



How the customer explained it



How the Project Leader understood it



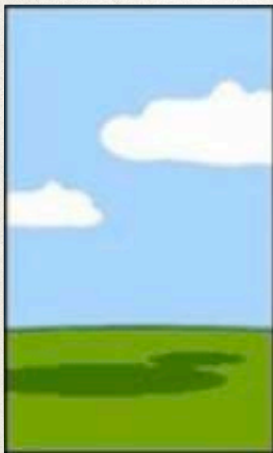
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



How the project was documented



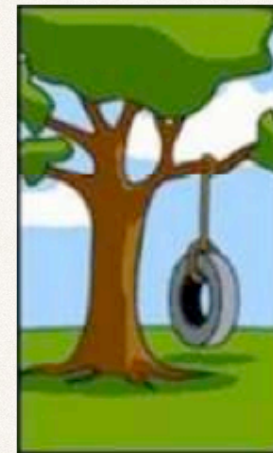
What operations installed



How the customer was billed



How it was supported



What the customer really needed

Ingegneria di Sistema

A. Gianoli

Dove

- ❖ Pressmann, solo qualcosa nel cap. 8
- ❖ Sommerville, cap. 2

Finora...

- ❖ L'Ingegneria del software si occupa di teorie, metodi e strumenti per sviluppare, produrre e mantenere prodotti software
- ❖ “Prodotto software” vuole dire i programmi e la relativa documentazione
- ❖ Un prodotto software può essere valutato rispetto ad attributi di qualità del prodotto
- ❖ Il processo software consiste nelle attività necessarie per sviluppare software, e anch'esso può essere valutato rispetto ad attributi di qualità di processo
- ❖ Il software è inutile se non è combinato con componenti hardware per formare un “sistema”

Ingegneria di Sistema

- ❖ Un sistema è un insieme di componenti correlate tra loro
 - ❖ software
 - ❖ hardware
 - ❖ risorse umane
 - ❖ dati (informazioni)
- ❖ tutti insieme sono finalizzati ad un obiettivo comune
- ❖ *“Un insieme di componenti correlati in modo da raggiungere uno scopo prefissato mediante l’elaborazione di dati”*

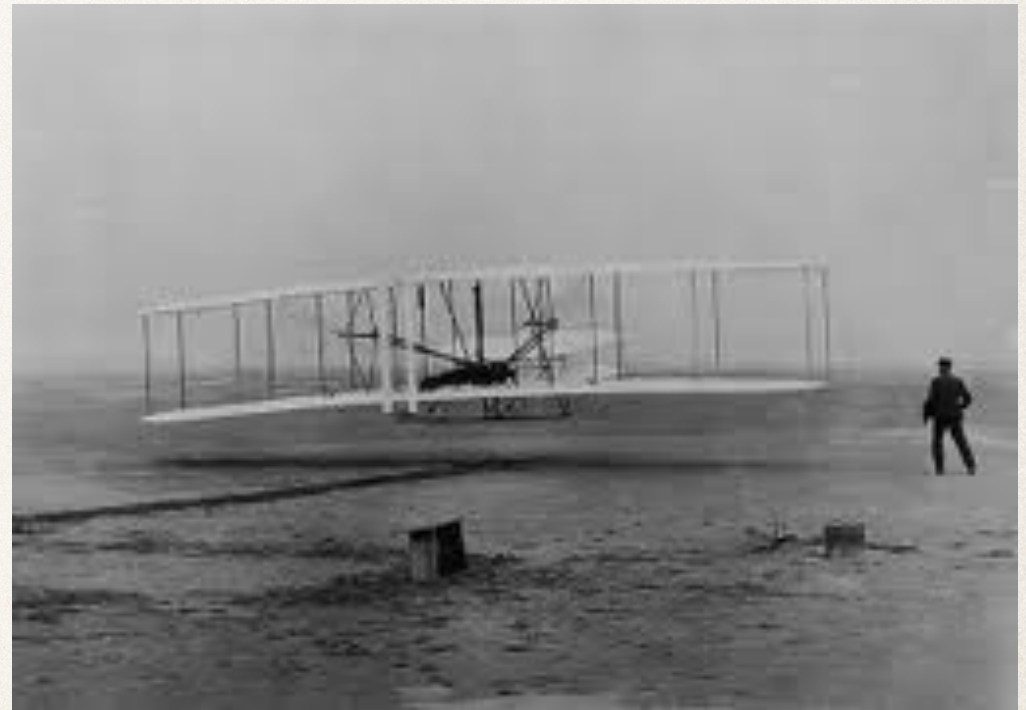
Un sistema per volare: 1

- ❖ Un primo approccio: decomposizione strutturale
- ❖ Mitologia greca: Icaro cerca di volare imitando gli uccelli, creando un sistema i cui elementi corrispondevano alle parti fisiche dei volatili
- ❖ Fallimento...



Un sistema per volare: 2

- ❖ Un secondo approccio:
decomposizione funzionale
- ❖ I fratelli Wright capiscono che restare in quota e muoversi sono due funzioni diverse, e che possono essere assegnate a due diverse componenti: ali e motore
- ❖ Successo...



