



Università
Ca' Foscari
Venezia

INGEGNERIA DEL SOFTWARE

AA. 2013/'14

Tino Cortesi

Dipartimento di Informatica
Università Ca' Foscari
Venezia



Università
Ca' Foscari
Venezia

Obiettivi del corso

- Il corso di Ingegneria del Software presenta i metodi, le tecniche e gli strumenti fondamentali di documentazione della specifica, analisi e progetto di sistemi software **complessi** da un punto di vista architetturale
- A lezione daremo particolare spazio alla parte di analisi/specifica dei requisiti e progettazione usando UML.
- Il corso sarà accompagnato dalla acquisizione di competenze relative all'utilizzo di ambienti di sviluppo e alla realizzazione di un progetto di gruppo che ha come obiettivo lo sviluppo di apps per WindowsPhone



Università
Ca' Foscari
Venezia

Competenze in Uscita

- Riconoscere un modello di processo
- Avere un'idea dei problemi affrontati dalla ingegneria dei sistemi
- Conoscere problemi, attività e prodotti della ingegneria dei requisiti
- Conoscere le principali attività e metodi di gestione dei progetti di sviluppo del software
- Avere le nozioni di base della definizione dei requisiti del software e saper specificare semplici requisiti
- Aver acquisito le nozioni di base della progettazione del software e della documentazione relativa
- Conoscere le principali attività e metodi di verifica e validazione
- Conoscere le principali attività e metodi di stima della qualità del software, del processo e della gestione del progetto.



Università
Ca' Foscari
Venezia

Strumenti di lavoro

Area wiki riservata:

http://blogs.unive.it/groups/software_engineering/

Lezioni: Lunedì e Giovedì dalle 9.00 alle 10.25 (puntuali)

Ricevimento: su appuntamento (via e-mail)



Università
Ca' Foscari
Venezia

Testi di riferimento

- Roger Pressmann,
**Principi di Ingegneria del
Software**, quinta ed.
McGraw Hill
ISBN 9788838662164

oppure

- Ian Sommerville,
Ingegneria del Software,
settima ed.
Pearson Education,
ISBN 9788871923543





Università
Ca' Foscari
Venezia

Testi integrativi

- Sulle metodologie object-oriented in generale:
 - L. Maciaszek, Sviluppo di Sistemi Informativi con UML, Addison Wesley.
- Sull'ingegneria dei requisiti:
 - Wieringa, Design Methods for Reactive Systems, Elsevier.
- Sulla verifica e la validazione e stima dei costi:
 - Ghezzi, Jazayeri, Mandrioli, Ingegneria del Software, 2a Edizione, Pearson;
 - Pfleeger, Atlee, Software Engineering: International Edition:3/e Prentice Hall, 2006.
- Sulla notazione UML: Fowler, UML Distilled, Pearson, 2004
- Esempi commentati su UML:
 - Binato, Fuggetta, Sfardini, Ingegneria del Software: Creatività e metodo, Pearson, 2006;
 - Baresi, Lavazza, Pianciamore, Dall'idea al codice con UML 2 Pearson, 2006; L. Maciaszek, Sviluppo di Sistemi Informativi con UML, Addison Wesley;
 - Collana SHAUM - UML.



Università
Ca' Foscari
Venezia

Modalità d'esame

Per chi frequenta (min 70% delle lezioni):

1. Essere presente alle lezioni e alle attività di laboratorio
2. Rispettare le consegne intermedie:
 - Piano di progetto (10/10/2013)
 - Documento di analisi e specifica (30/10/2013)
 - Documento di progettazione (30/11/2013)
3. caricare le apps su WindowsPhone marketplace (28/02/2014)

Per chi non frequenta:

Prova orale: discussione della documentazione completa relativa alla realizzazione di una app e sui contenuti del corso.



Università
Ca' Foscari
Venezia

Piano delle lezioni

- Introduzione all'IS
- Ingegneria di Sistema
- Project Management
- Struttura Organizzativa
- Il Piano di Business
- Il Piano di Progetto
- Modelli di processo
- Analisi dei Requisiti
- Documento dei Requisiti
- Modelli UML
- Principi di Progettazione del Software
- Progettazione Architeturale
- Modelli di progettazione Orientata ad Oggetti
- Progettazione di Interfacce Utenti
- Metodi di Verifica e Validazione
- Defect Testing - Tecniche di Inspection e Walkthrough
- Analisi Statica



Università
Ca' Foscari
Venezia

Perché un corso di Ingegneria del Software?

- E' importante distinguere:
 - i sistemi semplici (uno sviluppatore, un utente, prodotto sperimentale)
 - dai sistemi “hard” (molti sviluppatori, molti utenti, “prodotto vero”)
- L'esperienza acquisita nello sviluppo di sistemi semplici non è scalabile! Analogia con la costruzione di un ponte:
 - Su un ruscello = facile, basta una persona
 - Sul ponte di Messina ... ?
- Il problema è la complessità
 - UNIX contiene 4 milione di linee di codice
 - Windows 2000 contiene 10^8 linee di codice
- Focus sul lavoro di team

Perchè preoccuparsi?

Robert Cringely (giornalista scientifico):

“...se l'automobile avesse seguito lo stesso sviluppo del computer,
una Rolls-Royce costerebbe oggi 100 \$,
farebbe un milione di chilometri con 5 litri...

