

Web Application Engineering

analisi del dominio

cristian lucchesi
IIT-CNR

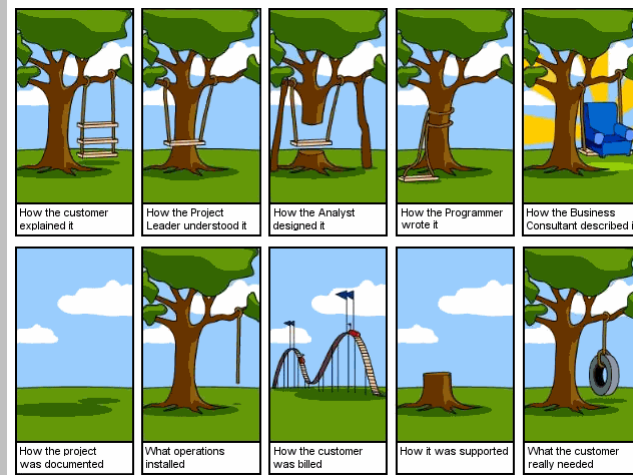
Pescara, 15-16 Maggio 2007
AleI – Ud'A



Analisi del dominio

- l'obiettivo è di arrivare alla definizione sufficientemente rigorosa dei requisiti di sistema
- si stabilisce cosa richiede il cliente ad un prodotto software (non come il prodotto verrà costruito)
- in genere, si analizzano prima gli aspetti più astratti poiché errori nelle astrazioni si propagano nelle soluzioni più concrete

L'altalena dei Requisiti



altre altalene

Alei/Ud'A - Pescara, 15-16 maggio 2007 - cristian lucchesi, IIT-CNR

3

Analisi del dominio: approcci di analisi

- gli approcci sono suddivisi in tre categorie principali

approccio informale

nessun modello del sistema viene costruito (uso di un linguaggio informale)
raccolta di informazioni attraverso:

- riunioni tra gli utenti/committenti ed analisti
- l'uso di questionari
- studio di documentazione esistente

si procede attraverso la formulazione e il successivo affinamento di vari documenti sottoposti a convalida

approccio basato sulla prototipazione

il problema viene analizzato ed i requisiti sono compresi grazie all'uso di un prototipo del sistema da parte di cliente ed utenti

approccio basato su modelli concettuali

produce rappresentazioni del dominio applicativo e del sistema

Alei/Ud'A - Pescara, 15-16 maggio 2007 - cristian lucchesi, IIT-CNR

4

Alcuni approcci basati su modelli concettuali

■ Analisi Strutturata

- basata sull'uso di Data Flow Diagrams e Dizionario dei Dati
- l'analisi del problema viene eseguita usando l'approccio della decomposizione delle funzioni - i dati e le relative relazioni sono modellati con linguaggi diversi (es. modello Entità-Relazioni)

■ Analisi Object-Oriented

- l'analisi del problema viene eseguita usando l'approccio della decomposizione in oggetti (entità/concetti del dominio del problema)
- si usano linguaggi di modellazione, es. UML

Modelli concettuali

- ogni modello può rappresentare un certo livello di dettaglio al fine di permettere successive analisi più specifiche
- si costruiscono modelli per gli aspetti
 - funzionale (diagrammi di attività, casi d'uso, diagrammi di sequenza e di collaborazione)
 - statico ovvero strutturale (diagramma delle classi e degli oggetti)
 - dinamico per classi di oggetti (statechart)

Analisi tramite modello concettuale

- L'analisi del dominio si distingue in tre fasi metodologiche:
 - Concettualizzazione
 - Analisi degli oggetti
 - Analisi della dinamica

Analisi tramite modello concettuale

cont.

Relazione tra fasi metodologiche:

- normalmente gli aspetti temporali sono i più difficili da modellare
- di conseguenza è spesso preferibile esaminare inizialmente gli aspetti statici di un sistema (cioè la struttura degli oggetti e le associazioni tra di essi)
- successivamente si analizza l'evoluzione temporale di oggetti e associazioni
- gli aspetti di un sistema che riguardano il tempo e l'evoluzione costituiscono il modello della dinamica del sistema

Analisi del dominio: concettualizzazione

- studio del dominio del problema per delimitarne i contorni ed evidenziare le realtà di interesse (Requisiti di Contesto)
 - possono essere stese, in linguaggio naturale, le specifiche del dominio: Problem Statement
 - si possono eseguire delle interviste con gli stakeholder per acquisire le specifiche del dominio - le specifiche possono tradursi in prototipi, diagrammi UML e/o casi d'uso