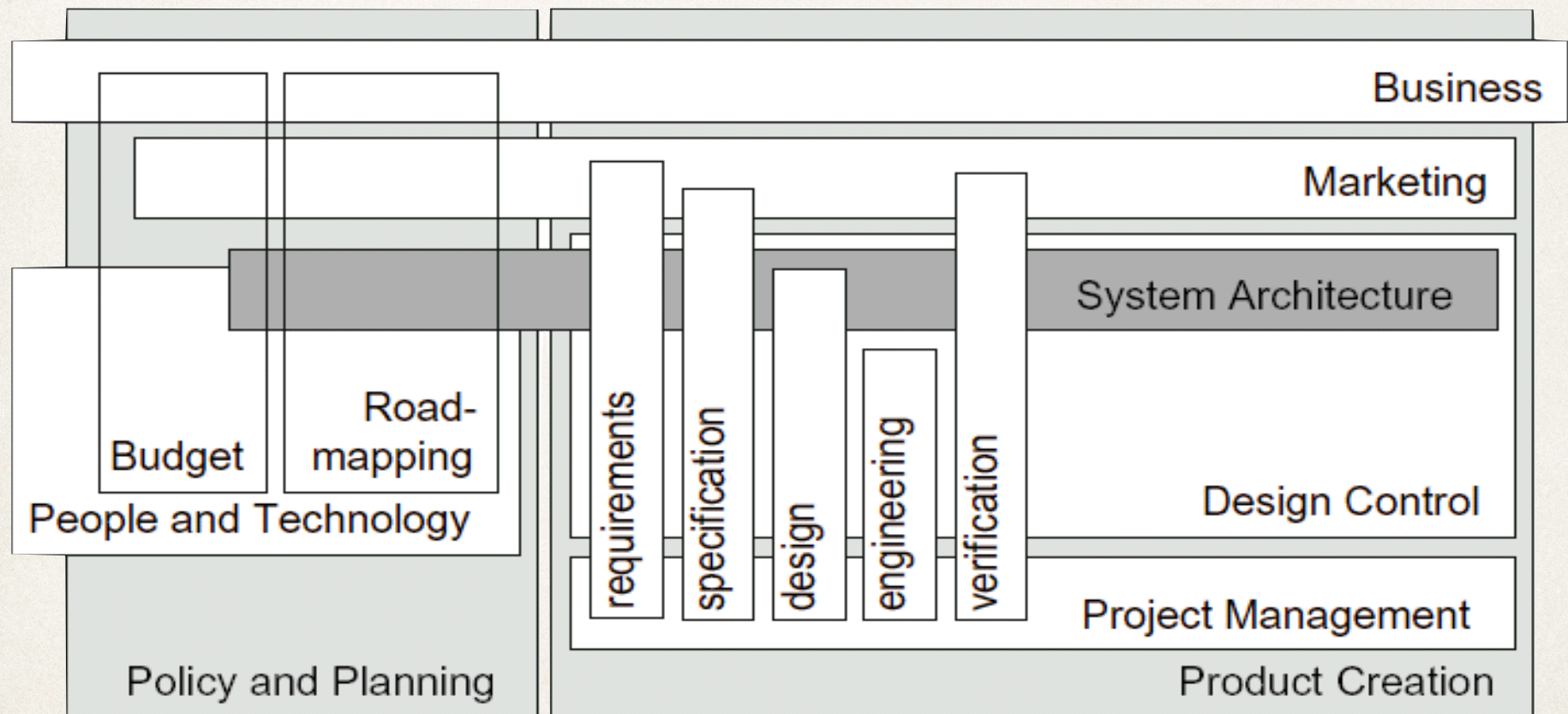
Ingegneria di Sistema

- ❖ Trasformare un bisogno operativo in una descrizione di parametri operativi e una configurazione di sistema attraverso un processo iterativo di analisi, sintesi, ottimizzazione, progetto e valutazione
- ❖ Integrare parametri tecnici correlati e assicurare la compatibilità di tutte le interfacce fisiche e funzionali al fine di ottimizzare il progetto complessivo
- ❖ Integrare affidabilità, manutenibilità, supporto logistico, sicurezza, fattibilità, integrità strutturale, fattori umani con l'obiettivo di ottimizzare il risultato

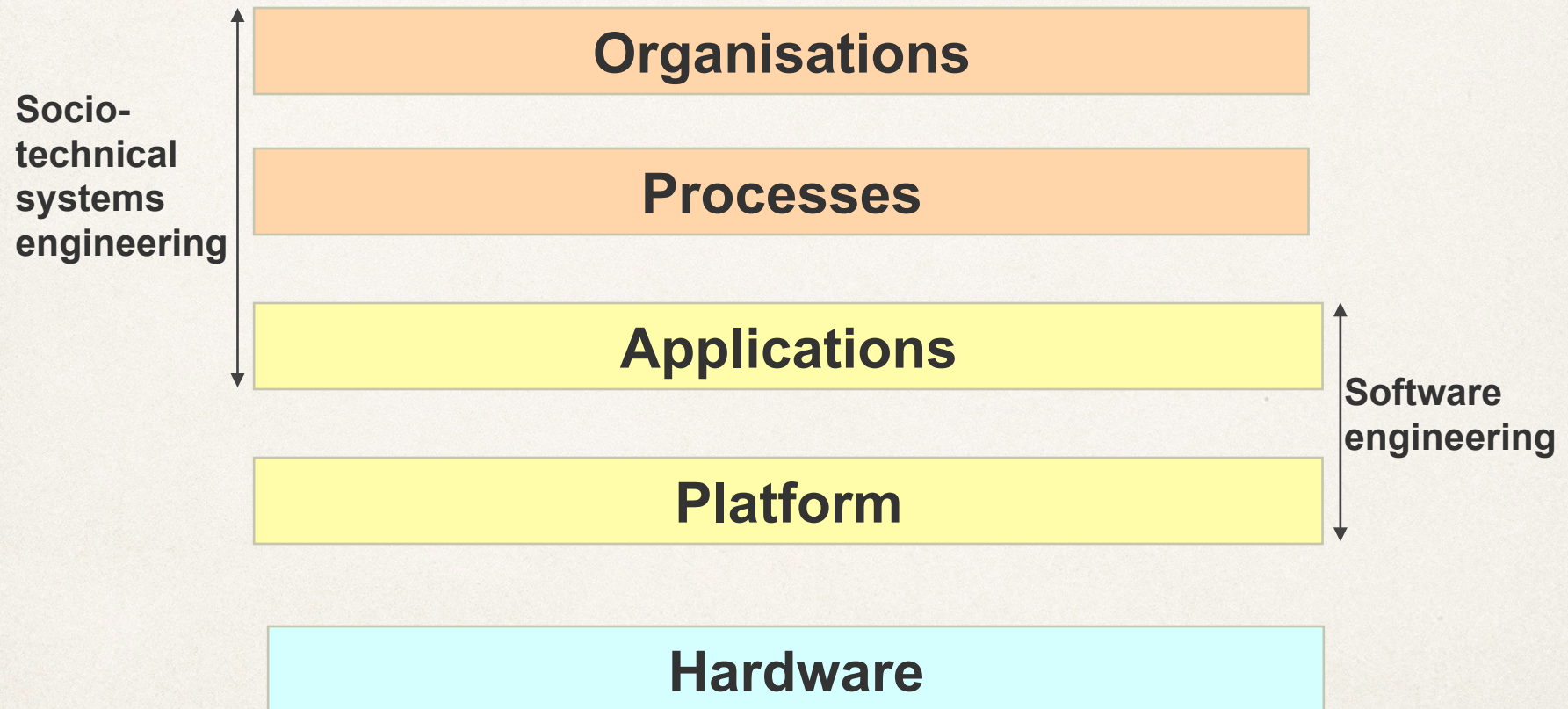
Ingegneria di Sistema

- ❖ significa
 - ❖ progettare
 - ❖ implementare
 - ❖ installare
- ❖ sistemi che includono
 - ❖ hardware (meccanico, elettronico, ...)
 - ❖ software
 - ❖ personale