**ed esploderebbe una volta all'anno
causando la morte di tutti i passeggeri**



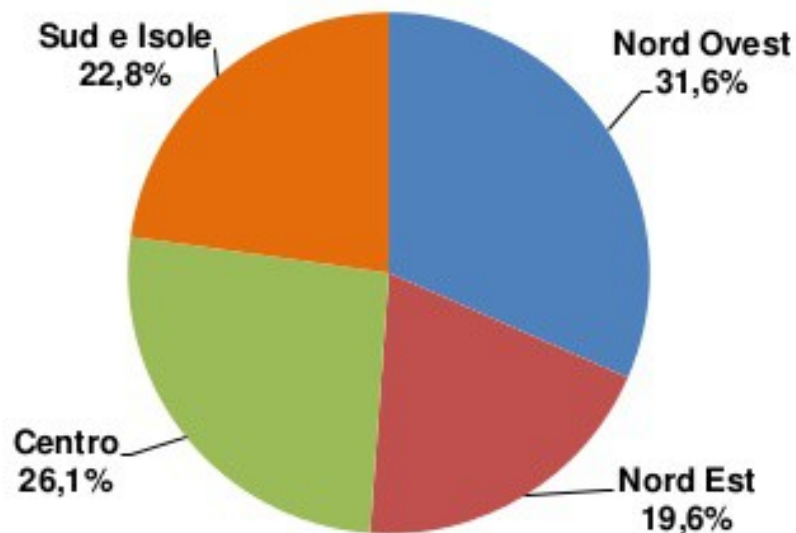
Il contesto

Caratteristiche strutturali del settore ICT in Italia (2012)

Numerosità

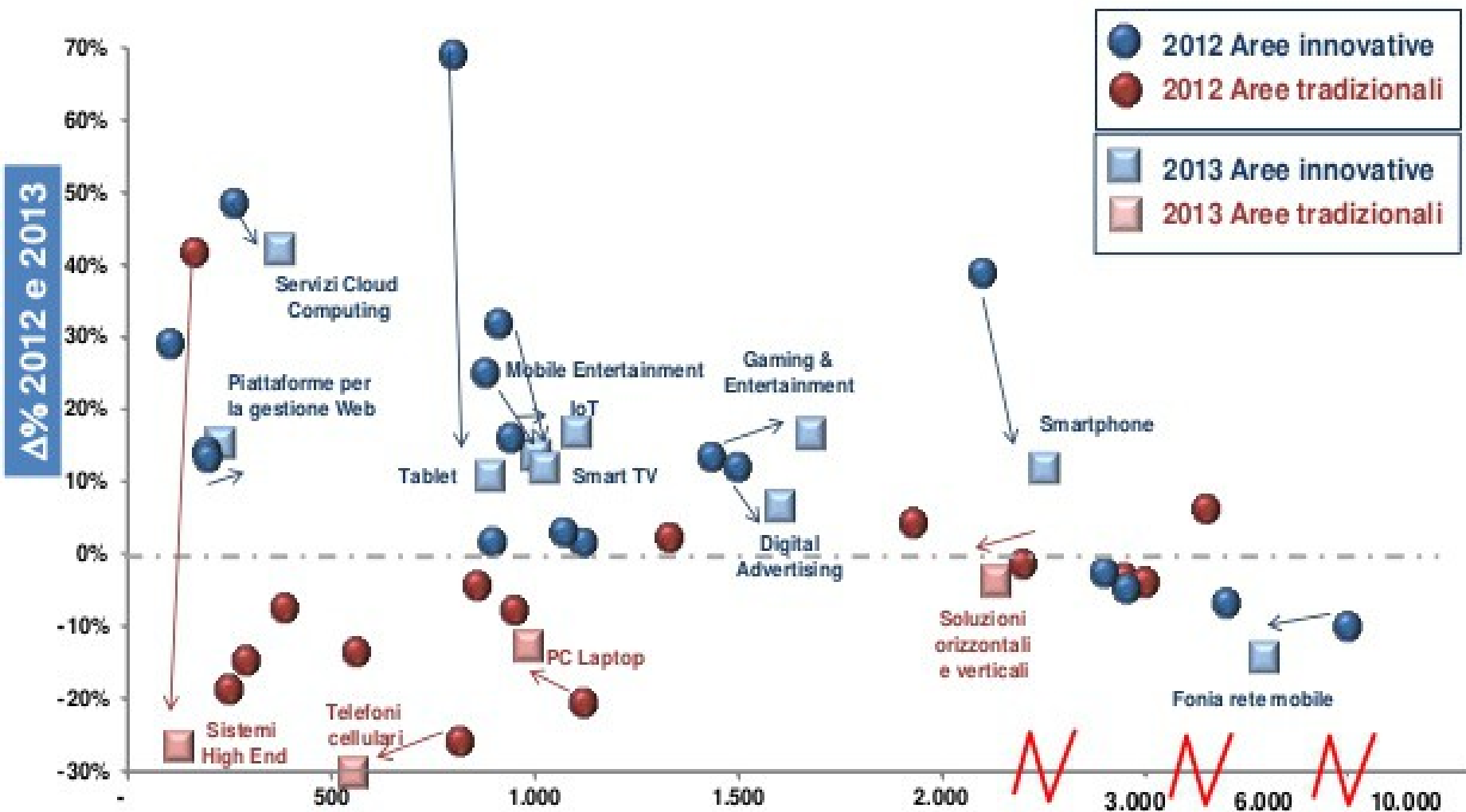
N° imprese attive	89.042
Addetti	464.562
Dimensione media (N° addetti per impresa)	5,2

