

Sistemi Informativi:

La determinazione dei requisiti

Sommario

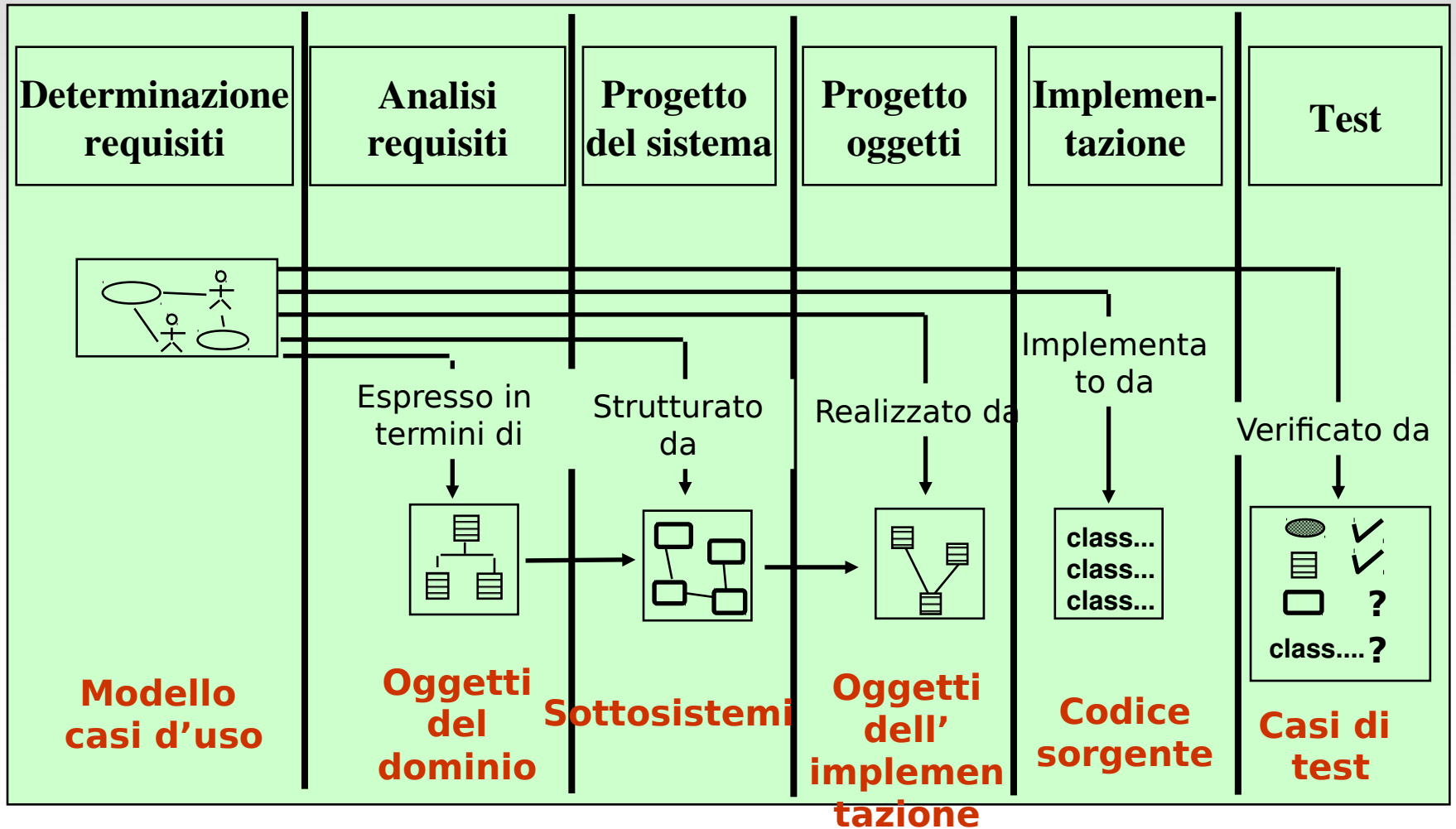
Che cosa sono i requisiti

Come ottenere i requisiti

Validazione dei requisiti

Documento dei requisiti

attività' del ciclo di vita SW



Determinazione requisiti

identificazione di

- attori
- scenari
- casi d'uso

relazioni tra casi d'uso

raffinamento casi d'uso

identificazione di

- requisiti non funzionali
- oggetti partecipanti

identificazione del sistema

un sistema non e' definito solo da un'istantanea del suo dominio

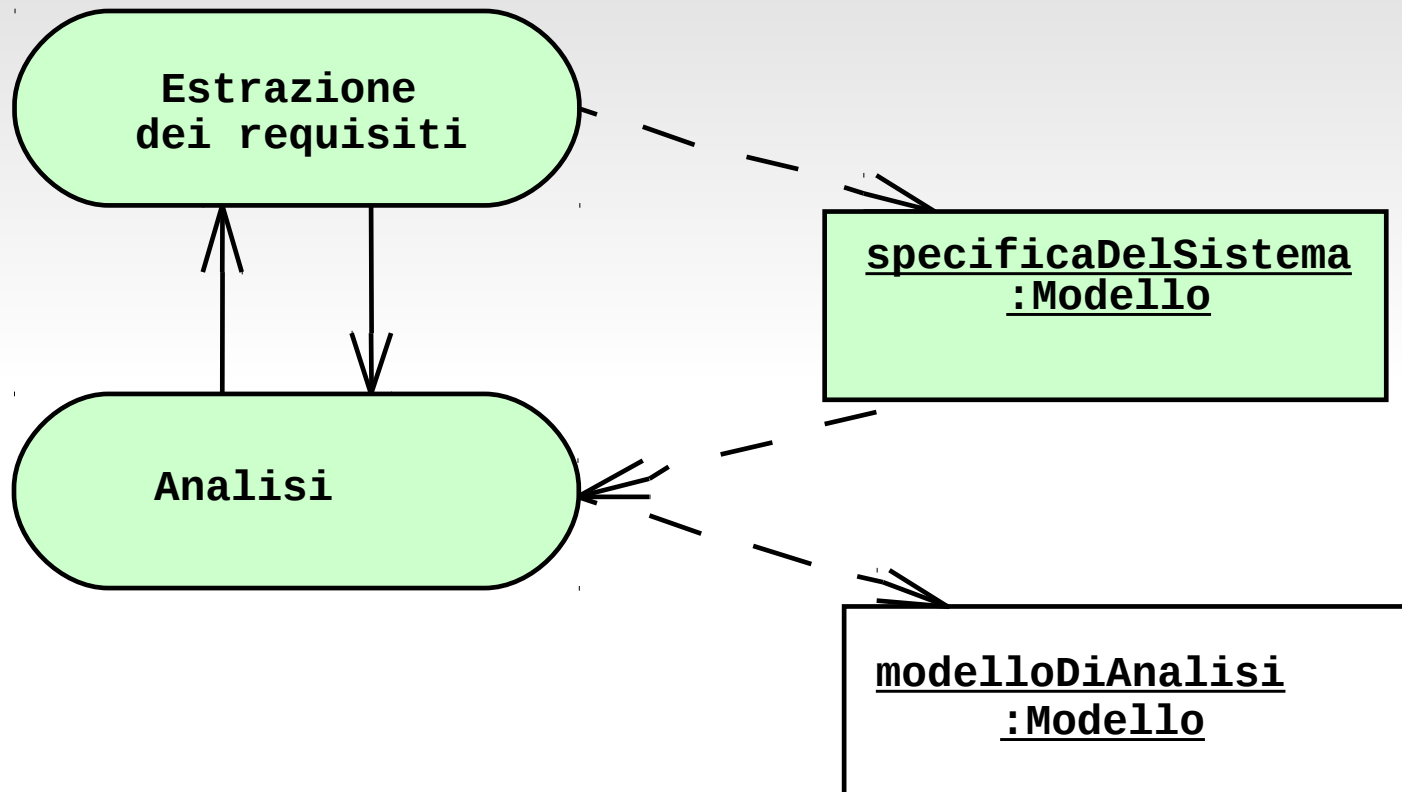
problemi importanti:

