delle applicazioni Web e dei sistemi informativi
- addetti alla funzione di gestione e controllo della qualità
- persone coinvolte nello sviluppo del software (capi progetto, analisti, ecc.)

PREREQUISITI

Conoscenza generale delle problematiche relative alla progettazione, manutenzione o assistenza di applicazioni e sistemi.

PROGRAMMA

La soddisfazione degli utenti Web

- pazienza e frustrazione, i fattori di insoddisfazione
- la percezione del tempo e le soglie di sopportazione
- l'incremental loading
- la frustrazione cumulativa e il punto di non-ritorno
- aspettative, motivazioni e alternative degli utenti

Prestazioni di un sistema Web

- pagina verso transazione Web
- tempo di risposta e throughput di un'applicazione
- definizione del livello di servizio

Come misurare le prestazioni end-to-end di applicazioni Web

- tecniche attive e passive
- packet level dump
- analisi dei log
- load testing
- definizione del campione significativo
- generazione di traffico artificiale
- reti di misurazione

Cosa è il Capacity Planning

- quali gli approcci
- costi-benefici
- il grado di profondità dell'analisi
- forecast e what-if analysis

Dimensionamento dei sistemi e teoria delle reti di code

- risorse e richieste
- classificazione delle risorse (a coda, a ritardo, multiservente, passive)
- modello del carico e visite
- calcolo dei parametri

Tecniche risolutive

- sistemi aperti e chiusi
- tecniche approssimate e tecniche esatte
- individuazione e rimozione dei bottleneck

L'attività di Capacity Planning

- definizione degli indicatori di Business (Business driver o Business key indicator)
- modello del sistema e modello del carico
- previsione del fabbisogno di risorse in funzione dell'evoluzione del Business

Case Studies di sistemi Web reali

Durata: 2 giorni

6

CAPACITY MANAGEMENT

TEST DI CARICO PER IL COLLAUDO E PER IL TUNING PRESTAZIONALE

DESCRIZIONE

Il processo di *collaudo* prestazionale è un'attività finalizzata a verificare, attraverso test di carico, se e in che misura, un'applicazione sia in grado di erogare un livello di servizio predeterminato su una specifica architettura hardware/software, prima del rilascio in produzione.

Lo scopo principale del test di carico per il *tuning* è quello di migliorare sistematicamente ogni livello di un sistema informatico complesso per garantire elevati standard qualitativi in termini di prestazioni. Questo obiettivo viene raggiunto mediante un processo ciclico che prevede di caricare il sistema, individuare le componenti che ne limitano le prestazioni ed eliminare i colli di bottiglia mediante interventi di tuning e upgrade.

Nella prima parte del corso viene descritta una metodologia che permette di sistematizzare il processo di collaudo prestazionale.

La seconda parte del corso si focalizza su due tipi di test di carico: il load test, che permette di valutare le prestazioni del sistema nelle normali condizioni operative, e lo stress test, che permette di valutare come si comporta il sistema in condizioni di massimo carico.

Durante il corso vengono mostrati i casi reali, utilizzando come strumento di riferimento LoadRunner® di Mercury.

PARTECIPANTI

- responsabili dei sistemi informativi
- responsabili delle applicazioni
- addetti alla funzione di gestione e controllo della qualità

PREREQUISITI

Nessuno.

PROGRAMMA

Definire i Service Level Objectives, la pianificazione

Il processo di test per il collaudo

- la progettazione del test
- la predisposizione degli ambienti
- la progettazione esecutiva
- l'esecuzione dei test

La validazione dei risultati

I limiti di una procedura di collaudo, errori comuni, un approccio safe

Le tipologie del test di carico

- load test e stress test

Modalità di esecuzione dello stress test

Principali problemi sotto carico

- memory leak
- resource locking
- transactional limits
- bandwidth limits
- hardware bottlenecks
- software bottlenecks

Il processo di test per il tuning

- la progettazione del test
- la predisposizione degli ambienti
- la progettazione esecutiva
- l'esecuzione test e l'analisi risultati
- il tuning hardware/software
- la rendicontazione delle attività

Durata: 3 giorni

DB2-MVS: ARCHITETTURA E FUNZIONI

DESCRIZIONE

Il corso presenta l'architettura, i servizi e le principali funzioni del sistema, con l'obiettivo di fornire una panoramica completa e dettagliata sull'ambiente DB2-MVS.

Vengono esaminate in dettaglio le diverse componenti del sistema, e viene messo in evidenza l'inquadramento del DB2 nell'ambito dell'architettura MVS.

