

Copyright © 2004 Moreno Marzolla

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

## Cosa è l'Ingegneria di Sistema

- Disegno, implementazione, installazione e messa in opera di sistemi che includono hardware, software e personale.
  - Introduremo il concetto di **ingegneria di sistema**
  - Descriveremo il processo di **acquisizione** e di **sviluppo** di un sistema
  - Vedremo come rappresentare l'architettura di un sistema
  - Introduremo il concetto di **affidabilità** di un sistema

## Ingegneria di Sistema

- Un **sistema** è un insieme di componenti correlate:
  - Software
  - Hardware
  - Risorse umane
  - Dati (Informazione)
- ...Insieme sono finalizzate ad un obiettivo comune
- *“Un insieme di elementi organizzati in modo da raggiungere uno scopo prefissato mediante l'elaborazione di dati”*

## Ingegneria di Sistema

- Ingegneria di sistema significa...
  - Progettare
  - Implementare
  - Installare
- ...sistemi che includono
  - hardware (meccanico, elettrico, elettronico),
  - software
  - personale

## Simulazione di sistemi / 1

- *“Costruiamo i sistemi come i fratelli Wright hanno costruito i loro aerei: costruiamo l'oggetto, lo spingiamo su un colle, lo facciamo schiantare e ricominciamo da capo”*

R. M. Graham

## Simulazione di sistemi / 2

- ...Ciò è stato talvolta preso troppo alla lettera



## Modellazione di un sistema

- Può essere utile per valutare in anticipo le caratteristiche quantitative o qualitative
- Come si procede
  - Definire una serie di processi che rappresentano entità della realtà fisica
  - Definire il comportamento di ciascun processo
  - Definire i dati che guidano il sistema
    - Dati esogeni (dati che provengono dall'esterno)
    - Dati endogeni (dati che il sistema si scambia al proprio interno)

## Simulazione di sistemi

- Gli strumenti di modellazione e simulazione di sistemi aiutano a eliminare le sorprese nella costruzione dei sistemi basati su computer
- Questi strumenti sono applicati durante il processo di ingegneria del sistema
  - Si può fare dal momento in cui viene specificato il ruolo dell'hardware, del software e delle persone

## Affidabilità di un sistema

- L'interdipendenza delle componenti fa sì che gli errori possano essere propagati in tutto il sistema. I fallimenti possono essere dovuti a interrelazioni tra componenti di cui non si è tenuto conto
- L'affidabilità dipende dall'affidabilità dell'hardware, del software e degli operatori
- La proporzione di software nei sistemi complessi è in crescita. Oggi il software fa molto di ciò che veniva fatto dall'hardware.

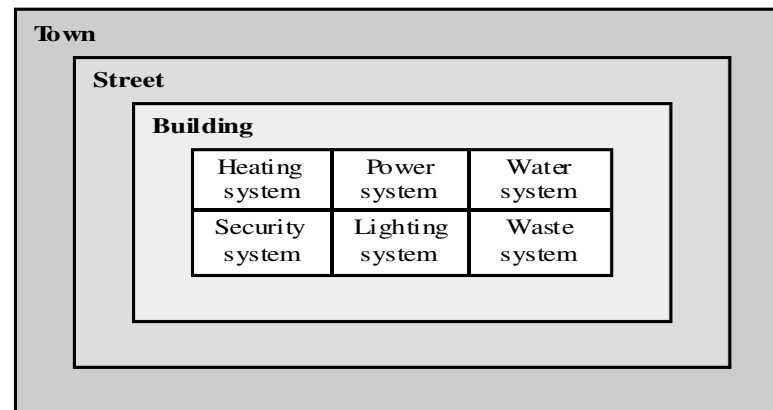
## Fattori che influenzano l'affidabilità

- **Affidabilità dell'hardware**
  - Quanto è probabile un fallimento hardware, e quanto tempo è richiesto per ripararlo?
- **Affidabilità del software**
  - Quanto è probabile che una componente software produca un risultato errato?
  - Differisce dall'affidabilità dell'hardware perché il software non è soggetto all'usura del tempo
- **Affidabilità dell'operatore**
  - Quanto è probabile che un operatore del sistema commetta un errore?