- identificare l'ambito di un sistema
 - casi d'uso e scenari
- identificazione degli oggetti:
 - a seconda dell' ambito del sistema , si possono individuare oggetti diversi:
- contesto di un sistema: un sistema interagisce o dipende quasi sempre da altri sistemi o dati preesistenti
- A volte sw progettato per sola migrazione dati

Il processo dei requisiti:

- estrazione dei requisiti: definire il sistema in termini comprensibili al cliente, soprattutto il linguaggio naturale
- analisi: specifica tecnica del sistema in termini comprensibili allo sviluppatore

prodotti del processo dei requisiti



Specifica del sistema vs. modello analitico

la specifica del sistema usa il linguaggio naturale (derivato dalla descrizione del problema che il sistema risolve)

il modello analitico usa un linguaggio formale o semi-formale (es UML)

Importanza dei requisiti

I requisiti riguardano quello che i partecipanti vogliono ottenere dal sistema
una scadente determinazione dei requisiti e' la **principale** causa del fallimento di un sistema
la seconda e' la scarsa partecipazione dell'utente

Ottenimento dei requisiti

attività impegnativa

richiede partecipanti con competenze diverse:

- utenti con conoscenza del dominio applicativo
- sviluppatori con conoscenza del dominio implementativo

come colmare il divario tra utente e sviluppatore?

- scenario (utente): esempio dell'uso del sistema in termini di una serie di interazioni tra utente e sistema
- caso d'uso (analista): astrazione che descrive una classe di scenari

» Un bravo analista cerca sempre di razionalizzare, astrarre e semplificare la descrizione dell'utente

Tipi di requisiti

Funzionali: descrivono interazione tra ambiente e sistema prescindendo dall'implementazione

- il bancomat deve mostrare la massima quantità prelevabile

Non Funzionali: aspetti visibili all'utente non derivabili direttamente dai casi d'uso

- tempo di risposta < 10 sec

Vincoli: imposti dall'utente o dall'ambiente operativo

- deve essere scritto in Javascript

Prima scriverli, **poi** catalogarli

fuori dai requisiti

Solitamente:

- struttura del sistema, tecnologia implementativa
- metodologia e ambiente di sviluppo
- linguaggio implementativo
- riusabilita'

sforzarsi il piu' possibile di **eliminare** i punti precedenti dai requisiti

- E' il Criterio di Separation Of Concerns (SOC)

Validazione dei requisiti

Passo critico:

- di solito dopo la analisi o la ingegnerizzazione dei requisiti; anche alla consegna

Proprietà' dei requisiti:

- correttezza: rappresentano la visione del cliente
- completezza: **tutti** gli scenari possibili sono contemplati, inclusi comportamenti anomali del sistema e dell'utente
- consistenza: nessun requisito e' in contraddizione con un altro
- chiarezza: non vi sono ambiguita'
- realismo: sono realizzabili e consegnabili
- tracciabilita': ogni funzione del sistema e' riconducibile ad un insieme di requisiti (la facciamo perche' serve)

Evoluzione dei requisiti

I requisiti cambiano rapidamente durante la loro definizione:

Tools SW appositi permettono di:

- creare un archivio con accesso multi utente
- creare automaticamente il documento dei requisiti
- tracciare e modificare i requisiti durante tutto il progetto

Dopo che un requisito e' stato approvato, un suo cambiamento potrebbe richiedere una approvazione formale da parte del management

- Piu' passa il tempo piu' costa cambiare un requisito

Tipi di progetto

Sviluppo tutto nuovo (greenfield engineering):

- sviluppo “from scratch”, nessun Sistema pregresso, requisiti estratti dai clienti ed utenti
- Sollecitato dai bisogni degli utenti

Reingegnerizzazione

- riprogettare o implementare un vecchio sistema con nuova tecnologia
- sollecitato dal gestore delle tecnologie

Ingegnerizzazione di una interfaccia

- fornitura dei servizi di un vecchio sistema in un nuovo ambiente
- sollecitato dal gestore delle tecnologie o da nuovi bisogni del mercato

Scenario

“A narrative description of what people do and experience as they try to make use of computer systems and applications” [M. Carrol, Scenario-based Design, Wiley, 1995]

una descrizione concreta e circoscritta di una funzione del sistema usata da un singolo utente utilizzabile in vari momenti del ciclo di vita

Caso d'uso

- a use case describes the system's behavior as it responds to a series of related requests from an actor.
- a use case is a list of action or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. The actor can be a human, an external system, or time.

Perche' scenari e casi d'uso

gli scenari sono comprensibili all'utente,
modellano il sistema dal suo punto di vista
(requisiti funzionali)

casi d'uso possono formare la base per lo
sviluppo di tutto il sistema:

- manuale utente
- progetto del sistema e degli oggetti
- implementazione
- specifica dei test
- test di accettazione del cliente

Perche' scenari e casi d'uso

In tempi passati il progettista, ascoltando il cliente, concepiva un modello astratto di funzioni la cui combinazione avrebbe soddisfatto I requisiti richiesti.

Non e' detto che questa realizzazione, anche se tecnicamente corretta, avrebbe prodotto un sistema usabile e comprensibile all'utente.

I casi d'uso obbligano a creare un sistema che si comporti secondo le aspettative dell'utente

Tipi di scenario

caso normale (as-is):

- di solito nella reingegnerizzazione, l'utente descrive il funzionamento del sistema

visione:

- il sistema futuro, in nuovi sviluppi o reingegnerizzazione, spesso realizzato in collaborazione tra utente ed sviluppatore

Addestramento/tutorial:

- istruzioni passo dopo passo per un utente nuovo del sistema

Come trovare gli scenari?

non pensate che l'utente ne parli se il sistema è un nuovo sviluppo

approccio incrementale, evolutivo

- aiutare il cliente nella formulazione dei requisiti
- farsi aiutare dal cliente a capire i requisiti
- I requisiti evolvono durante lo sviluppo degli scenari

tecniche per trovare gli scenari

chiedere a se' o al cliente:

- quali sono I compiti principali del sistema?
- Quali dati un attore crea , modifica, aggiunge o rimuove dal sistema?
- Di quali cambiamenti esterni deve essere informato il sistema?
- di quali eventi del sistema bisogna informare l'attore?