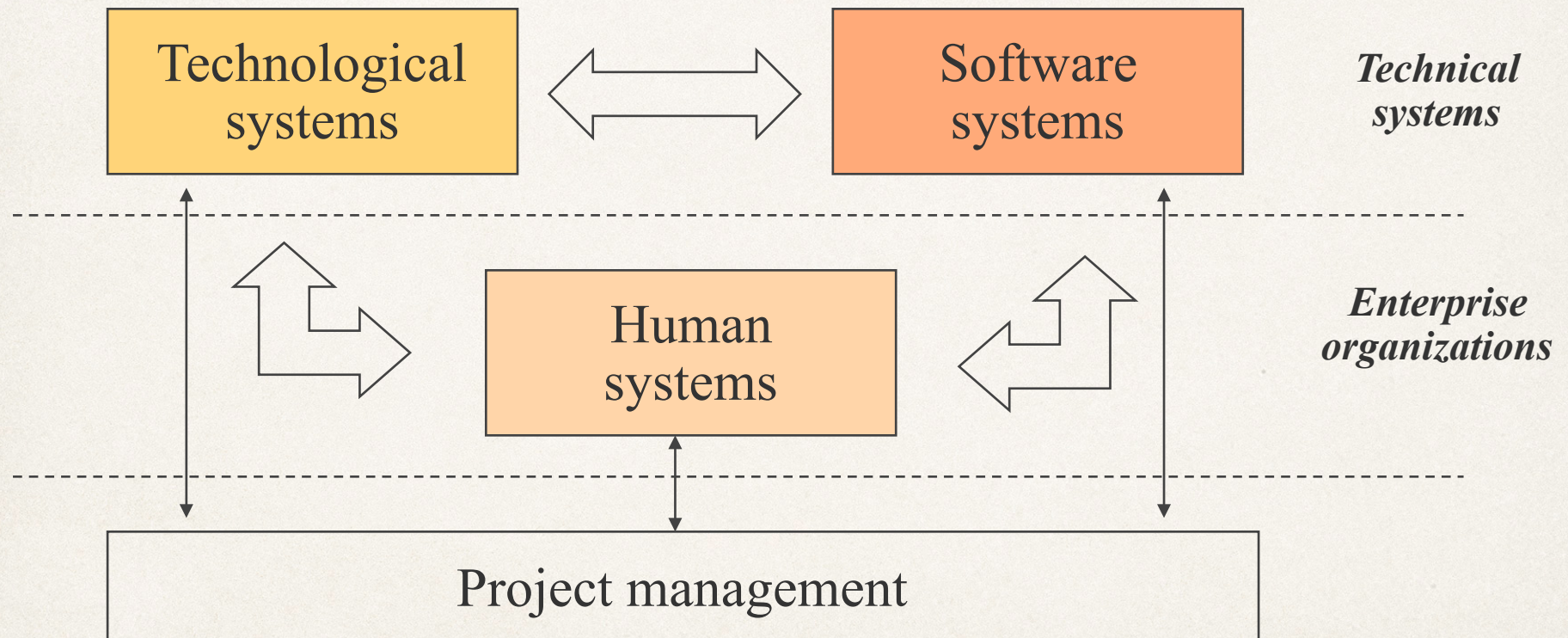
Il contesto



Il contesto

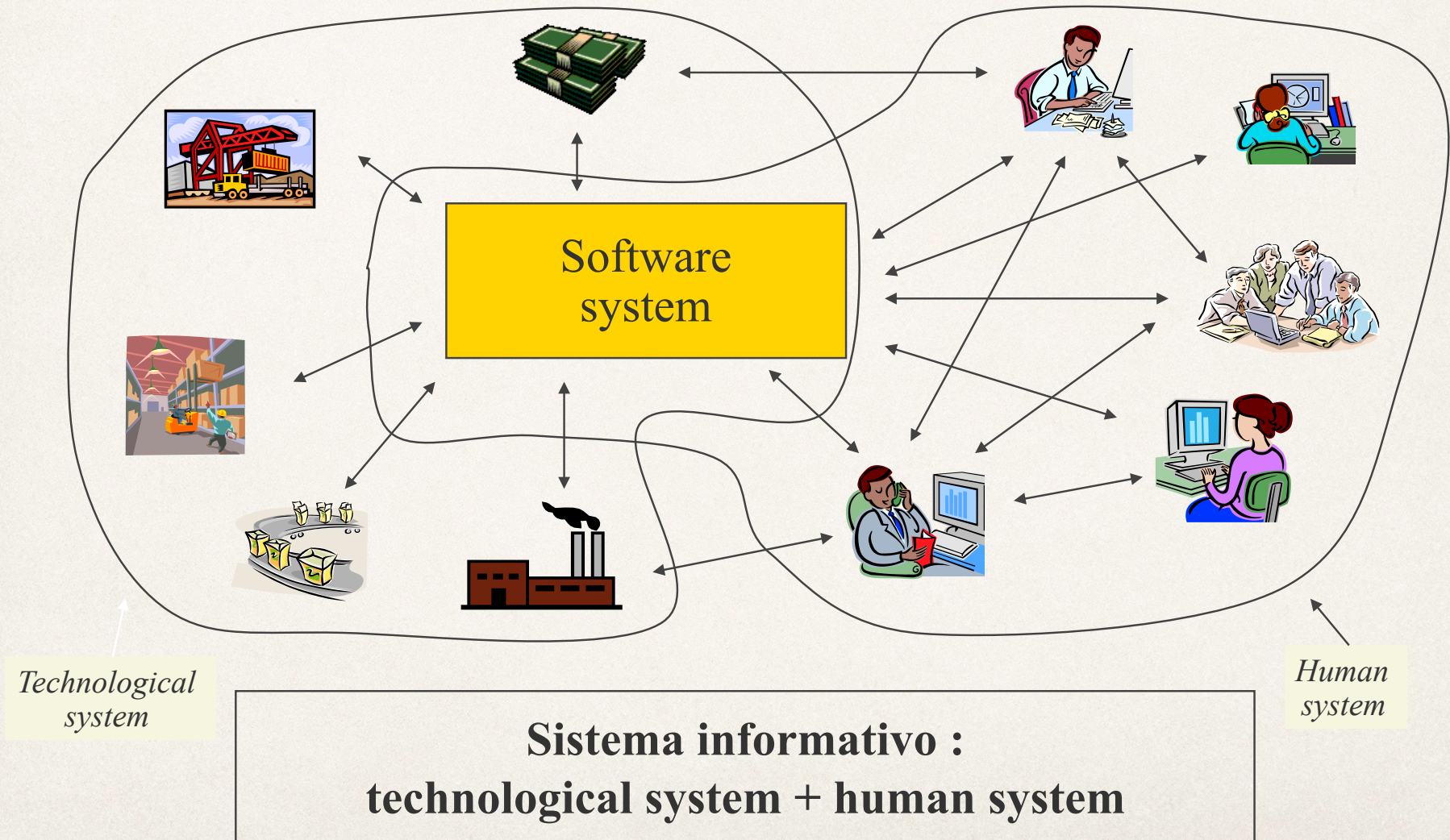


Sistemi industriali

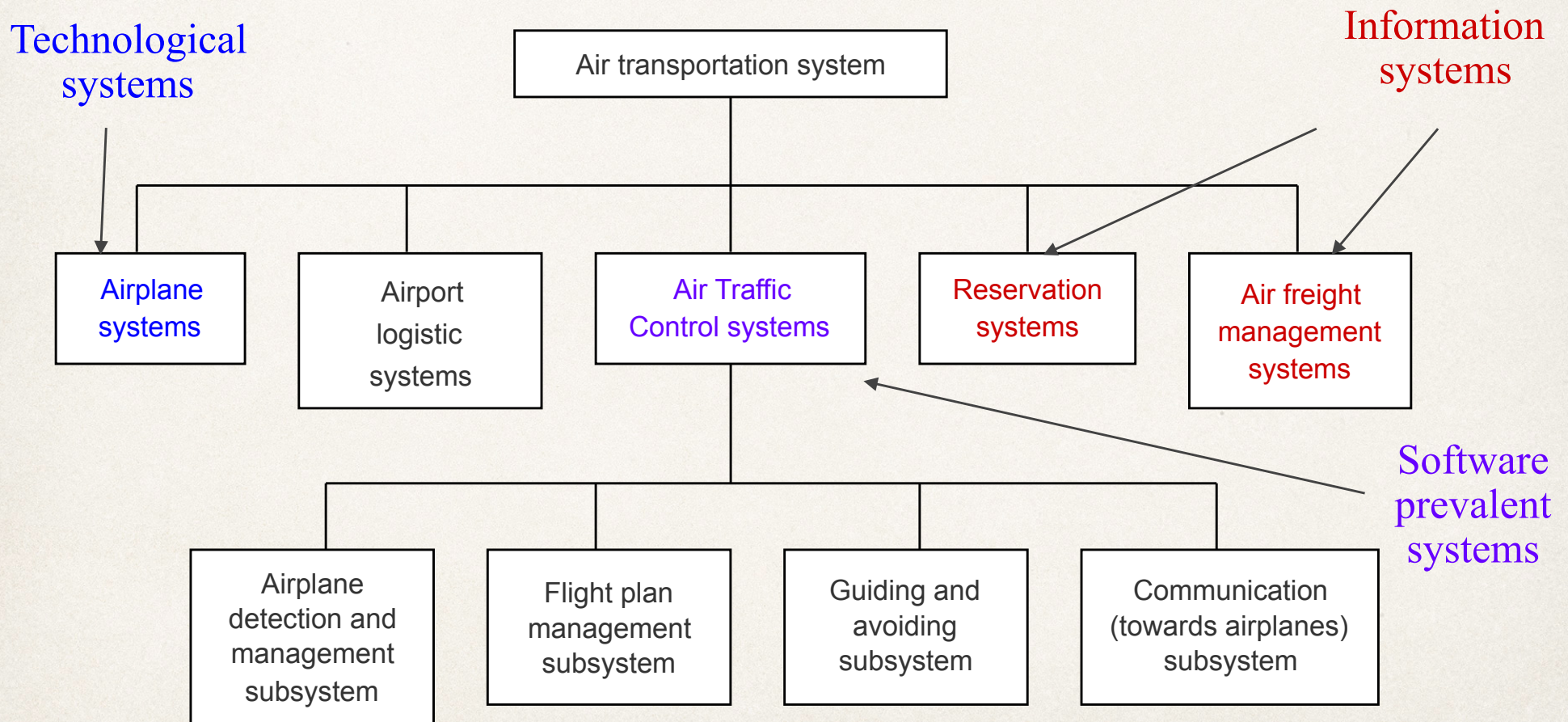


Struttura di un tipico sistema industriale

Sistemi informativi (ERP, SCM, BI, ...)



Esempio: controllo traffico aereo



Complessità dei sistemi

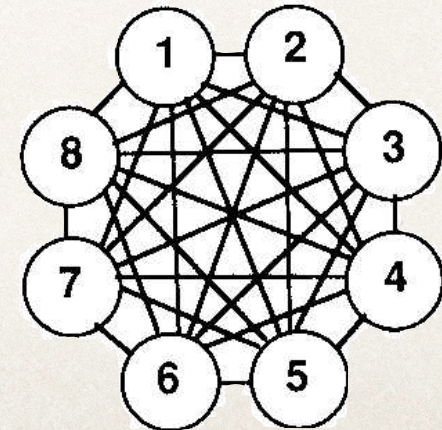
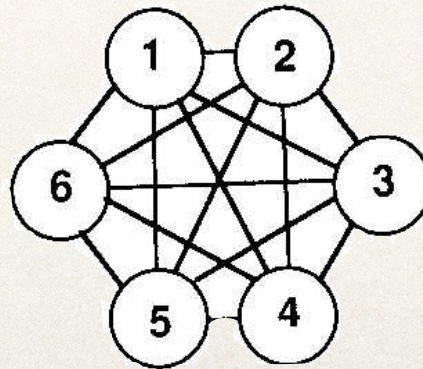
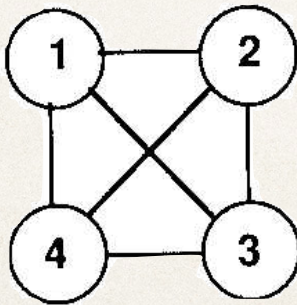
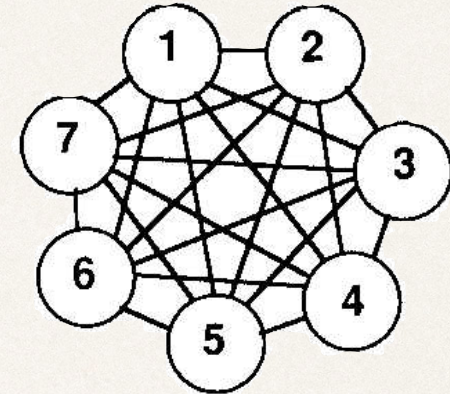
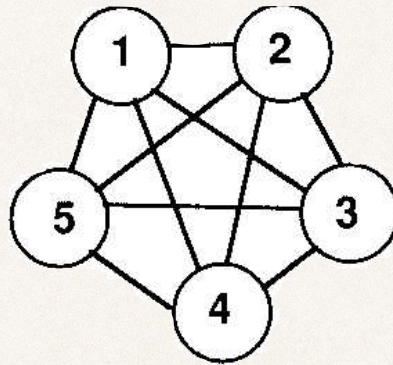
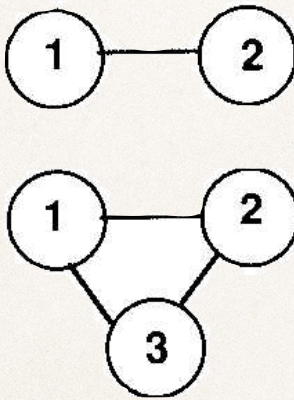
- ❖ Numero dei sottosistemi
 - ❖ ampiezza dell'albero di decomposizione del sistema
 - ❖ profondità dell'albero di decomposizione del sistema
- ❖ Eterogeneità di tipo dei sottosistemi

Type of system	Scientific bases
Software system	Computer science
Technological system	Physics
Human system	Human sciences

+
Signal
processing
Control theory

Difficoltà teorica nel definire la relazione tra i sottosistemi

Problema di scalabilità dei sottosistemi



Approccio multidisciplinare

Types of systems	Formal models
<i>Software systems</i>	Computer science modeling <ul style="list-style-type: none">■ Discrete models■ Specification
<i>Technological systems</i>	Continuous modeling
Human systems	Human sciences

Signal
+ processing
Control theory



Objective = technical architecture



- Management
- System engineering

Servono capacità scientifiche E manageriali