## Affidabilità di un sistema

- Il Software è spesso visto come il problema. Molti sistemi complessi sono falliti a causa di problemi col software.
- **Resilience** (capacità di recupero) = abilità del sistema di continuare ad operare correttamente in presenza di fallimenti di una o più componenti

## Sistemi e sottosistemi



## Il tutto non è la somma delle parti

- Le componenti di un sistema possono operare in modo indipendente, ma quando sono integrate in un sistema dipendono da altre componenti
- Esempi
  - Un sistema di gestione del traffico aereo
  - Un sistema di allarme

## Proprietà emergenti

- Sono le proprietà del sistema **nella sua globalità**, che possono essere o non essere direttamente derivate dalle proprietà di sue singole componenti
- Le proprietà emergenti derivano dall'**interazione** tra le componenti
- Le proprietà emergenti si possono misurare quando il sistema è stato assemblato.

## Esempi di proprietà emergenti

- Il **“peso”** globale del sistema
  - Questa proprietà emergente si può calcolare direttamente dalle singole parti
- L'**affidabilità** (*reliability*)
  - Dipende dall'affidabilità delle singole componenti, e da come sono in relazione tra di loro
- L'**usabilità** del sistema
  - Proprietà complessa che non dipende solo dall'hardware/software, ma anche dall'utente e dall'ambiente in cui il sistema opera
- La **riparabilità** del sistema
  - Dipende dalla possibilità di identificare facilmente il guasto, localizzare il componente responsabile e modificare o rimpiazzare tali componenti

## Tipi di proprietà

- Proprietà *funzionali*
  - Queste proprietà appaiono quando il sistema viene assemblato per compiere un determinato obiettivo (**“Cosa il sistema deve fare”**)
  - Es: un aereo ha la proprietà funzionale di essere un mezzo di trasporto volante
- Proprietà *non funzionali*
  - **proprietà** del sistema, ad es. sicurezza, efficienza...
  - **vincoli** sul sistema, ad es. vincoli d'ambiente
  - caratteristiche che il sistema **non** deve esibire

## Affidabilità di un sistema

- A causa delle interdipendenze all'interno di un sistema, gli errori si possono propagare da una componente ad un'altra
- Causa di problemi può essere una imprevista interrelazione tra componenti
- E' generalmente impossibile prevedere tutte queste interrelazioni

## Relazioni tra l'affidabilità delle componenti

- Guasti hardware possono causare segnali “spuri” nel software che a loro volta causano la produzione di risultati non corretti
- Errori del software possono causare “stress” nell'operatore, aumentando la sua propensione a commettere errori
- L'ambiente in cui il sistema opera può influenzare la sua affidabilità

## I sistemi e il loro ambiente

- I sistemi non sono indipendenti, ma sono inseriti in un ambiente, la cui conoscenza va inclusa nella specifica
- L'obiettivo di un sistema può essere di modificare il proprio ambiente (es. sistema di riscaldamento)
- L'ambiente può condizionare il comportamento del sistema (es. blackout)
- Sull'ambiente si devono fare delle assunzioni

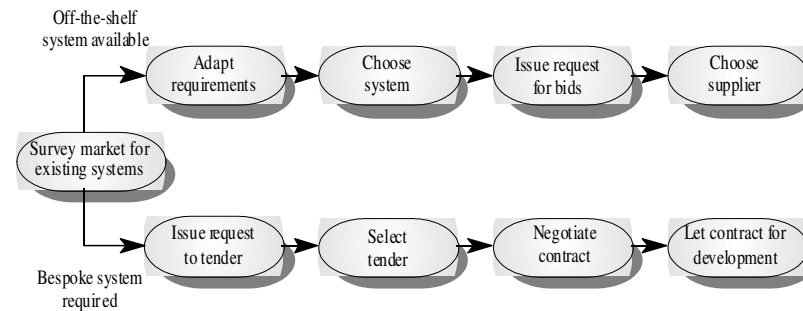
## Proprietà “in negativo”

- Alcune proprietà possono essere misurate
  - Le prestazioni, ad esempio
- Altre proprietà che il sistema *non* deve esibire possono porre problemi
  - **Safety** – Il sistema non deve comportare rischi per l'operatore o l'ambiente;
  - **Security** – Il sistema non deve consentire l'uso non autorizzato.
- Misurare questo secondo tipo di proprietà può essere assai complicato.

