

# Ingegneria del Software L-A

## 2. Analisi orientata agli oggetti

## Analisi

- Per effettuare correttamente l'analisi, è necessario
  - Comunicare con l'utente
  - Ottenere una buona conoscenza dell'area applicativa
  - Determinare in dettaglio i requisiti del sistema
- Produce un documento dei requisiti base di partenza della fase successiva del processo di sviluppo del software, cioè la progettazione
- Se non viene svolta in modo accurato, può avere conseguenze molto serie: i costi per gestire requisiti omessi o errati possono diventare insostenibili

## Analisi

- Raccolta dei requisiti
- Analisi del dominio  
Produce un modello che descrive il dominio del problema da affrontare:  
la porzione del mondo reale, rilevante per il sistema
  - Su cui si devono mantenere informazioni
  - Con cui si deve interagire
- Analisi dei requisiti  
Produce un modello che descrive le responsabilità del sistema
  - Che cosa deve fare il sistema per soddisfare il cliente
  - Non come il sistema va realizzato

## Analisi

- Analisi orientata agli oggetti

Produce

- un modello dei dati o modello statico
  - la parte fondamentale dell'OOA, su cui si basa tutto il resto
  - descrive le entità del mondo reale rilevanti per il sistema
- un modello comportamentale o modello dinamico
  - completa il modello dei dati
  - descrive le funzionalità del sistema

L'insieme dei due modelli costituisce il punto di partenza della successiva fase di progettazione

## Analisi e gestione dei rischi

- Analisi sistematica e completa di tutti i possibili rischi che possono fare fallire o intralciare la realizzazione del sistema in una qualsiasi fase del processo di sviluppo
- Ogni rischio presenta due caratteristiche:
  - Probabilità che avvenga  
non esistono rischi con una probabilità del 100% (sarebbero vincoli al progetto)
  - Costo  
se il rischio si realizza, ne seguono effetti indesiderati e/o perdite

## Analisi e gestione dei rischi

- Rischi relativi ai requisiti  
i requisiti sono perfettamente noti?  
Il rischio maggiore è quello di costruire un sistema che non soddisfa le esigenze del cliente
- Rischi relativi alle risorse umane  
è possibile contare sulle persone e sull'esperienza necessarie per lo sviluppo del progetto?

## Analisi e gestione dei rischi

- Rischi tecnologici  
si sta scegliendo la tecnologia corretta?  
si è in grado di aggregare correttamente i vari componenti del progetto (ad es., GUI, DB, ...)?  
quali saranno i possibili futuri sviluppi della tecnologia?
- Rischi politici  
ci sono delle forze politiche (anche in senso lato) in grado di intralciare lo sviluppo del progetto?

## Analisi e gestione dei rischi

- **Strategia reattiva** o "*alla Indiana Jones*"
  - "Niente paura, troverò una soluzione"
- **Strategia preventiva**
  - Si mette in moto molto prima che inizi il lavoro tecnico
  - Si individuano i rischi potenziali, se ne valutano le probabilità e gli effetti e si stabilisce un ordine di importanza
  - Si predispone un piano che permetta di reagire in modo controllato ed efficace
    - Più grande è un rischio
    - Maggiore sarà l'attenzione che bisognerà dedicargli

## Analisi Raccolta dei requisiti

- Raccolta di tutte le informazioni su cosa il sistema deve fare secondo le intenzioni del cliente
- Obiettivi
  - Definire la ragione d'essere e gli obiettivi del sistema
  - Definire i limiti del dominio del problema
  - Identificare le funzioni principali del sistema

Ingegneria del Software L-A

2.9

## Analisi Raccolta dei requisiti

- Non prevede passi formali, né ha una notazione specifica, perché dipende moltissimo dal particolare tipo di problema
- Deve produrre
  - Un documento (testuale)
    - Scritto dall'analista
    - Discusso e approvato dal cliente
  - Un dizionario o glossario contenente la definizione di tutti i termini e i concetti utilizzati