Distribuzione geografica



Fonte: Elaborazioni NetConsuting su fonti varie

L'andamento dei principali prodotti e servizi del Global Digital Market (2012/2013)



Fonte: Assinform / NetConsuting

Valore del mercato 2012/2013 M€

Milano, 13 giugno 2013

5



Assinform

L'Universo delle Apps

N° Apps scaricate (mld)

Valore mercato Apps nel mondo (Mld\$)

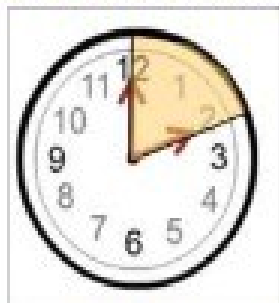


\$ 15,5
2012

\$ 25
2013

+62%

Ore spese in media / gg per utente



63%
Tasso
refresh Apps

N° medio Apps usate
8

Categorie
più sviluppate



A quali device si rivolgono



11 Mln utenti

Fonte: Elaborazioni NetConsulting su Gartner e GIGAOM Pro



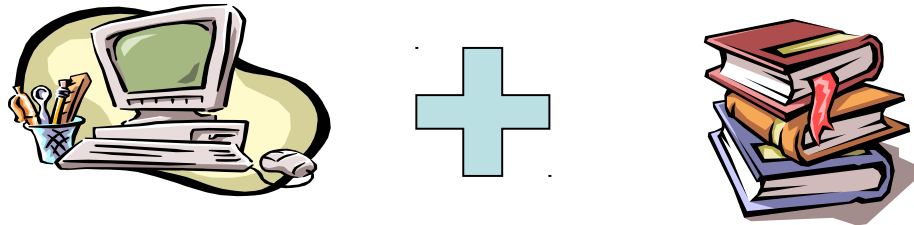
Università
Ca' Foscari
Venezia

Ingegneria del Software

Necessità di ingegnerizzazione	Categoria	Numero tipico di programmatori	Durata tipica	Dimensione del prodotto in righe di codice	Esempi	Analogia con l'edilizia
	Banale	1	1–2 settimane	< 500	Compiti a casa per studenti	Piccola ristrutturazione della casa
	Piccolo	1–3	Poche settimane o mesi	500–2 000	Progetti di gruppo per studenti, compiti avanzati	Aggiunta di un locale
	Medio	2–5	Da qualche mese a un anno	2 000–10 000	Progetti di ricerca, semplice software di produzione come assemblatori, editor, applicazioni ricreative ed educative	Casa monofamiliare
	Grande	5–25	1–3 anni	10 000–100 000	La maggior parte delle applicazioni attuali – elaboratori di testi, fogli elettronici, sistemi operativi per piccoli computer, compilatori	Piccolo supermercato
	Molto grande	25–100	3–5 anni	100 000–1 milione	Sistemi di prenotazione per aerei, sistemi di controllo del magazzino per multinazionali	Grande palazzo di uffici
	Grandissimo	>100	> 5 anni	> 1 milione	Sistemi operativi real-time su larga scala, sistemi militari avanzati, reti di telecomunicazioni internazionali	Grattacielo imponente

Cos'è un prodotto software?

- Qualcosa di più di un insieme di linee di codice...



- Un insieme di linee di codice, strutturato in packages
- Tutta la documentazione che descrive la struttura del sistema
- I dati di configurazione, che permettono di installarlo
- Il manuale utente



Prodotti software: una prima classificazione

1. System software – operating systems, drivers, compilers, etc.
2. Application software – Custom business apps.
3. Engineering/Scientific software – Mentor Graphics, ANSYS.
4. Embedded software – Cell phones, PDAs.
5. Product software – Word, Excel.
6. Web Applications
7. Artificial Intelligence software



Università
Ca' Foscari
Venezia

Prodotti Software: un'altra classificazione

- **Prodotti Generici**
 - Sistemi stand-alone prodotti da un'organizzazione di sviluppo e venduti sul mercato ad ogni cliente
- **Prodotti Dedicati**
 - Sistemi che sono commissionati da un cliente specifico e sviluppati appositamente
- La maggior spesa di software riguarda sistemi generici, ma il maggior sforzo di sviluppo è su prodotti dedicati
- La differenza principale? Chi dà la specifica del prodotto (il produttore o il consumatore).