Modellazione di un sistema

- ❖ Può tornare utile per valutare in anticipo le caratteristiche quantitative o qualitative
- ❖ Come si procede
 - ❖ si definisce una serie di processi che rappresentano entità della realtà fisica
 - ❖ si definisce il comportamento di ciascun processo
 - ❖ si definiscono i dati che guidano il sistema
 - ❖ dati esogeni (che provengono dall'esterno)
 - ❖ dati endogeni (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 si specifica il ruolo dell'hardware, del software e delle persone

Affidabilità di un sistema

- ❖ L'interdipendenza delle componenti fa sì che gli errori possano propagarsi in tutto il sistema
- ❖ I fallimenti possono essere dovuti anche a interrelazioni tra componenti di cui non si è tenuto conto
- ❖ L'affidabilità complessiva dipende da quella dell'hardware, quella del software e quella degli operatori
- ❖ L'affidabilità è un parametro sempre più importante: la proporzione della componente software in sistemi complessi è in forte aumento. Il sw fa molto di ciò che si faceva fare all'hw

Fattori che influenzano l'affidabilità

- ❖ Affidabilità dell'hw
 - ❖ quanto è probabile un fallimento hw e quanto tempo è richiesto per ripararlo?
- ❖ Affidabilità del sw
 - ❖ quanto è probabile che una componente sw produca un output sbagliato?
 - ❖ si differenzia da quella hw perché il sw non è soggetto a usura
- ❖ Affidabilità operatore
 - ❖ quanto è probabile che un utilizzatore commetta un errore?