Osservazione dell'attività dell 'utente (sistema preesistente)

- parlare direttamente all'utilizzatore,
- aspettarsi resistenze e superarle

Riassumendo

estrazione dei requisiti:

- nuovo sviluppo, reingegnerizzazione, nuova interfaccia

scenario

- modo di comunicare con l'utente
- normale, visione, valutazione, addestramento
- caso d'uso: astrazione di scenario; lo scenario rappresenta un singolo cammino all'interno di un caso d'uso

corretto modo di analisi:

- partire da casi d'uso e trovare gli oggetti coinvolti
- alla domanda: “che cos'è questo oggetto”? rispondere qual'è la sua funzione

I requisiti nella fase iniziale del progetto

Estrazione dei requisiti

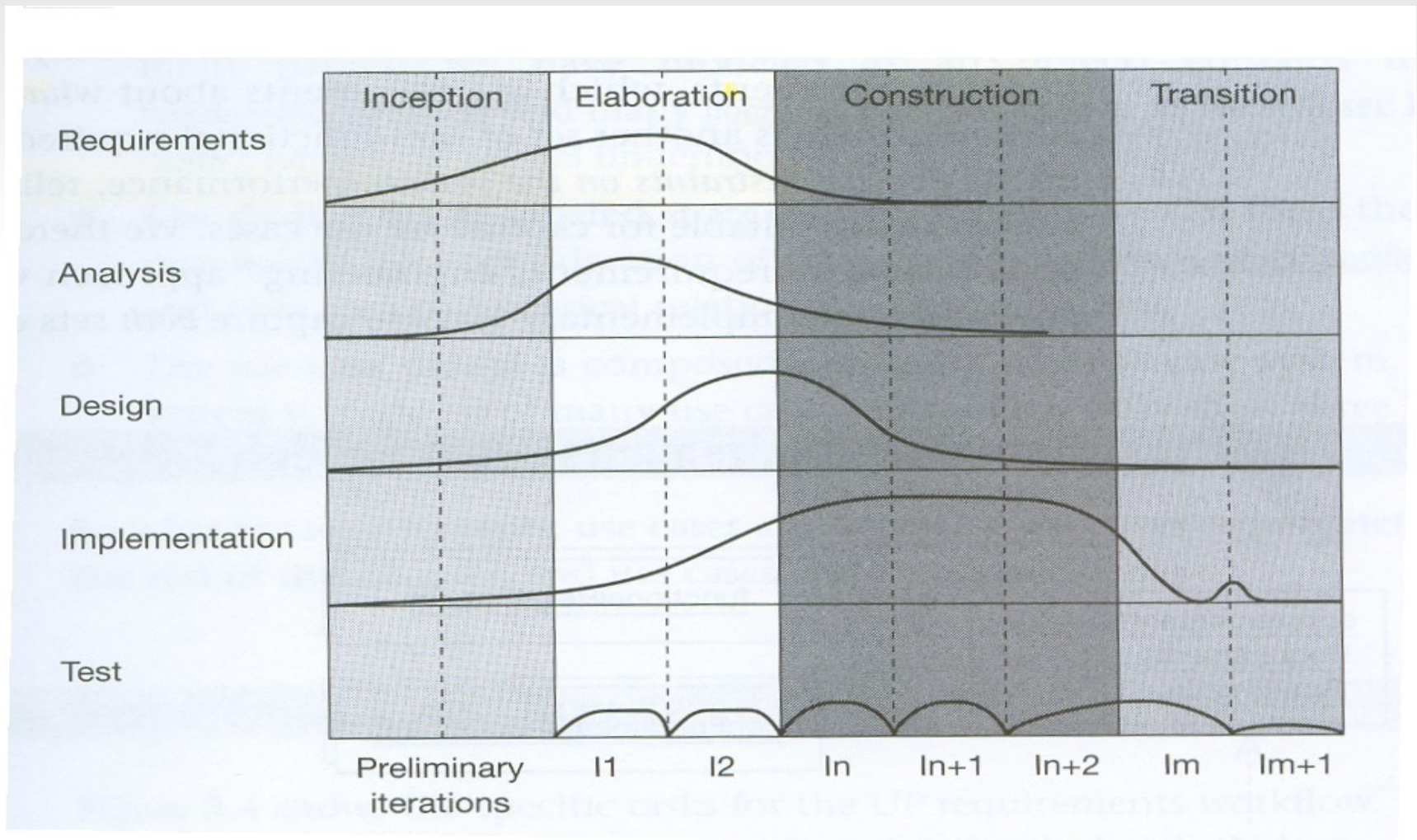
Gestione dei requisiti

Casi d'uso

Modello ad alto livello dei requisiti

Documento dei requisiti

Gran parte del lavoro sui requisiti avviene nelle fasi iniziali (Arlow..)