Ingegneria del Software L-A

2.10

## Analisi Raccolta dei requisiti

- Tipologie di persone coinvolte
  - Analista
  - Utente
  - Esperto del dominio (non sempre indispensabile)
- Metodi utilizzati
  - Interviste, questionari
  - Studio di documenti che esprimono i requisiti in forma testuale
  - Osservazione passiva o attiva del processo da modellare
  - Studio di sistemi software esistenti
  - Prototipi

Ingegneria del Software L-A

2.11

## Analisi Raccolta dei requisiti

- La gestione delle interviste è molto complessa
  - ▶ i clienti potrebbero
    - Avere solo una vaga idea dei requisiti
    - Non essere in grado di esprimere i requisiti in termini comprensibili
    - Chiedere requisiti non realizzabili o troppo costosi
    - Fornire requisiti in conflitto
    - Essere poco disponibili a collaborare

Ingegneria del Software L-A

2.12

## Analisi Raccolta dei requisiti

- Confronto con altri sistemi  
Posizionare il sistema che si sta progettando rispetto allo stato dell'arte
- Quali sono
  - Le caratteristiche migliori
  - I problemi più seri
  - Le caratteristiche non necessarie del nostro sistema e dei sistemi della concorrenza?

Ingegneria del Software L-A

2.13

## Analisi Validazione dei requisiti

- Ogni requisito deve essere validato e negoziato con i clienti prima di essere riportato nel documento dei requisiti
- Attività svolta in parallelo alla raccolta
- **Validità** – il nuovo requisito è inerente il problema da risolvere?
- **Consistenza** – il nuovo requisito è in sovrapposizione e/o in conflitto con altri requisiti?
- **Realizzabilità** – il nuovo requisito è realizzabile con le risorse disponibili (hardware, finanziamenti, ...)?
- **Completezza** – esiste la possibilità che ci siano requisiti rimasti ignorati?

Ingegneria del Software L-A

2.14

## Analisi Cambiamento dei requisiti

- È normale che i requisiti subiscano modificazioni ed evolvano nel tempo
  - Requisiti esistenti possono essere rimossi o modificati
  - Nuovi requisiti possono essere aggiunti in una fase qualsiasi del ciclo di sviluppo
- Tali cambiamenti
  - Sono la norma, non l'eccezione
  - Possono diventare un grosso problema se non opportunamente gestiti

Ingegneria del Software L-A

2.15

## Analisi Cambiamento dei requisiti

- Più lo sviluppo è avanzato, più il cambiamento è costoso
  - Modificare un requisito appena definito è facile
  - Modificare lo stesso requisito dopo che è stato implementato nel software potrebbe essere molto costoso
- Ogni cambiamento deve essere accuratamente analizzato per valutare
  - La fattibilità tecnica
  - L'impatto sul resto del sistema
  - Il costo

Ingegneria del Software L-A

2.16

## Analisi Cambiamento dei requisiti

- Consiglio – sviluppare sistemi che
  - Siano il più possibile resistenti ai cambiamenti dei requisiti
  - Inizialmente, eseguano esclusivamente e nel modo migliore i soli compiti richiesti
  - In seguito, siano in grado di sostenere l'aggiunta di nuove funzionalità senza causare "disastri" strutturali e/o comportamentali

Ingegneria del Software L-A

2.17

## Analisi Analisi del dominio

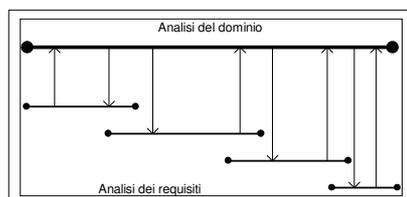
- **Obiettivo** – identificare strutture e comportamenti comuni a tutti i sistemi software di una particolare area applicativa
- Non è legata a un progetto specifico
- Scopo primario è la riusabilità di schemi di progettazione e di componenti software per tutte le applicazioni che operano su un dato dominio
- Esempi di domini sono:
  - Il controllo del traffico aereo
  - La gestione aziendale
  - Le operazioni bancarie
  - ...