Affidabilità di un sistema

- ❖ Il software spesso viene visto come il problema. Molti sistemi complessi sono considerati un fallimento a causa di problemi col software
- ❖ Resilience = abilità del sistema a continuare ad operare correttamente in presenza di fallimenti di uno o più componenti (spesso è una qualche forma di ridondanza)

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 dalle altre componenti
- ❖ Esempi:
 - ❖ sistema di gestione del traffico aereo
 - ❖ sistema di allarme
 - ❖ sistemi HVAC (*Heating, Ventilation and Air Cooling*)

Proprietà emergenti

- ❖ Proprietà del sistema visto globalmente, che non possono necessariamente essere derivate dalle proprietà delle singoli componenti del sistema
- ❖ Possono essere conseguenza delle relazioni che intercorrono tra le componenti del sistema (derivano dall'integrazione delle componenti)
- ❖ Sono proprietà che possono essere valutate e misurate solo in seguito all'integrazione delle componenti del sistema

Esempi di proprietà di un sistema

- ❖ Peso complessivo del sistema
 - ❖ Può essere calcolato a partire dalle proprietà delle componenti del sistema
- ❖ L'affidabilità del sistema
 - ❖ dipende dall'affidabilità delle singole componenti E dalle relazioni che intercorrono tra le componenti
- ❖ L'usabilità di un sistema
 - ❖ non dipende solo dalle componenti hw e sw, ma anche dall'ambiente e dagli operatori

Tipi di proprietà emergenti

- ❖ **Proprietà funzionali**

- ❖ Queste appaiono quando le varie componenti vengono assemblate e lavorano per uno scopo comune (“quello che il sistema deve fare”)
 - ❖ p.e.: una bicicletta ha la proprietà funzionale di essere un mezzo di trasporto una volta che le sue componenti vengono assemblate

- ❖ **Proprietà non-funzionali**

- ❖ Queste proprietà hanno a che fare con il comportamento del sistema nel suo ambiente operativo; possono anche essere vincoli sul sistema
 - ❖ p.e.: reliability, performance, safety, security..

Affidabilità di un sistema

- ❖ A causa delle interdipendenze all'interno di un sistema gli errori si possono propagare da una componente all'altra
- ❖ La causa di un problema può anche essere una imprevista interrelazione tra componenti
- ❖ E' generalmente impossibile prevedere tutte queste interazioni

Relazioni tra l'affidabilità delle componenti

- ❖ Guasti hw possono causare segnali “spuri” nel software che a loro volta causano la produzione di risultati non corretti
- ❖ Errori nel sw 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 (p.e. sistema di riscaldamento)
- ❖ L'ambiente può condizionare il comportamento del sistema (p.e. blackout)
- ❖ Sull'ambiente si devono fare delle assunzioni

Tipi di proprietà: proprietà in negativo

- ❖ Proprietà quali performance, reliability possono essere misurate
- ❖ Tuttavia altre proprietà sono quelle che il sistema *non* deve esibire
 - ❖ safety: il sistema non deve comportare rischi (per operatore, ambiente, ...)
 - ❖ security: il sistema non deve consentire l'uso non autorizzato
- ❖ Misurare questo tipo di proprietà può essere molto complicato

Acquisizione di un sistema

- ❖ Un sistema può essere costruito o acquisito
- ❖ Per acquisire un sistema per una azienda per soddisfare una particolare necessità è necessario dare almeno la specifica del sistema (cosa è richiesto al sistema) e l'architettura di progetto
- ❖ Occorre scegliere tra sistemi/sottosistemi quali comprare "off the shelf" e quali sviluppare in modo specifico su contratto
- ❖ Fornitori e sotto-fornitori...

Il processo di acquisizione

sistemi *"off the shelf"*

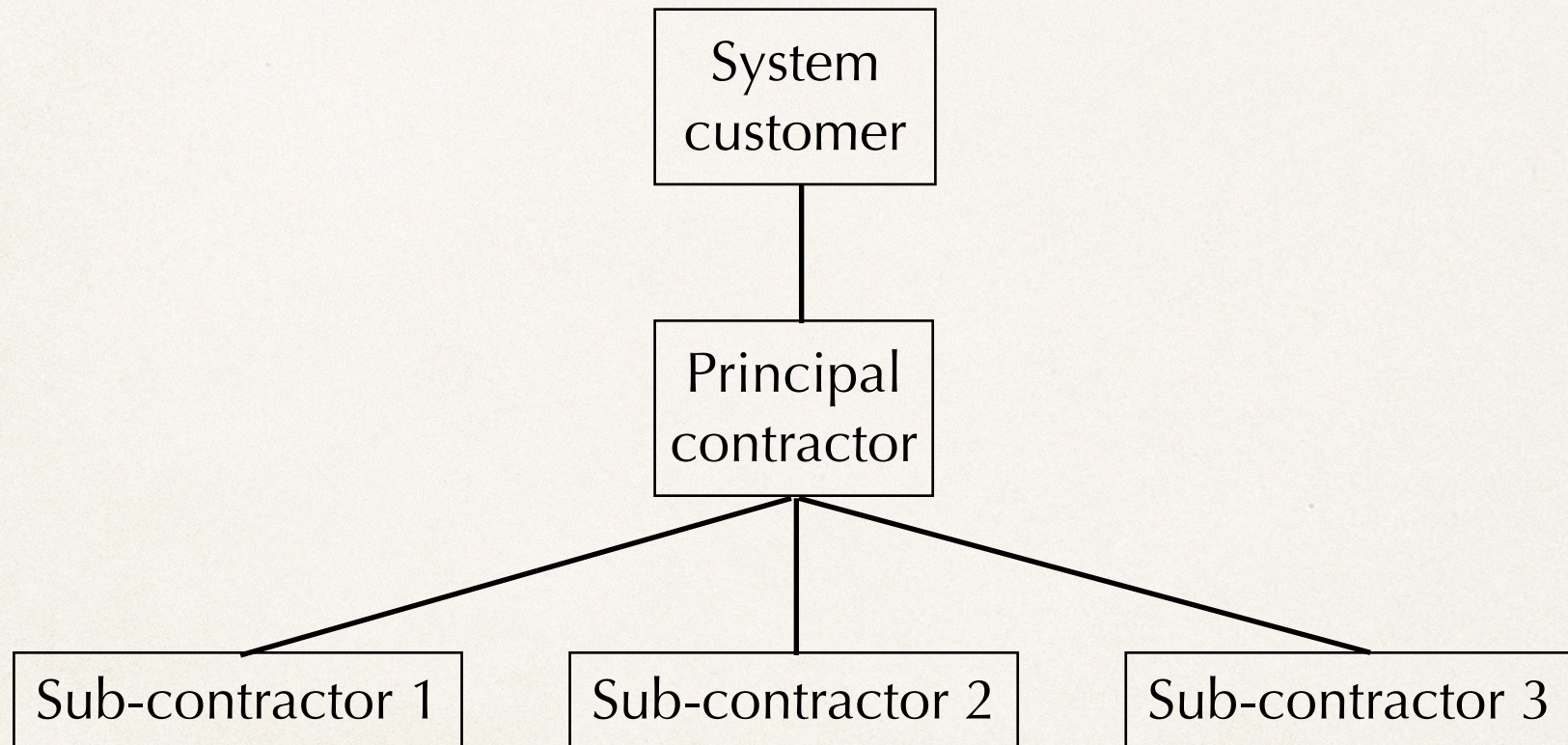


sistemi dedicati

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 si prevede un certo intervallo di tempo per modificare i requisiti: trascorso questo intervallo si inizia il processo di sviluppo