Ruoli nell'estrazione dei requisiti

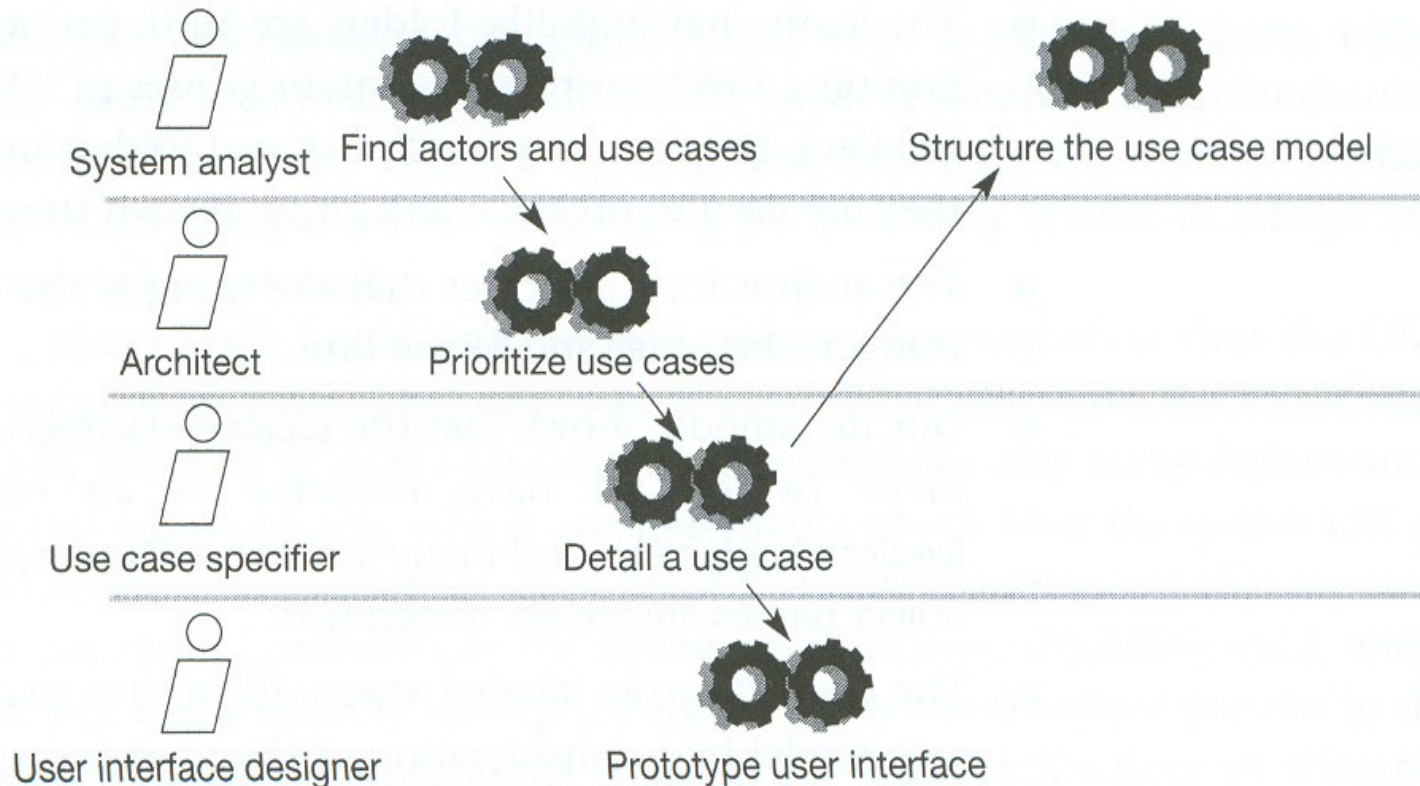
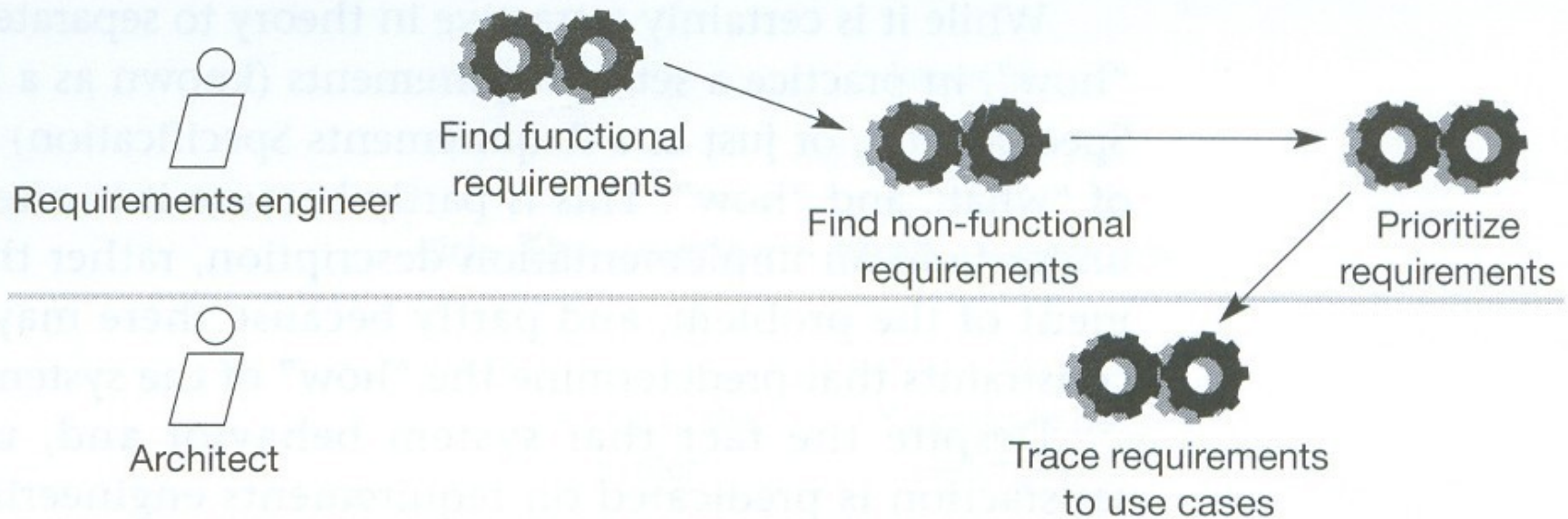
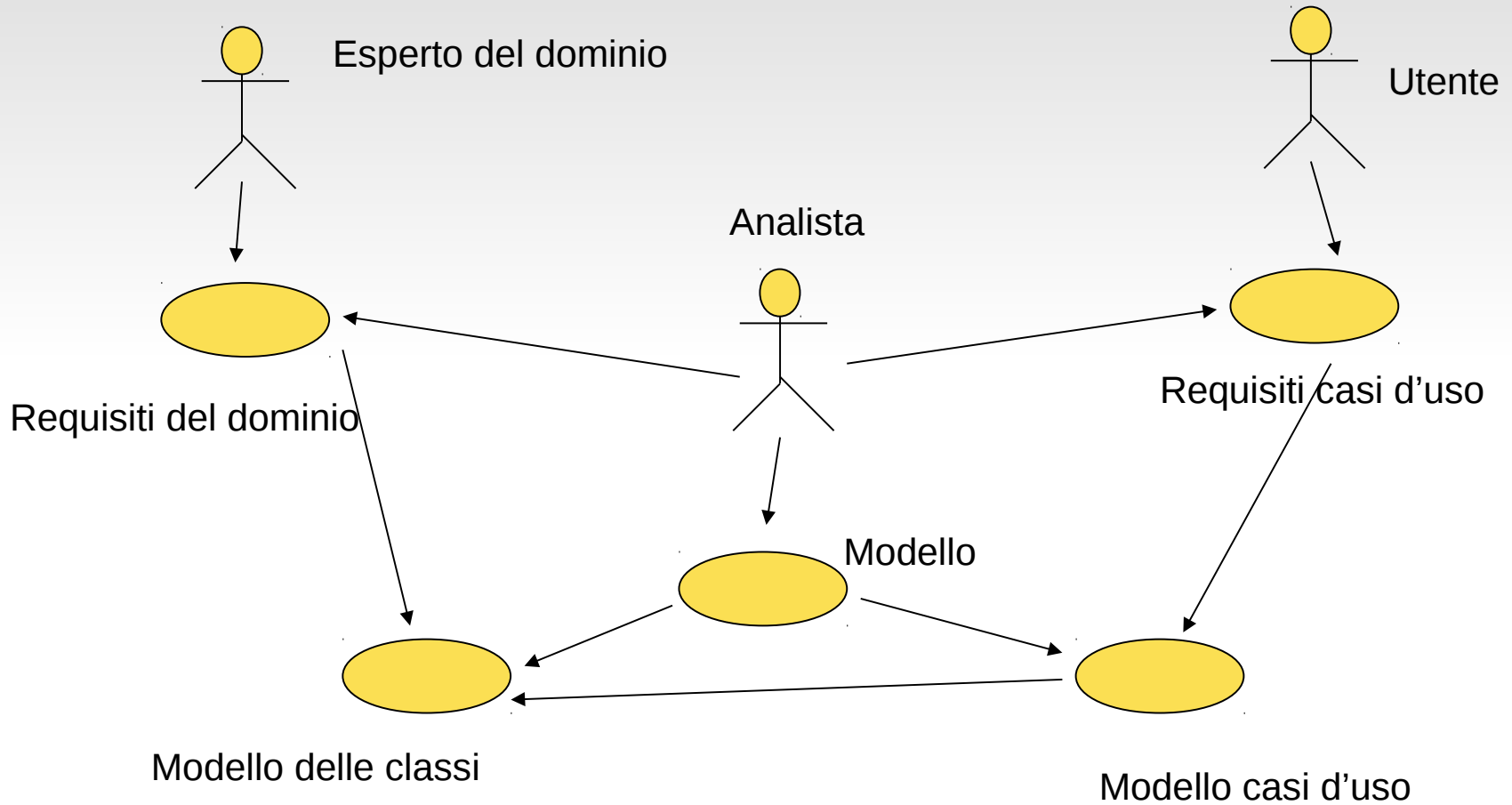


Figure 3.4 Reproduced from Figure 7.10 [Jacobson 1] with permission from Addison-Wesley

Altri compiti: trovare requisiti funzionali e non , prioritizzare, tracciare i requisiti ai casi d'uso (Arlow...)