Ingegneria del Software L-A

2.18

## Analisi Analisi del dominio

- Deriva dall'analisi dei requisiti dei vari sistemi che operano in un dominio
- Aiuta ad effettuare le analisi dei nuovi sistemi, ed è da queste continuamente migliorata



Ingegneria del Software L-A

2.19

## Analisi Analisi dei requisiti

- **Obiettivo** – definire i requisiti funzionali e descrivere il comportamento del sistema da realizzare
- Si concentra sulla descrizione del sistema, del mondo esterno e dei loro costituenti fondamentali, e non sui dettagli di come tale sistema funziona
- Prima considera le entità importanti dell'ambiente esterno e del sistema, poi le raffina alla luce delle responsabilità del sistema

Ingegneria del Software L-A

2.20

## Analisi Analisi dei requisiti

- Deve produrre un documento dei requisiti, nel quale i requisiti funzionali devono essere specificati in modo chiaro e conciso, in modo da poter essere letti e compresi
- Il documento dei requisiti:
  - Formalizza le necessità del cliente
  - Stabilisce un elenco di obblighi
  - Fornisce la base per il successivo sviluppo del sistema

Ingegneria del Software L-A

2.21

## Esempio: Villaggio Turistico Raccolta dei Requisiti

- In un villaggio turistico, gli ospiti fanno spesa nei diversi negozi e pagano i diversi servizi sempre e solo servendosi di una carta (simile a un bancomat) denominata Guest Card
- La valuta di riferimento è sempre l'euro
- Al termine della vacanza, ad ogni ospite viene consegnato un estratto conto con la lista delle spese effettuate, nella valuta scelta dal cliente
- Per ogni spesa, l'elenco deve riportare la data e l'ora, il punto vendita, il tipo di acquisto e l'importo addebitato
- Al termine di ogni settimana, ad ogni negozio deve essere consegnato l'elenco degli acquisti effettuati presso i vari punti vendita associati

Ingegneria del Software L-A

2.22

## Esempio: Villaggio Turistico Analisi dei Requisiti

- In un villaggio turistico, gli ospiti fanno spesa nei diversi negozi e pagano i diversi servizi sempre e solo servendosi di una carta (simile a un bancomat) denominata Guest Card
- Villaggio Turistico
- Ospite
- Spesa
- Negozio
- Servizio
- Carta ► Guest Card

Ingegneria del Software L-A

2.23

## Esempio: Villaggio Turistico Analisi dei Requisiti

- In un villaggio turistico, gli ospiti fanno spesa nei diversi negozi e pagano i diversi servizi sempre e solo servendosi di una carta (simile a un bancomat) denominata Guest Card
- Ospite
  - Può acquistare un servizio in un negozio
  - Deve pagare il servizio con la Guest Card
- Negozio
  - Eroga servizi
  - Incassa il pagamento del servizio mediante Guest Card
- Servizio
  - Ha un costo
- Guest Card
  - Unico mezzo per effettuare i pagamenti

Ingegneria del Software L-A

2.24

## Esempio: Villaggio Turistico Analisi dei Requisiti

- La valuta di riferimento è sempre l'euro
- Ospite
  - Può acquistare un servizio in un negozio
  - Deve pagare il servizio con la Guest Card in euro
- Negozio
  - Eroga servizi il cui costo è in euro
  - Incassa il pagamento del servizio mediante Guest Card in euro
- Servizio
  - Ha un costo in euro
- Guest Card
  - Permette di effettuare i pagamenti in euro
- Valuta di riferimento
  - Unica in tutto il Villaggio Turistico
  - In euro