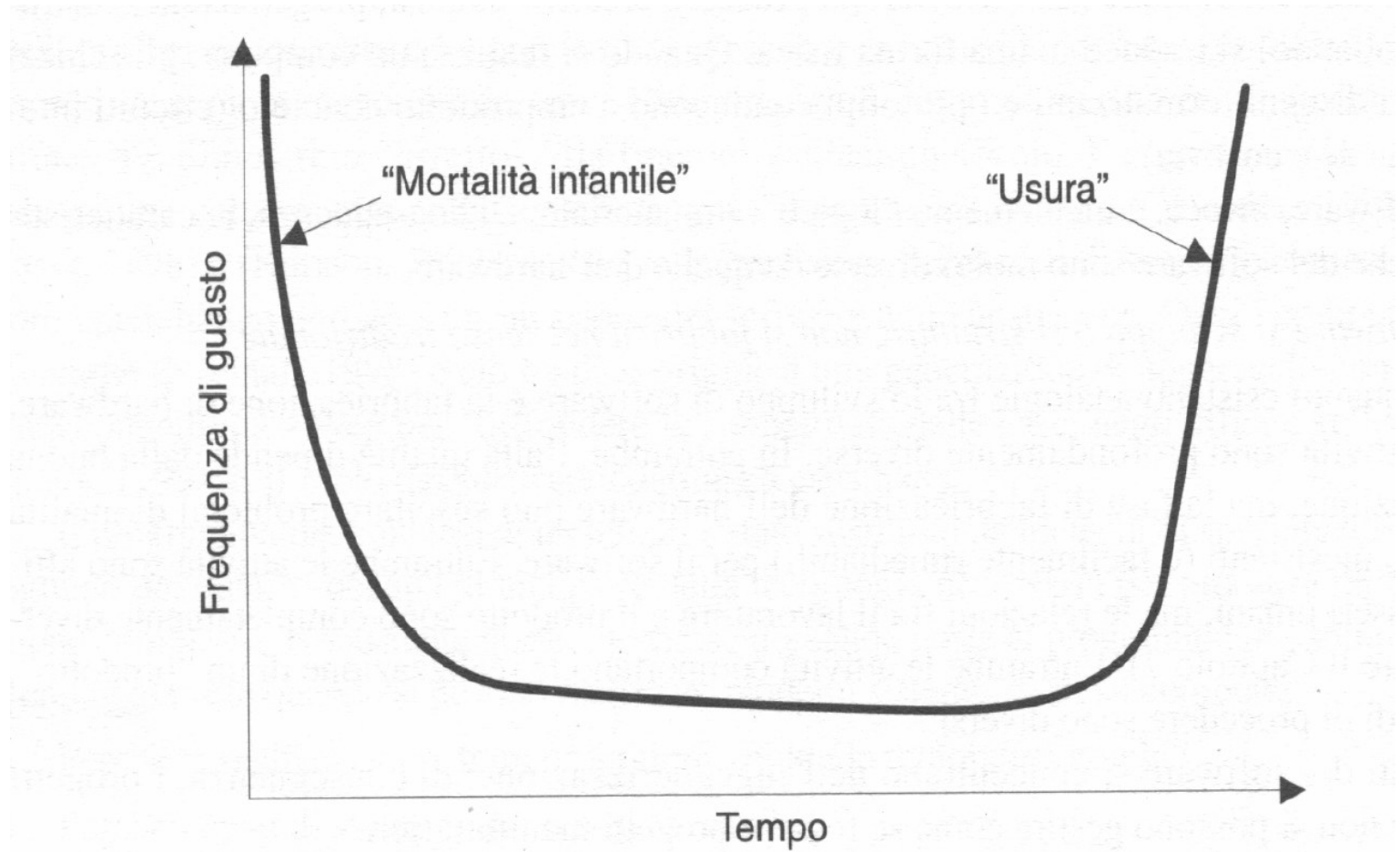


Università
Ca' Foscari
Venezia

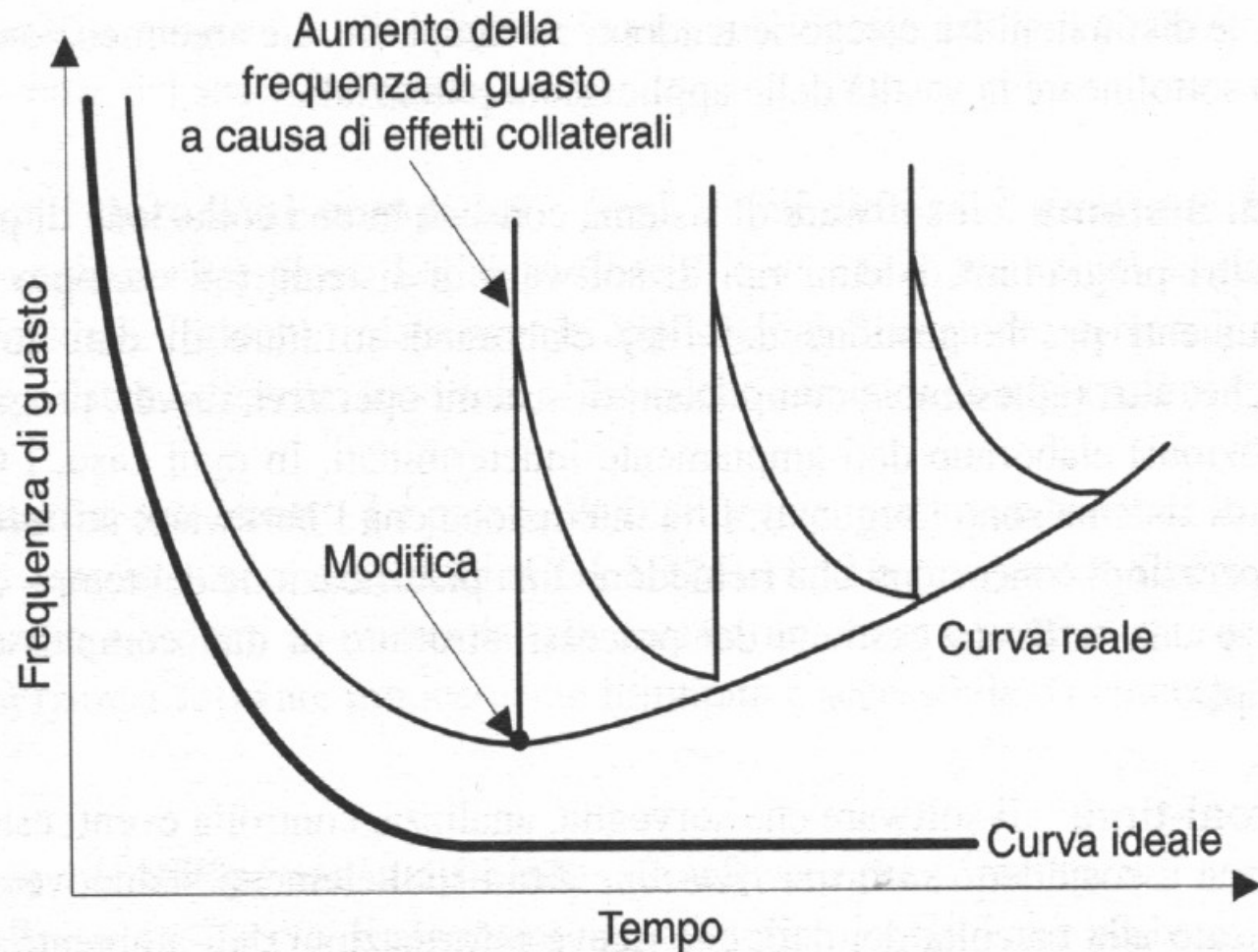
Le caratteristiche del prodotto software