Ottenimento dei requisiti



Metodi tradizionali di estrazione dei requisiti

- Intervistare utenti ed esperti del dominio
- L'intervistato potrebbe non essere in grado di collaborare o volerlo
- Spedire all'intervistato il documento dell'intervista subito dopo
- Questionari
- Osservazione
- Studio dei documenti e dei sistemi software

Trovare i requisiti

Conduct a brainstorming session

Interview users

Send questionnaires

Work in the target environment

Study analogous systems

Examine suggestions and problem reports

Talk to support teams

Study improvements made by users

Look at unintended uses

Conduct workshops

Demonstrate prototypes to stakeholders

Intervistare utenti ed esperti del dominio

Intervista strutturata

- Domande aperte
- Domande chiuse (a scelta fissa)

Intervista libera

Come ben si sa, evitare domande che:

- Esprimano l'opinione dell'intervistatore
- Privilegino specifiche risposte
- Contengano la risposta nella domanda

Questionari

In aggiunta alle interviste

Domande a risposta predefinita:

- A scelta multipla
- Di valutazione (rating)
- Di ordinamento (ranking)

Non e' detto che le persone rispondano allo stesso questionario sempre nello stesso modo

in quali casi questionari chiusi potrebbero essere utili?
(votazione)

Osservazione dell' utente

- Passiva
- Attiva
- Prolungata nel tempo
- Gli individui tendono a comportarsi in modo differente quando si sentono osservati

Studio dei documenti e dei sistemi software

Requisiti dei casi d'uso

- Documenti dell'organizzazione
- Moduli e relazioni relative al sistema

Requisiti del dominio

- Libri e riviste
- Documentazione di sistemi software

Prototipazione

Prototipo usa e getta

- Importante per l'interfaccia utente

Prototipo evolutivo

- Meno usato, costoso da modificare, crea sistemi limitati

Metodi recenti di estrazione dei requisiti

Prototipazione

Joint Application Development – sviluppo congiunto dell'applicazione (JAD)

Rapid Application Development – sviluppo rapido dell'applicazione (RAD)

In generale piu' costosi, per progetti ad alto rischio

Joint Application Development

“Un gruppo lavora meglio di tanti singoli”

– Partecipanti

- Leader
- Segretario
- Clienti:
 - Utenti
 - Managers
- Sviluppatori: raccolgono informazioni

Rapid Application Development

Produrre velocemente una soluzione, anche se non perfetta

– 5 strumenti

- Prototipo evolutivo
- Strumenti di CASE
- Specialists With Advanced Tools (SWAT) - i migliori disponibili
- Sessione JAD: segretario il gruppo SWAT
- Tempo contingentato: piuttosto si tagliano funzionalità' (analogia con extreme programming)

Rapid Application Development

Svantaggi (presto e bene

- GUI inconsistente
- Soluzione specifiche anziche' generali
- Poca documentazione
- Difficolta' di manutenzione ed estensione

Negoziazione e validazione

Condotte in parallelo alla loro scoperta

Requisiti fuori dall'ambito del sistema:

- Perché fuori dai vincoli di budget o tempo
 - → Release successive
- Perché in conflitto o sovrapposti ad altre realtà preesistenti

Importanza/priorità dei requisiti (indispensabile, ..., immediato...)

Tabella di dipendenza dei requisiti

Tabella di dipendenza tra requisiti

Requisito	R1	R2	R3
R1	X	X	Conflitto
R2	X	X	X
R3		Sovrapposizione	X

Rischi associati ai requisiti

- Tecnico (es difficile da implementare)
- Prestazionali (es rallenta il sistema)
- Sicurezza (es espone dati sensibili)
- Integrità' della base di dati (es di difficile validazione)
- Processo di sviluppo (es uso di tools poco affidabili)
- Politici/Legali (cambiano le leggi)
- Volatilità' (cambia frequentemente)

Gestione dei requisiti

Identificazione e classificazione dei requisiti

- Numero sequenziale
- Numero di riferimento al paragrafo del documento
- Numero di riferimento all'interno della categoria
- ID generato da Base Dati

Gerarchie di requisiti

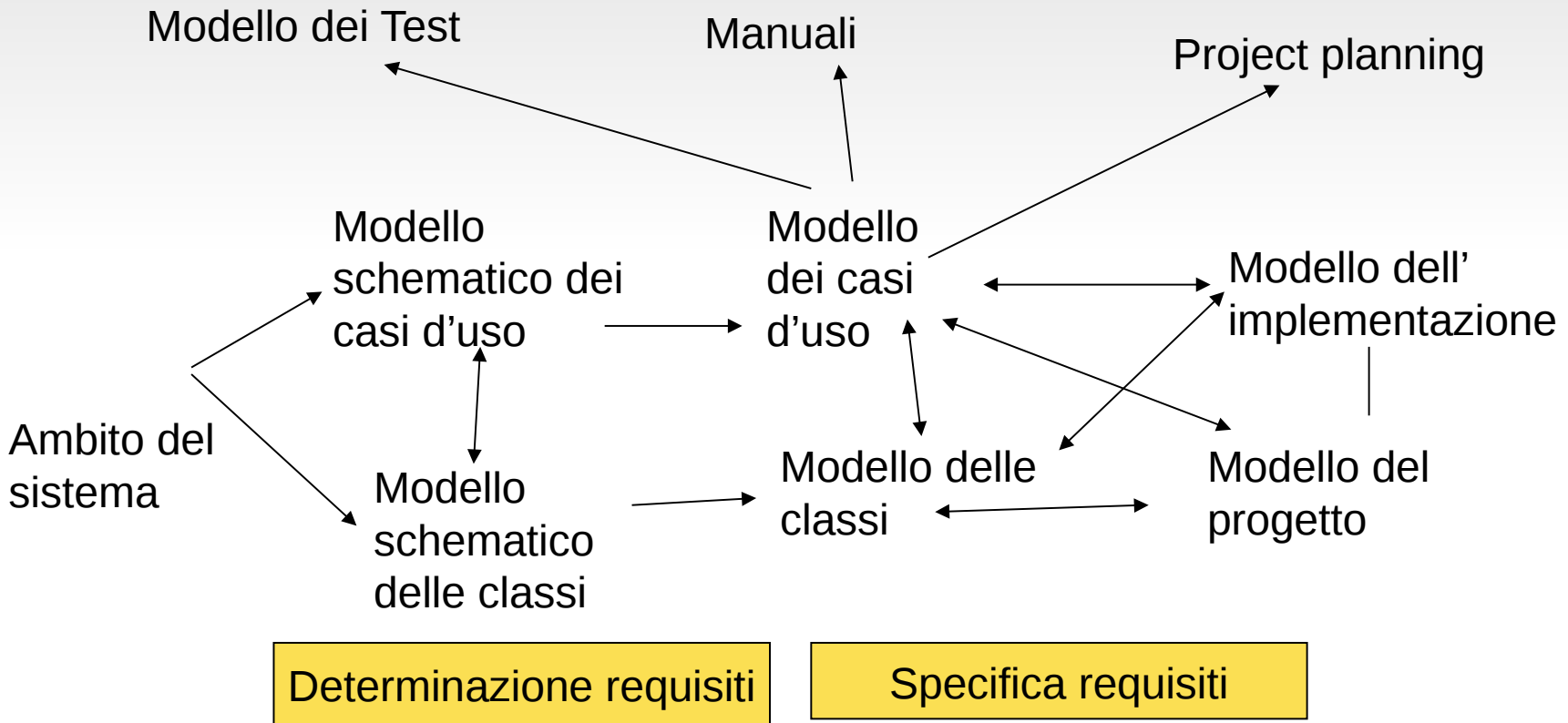
Gestione dei cambiamenti (processo di approvazione)

Tracciabilità dei requisiti rispetto a tutti gli altri elementi del progetto (requisiti inclusi)

Gerarchie di requisiti / livello di astrazione

- 1 “modalita’ della consegna”
 - »1.1 ritiro in sede
 - »1.2 spedizione
- 2 “costo della consegna ”
 - »2.1 se l’importo delle merci e’ \geq 200 Euro la spedizione (R1.2) e’ gratuita
 - »2.2 se l’importo e’ $<$ 200 Euro il costo della spedizione (R1.2) e’ 7 Euro
 - »2.3 per ritiro in sede (R1.1) il costo e’ 0

Modello schematico, ad alto livello (business model) dei requisiti



Relazioni tra requisiti e casi d'uso

	C1	C2	C3	C4
R1	X		X	
R2		X		
R3				X
R4	X			

problem statement: telemarketing

Una società di beneficenza vende biglietti di lotteria per raccogliere finanziamenti attraverso campagne per scopi benefici. La società mantiene un elenco di sostenitori da contattare ad ogni campagna. Per supportare i venditori nelle telefonate occorre un sistema che consenta di organizzarle secondo date e priorità. Le chiamate a vuoto devono essere ritentate, come peraltro occorre prevedere chiamata e risposta. Occorre registrare inoltre l'esito della telefonata ed il numero di biglietti venduti.