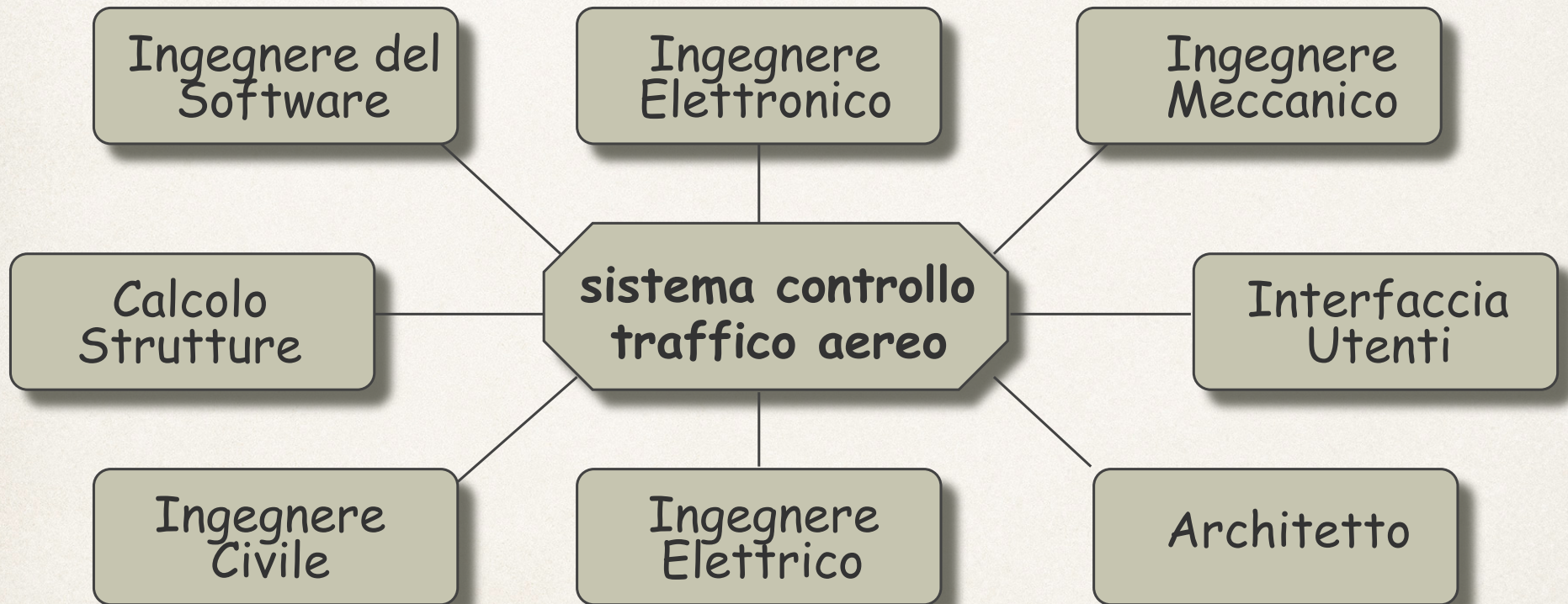
Modello contraente/sottofornitori



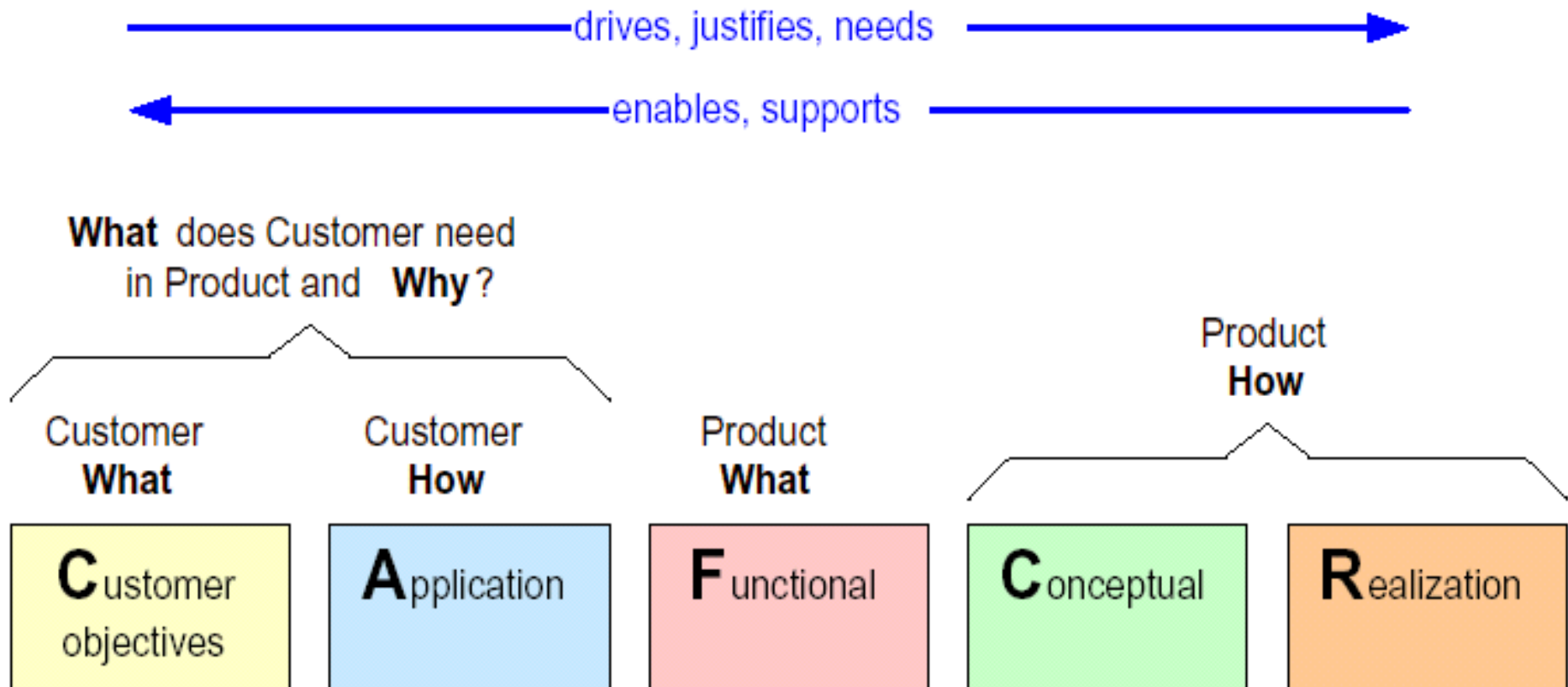
Progettazione di un sistema

- ❖ Coinvolge inevitabilmente tecnici di aree diverse, con problemi di “vocabolario” e di metodologia
- ❖ Di solito segue un modello di sviluppo a cascata, per poter sviluppare parallelamente le diverse componenti del sistema
- ❖ C'è poco spazio per iterazioni tra le diverse fasi, dati gli alti costi di modifica
- ❖ Il sottosistema “software” è quello più flessibile (viene visto come collante)
 - ❖ Modifiche hardware sono in generale molto costose e complesse: il sw può dover compensare problemi hw

Multidisciplinarietà



Architettura di riferimento



Sviluppo di un sistema



Definizione dei requisiti

- * Quali sono i requisiti globali di un sistema?
- * **Requisiti funzionali**: cosa il sistema deve fare
- * **Requisiti non funzionali**:
 - * proprietà del sistema: sicurezza, efficienza, ...
 - * vincoli su sistema: vincoli sull'ambiente, piattaforme da usare, ...
- * *Requisiti che un sistema non deve avere*