- Il software si sviluppa o si struttura, non si “costruisce” come negli altri processi manifatturieri
- Lo sviluppo del software e la produzione dell'hardware sono due processi profondamente diversi
- In comune c'è la tensione a semplificare la produzione riducendola ad un “Assemblaggio di componenti”

Curva dei guasti per l'hardware



Curva dei guasti per il software





Università
Ca' Foscari
Venezia

Cos'è l'Ingegneria del Software?

- Definizione IEEE:

“The application of systematic, disciplined,
quantifiable approach to the development,
operation, and maintenance of software;
that is,
the application of engineering to software.”

Cos'è l'ingegneria del software?

- “Software engineering” è una disciplina che cerca di fornire le **regole** per il processo di produzione del software
- Un ingegnere del software dovrebbe:
 - adottare un approccio sistematico e organizzato al proprio lavoro
 - usare strumenti e tecniche appropriate, che dipendono dal problema che deve essere risolto, dai vincoli presenti e dalle risorse disponibili.



Attributi di qualità di un prodotto software

Le qualità su cui si basa la valutazione di un sistema software possono essere

- *interne*, se riguardano le caratteristiche legate alle scelte implementative e non sono visibili agli utenti;
- *esterne*, se riguardano le funzionalità fornite dal sistema e sono visibili agli utenti.

Le due categorie sono legate, infatti non è possibile ottenere qualità esterne se il sistema non gode di qualità interne.



Università
Ca' Foscari
Venezia

Attributi di qualità di un prodotto software

Correttezza - un sistema è corretto se rispetta le specifiche.

Affidabilità (reliability) - un sistema è affidabile se l'utente può dipendere da esso.

Robustezza - un sistema è robusto se si comporta in modo ragionevole anche in circostanze non previste dalle specifiche.

Efficienza - un sistema è efficiente se usa bene le risorse di calcolo.

Facilità d'uso - un sistema è facile da usare se l'interfaccia che presenta all'utente gli permette di operare in modo naturale.



Università
Ca' Foscari
Venezia

Qualità del software (ctd.)

Verificabilità - un sistema è verificabile se le sue caratteristiche sono verificabili.

Riusabilità - un sistema è riusabile se può essere usato, in tutto o in parte, per costruire nuovi sistemi.

Portabilità - un software è portabile se può funzionare su più piattaforme hardware/software.