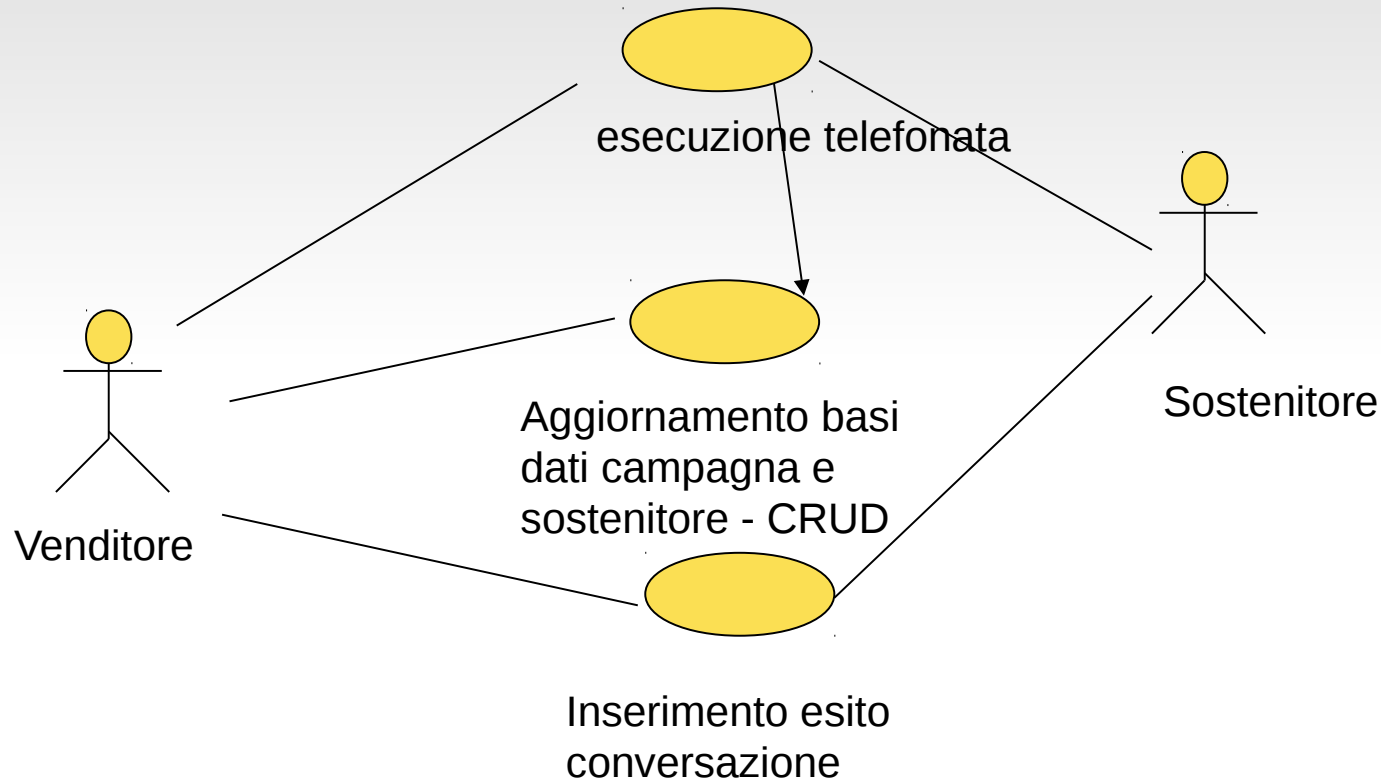
Esempio di telemarketing

- Telemarketing
 - Le campagne sono pianificate su richiesta del comitato etico della società
 - Ogni campagna deve essere approvata dall'amministrazione locale
 - La pianificazione e progettazione delle campagne è supportata da una base dati delle campagne separata
 - Esiste anche una base dati dei sostenitori da selezionare durante le campagne
 - Gli ordini dei biglietti della lotteria sono registrati per l'uso da parte del sistema di gestione degli ordini
 - Il sistema di gestione degli ordini conserva lo stato degli ordini nella base dati dei sostenitori

Diagramma di contesto del sistema



Modello schematico dei casi d'uso



Altro esempio attori e casi d'uso

diagramma che illustra i confini del sistema, i casi d'uso, gli attori

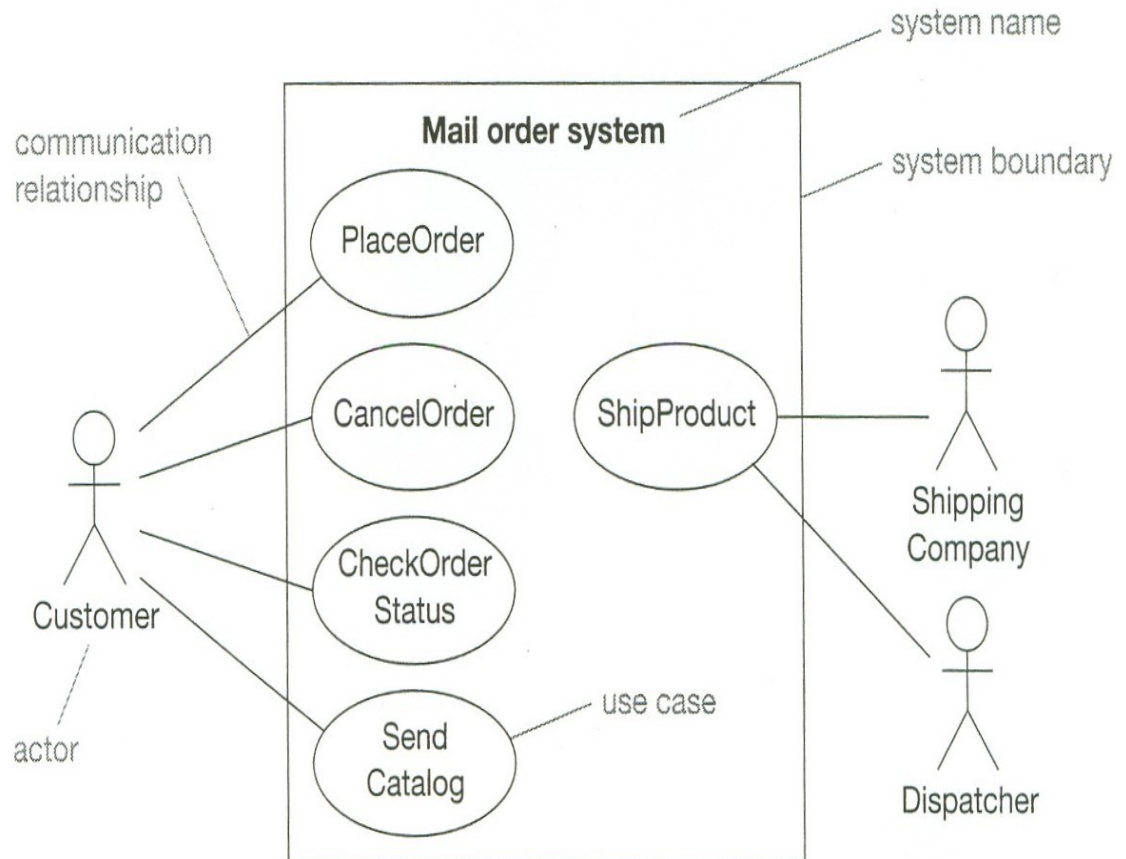
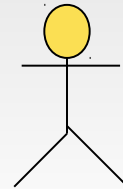
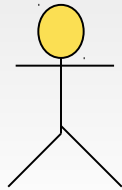