Vengono inoltre esaminate le problematiche dell'ottimizzazione e le possibilità offerte dal DB2 distribuito.

L'esposizione teorica è corredata da esercitazioni che favoriscono l'assimilazione dei concetti esposti.

PARTECIPANTI

- database administrator
- analisti
- analisti programmatori
- programmatori
- sistemisti

PREREQUISITI

Conoscenze di base sui sistemi relazionali.

PROGRAMMA

Caratteristiche e configurazione del sistema

Catalogo DB2

- organizzazione e utilizzo delle informazioni

Prestazioni e problematiche di ottimizzazione

Integrità fisica dei dati

- problematiche di backup/recovery

Componenti e funzionalità del DB2 distribuito

Organizzazione dei dati

- caratteristiche e utilizzo degli oggetti DB2

Integrità logica dei dati

Funzioni di utilità

Modifiche al disegno del database

Gestione della concorrenza

Riservatezza dei dati

Elementi relativi alle features delle nuove versioni

Durata: 3 giorni

DB2-MVS: SVILUPPO DI APPLICAZIONI ORIENTATO ALLE PERFORMANCE

DESCRIZIONE

Il corso ha come obiettivo quello di fornire ai partecipanti le conoscenze necessarie per sviluppare applicazioni DB2 ottimizzate in termini di prestazioni.

Vengono esaminati in dettaglio gli aspetti che influenzano le prestazioni delle applicazioni e gli interventi da effettuare per ottenere i risultati desiderati.

Vengono inoltre forniti criteri e norme per l'ottimizzazione delle query SQL e una chiave di lettura per valutare le scelte effettuate dall'ottimizzatore.

L'esposizione dei relatori è corredata da esercizi che affrontano problematiche di graduale complessità.

PARTECIPANTI

- database administrator
- analisti
- analisti programmatori
- programmatori

PREREQUISITI

Conoscenza dettagliata dell'SQL; partecipazione al corso "DB2 MVS: architettura e funzioni" o conoscenze equivalenti.

PROGRAMMA

Analisi delle problematiche e individuazione delle aree di intervento

Il processo di ottimizzazione

- componenti e fasi fondamentali
- tecniche di accesso ai dati
- analisi dei predicati

Analisi delle prestazioni

- analisi delle informazioni che compongono il piano di accessi
- correlazione tra informazioni statistiche e piano di accessi
- impatto delle informazioni statistiche contenute nel catalogo DB2 sull'ottimizzazione

Disegno dei dati

- modifiche da apportare al modello dei dati per ottimizzare le prestazioni

Utilizzo degli indici per migliorare le performance

Integrità logica dei dati

- implicazioni applicative e studio delle alternative

Linee guida per il disegno delle applicazioni DB2 distribuite e non distribuite

Utilizzo efficiente delle funzioni SQL

- criteri e norme per individuare l'alternativa più conveniente

Elementi relativi alle features delle nuove versioni

Durata: 3 giorni



ORACLE

ORACLE 10g: ARCHITETTURA E FUNZIONI

DESCRIZIONE

Lo scopo del corso è quello di fornire una visione il più completa possibile dell'architettura di un dbms Oracle e delle funzioni che mette a disposizione.

Viene fatta una panoramica sulla gestione delle reti e i parametri di configurazione, mentre vengono illustrate più in dettaglio le diverse componenti del sistema.

Sono inoltre esaminate le varie tecniche di Backup e Recovery e l'utilizzo di RMAN.

PARTECIPANTI

- database administrator
- analisti
- analisti programmatori
- programmatori
- sistemisti

PREREQUISITI

Conoscenze di base sui sistemi relazionali e sul linguaggio SQL.

PROGRAMMA

RDBMS

- introduzione (architettura e processi)
 - * Spfile
 - * OMF
 - * SQL e iSQL*PLUS
 - * Cenni su Oracle Enterprise Manager Grid Control

Panoramica e configurazione di Oracle Net

Dizionario Dati: organizzazione e utilizzo delle informazioni

Definizione di uno schema

- Edo Log File
- Datafile
- Tablespace
- Tablespace Sysaux
- Tablespace Undo
- Extend
- Segmenti e tipi
- Rollback Segment
- Undo Data
- Tabelle
- Indici
- Vista
- Materialized view
- Sequence

Partizionamento

- tipi di partizionamento: Range, Hash, List e Composite
- indici partizionati
- gestione delle partizioni