Facilità di manutenzione - un sistema è facile da mantenere se

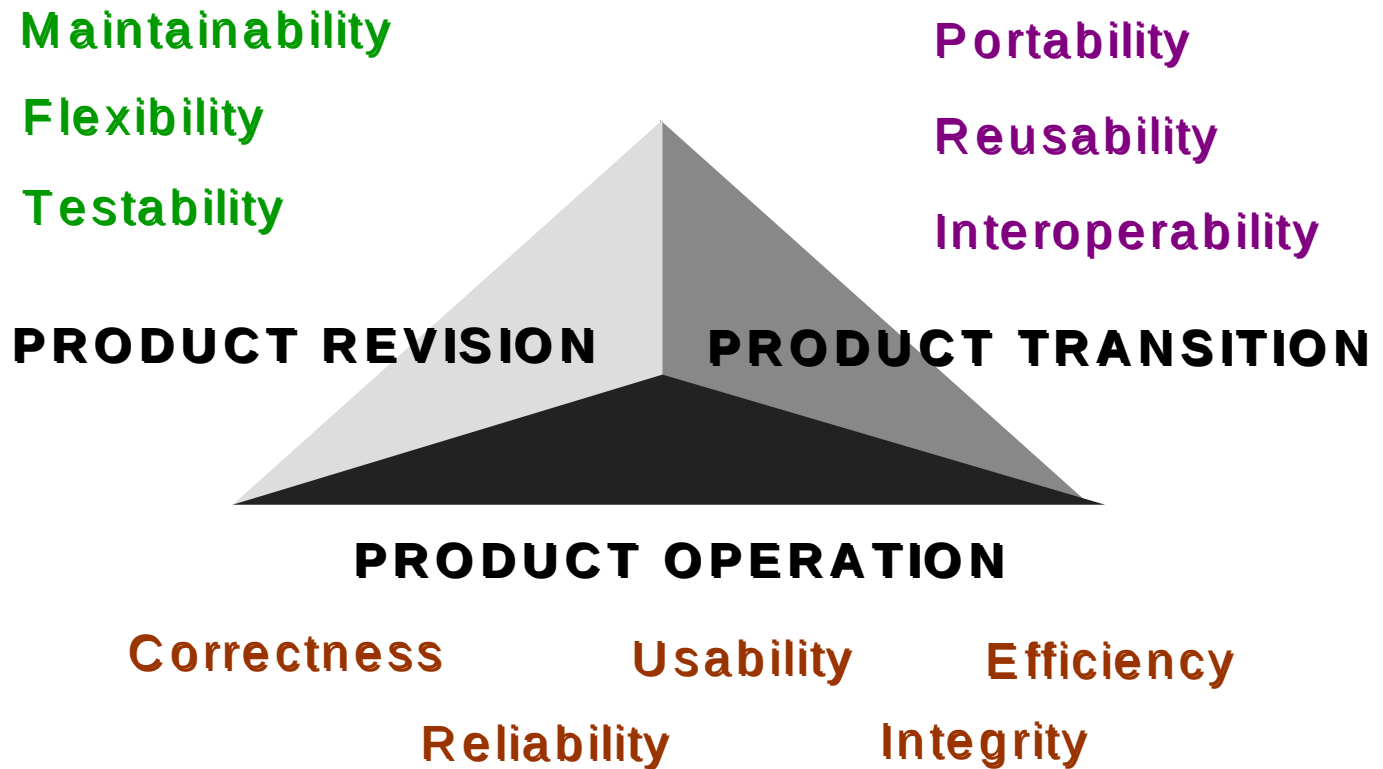
- è strutturato in modo tale da facilitare la ricerca degli errori,
- la sua struttura permette di aggiungere nuove funzionalità al sistema,
- la sua struttura permette di adattarlo ai cambiamenti del dominio applicativo.

Interoperabilità - abilità di un sistema di cooperare con altri sistemi, anche di altri produttori.



Università
Ca' Foscari
Venezia

Il triangolo di McCall





Università
Ca' Foscari
Venezia

Ingegneria del software e informatica

- L'informatica è una scienza: il “cuore” sono i fondamenti teorici: linguaggi – algoritmi – complessità – formalismi ecc.
- L'ingegneria del software ha a che fare con aspetti più “pratici”: come pianificare e sviluppare la produzione di software di qualità.
- Ad un ingegnere del software le conoscenze di base dell'informatica servono quanto la fisica ad un ingegnere elettrico



Università
Ca' Foscari
Venezia

Processo di produzione software

- Il processo di produzione software è un insieme di attività il cui fine complessivo è
 - lo **sviluppo** di un prodotto software oppure
 - la **modifica** di un prodotto software



Università
Ca' Foscari
Venezia

Attributi di qualità del processo di produzione software

- Comprensibilità
- Visibilità
- Supportabilità (CASE)
- Accettabilità
- Robustezza
- Mantenibilità
- Rapidità



Università
Ca' Foscari
Venezia

Problemi nel processo di sviluppo del software

- Specifiche incomplete/incoerenti
- Mancanza di distinzione tra specifica, progettazione e implementazione
- Assenza di un sistema di validazione
- Il software non si consuma: la manutenzione non significa riparare alcune componenti “rotte”, ma modificare il prodotto rispetto a nuove esigenze