Figure 4.6

domanda

come ripartire i seguenti casi d'uso?

cliente



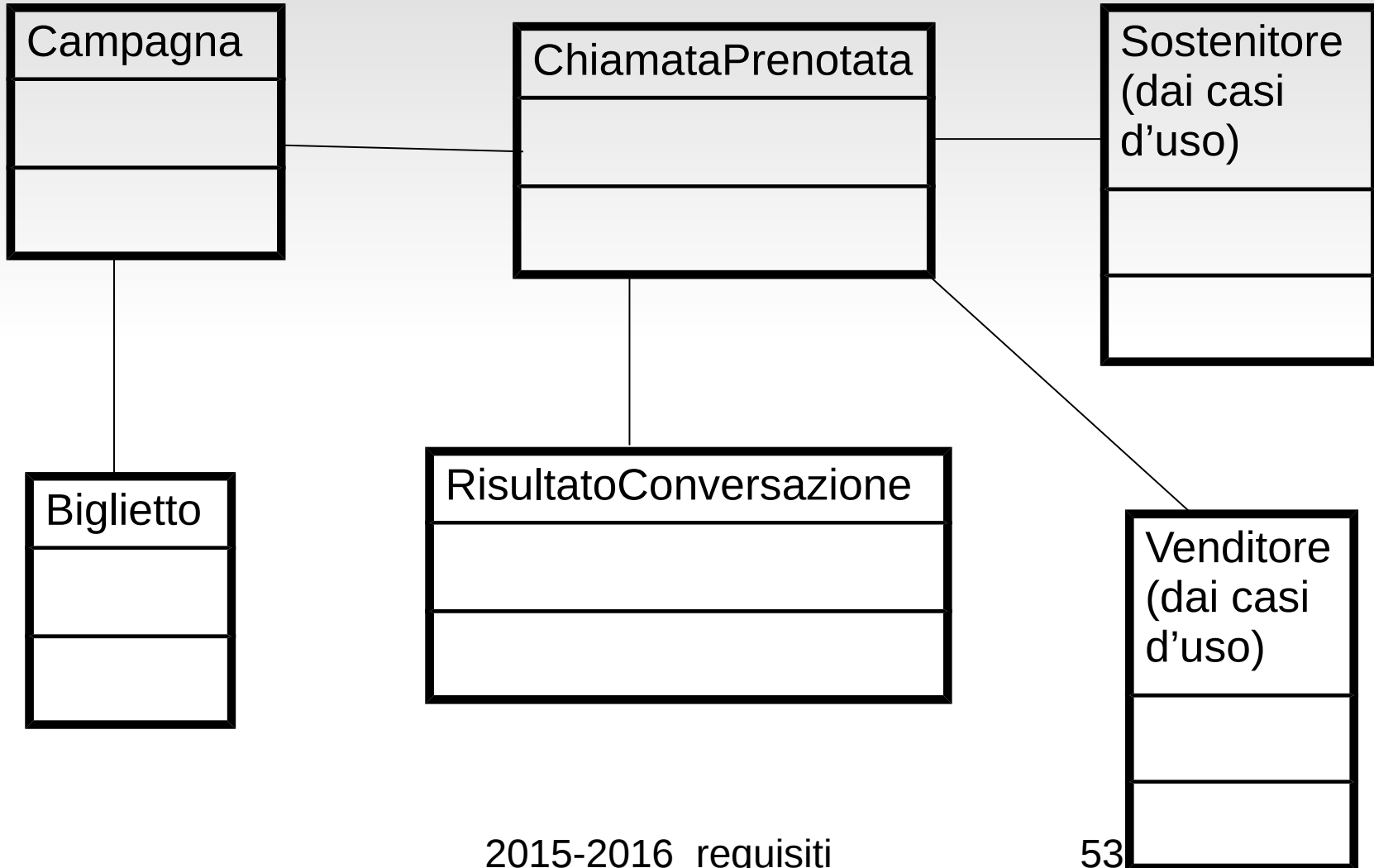
impiegato

prelievo bancomat

chiudere conto

aprire conto corrente

Modello schematico delle classi telemarketing



Requisiti non funzionali

fattori umani e di interfaccia

documentazione

hardware

prestazioni

errori e condizioni al limite

interfaccia di sistema

qualita'

modifiche al sistema

ambiente fisico

sicurezza

risorse e gestione

Domande

Interfaccia utente e fattori umani

- tipi di utente
- loro addestramento
- il sistema deve essere facile da apprendere?
- e' importante che l'utente non faccia errori?
- che strumenti per I/O per l'utente?

Documentazione

- che tipo e che pubblico per essa

Hardware

- che tipo, caratteristiche memoria e velocita'

Domande (2)

prestazioni

- limiti imposti di velocità o quantità di lavoro svolto?
- limiti sulla quantità di dati?

errori e condizioni limite

- risposta ad errori in input?
- risposta a condizioni estreme

interfacce di sistema

- l'input/output provengono da altri sistemi?
- restrizioni su formato e supporto di input/output

Domande (3)

- qualita'
 - requisiti di affidabilita'
 - quanto tempo on line?
 - massima pausa accettata
 - portabilita' ad altri ambienti
- modifiche
 - componenti candidati a modifiche future
 - modifiche previste
- ambiente fisico
 - ambiente in cui le macchine e persone lavoreranno
 - una o piu' localita'?
 - Condizioni estreme di temperatura, campi magnetici..?

Domande (4)

Sicurezza

- Controllare/restringere l'accesso ai dati?
- E' un problema la sicurezza fisica?

Risorse e gestione

- frequenza del salvataggio
- responsabilita' del salvataggio
- chi installa il sistema
- chi lo mantiene

Vincoli

Vincolo

- qualunque restrizione del cliente all'insieme delle possibili soluzioni

Esempio

- la piattaforma deve essere .NET
- Linguaggio Java
- Usare lo standard di documentazione “X”
-

Classificazione requisiti FURPS

Functionality - Feature set, Capabilities, Generality,
Security

Usability - Human factors, Aesthetics, Consistency,
Documentation

Reliability - Frequency/severity of failure, Recoverability,
Predictability, Accuracy, Mean time to failure

Performance - Speed, Efficiency, Resource consumption,
Throughput, Response time

Supportability - Testability, Extensibility, Adaptability,
Maintainability, Compatibility, Configurability,
Serviceability, Installability, Localizability,

Possibili problemi

Mentalita' “carrello della spesa”

- Il prodotto deve essere localizzato?
 - Si, dovremmo pensare ai mercati stranieri
- Riguardo la sicurezza?
 - Certo, il prodotto deve essere sicuro
- Esigenze di affidabilita?
 - Si, 24x7, cosi' batteremo la concorrenza

Altre liste di requisiti

Vedi documenti sul sito (tratti da IBM ed altri documenti di progetto)

Distinguere fra:

- requisiti funzionali (che l'utente dovrebbe conoscere)
- Requisiti **architetturali** (SW) che l'utente potrebbe non conoscere

Requisiti architetturali

Dato un insieme di requisiti, il progettista crea una soluzione SW che li soddisfa

L'architettura della soluzione descrive quali sono I **componenti** , su quali macchine risiedono, come collaborano