## Acquisizione di un sistema

- Un sistema può essere costruito o acquisito
- Per acquisire un sistema per una azienda per soddisfare una qualche necessità è necessario dare almeno la specifica del sistema (cosa è richiesto dal sistema)
- Scegliere tra sistemi o sottosistemi da comprare *off the shelf* e quelli da sviluppare in modo specifico su contratto.

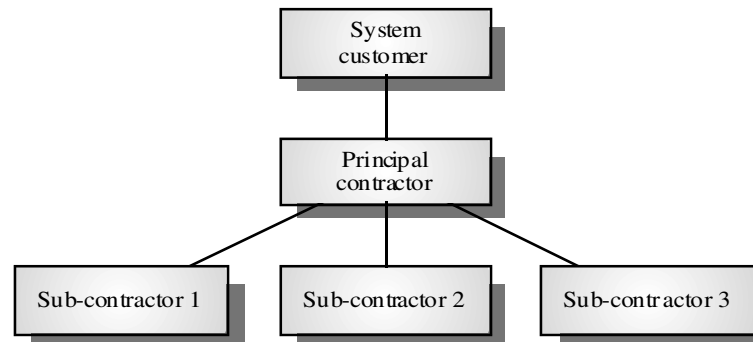
## Il processo di acquisizione



## Alcuni problemi del processo di acquisizione

- Un prodotto *off-the-shelf* potrebbe non adattarsi perfettamente alle richieste
  - Valutare la possibilità di modificare i requisiti
- La specifica dei requisiti può far parte del contratto stipulato con il produttore del software
- Di solito è previsto un certo tempo per modificare i requisiti. Trascorso questo tempo, il processo di sviluppo inizia

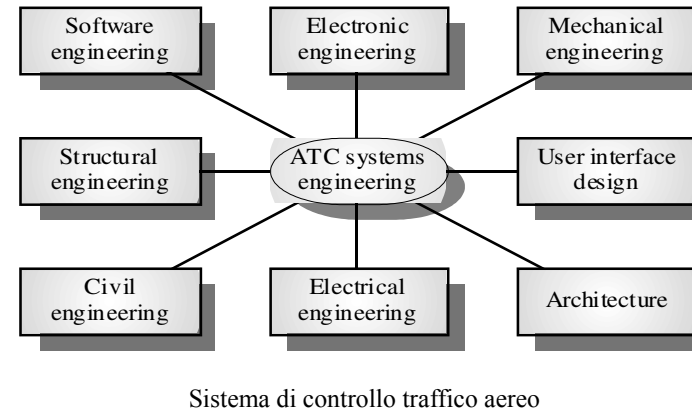
## Contraenti e Sotto-contraenti



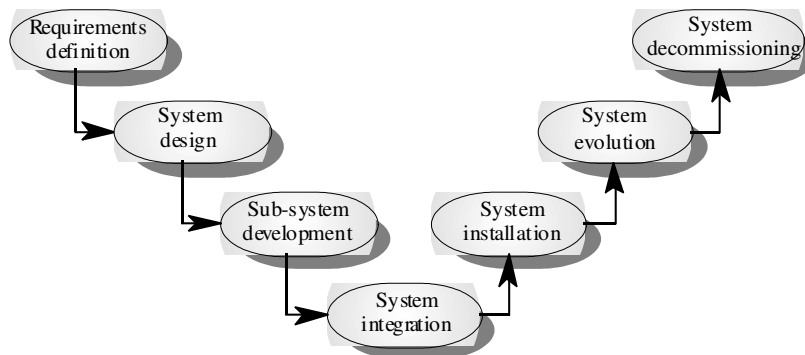
## Progettazione di un sistema

- Coinvolge tecnici di aree diverse, con problemi di vocabolario e metodologia
- Usualmente segue un modello di sviluppo a *cascata*
  - C'è poco spazio per iterazioni tra le varie fasi, per gli alti costi di modifica
- Il sottosistema software è quello più flessibile
  - Modifiche hardware sono molto costose e complesse. Il software può dover compensare problemi hardware