Ingegneria del Software L-A

2.25

## Esempio: Villaggio Turistico Analisi dei Requisiti

- Al termine della vacanza, ad ogni ospite viene consegnato un estratto conto con la lista delle spese effettuate, nella valuta scelta dal cliente
- Termine della vacanza – evento temporale
- Estratto conto = lista delle spese effettuate
  - *report* di stampa
- Spesa effettuata = Servizio acquistato dall'ospite
- Cliente = Ospite
- Valuta scelta dall'ospite
  - Può essere differente dalla valuta di riferimento

Ingegneria del Software L-A

2.26

## Esempio: Villaggio Turistico Analisi dei Requisiti

- Al termine della vacanza, ad ogni ospite viene consegnato un estratto conto con la lista delle spese effettuate, nella valuta scelta dal cliente
- Ospite
  - Deve scegliere la valuta x pagamento finale
- Termine della vacanza – evento
  - Generazione dell'estratto conto acquisti
  - Consegna all'ospite dell'estratto conto acquisti
  - Pagamento finale nella valuta scelta dall'ospite
- NOTA: Sarà necessario effettuare conversioni tra valute diverse

Ingegneria del Software L-A

2.27

## Esempio: Villaggio Turistico Analisi dei Requisiti

- Per ogni spesa, l'elenco deve riportare la data e l'ora, il punto vendita, il tipo di acquisto e l'importo addebitato
- Spesa = Acquisto o Movimento
  - Data e ora del movimento
  - Punto di vendita (NON coincide con Negozio!)
  - Tipo di acquisto
  - Importo in euro
- Punto Vendita
- Catena Punti Vendita (ex Negozio)
- Tipo di Acquisto = Servizio

Ingegneria del Software L-A

2.28

## Esempio: Villaggio Turistico

### Analisi dei Requisiti

- Al termine di ogni settimana, ad ogni negozio deve essere consegnato l'elenco degli acquisti effettuati presso i vari punti vendita associati
- Termine di ogni settimana – evento temporale
  - Generazione dell'estratto conto vendite x Punto Vendita
  - Consegna alla Catena Punti Vendita

Ingegneria del Software L-A

2.29

## Analisi

### Casi d'uso e scenari

- I requisiti funzionali descrivono il comportamento del sistema
- I casi d'uso e i relativi scenari permettono
  - Di formalizzare i requisiti funzionali
  - Di comprendere meglio il funzionamento del sistema (e quindi di metterne in evidenza eventuali carenze)
  - Di comunicare meglio con il cliente

Ingegneria del Software L-A

2.30

## Analisi

### Casi d'uso e scenari

- **Caso d'uso**
  - Descrizione di una richiesta fatta al sistema da una qualsiasi entità esterna al sistema stesso (attore)
  - Insieme di scenari legati da un obiettivo comune
- **Scenario** – sequenza di passi che descrive
  - sia l'interazione tra l'attore e il sistema
  - sia le elaborazioni necessarie per soddisfare la richiesta dell'attore

Ingegneria del Software L-A

2.31

## Analisi

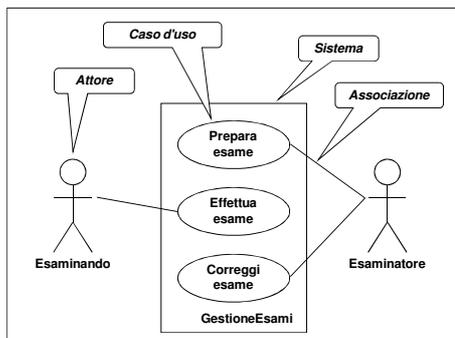
### Casi d'uso e scenari

- **Passi da intraprendere**
  - Individuare il confine del sistema
  - Individuare gli attori
  - Individuare i casi d'uso
    - Specificare il caso d'uso
    - Specificare gli scenari associati al caso d'uso
- L'insieme di tutti i casi d'uso costituisce l'immagine del sistema verso l'esterno