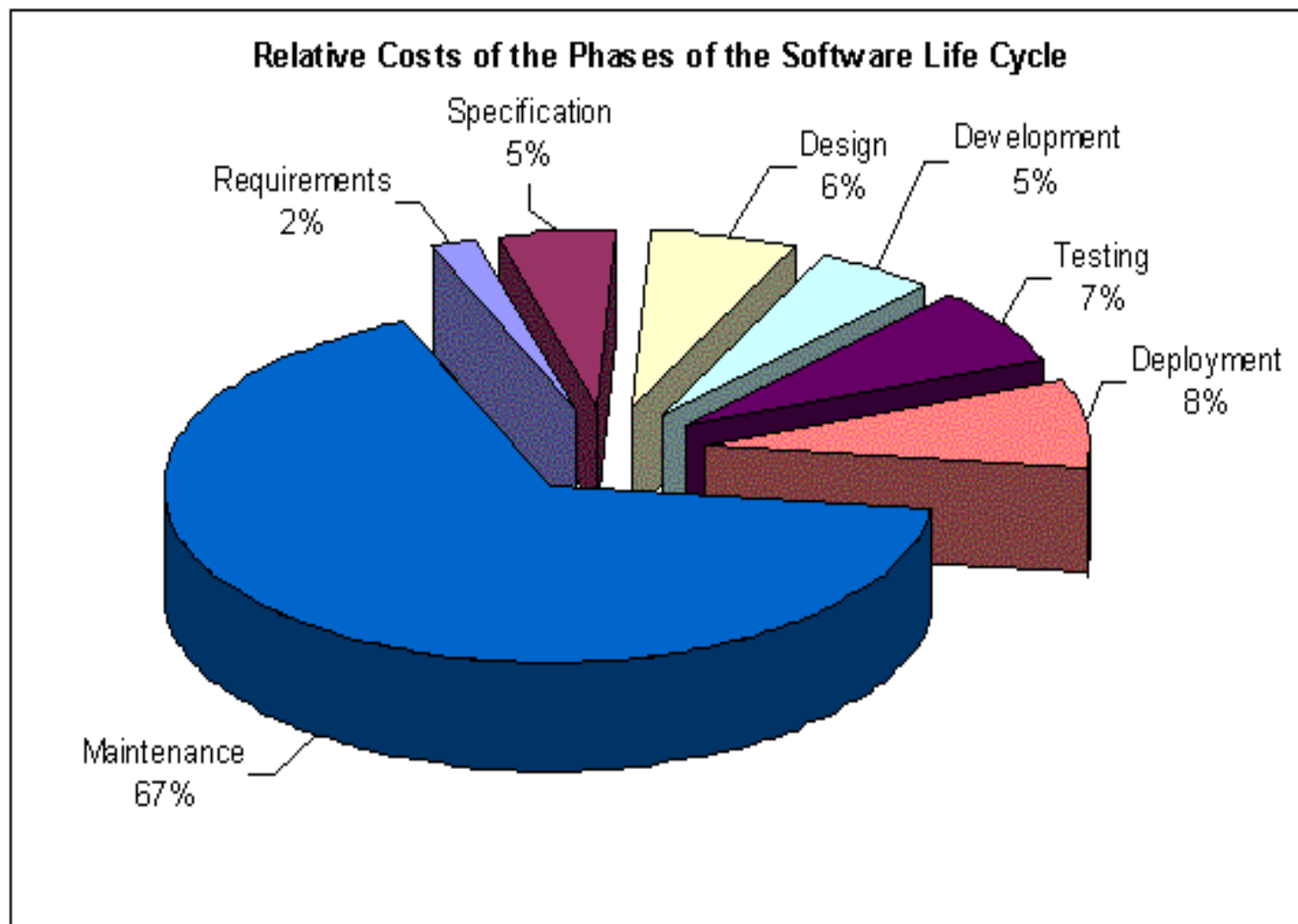


Università
Ca' Foscari
Venezia

I costi di un prodotto software

- All'incirca il 60% dei costi è legato allo sviluppo, il 40% sono costi per la verifica e validazione (testing).
- I costi variano a seconda del tipo di sistema che deve essere sviluppato e da requisiti quali la performance o l'affidabilità del sistema.
- La distribuzione di costi nelle varie fasi del processo di produzione del software dipende dal modello di processo.