## Multidisciplinarietà



## Sviluppo di un sistema



## Definizione dei Requisiti

- **Requisiti Funzionali**
  - Cosa il sistema deve fare
- **Proprietà del Sistema (Requisiti non funzionali)**
  - Proprietà emergenti del sistema: availability, prestazioni, safety...
- **Caratteristiche che il sistema **non** deve avere**
  - Es: un sistema per il controllo traffico aereo *non* deve sovraccaricare l'operatore mostrando troppe informazioni contemporaneamente

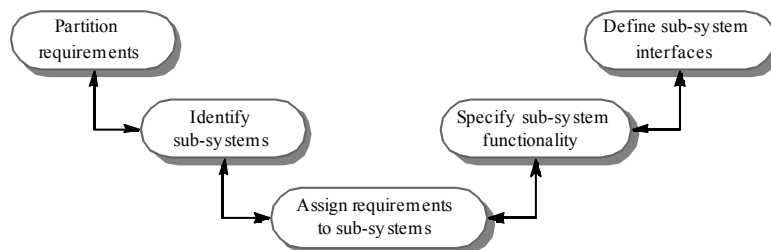
## Esempio

- Possibili requisiti per un sistema per la sicurezza di un edificio
  - Costruire un sistema di allarme contro incendi e furti per l'edificio che segnali all'interno ed all'esterno la presenza di un incendio o di un'intrusione non autorizzata.
  - Costruire un sistema che assicuri che il normale funzionamento del lavoro svolto nell'edificio non sia turbato da eventi come incendi o intrusioni non autorizzate.

## Disegno del Sistema

- La fase di disegno definisce il modo in cui le funzionalità del sistema devono essere fornite dalle diverse componenti.
  - Partizionamento dei requisiti in gruppi correlati
  - Identificazione dei sottosistemi; ogni sottosistema di solito soddisfa ad un gruppo di requisiti
  - Assegnare requisiti ai sottosistemi
  - Specificare la funzionalità dei sottosistemi (di solito ciò è incluso nella precedente fase di specifica dei requisiti)
  - Definire le interfacce dei sottosistemi

## La fase di disegno



## Implementazione dei sottosistemi

- Implementare ciascuno dei sottosistemi individuati nella fase precedente
  - Spesso in parallelo, da parte di team diversi
  - Problemi di comunicazione fra team
- Può richiedere a sua volta un nuovo processo di sviluppo
- Spesso (ma non sempre) i sottosistemi sono componenti off-the-shelf che vengono integrati
  - Questa operazione spesso è non banale



## Integrazione del sistema

- Ciascuno dei sottosistemi deve ora essere integrato in una entità completa
- Possono emergere problemi a causa di assunzioni errate sui sottosistemi
  - Es: A e B “dovevano” utilizzare lo stesso protocollo di comunicazione, e in realtà usano ciascuno un “dialetto” diverso e incompatibile
- Spesso è impossibile completare lo sviluppo di tutti i sottosistemi allo stesso tempo
- Conviene integrare i sottosistemi in modo *incrementale*

## Installazione del sistema

- Mettere il sistema completo nel suo ambiente operativo
- Attività non priva di problemi:
  - L'ambiente effettivo può differire da quello ipotizzato
  - Resistenze da parte degli utenti all'introduzione di un nuovo sistema
  - Il nuovo sistema può dover cooperare/convivere con un sistema precedente.
  - Problemi di ordine pratico (cablature, consumo di corrente, condizionamento ambientale...)

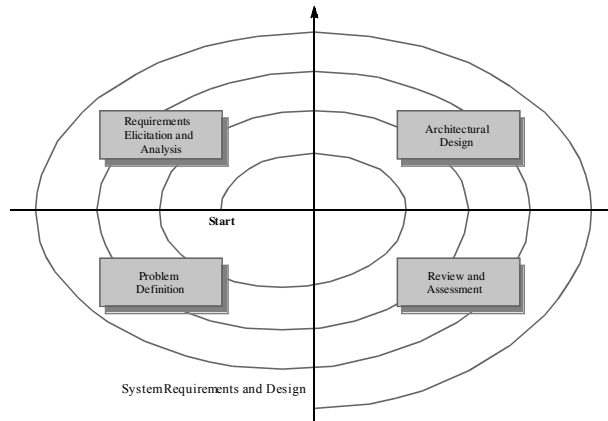
## Messa in opera del sistema

- Mette in luce i requisiti non contemplati
- Gli utenti possono usare il sistema in modo difforme da quello anticipato dai progettisti
- Può rivelare problemi nell'interazione con altri sistemi
- Problemi nell'uso di interfacce diverse che possono indurre in errore gli operatori
- Il personale deve essere adeguatamente addestrato