Ingegneria del Software L-A

2.32

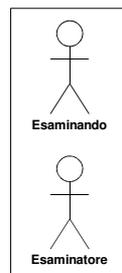
## Analisi Casi d'uso e scenari



Ingegneria del Software L-A

2.33

## Analisi Casi d'uso e scenari

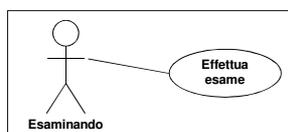


- **Attore**  
ruolo interpretato da un utente (persona o sistema esterno) nei confronti del sistema
- Tutti gli esaminandi interpretano lo stesso ruolo
- Tutti gli esaminatori interpretano lo stesso ruolo

Ingegneria del Software L-A

2.34

## Analisi Casi d'uso e scenari



Scenario principale del caso d'uso "Effettua esame"

1. L'esaminando entra nel sistema (*login*)
2. L'esaminando inizia l'esame
3. L'esaminando naviga tra le domande e risponde
4. L'esaminando termina l'esame
5. L'esaminando esce dal sistema (*logout*)

Ingegneria del Software L-A

2.35

## Analisi Casi d'uso e scenari

- Un caso d'uso
  - viene sempre **avviato** direttamente o indirettamente dall'intervento di un **attore** che si pone un dato **obiettivo**
    - l'esaminando vuole fare l'esame
  - si **conclude con successo** quando l'obiettivo viene raggiunto
    - l'esaminando ha fatto l'esame
  - si **conclude con fallimento** quando l'obiettivo NON viene raggiunto
    - l'esaminando non è riuscito a fare l'esame – ad es. non è riuscito ad effettuare il login (in questo contesto, il fatto che l'attore abbia superato o no l'esame è irrilevante)

Ingegneria del Software L-A

2.36

## Analisi Casi d'uso e scenari

- Un caso d'uso viene sempre descritto dal punto di vista di un attore e comprende
  - **0+ Precondizioni** – condizioni che devono essere tutte verificate prima che il caso d'uso possa essere eseguito – vincoli sullo stato iniziale del sistema
  - **1+ Sequenze di passi** – cioè **sequenze di interazioni** tra l'attore e il sistema necessarie a raggiungere l'obiettivo richiesto – potrebbero comprendere ramificazioni (**if**) e iterazioni (**for**, **foreach** e **while**)
  - **0+ Postcondizioni** – condizioni che devono essere tutte vere quando il caso d'uso termina l'esecuzione di norma con successo

## Analisi Casi d'uso e scenari

- Ogni sequenza di passi deve
  - essere scritta in una **forma narrativa strutturata**
  - utilizzare il **vocabolario di dominio**
- In tal modo, il committente
  - potrà comprendere facilmente la descrizione dei casi d'uso e di conseguenza
  - non solo sarà in grado di validare i casi d'uso
  - ma sarà anche incoraggiato a partecipare attivamente alla loro definizione

## Analisi Casi d'uso e scenari

- Tipicamente un caso d'uso comprende
  - uno **scenario principale**
  - eventuali **scenari alternativi**
    - che rappresentano possibili varianti del flusso
    - che sono fatti "scattare" da opzioni, condizioni d'errore, violazione della sicurezza, ...
- Ogni scenario
  - potrebbe essere descritto formalmente mediante un diagramma di sequenza,
  - ma viene descritto mediante il linguaggio naturale, che è più adeguato nella fase di raccolta dei requisiti

## Definizione dei casi d'uso e degli scenari

1. Identificare tutti i possibili utilizzi del sistema
2. Definire un attore per ogni categoria di utenza e per ogni ruolo che l'utente gioca e che ha rilevanza per il sistema
3. Per ciascun attore identificare tutti gli obiettivi significativi che gli utenti si pongono e che il sistema dovrà supportare
4. Per ciascun obiettivo definire un caso d'uso

## Definizione dei casi d'uso e degli scenari

5. Definire i casi d'uso
  - senza eccedere nella complessità e nei dettagli
  - mantenendo sempre lo stesso livello di astrazione: singoli passi in un caso d'uso d'alto livello d'astrazione possono essere trattati come obiettivi da casi d'uso di livello d'astrazione più basso
6. Ricontrollare e validare i casi d'uso insieme agli utenti e ai committenti