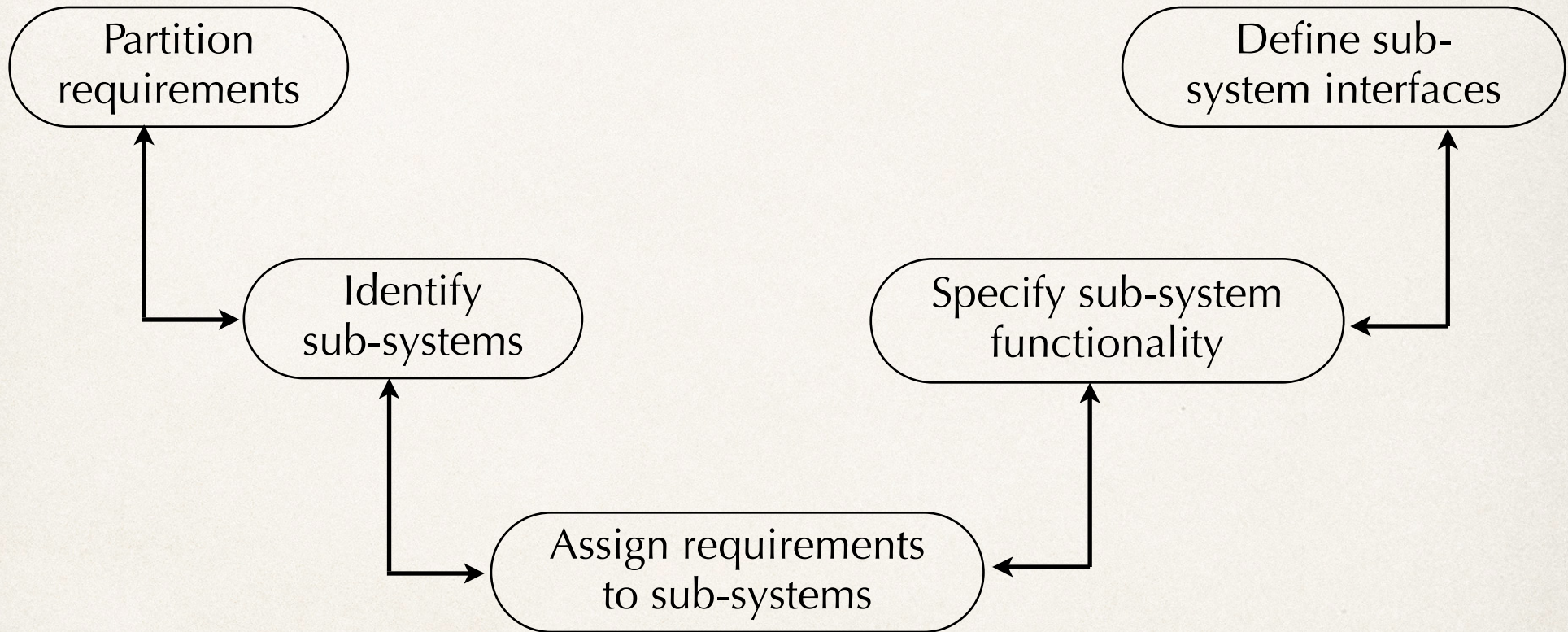
Esempio: sicurezza in un edificio

- ❖ Costruire un sistema di allarme contro incendi e furti per l'edificio, che segnali all'interno e all'esterno la presenza di un incendio o di una 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

- * Questa fase definisce il modo in cui le funzionalità del sistema devono essere fornite dalle diverse componenti
 - organizzare i requisiti, separandoli in gruppi correlati
 - identificare i sottosistemi: ogni sottosistema di solito soddisfa un gruppo di requisiti
 - assegnare i requisiti ai sottosistemi
 - specificare le funzionalità dei sottosistemi (di solito incluso fase precedente di specifica requisiti)
 - definire le interfacce dei sottosistemi
 - * punto critico per parallelizzazione

La fase di disegno



Sviluppo dei sottosistemi

- ❖ Implementare ciascuno dei sottosistemi individuati nella fase precedente
 - ❖ spesso viene fatto in parallelo da parte di team diversi
 - ❖ problema di comunicazione tra i team
- ❖ Può richiedere a sua volta un nuovo processo di sviluppo
- ❖ I sottosistemi possono essere componenti off-the-shelf che vengono integrati
 - ❖ non è detto che sia banale

Integrazione del sistema

- ❖ E' il processo di mettere insieme hw, sw e risorse umane
 - ❖ non è possibile un approccio “big bang”
- ❖ Conviene integrare i sottosistemi in modo incrementale
 - ❖ i sottosistemi vanno ingrati uno alla volta
 - ❖ questo riduce i costi di individuazione degli errori
- ❖ In questa fase possono emergere problemi di interfaccia tra le diverse componenti
- ❖ Spesso non è possibile completare lo sviluppo di tutti i sottosistemi allo stesso tempo

Installazione del sistema

- ❖ Mettere il sistema completo nel suo ambiente operativo
- ❖ Possibili problemi
 - ❖ l'ambiente finale può essere diverso da quello di sviluppo/test
 - ❖ resistenza da parte utilizzatori al nuovo sistema
 - ❖ coesistenza (cooperazione/convivenza) del nuovo sistema con il precedente
 - ❖ problemi “pratici” (cablaggio, corrente, condizionamento, ...)
 - ❖ formazione utenti