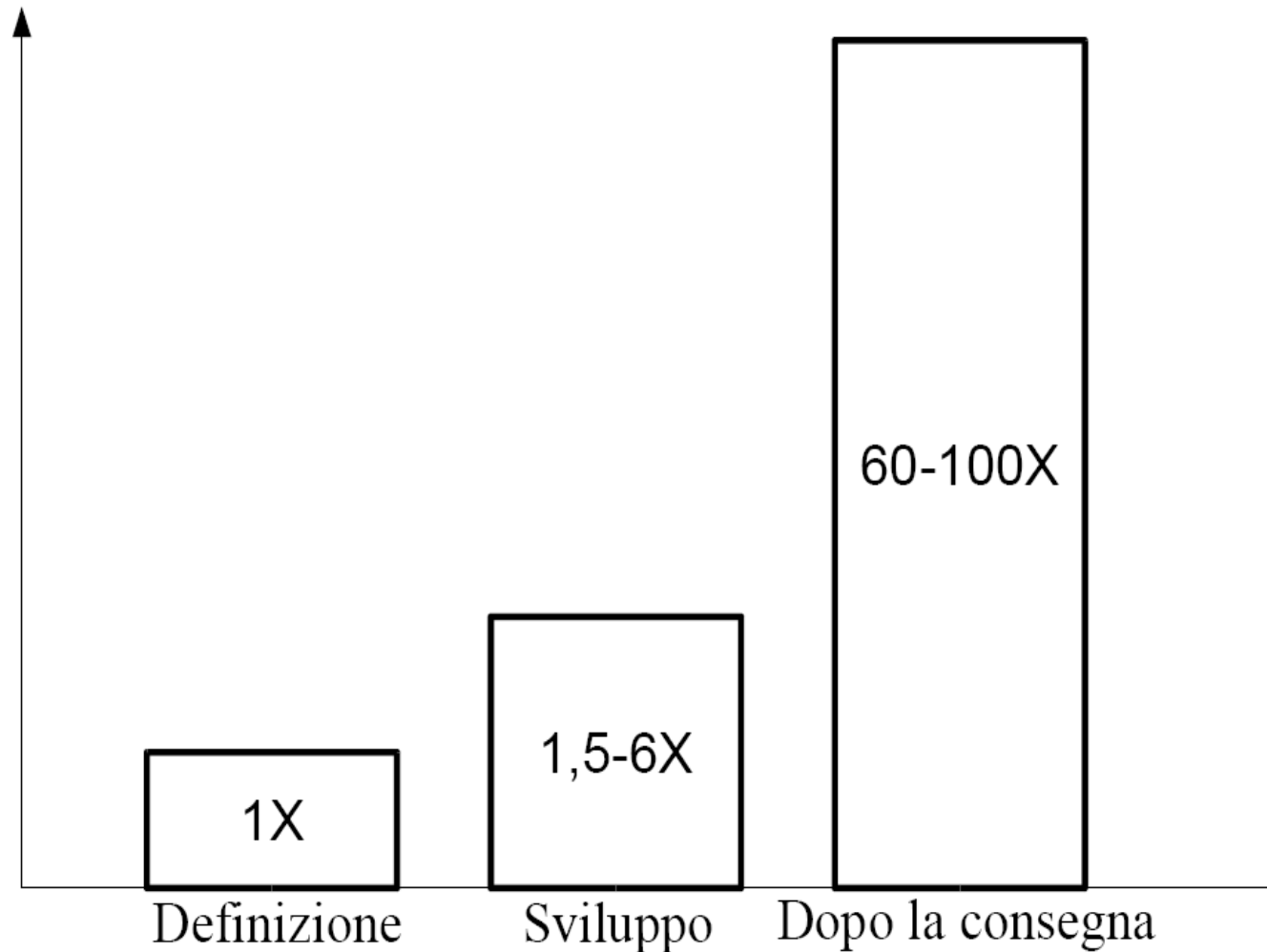
Costi relativi nel ciclo di vita del sw





Università
Ca' Foscari
Venezia

Costo di una modifica





Università
Ca' Foscari
Venezia

Le sfide ed i problemi

- **Skill shortage**

Secondo una ricerca condotta da EITO nel 2003, in Europa ci sono 1 milione e 700 mila posti nell'Information & Communication Technology che non riescono ad essere coperti perché mancano le giuste risorse. In Italia la carenza di risorse è calcolata in 167 mila unità.

- **Skill gap**

Problema ancor più grave è lo Skill gap, ossia la distanza fra le competenze tecniche richieste dal mercato ICT e la professionalità degli operatori del settore.

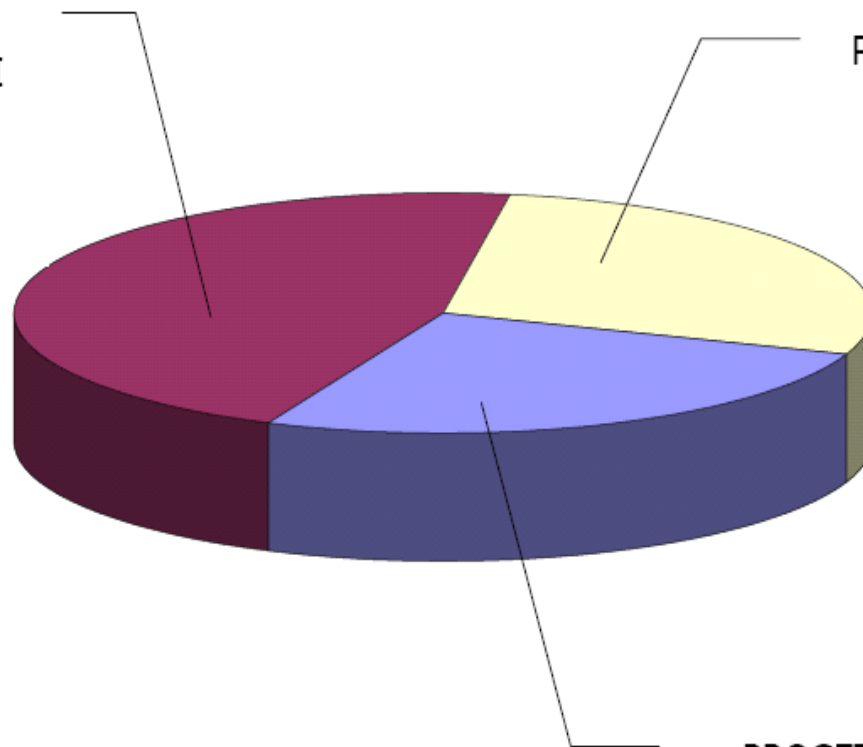
- **Alta percentuale di fallimento**

Indagine dello Standish Group, basata su un campione di 28.000 progetti e pubblicata da Computer Weekly il 9 luglio 1998:



Università
Ca' Foscari
Venezia

PROGETTI IN RITARDO
CON COSTI IMPREVISTI
CON FUNZIONALITA'
INADEGUATE
46%



PROGETTI FALLITI
28%

PROGETTI RIUSCITI
26%



Università
Ca' Foscari
Venezia

Concludendo

L'**ingegneria del software** tratta della realizzazione di sistemi software di dimensioni e complessità tali da richiedere uno o più team di persone.

L'ingegneria del software è la disciplina tecnologica e manageriale che riguarda la *produzione sistematica* e la *manutenzione* dei prodotti software che vengono sviluppati e modificati entro tempi e costi preventivati.

L'ingegneria del software è un insieme di teorie, metodi e strumenti, sia di tipo tecnologico che organizzativo, che consentono di produrre applicazioni con le desiderate caratteristiche di *qualità*.