Problem statement

- regole empiriche per la stesura di un Problem Statement corretto
 - la sua formulazione deve stabilire cosa deve essere fatto e non come (specifica delle esigenze, non una soluzione proposta)
 - brevità del documento
 - buona capacità di astrazione
 - evitare, per quanto possibile, le ambiguità
 - considerare il fatto che è attraverso i diagrammi UML che i requisiti dovrebbero essere sviluppati

Scenari

- nelle agile development si preferisce l'utilizzo della comunicazione verbale con la definizione dei casi d'uso
 - gli stackholder presentano gli obiettivi da raggiungere descrivendo una serie di scenari
 - gli sviluppatori possono preparare un prototipo per discutere in maniera più approfondita dello scenario
 - può essere preparato un caso d'uso che descrive lo scenario

Casi d'uso

- un caso d'uso rappresenta una sequenza di passi che descrivono l'interazione tra un sistema ed uno o più utenti
- si tratta, essenzialmente, di una tecnica per:
 - scoprire
 - chiarire
 - concordare
- i requisiti di un sistema.

Casi d'uso

cont.

- usare i casi d'uso significa:
 - individuare chi dovrà utilizzare il sistema
 - chiedersi quali sono gli obiettivi che si intendono conseguire utilizzando il sistema
 - approfondire, in termini di descrizione di scenari concreti, ciascuna modalità di utilizzo, chiarendo:
 - il modo in cui inizia
 - le risposte che l'utilizzatore si attende dal sistema
 - la sequenza di passi con cui l'interazione si svolge
 - eventuali altri soggetti (esterni al sistema) coinvolti

Struttura dei casi d'uso

- Titolo
- Sommario
- Attori
- Precondizioni
- Descrizione (passi principali)
- Alternative (passi alternativi)
- Postcondizioni
- non esiste un modo univoco per descrivere i casi d'uso, tipicamente si tratta di una descrizione testuale dei passi compiuti dall'utente
- molto spesso per i casi d'uso di utilizza una struttura del tipo:

Esempio di caso d'uso

Titolo:

validazione PIN del bancomat

Sommario:

il sistema controlla e convalida il PIN del client

Attori:

cliente ATM

Precondizioni:

il sistema è in attesa e mostra sul display un messaggio di benvenuto

Esempio di caso d'uso

Descrizione:

1. l'utente inserisce il bancomat
2. il sistema riconosce il bancomat e ne legge il numero
3. il sistema chiede il PIN
4. il client inserisce il PIN
5. il sistema controlla la data di scadenza e se la Card risulta in stato PrelievoContanti di smarrita o rubata
6. se la Card è valida, il sistema controlla se il PIN è corretto
7. se il PIN è corretto, il sistema controlla quali conti correnti sono accessibili
8. il sistema mostra al cliente le possibili transazioni (menu): prelievo, saldo, lista movimenti o trasferimento

Alternative:

1. il sistema non riconosce la Card, la Card viene espulsa
2. il sistema determina che la Card è scaduta: la Card viene confiscata
3. il sistema determina che la Card risulta smarrita o rubata: la Card viene confiscata;
4. il cliente digita un PIN non corretto...

Postcondizioni:

il PIN è stato validato

Casi d'uso, analisi e design

Utilizzare i casi d'uso permette di:

- ragionare con i committenti (e le altre parti interessate al sistema, come gli utilizzatori) in termini di casi d'uso (cioè di "storie" concrete di utilizzo)
- agevola notevolmente la scoperta dei requisiti ed il loro progressivo chiarimento
- costituisce un ottimo punto di partenza per le attività di analisi e design, implementazione e test del sistema.

Prototipi e casi d'uso

- per ogni caso d'uso avviene un'interazione tra attori e sistema
- l'interazione avviene per gli attori umani tramite le interfacce utente
- il prototipo può essere definito in parallelo ai casi d'uso
 - sono forme complementari di rappresentazione di uno scenario di interazione
 - il prototipo aiuta a chiarire la sequenza dei casi d'uso

Riferimenti

- **www.analisi-disegno.com**
materiale su gestione dei requisiti, casi d'uso, gestione dei progetti software, UML, ...
- Best Practice Software Engineering:
<http://best-practice-software-engineering.ifs.tuwien.ac.at/>

**grazie per
l'attenzione**