## Evoluzione e dismissione del sistema

- Può essere richiesta per adeguarsi all'evoluzione dei requisiti, o dell'hardware, o dell'ambiente in cui il sistema opera...
- Procedimento molto costoso:
  - Ogni cambiamento deve essere accuratamente esaminato
  - I sottosistemi *non* sono sempre indipendenti. Una modifica in uno di essi si può ripercuotere negativamente su tutto il sistema
  - Man mano che il sistema invecchia, i cambiamenti effettuati si accumulano

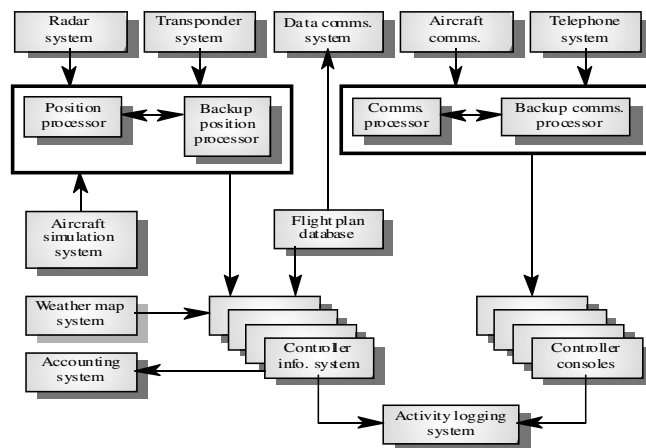
## Un modello “a spirale” del processo di sviluppo



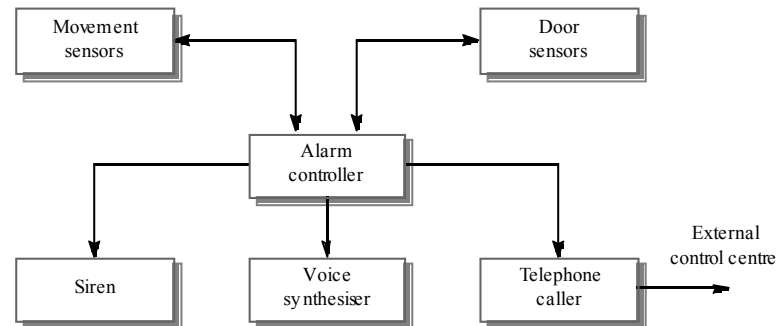
## Modellare l'architettura di un sistema

- Il modello architetturale di un sistema mostra in modo astratto la struttura in sottosistemi
- Dovrebbe rappresentare i flussi di informazione tra i vari sottosistemi
- Usualmente il modello architetturale è presentato sotto forma di diagramma a blocchi
- Dal modello si dovrebbero identificare i diversi tipi di componenti del sistema

## Esempio: sistema per il controllo traffico aereo



## Esempio: sistema d'allarme



## Descrizione dei sottosistemi

Sub-system	Description
Movement sensors	Detects movement in the rooms monitored by the system
Door sensors	Detects door opening in the external doors of the building
Alarm controller	Controls the operation of the system
Siren	Emits an audible warning when an intruder is suspected
Voice synthesizer	Synthesizes a voice message giving the location of the suspected intruder
Telephone caller	Makes external calls to notify security, the police, etc.

## Individuare le componenti funzionali

- Sono quelle componenti che svolgono una singola funzione ben definita. Un sottosistema è solitamente multifunzione.
- Tipi di componenti funzionali:
  - *Sensori*
  - *Attuatori*
  - *Componenti per Computazione*
  - *Componenti per Comunicazione*
  - *Coordinamento*
  - *Componenti Interfaccia*

## Esempio