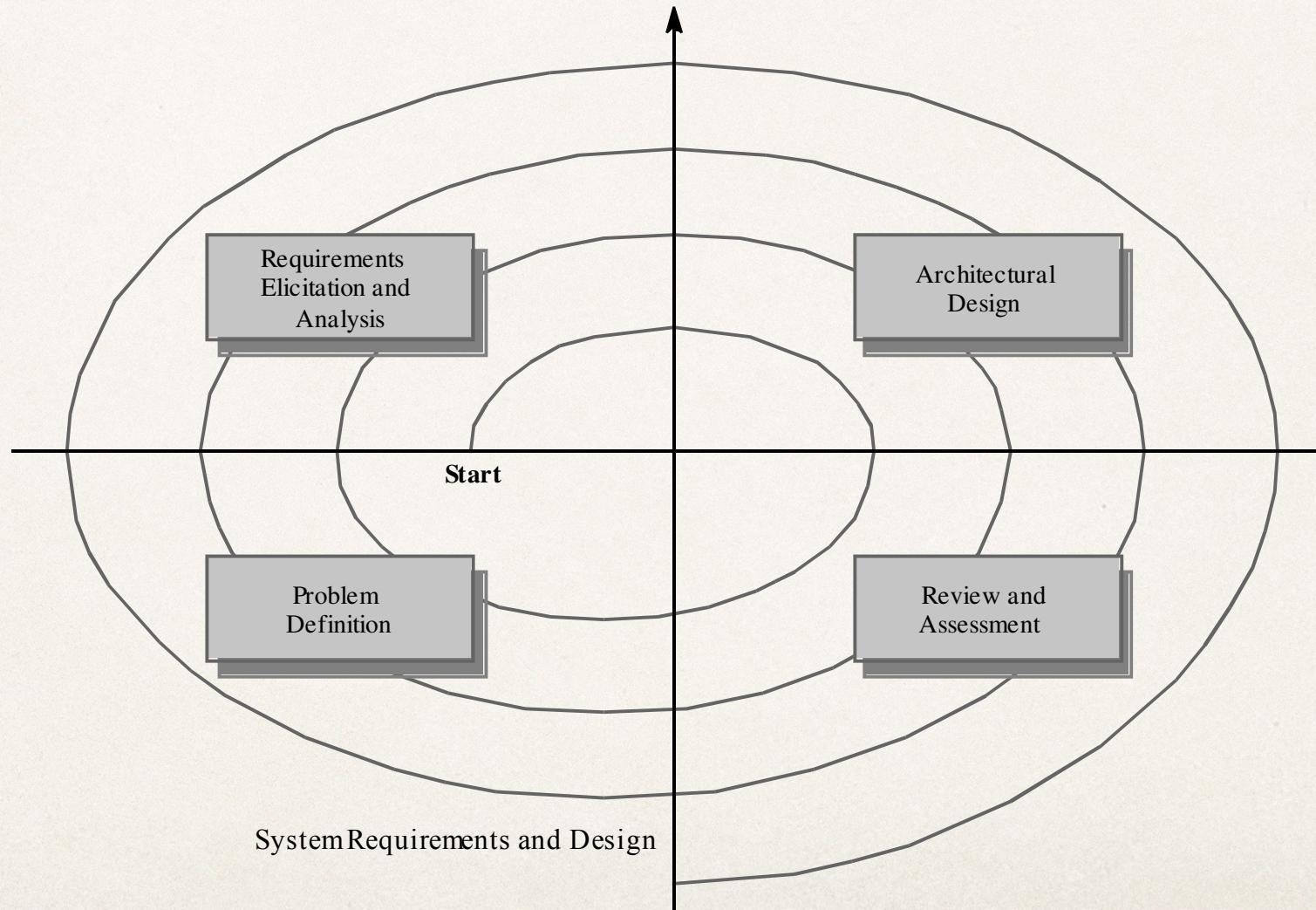
Messa in opera

- ❖ Mette in luce i requisiti non contemplati
- ❖ Gli utenti possono utilizzare il sistema in modo diverso da quello previsto dai progettisti
- ❖ Si possono verificare problemi con l'interazione con altri sistemi
- ❖ Problemi nell'uso di interfacce diverse che possono indurre in errore gli operatori

Mantenimento e smantellamento

- ❖ Qualunque sistema ha una sua durata (di solito legata alla dimensione)
 - ❖ deve evolvere per soddisfare l'evoluzione dei requisiti
- ❖ L'evoluzione è intrinsecamente costosa
 - ❖ ogni cambiamento deve essere accuratamente esaminato (ricordate i discorsi sull'affidabilità?)
 - ❖ va verificata l'interdipendenza dei sottosistemi
 - ❖ più passa il tempo e più i cambiamenti effettuati si accumulano
- ❖ Obsolescenza dei sistemi (hw E sw)
- ❖ Può richiedere la riconversione dei dati per l'uso con altri sistemi

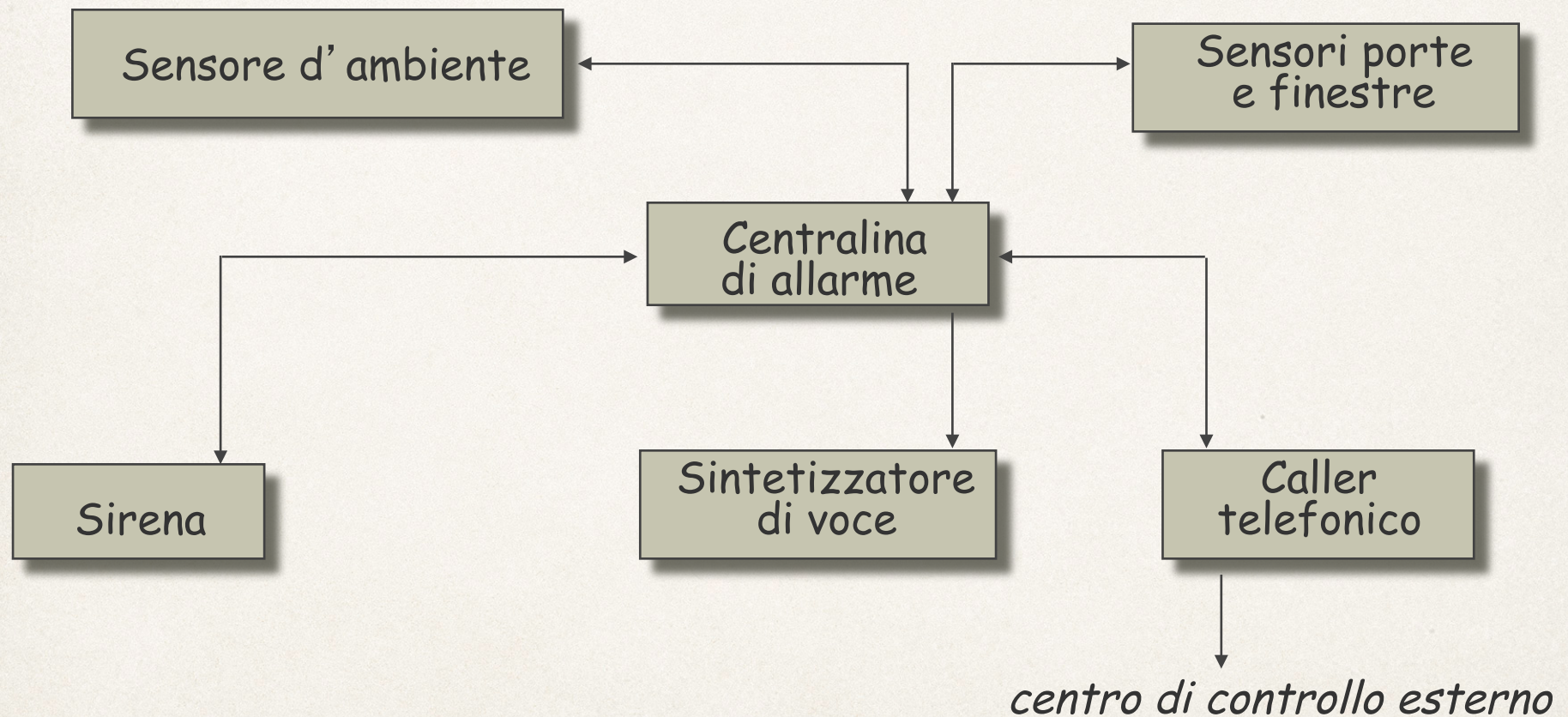
Un modello “a spirale” del processo di sviluppo



Modellare l'architettura di un sistema

- ❖ Il modello architetturale mostra in modo astratto la struttura in sottosistemi
- ❖ Modelli gerarchici: organizzazione ad albero (il controllo traffico aereo visto in precedenza è un modello gerarchico statico)
- ❖ Modelli funzionali: rappresentano i flussi di informazione tra i vari sottosistemi
 - ❖ di solito mediante un diagramma a blocchi
 - ❖ dal modello si dovrebbero identificare i diversi tipi di componenti del sistema
 - ❖ vari modi: data flow diagram (DFD), behaviour diagram, ..

Esempio: sistema d'allarme



Esempio: sistema d'allarme

(sottosistemi)

- ❖ **Sensori**: ottengono informazioni dall'ambiente
- ❖ **Attuatori**: determinano cambiamenti nell'ambiente
- ❖ Componenti di **calcolo**: eseguono delle operazioni input->output
- ❖ Componenti di **comunicazione**: permettono alle componenti del sistema di comunicare
- ❖ Componenti di **coordinamento**: coordinano le operazioni delle diverse componenti
- ❖ **Interfacce**: trasformano rappresentazioni dei dati usate da una componente del sistema in un'altra rappresentazione

Esempio: sistema di allarme