Componenti PL/SQL

- Function
- Procedure
- Package
- Trigger

Le estensioni Object Relational

Integrità referenziale

- Entity Integrity
- Referential Integrity
- Check Constraints
- Deferred Constraints

Gestione della concorrenza

- modalità, durata e tipologia di lock

Gestione della sicurezza

- utenti
- profili
- ruoli

Utility

- Data Pump Export
- Data Pump Import
- SQL *loader
- External Table

Backup e recovery

- gestioni dall'utente
- panoramica e Configurazione di RMAN

Durata: 3 giorni

INFORMAZIONI

MODALITÀ DI ISCRIZIONE

Il pagamento della quota, iva inclusa, dovrà essere effettuato tramite bonifico bancario presso la Banca Intesa Sanpaolo Ag. 6787 di Roma Codice IBAN: IT 34 Y 03069 05039 048890270110 intestato alla Technology Transfer S.r.l. e inviato insieme alla scheda di iscrizione a:

Technology Transfer S.r.l.
Piazza Cavour, 3 - 00193 Roma
Tel. 06-6832227 Fax 06-6871102

Le iscrizioni devono pervenire entro e non oltre 15 giorni prima della data di inizio del corso.

Vi consigliamo di far precedere la scheda di iscrizione da una prenotazione telefonica.

ISCRIZIONI MULTIPLE

Nel caso di iscrizioni multiple alla stessa edizione di un corso, verranno applicate le seguenti condizioni:

- I° e II° iscritto tariffa intera
- dal III° iscritto sconto del 15%

LINGUA

I corsi di questo programma saranno tenuti in lingua italiana.

LUOGO

I corsi si svolgeranno a Roma presso:
Hotel Visconti Palace, Via Federico Cesi, 37.

ORARIO

9.30-13.00 14.00-17.00

CONDIZIONI GENERALI

In caso di rinuncia con preavviso inferiore a 15 giorni verrà addebitato il 50% della quota di partecipazione.

In caso di rinuncia con preavviso inferiore ad una settimana verrà addebitata l'intera quota.

In caso di cancellazione del corso, per qualsiasi causa, la responsabilità della Technology Transfer si intende limitata al rimborso delle quote di iscrizione già pervenute.

Sostituzioni possono essere effettuate in qualsiasi momento solamente prima dell'inizio del corso. Poiché il numero dei partecipanti è limitato, le iscrizioni verranno accettate secondo l'ordine cronologico di arrivo.

CORSI "IN HOUSE"

Tutti i corsi elencati in questo programma possono essere effettuati, a richiesta, presso il cliente con modalità e costi da concordare.

Possiamo inoltre organizzare presso il cliente corsi, non presenti in questo programma, che affrontano importanti problematiche dell'Information Technology. Vi invitiamo pertanto a consultarci e a discutere con noi i Vostri piani di formazione.

SCHEDA D'ISCRIZIONE

TITOLO DEL CORSO

DATA

LUOGO

NOME

COGNOME

FUNZIONE AZIENDALE

AZIENDA

PARTITA IVA

CODICE FISCALE

INDIRIZZO

CITTÀ

CAP

PROVINCIA

TELEFONO

FAX

E-MAIL

TIMBRO E FIRMA

ISCRIZIONE MULTIPLE

Nel caso di iscrizioni multiple alla stessa edizione di un corso, verranno applicate le seguenti condizioni:

- I° e II° iscritto *tariffa intera*
- dal III° iscritto *sconto del 15%*

TUTELA DATI PERSONALI

Ai sensi dell'art. 13 della legge n. 196/2003, il partecipante è informato che i suoi dati personali acquisiti tramite la scheda di partecipazione al seminario saranno trattati da Technology Transfer anche con l'ausilio di mezzi elettronici, con finalità riguardanti l'esecuzione degli obblighi derivati dalla Sua partecipazione al seminario, per finalità statistiche e per l'invio di materiale promozionale dell'attività di Technology Transfer. Il conferimento dei dati è facoltativo ma necessario per la partecipazione al seminario. Il titolare del trattamento dei dati è Technology Transfer, Piazza Cavour, 3 - 00193 Roma, nei cui confronti il partecipante può esercitare i diritti di cui all'art. 13 della legge n. 196/2003.





TECHNOLOGY TRANSFER S.r.l.

*Piazza Cavour, 3 - 00193 Roma
Tel. 06-6832227 Fax 06-6871102
e-mail: info@technologytransfer.it*

www.technologytransfer.it