Ingegneria del Software L-A

2.41

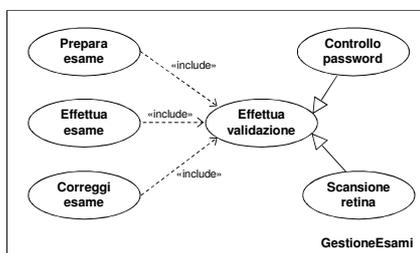
## Analisi Relazioni tra casi d'uso

- **Generalizzazione / Specializzazione**
  - Si utilizza quando un caso d'uso è simile ad un altro, ma fa qualcosa di più
  - È applicabile anche agli attori – un attore può essere la specializzazione di un altro attore
- **Inclusione «include»** (o «uses»)  
 • Si utilizza quando un caso d'uso “usa” almeno una volta un altro caso d'uso
- **Estensione «extend»** (o «extends»)  
 • Si utilizza quando è necessario aggiungere un comportamento opzionale a un caso d'uso esistente

Ingegneria del Software L-A

2.42

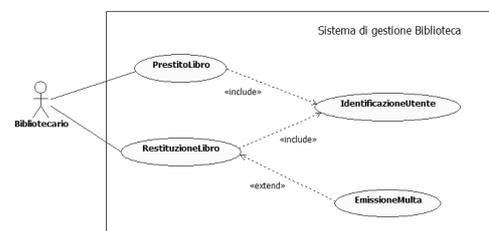
## Analisi Relazioni tra casi d'uso



Ingegneria del Software L-A

2.43

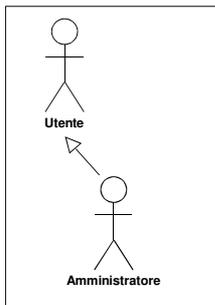
## Analisi Relazioni tra casi d'uso



Ingegneria del Software L-A

2.44

## Analisi Relazioni tra attori

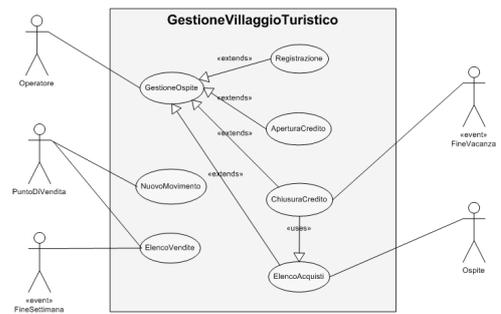


- L'attore Amministratore
- Eredita tutti i casi d'uso dell'attore Utente
- Ha casi d'uso propri

Ingegneria del Software L-A

2.45

## Esempio: Villaggio Turistico Casi d'uso



Ingegneria del Software L-A

2.46

## Analisi Definizione degli scenari - 1

- **Introduction**
  - Describe a quick background of the use case
- **Actors**
  - List the actors that interact and participate in this use case
- **Pre-conditions**
  - Pre-conditions that need to be satisfied for the use case to perform
- **Post-conditions**
  - Define the different states in which you expect the system to be in, after the use case executes
- **Basic Flow**
  - List the primary events that will occur when this use case is executed

Ingegneria del Software L-A

2.47

## Analisi Definizione degli scenari - 1

- **Alternative flows**
  - Any subsidiary events that can occur in the use case should be separately listed
  - List each such event as an alternative flow
  - A use case can have as many alternative flows as required
- **Special Requirements**
  - Business rules for the basic and alternative flows should be listed as special requirements in the use case narration
  - These business rules will also be used for writing test cases
  - Both success and failure scenarios should be described here
- **Use case relationships**
  - For complex systems, it is recommended to document the relationships between use cases

Ingegneria del Software L-A

2.48

## Analisi Definizione degli scenari - 2

Titolo	
Descrizione	
Relazioni	
Attori	
Precondizioni	
Scenario Principale	
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	