Esempio: architettura client-server

Metodo per requisiti architettonici

Mantenere sempre una lista completa (anche quelli magari irrilevanti)

Per ognuno creare domande comprensibili ai clienti

Aiutarli con esempi di risposte ed il loro significato

Acquisire le risposte

Aiutare i clienti ad assegnare loro priorità per compromessi fra richieste in conflitto

Possibili problemi

Questo e' troppo tecnico per me

- Far notare se mancasse la risposta

Tutti i requisiti sono uguali

- Alta e bassa prioritá'

Raccolti e parcheggiati da qualche parte

- Farne capire l'importanza

Il requisito non puo' essere misurato

- Specificare il meglio possibile

Possibili problemi

Non c'e' tempo sufficiente

- Importanza per il progetto

Mancanza di proprietario del documento

- Collaborazione tra analista ed architetto

Tutti gli “Stakeholder” sono uguali

- Domande adatte ad ogni ruolo

Requisiti troppo generali

- Non esagerare nell'ingegnerizzazione

Accordo sul progetto

Rappresenta l'accettazione del progetto da parte del cliente in base a quanto rappresentato nel documento dei requisiti

Cliente e progettisti convergono su una singola idea di soluzione, sulle funzionalità e caratteristiche che il sistema possiederà. inoltre ci si accorda su:

- lista di priorità
- processo di revisione
- lista di criteri per accettare o rifiutare il progetto
- spese e tempistica

dare prioritá ai requisiti

Alta (“Core requirements”)

- da affrontare durante analisi, progetto, implementazione
- una caratteristica importante deve essere esibita con successo nei test di accettazione da parte del cliente

Media (“Optional requirements”)

- da affrontare durante analisi, progetto,
- Di solito sviluppati e mostrati in una seconda fase di sviluppo del sistema

– Bassa (“Fancy requirements”)

- da affrontare durante analisi
- Mostrano un uso futuro del sistema

Documento dei requisiti

- Il documento di specifica dei requisiti (System Requirements Specification) SRS contiene sia requisiti funzionali che non funzionali
- Realizzabile in molti modi
- Input per analisi e progetto
- Mi aiuta a capire che cosa deve fare il sistema?

Esempio di documento dei requisiti

Requirements Document	
<i>Table of Contents</i>	
1.	Project Preliminaries
1.1	Purpose and Scope of the Product
1.2	Business Context
1.3	Stakeholders
1.4	Ideas for Solutions
1.5	Document Overview
2.	System Services
2.1	The Scope of the System
2.2	Function Requirements
2.3	Data Requirements
3.	System Constraints
3.1	Interface Requirements
3.2	Performance Requirements
3.3	Security Requirements
3.4	Operational Requirements
3.5	Political and Legal Requirements
3.6	Other Constraints
4.	Project Matters
4.1	Open Issues
4.2	Preliminary Schedule
4.3	Preliminary Budget
Appendices	
	Glossary
	Business Documents and Forms
	References

Preliminari del progetto

- Indirizzato al management
- Inizia con gli obiettivi del progetto
- Fornisce una motivazione economica (business case) che si ricollega al system planning dell'organizzazione
- Identifica i gruppi interessati e le persone che dovranno approvare il sistema finale
- Offre idee iniziali per la soluzione, analizzando possibili prodotti da acquistare sul mercato.
- Include un sommario del resto del documento

Servizi del sistema

- Descrive il contesto e i servizi che devono essere forniti dal sistema
- Spesso piu' di meta' dell'intero documento
- Contiene i modelli ad alto livello dei requisiti
 - Diagramma del contesto (system scope)
 - Requisiti funzionali -> Diagramma sintetico dei casi d'uso
 - Requisiti informativi -> Diagramma sintetico delle classi (dati persistenti)

Vincoli del sistema

- Definisce i vincoli che devono essere rispettati dal sistema durante il funzionamento riguardo
 - Interfacce
 - Prestazioni (tempo risposta, throughput ..)
 - Sicurezza – privilegi per l'esecuzione di certe operazioni..
 - Operazionali - hw e sw
 - Politico – legali
 - Usabilita'
 - Manutenibilita'

Aspetti rilevanti per la realizzazione del sistema

- Problemi aperti
 - Requisiti futuri
 - Variazione dell' importanza dei requisiti attuali
 - Requisiti attuali da implementare in futuro – miglioramenti
 - Problemi potenziali quando il sistema diventa operativo
- Tempistica
 - Personale ed altre risorse
 - Diagrammi di pianificazione (PERT, Gantt)
- Budget preventivo
 - Un intervallo e non una cifra esatta

Appendici

- **Glossario**
 - Termini
 - Acronimi
 - Abbreviazioni
- Documenti e moduli
 - Esempi di moduli completati
- Riferimenti
 - Libri e documenti di pubblico dominio
 - Documenti interni dell'organizzazione ,di riunioni

Il glossario del progetto

contiene il dizionario dei termini e del gergo del contesto aziendale

disambigua omonimi e sinonimi

comprensibile da tutti i partecipanti

UML non stabilisce uno standard

deve essere sincronizzato col resto del progetto

Unified Modeling Language

Linguaggio visuale per specifica, costruzione, documentazione di un sistema software

Orientato agli oggetti

Nato nel 1997 dalla fusione di molti altri linguaggi di modellazione

Unified Modeling Language

3 modi d'uso

Abbozzo (molto in modalita' agile)

Progetto

Linguaggio di programmazione

Unified Modeling Language

13 diagrammi! Useremo un sottinsieme:

Classi

Attivita'

Casi d'uso

Stato

Sequenza

(Forse) Collaborazione

+ diagramma di contesto

Documenti principali

Software Project Management Plan

- Scadenze e oggetti da consegnare

Software Requirement Specification (o URD)

- Requisiti su cui basare design e test

Software Design Document

- Design e decisioni su cui basare implementazione e test

Software Test Document

- Test e risultati del test

Tutti questi documenti?

Non e' detto, dipende dalle dimensioni del progetto e dal modello di management

Per certi progetti e metodi di management esiste semplicemente una lista di requisiti prioritizzata

Il documento aiuta a creare un sistema funzionante o fa piu' che altro perdere tempo?

E' richiesto dai progettisti o dai controllori?

Tutti questi documenti?

E' un problema di organizzazione del lavoro;
tutto si semplifica se

- e' un progetto con un solo team
- il cliente e' sempre disposto a fornire supporto

Altrimenti:

- i tecnici devono essere autonomi e basarsi sulla qualita' dei documenti per la realizzazione.

Digressione: Il team

- Occorrono personalita' differenti:
 - Controller
 - Promoter
 - Analyzer
 - Supporter

Sommario

- La determinazione dei requisiti riguarda la loro scoperta e documentazione
- Scoperta attraverso la conoscenza del dominio e dei casi d'uso
- L'ottenimento dei requisiti puo' avvenire tramite interviste ad utenti ed esperti, questionari, osservazione ,prototipazione , esame di documenti, JAD, RAD, ..
- Negoziazione e validazione eliminano conflitti e sovrapposizioni dei requisiti
- I requisiti devono essere gestiti
- 3 tipi di diagrammi ad alto livello: di contesto, dei casi d'uso, delle classi persistenti
- Riferimenti
 - Libri e documenti di pubblico dominio
 - Documenti interni dell'organizzazione ,di riunioni
- Il documento risultante e' il documento dei requisiti
- UML

esercizio

- documento requisiti di un albergo
- funzionali e non, glossario
- casi d'uso
- Useremo I templates su moodle

Per ogni requisito

ID:

Descrizione:

Essenziale:

Priorita':

Stabile:

Fonte:

Test:

Impatto: