



# 1. Introduzione

## Cosa, Quando, Come e Perché

Andrea Polini

Ingegneria del Software  
Corso di Laurea in Informatica

# Sommario

- 1 Informazioni Generali
- 2 Ingegneria del Software - generalità
- 3 4 P's

# Sommario

- 1 Informazioni Generali
- 2 Ingegneria del Software - generalità
- 3 4 P's

# Corso e Docente

- Andrea Polini

- e-mail: [andrea.polini@unicam.it](mailto:andrea.polini@unicam.it)
- ricevimento
  - il Giovedì dalle ore 16:00
  - ancor più facile concordando appuntamento con il docente

- Ingegneria del Software

- orario:
  - Lunedì dalle 14 alle 16
  - Mercoledì dalle 14 alle 16 (Prof. Morichetta - inizio sarà comunicato)
  - Giovedì dalle 14 alle 16
- web: [Wiki didattica pagina dedicata](#)

- Esame:

- 2 e 23 febbraio 2023 – ore 15
- 8 giugno 2023, 6 luglio 2023 – ore 15
- 7 e 28 settembre 2023 – ore 15
- 8 e 29 febbraio 2024 – ore 15

# Obiettivi Formativi

- Al termine del corso il vostro bagaglio di competenze dovrebbe essersi arricchito con:
  - Metodologie, tecniche e strumenti per la modellazione e lo sviluppo di **sistemi software complessi**
  - Capacità di collaborare in team per lo sviluppo di un sistema complesso
- Al termine del corso sarete capaci di:
  - Comprendere ed identificare le problematiche che influenzano il progetto di un sistema software complesso
  - Selezionare e giudicare gli approcci, le tecniche e gli strumenti migliori per lo sviluppo di un sistema software complesso
  - Utilizzare UML per la modellazione di un sistema software

# Materiale di studio

## ● Testi di riferimento:



Craig Larman

*Applicare UML e i Pattern*, 5<sup>a</sup> Ed. Italiana  
Pearson, 2020.



Michael Lewrick, Patrick Link, Larry Leifer

*Manuale di Design Thinking – Progettare la trasformazione digitale di team, prodotti, servizi ed ecosistemi*  
Edizioni LSWR

## ● Testi di consultazione:



Ian Sommerville

*Ingegneria del Software*, 10<sup>a</sup> Ed. Italiana  
Pearson, 2017.



Jim Arlow, Ila Neustadt

*UML2 e Unified Process - Analisi e Progettazione Object-Oriented*, 2<sup>a</sup> Ed. Italiana  
McGraw-Hill, 2006.

## ● Materiale fornito/indicato dal docente

# Esame!!!

- I corsi di Ingegneria del Software (IdS) e di Laboratorio di Ingegneria del Software (Lab) danno luogo ad un solo esame di 9 crediti (CFU).
- **Modalità di Esame:**
  - **Progetto:** la prova di progetto si svolge in gruppi da 2 o 3 (max) elementi e richiederà lo sviluppo di un sistema software applicando le metodologie e le tecniche apprese a lezione.
    - UML, Java, JavaScript e framework concordati con il docente
    - Eclipse, Git
  - **Orale:** la prova orale si svolge nelle date che saranno concordate con il docente. In particolare nella data indicata in ESSE3 sarà definito un calendario per la discussione dei progetti di gruppo e degli orali individuali. L'orale individuale richiederà allo studente di svolgere semplici esercizi e di rispondere a quesiti relativi ai contenuti del corso. **Possono svolgere la prova orale solo gli studenti che siano membri di gruppi che abbiano già svolto la presentazione/discussione del progetto.**

# Suggerimenti per la prova orale

## Raccomandazioni per la prova orale:

- considerare sempre che "...le parole sono il mezzo per giungere al significato..." "V per Vendita"
- <https://www.youtube.com/watch?v=qtP3FWRo60w>



# Suggerimenti per la prova orale

## Raccomandazioni per la prova orale:

- considerare sempre che “...le parole sono il mezzo per giungere al significato ...”  
“V per Vendetta”
- <https://www.youtube.com/watch?v=qtP3FWRo6Ow>

# Suggerimenti per la prova orale

## Raccomandazioni per la prova orale:

- considerare sempre che “...le parole sono il mezzo per giungere al significato ...”  
“V per Vendetta”
- <https://www.youtube.com/watch?v=qtP3FWRo6Ow>

# Contenuto del Corso

## Parte Teorica

- Introduzione e caratterizzazione dell'Ingegneria del Software (IdS)
- Processi di Sviluppo Software
- Qualità del Software e dei processi di sviluppo
- Ingegneria dei Requisiti
  - Design Thinking
- Design
  - UML
  - Architettura del Software
  - Sistemi a Componenti
  - Sistemi Distribuiti
- Sviluppo del Software
  - Programmazione Orientata agli oggetti
  - Design Patterns
- Verifica e Validazione (Cenni)
  - Verifica Statica
  - Verifica Dinamica
- Manutenzione del Software (Cenni)

# Sommario

- 1 Informazioni Generali
- 2 Ingegneria del Software - generalità
- 3 4 P's

# Cos'è il software?

- Con il termine software in generale si identifica un programma che specifica le istruzioni che un calcolatore dovrà eseguire al fine di raggiungere uno scopo. In questo contesto il termine identifica anche **tutti i documenti** (manuali, file di configurazione, file di installazione, etc..) che lo descrivono e che sono stati messi a punto durante le **varie fasi della produzione** del sistema. Includiamo nella definizione di software anche le sue caratteristiche strutturali, funzionali, non-funzionali . . . , e le sue possibili configurazioni
- l'ingegneria del software è la disciplina che agirà globalmente su tutte le possibili componenti del software al fine di poter produrre software di "qualità"
- Qualità??

# Cos'è il software?

- Con il termine software in generale si identifica un programma che specifica le istruzioni che un calcolatore dovrà eseguire al fine di raggiungere uno scopo. In questo contesto il termine identifica anche **tutti i documenti** (manuali, file di configurazione, file di installazione, etc..) che lo descrivono e che sono stati messi a punto durante le **varie fasi della produzione** del sistema. Includiamo nella definizione di software anche le sue caratteristiche strutturali, funzionali, non-funzionali . . . , e le sue possibili configurazioni
- l'ingegneria del software è la disciplina che agirà globalmente su tutte le possibili componenti del software al fine di poter produrre software di “qualità”
- Qualità??

# Cos'è il software?

- Con il termine software in generale si identifica un programma che specifica le istruzioni che un calcolatore dovrà eseguire al fine di raggiungere uno scopo. In questo contesto il termine identifica anche **tutti i documenti** (manuali, file di configurazione, file di installazione, etc..) che lo descrivono e che sono stati messi a punto durante le **varie fasi della produzione** del sistema. Includiamo nella definizione di software anche le sue caratteristiche strutturali, funzionali, non-funzionali . . . , e le sue possibili configurazioni
- l'ingegneria del software è la disciplina che agirà globalmente su tutte le possibili componenti del software al fine di poter produrre software di “qualità”
- Qualità??

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- Software embedded
- Software di produttività
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...



# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- Software embedded
- Software di produttività
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- **Software scientifico e tecnico**
- Software embedded
- Software di produttività
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- **Software embedded**
- Software di produttività
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- Software embedded
- **Software di produttività**
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- Software embedded
- Software di produttività
- **Applicazioni Web**
- Software per l'intelligenza artificiale
- ...

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- Software embedded
- Software di produttività
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...

# Software Categorie di Massima

Si possono dare molte classificazioni del software. In modo grossolano considerando il loro scopo ed ambito applicativo possiamo suddividere il software in:

- Software di sistema
- Software applicativo
- Software scientifico e tecnico
- Software embedded
- Software di produttività
- Applicazioni Web
- Software per l'intelligenza artificiale
- ...

# Un po' di storia...

- **Therac-25 (1985-1987)**

- Diversi pazienti furono irradiati con dosi massicce di raggi X da un'apparecchiatura per terapie contro il cancro. Sistema controllato da sistema di input software contenente un baco. Almeno 5 persone furono uccise.

- London Ambulance Service (1992)

- Il LAS introdusse un sistema automatico di selezione e invio delle unità mediche sostituendo le procedure manuali fino a quel punto in uso, introducendo meccanismi di selezione più avanzati. L'appalto fu bandito a giugno 1991 ed assegnato ad un costo di £1,1 milioni. Furono fissate precise scadenze non negoziabili. Il sistema mal progettato e realizzato causò, quando messo in opera, altissimi ritardi nell'arrivo delle unità (anche 11 ore) e circa 30 persone morirono a causa dei ritardi. Il software fu poi dismesso.



# Un po' di storia...

- **Therac-25 (1985-1987)**
  - Diversi pazienti furono irradiati con dosi massicce di raggi X da un'apparecchiatura per terapie contro il cancro. Sistema controllato da sistema di input software contenente un baco. Almeno 5 persone furono uccise.
- **London Ambulance Service (1992)**
  - Il LAS introdusse un sistema automatico di selezione e invio delle unità mediche sostituendo le procedure manuali fino a quel punto in uso, introducendo meccanismi di selezione più avanzati. L'appalto fu bandito a giugno 1991 ed assegnato ad un costo di £1,1 milioni. Furono fissate precise scadenze non negoziabili. Il sistema mal progettato e realizzato causò, quando messo in opera, altissimi ritardi nell'arrivo delle unità (anche 11 ore) e circa 30 persone morirono a causa dei ritardi. Il software fu poi dismesso.

# Un po' di storia...

...continua

- **Denver Airport Baggage System (1995)**

- L'aeroporto di Denver doveva essere inaugurato nel 1995 e doveva basarsi su di un avveniristico sistema di smistamento dei bagagli. Il sistema software avrebbe dovuto controllare 26 miglia di nastri trasportatori indirizzando correttamente i bagagli. Il sistema però non riuscì mai a fornire le garanzie necessarie e ritardò l'apertura dello scalo per mesi al costo di un milione di dollari al giorno. Alla fine il progetto fu abbandonato.

- **Il vettore spaziale Ariane 5 (1996)**

Il vettore Ariane 5 dell'agenzia spaziale europea esplose durante il volo inaugurale il 4 giugno 1996 dopo 39 secondi dal decollo. L'esplosione fu causata da un segnale di autodistruzione emesso dal sistema di controllo che erroneamente, a causa di buffer overflow, pensava di essere fuori rotta. La progettazione e costruzione del vettore era costata 500 milioni di dollari.

# Un po' di storia...

...continua

- **Denver Airport Baggage System (1995)**

- L'aeroporto di Denver doveva essere inaugurato nel 1995 e doveva basarsi su di un avveniristico sistema di smistamento dei bagagli. Il sistema software avrebbe dovuto controllare 26 miglia di nastri trasportatori indirizzando correttamente i bagagli. Il sistema però non riuscì mai a fornire le garanzie necessarie e ritardò l'apertura dello scalo per mesi al costo di un milione di dollari al giorno. Alla fine il progetto fu abbandonato.

- **Il vettore spaziale Ariane 5 (1996)**

- Il vettore Ariane 5 dell'agenzia spaziale europea esplose durante il volo inaugurale il 4 giugno 1996 dopo 39 secondi dal decollo. L'esplosione fu causata da un segnale di autodistruzione emesso dal sistema di controllo che erroneamente, a causa di buffer overflow, pensava di essere fuori rotta. La progettazione e costruzione del vettore era costata 500 milioni di dollari.

# Un po' di storia

...continua

- **Mars Climate Orbiter (1999)**

- La sonda Mars Climate Orbiter fu “smarrita” dopo il decollo con una perdita economica di circa 125 milioni di dollari. Successive investigazioni sulle responsabilità appurarono che in fase di progettazione non fu specificato il sistema di misura da utilizzare. Differenti team di sviluppo, nello sviluppo dei vari moduli fecero differenti assunzioni sul sistema di misurazione da applicare (metrico decimale ed imperiale).

- **Toyota Prius (2005)**

- Errori nella programmazione delle schede di controllo dell'auto causarono il richiamo di oltre 160.000 unità. L'auto passava all'alimentazione elettrica e proseguiva per pochi chilometri prima di arrestarsi, quando lanciata ad alta velocità in particolari situazioni di marcia.

# Un po' di storia

...continua

- **Mars Climate Orbiter (1999)**

- La sonda Mars Climate Orbiter fu “smarrita” dopo il decollo con una perdita economica di circa 125 milioni di dollari. Successive investigazioni sulle responsabilità appurarono che in fase di progettazione non fu specificato il sistema di misura da utilizzare. Differenti team di sviluppo, nello sviluppo dei vari moduli fecero differenti assunzioni sul sistema di misurazione da applicare (metrico decimale ed imperiale).

- **Toyota Prius (2005)**

- Errori nella programmazione delle schede di controllo dell'auto causarono il richiamo di oltre 160.000 unità. L'auto passava all'alimentazione elettrica e proseguiva per pochi chilometri prima di arrestarsi, quando lanciata ad alta velocità in particolari situazioni di marcia.

# Un po' di storia

...continua

- **Knight Capital Group (2012)**

- The Knight Capital Group was an American global financial services firm engaging in market making, electronic execution, and institutional sales and trading. With its high-frequency trading algorithms Knight was the largest trader in U.S. equities, with a market share of 17.3% on NYSE and 16.9% on NASDAQ. The company agreed to be acquired by Getco LLC in December 2012 after an August 2012 trading error lost \$460 million.

- **Boeing**

- (2015) 787 Dreamliner - To keep a Boeing Dreamliner flying, reboot once every 246 days - that can cause its electrical system to fail and, as a result, lead to "loss of control" of the plane.
- 2019 737 MAX - The 737 MAX suffered a recurring failure in the Maneuvering Characteristics Augmentation System (MCAS) causing two fatal accidents in which 346 people died. It was subsequently grounded worldwide from March 2019 to November 2020.

# Un po' di storia

...continua

- **Knight Capital Group (2012)**
  - The Knight Capital Group was an American global financial services firm engaging in market making, electronic execution, and institutional sales and trading. With its high-frequency trading algorithms Knight was the largest trader in U.S. equities, with a market share of 17.3% on NYSE and 16.9% on NASDAQ. The company agreed to be acquired by Getco LLC in December 2012 after an August 2012 trading error lost \$460 million.
- **Boeing**
  - (2015) 787 Dreamliner - To keep a Boeing Dreamliner flying, reboot once every 248 days . . . that can cause its electrical system to fail and, as a result, lead to "loss of control" of the plane.
  - 2019 737 MAX - The 737 MAX suffered a recurring failure in the Maneuvering Characteristics Augmentation System (MCAS) causing two fatal accidents in which 346 people died. It was subsequently grounded worldwide from March 2019 to November 2020.

# Un po' di storia

...continua

Per chi volesse divertirsi a leggere altre storie:

[https://en.wikipedia.org/wiki/List\\_of\\_software\\_bugs](https://en.wikipedia.org/wiki/List_of_software_bugs)

*Standish Chaos Report 2009:*

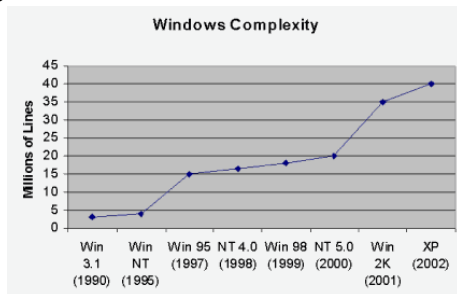
- Failed Projects: 24%
- Challenged Projects: 44%
- Successful Projects: 32%

La situazione porta alla stima di diverse centinaia di miliardi di euro persi annualmente nel settore ICT.



# Sfide nella produzione di SW

- Dimensione



- Distribuzione, Integrazione, Comunicazione
- Sicurezza (Safety) e Trust
- Eterogeneità dei linguaggi e delle piattaforme
- Integrazione e riuso di software pre-esistente
- Time-to-Market
- . . .

# Sulla dimensione del software – focus

Dimensioni del codebase di alcuni software e composizione del codebase di RedHat 7.2:

Prodotto	SLOC (righe di codice sorgente)
NASA Space Shuttle Flight Control	430K(shuttle) + 1.4M (ground)
Sun Solaris 1998-2000	7-8M
Microsoft Windows 3.1 (1992)	3M
Microsoft Windows 95	14M
Microsoft Windows 98	18M
Microsoft Windows NT (1992)	4M
Microsoft Windows NT5.0 (1998)	20M
RedHatLinux 6.2 (2000)	17M
MacOS 10.4 (2005)	86M
Linux kernel 4.2 (2016)	20.2M
Debian 7.0 (2012)	419M

Linguaggio	SLOC (%)
C	21461450 (71.18%)
C++	4575907 (15.18%)
ShellBourne	793238 (2.63%)
Lisp	722430 (2.40%)
Assembly	565536 (1.88%)
Perl	562900 (1.87%)
Fortran	493297 (1.64%)
Python	285050 (0.95%)
Tcl	213014 (0.71%)
Java	147285 (0.49%)
yacc/bison	122325 (0.41%)
Expect	103701 (0.34%)
lex/flex	41967 (0.14%)
awk/gawk	17431 (0.06%)
Objective-C	14645 (0.05%)
Ada	13200 (0.04%)
C shell	10753 (0.04%)
Pascal	4045 (0.01%)
sed	3940 (0.01%)

link

# Prodromi alla nascita dell'Ingegneria del Software

- Anni '50 appaiono i primi linguaggi di alto livello (COBOL)
  - I programmi cominciano a diventare via via più complessi
  - Programmare comincia a diventare un lavoro
- Anni '60-'70...la crisi del software
  - Le tecniche di sviluppo adottate fino a quel punto risultano non scalare
  - Gli sviluppatori sembrano incapaci di sviluppare software capace di utilizzare a fondo l'aumento nella potenza dei dispositivi hardware.
- Conferenze NATO (1968 Garmisch-D, 1969 Roma-I)
  - Vengono poste le basi per una nuova disciplina
  - sono necessarie nuove metodologie, strumenti formali, nuovi processi, nuovi strumenti di sviluppo, nuovi modi di organizzare il lavoro, etc..
  - la produzione del software si deve trasformare da un'attività artigianale ad un'attività "ingegneristica"

# Da allora....

- Molti passi sono stati fatti da allora, ma....
  - La **complessità** dei sistemi **continua a crescere** vertiginosamente
  - il software viene continuamente applicato in **nuovi domini**
  - le richieste sulle caratteristiche “**qualitative**” del software seguono lo stesso trend
  - moltissimi sistemi software falliscono o vengono cancellati dopo esser partiti e trovandosi spesso anche in una fase avanzata dello sviluppo
  - spessissimo il rilascio del software avviene in **ritardo rispetto alle scadenze stabilite**
  - altrettanto spesso la **qualità** del software rilasciato è “discutibile”

# Il progetto del software oggi - versione semiseria



How the customer explained it



How the Project Leader understood it



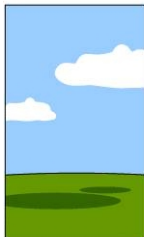
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



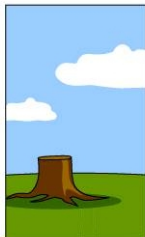
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

# Qualche definizione

- IEEE:

Applicazione di un approccio **sistematico, disciplinato e quantificabile** allo sviluppo, supporto e manutenzione del software

- Sommerville:

L'Ingegneria del Software è una disciplina ingegneristica che riguarda tutti gli aspetti della produzione del software. L'ingegnere del software deve adottare un approccio sistematico ed organizzato al suo lavoro ed utilizzare gli strumenti e le tecniche più appropriate dipendentemente dal problema da risolvere, dai vincoli di sviluppo e dalle risorse disponibili.


- Ghezzi, Jazayeri, Mandholi:

L'Ingegneria del Software è la branca della scienza dell'informazione che riguarda lo sviluppo di sistemi software le cui dimensioni richiedono l'intervento di uno o più team di sviluppo. ... programmare è principalmente un'attività personale, mentre l'ingegneria del software è essenzialmente

# Qualche definizione

- IEEE:  
Applicazione di un approccio **sistematico, disciplinato e quantificabile** allo sviluppo, supporto e manutenzione del software
- Sommerville:  
L'Ingegneria del Software è una disciplina ingegneristica che riguarda **tutti gli aspetti della produzione del software**. L'ingegnere del software deve adottare un approccio **sistematico ed organizzato** al suo lavoro ed utilizzare gli **strumenti e le tecniche più appropriate** dipendentemente dal problema da risolvere, dai vincoli di sviluppo e dalle **risorse disponibili**.
- Ghezzi, Jazayeri, Mandrioli:  
L'Ingegneria del Software è la branca della scienza dell'informazione che riguarda lo sviluppo di sistemi software le cui dimensioni richiedono l'intervento di uno o più team di sviluppo . . . programmare è principalmente un'attività personale, mentre l'ingegneria del software è essenzialmente **un'attività di team**.

## Qualche definizione

- IEEE:  
Applicazione di un approccio **sistematico, disciplinato e quantificabile** allo sviluppo, supporto e manutenzione del software
- Sommerville:  
L'Ingegneria del Software è una disciplina ingegneristica che riguarda **tutti gli aspetti della produzione del software**. L'ingegnere del software deve adottare un approccio **sistematico ed organizzato** al suo lavoro ed utilizzare gli **strumenti e le tecniche più appropriate** dipendentemente dal problema da risolvere, dai vincoli di sviluppo e dalle **risorse disponibili**.
- Ghezzi, Jazayeri, Mandrioli:  
L'Ingegneria del Software è la branca della scienza dell'informazione che riguarda lo sviluppo di sistemi software le cui **dimensioni** richiedono l'intervento di **uno o più team di sviluppo** . . . programmare è principalmente un'attività personale, mentre l'ingegneria del software è essenzialmente un'**attività di team**. 



# Qualche definizione

... continua

- Emmerich:

L'Ingegneria del Software è una branca dell'ingegneria dei sistemi che riguarda lo sviluppo di sistemi **software complessi** e di grandi dimensioni. Essa si focalizza su: obiettivi e limiti reali per i servizi forniti dai sistemi software; la precisa specifica della struttura di questi sistemi, del loro comportamento e l'implementazione di tali specifiche; le attività richieste al fine di garantire che le specifiche e gli **obiettivi siano raggiunti**; l'evoluzione di tali sistemi nel tempo. Infine essa riguarda anche i **processi, i metodi e gli strumenti** per lo sviluppo **economicamente vantaggioso e pianificato** del software.

# In evidenza

- Metodologie per lo sviluppo di sistemi di **medio/grandi dimensioni**
- Raccomanda **disciplina e sistematicità**
- non solo **il cosa viene prodotto ma anche il come viene prodotto** è di interesse per la disciplina
- Raccomanda l'introduzione di **metodi quantificabili al fine di poter paragonare differenti soluzioni possibili**
- riguarda lo sviluppo di sistemi che richiedono l'intervento **di team di sviluppo**. Dunque **comunicazione** diventa uno degli aspetti più importanti.

# In evidenza

... continua

- non riguarda soltanto la programmazione (per certi versi aspetto marginale)
- **Costi di sviluppo e tempi** sono un aspetto fondamentale dello sviluppo
- Si richiedono **capacità di gestione e di pianificazione**
- Obiettivo focale è la spinta verso la produzione di **qualità**
- Necessità di applicare **strumenti di supporto** – Computer Aided Software Engineering (CASE)

*The software developer must reconcile the abstract concepts with their concrete implementations, the mathematics of correct computation with time and space constraints deriving from physical laws and from limitation of current hardware technology*

*Bertrand Meyer*

# Why software fails

## Tipiche casistiche di fallimento:

- ▶ Over budget
- ▶ Non rispetto delle scadenze
- ▶ Non soddisfazione delle esigenze degli acquirenti
- ▶ Bassa qualità
- ▶ Difficoltà di utilizzo

## Motivazioni ricorrenti di fallimento

- ▶ Obiettivi non realistici o non ben articolati
- ▶ Scarsa qualità nella gestione del progetto
- ▶ Stime errate su costi e risorse
- ▶ Requisiti definiti in modo approssimativo
- ▶ Scarsa e non buona visibilità del processo
- ▶ Rischi non gestiti
- ▶ Scarsa comunicazione tra gli stakeholders
- ▶ Incapacità nella gestione della complessità del progetto

# Sommario

- 1 Informazioni Generali
- 2 Ingegneria del Software - generalità
- 3 4 P's**

# 4 P's

- ▶ **People** – project stakeholders
- ▶ **Product** – la codebase e relativi artefatti
- ▶ **Project** – le attività messe in campo per la produzione del prodotto
- ▶ **Process** – l'organizzazione delle attività

Nello sviluppo di un software complesso sono coinvolte molte persone a grandi linee appartenenti alla seguenti categorie:

- Business Management
- Project Management
- Development Team
- Customers
- End Users

# Prodotto

Cosa?

- Il prodotto è certamente l'obiettivo finale e più importante dello sviluppo e risponde alla domanda sul **cosa si vuole "costruire"**.
- con riferimento al prodotto il focus principale dell'Ingegneria del Software è quello di fornire metodi e strumenti affinché la qualità di ogni artefatto risulti "massima". (e.g. introduzione delle tecniche di programmazione orientata agli oggetti). **Obiettivo dell'ingegneria del software non è quello di produrre documentazione!**



Specifica le attività da svolgere per realizzare il prodotto. Tipicamente:

- Planning
- Requirements
- Design
- Implementation
- Testing
- Maintenance

# Processo

Come?

- Il processo definisce quali sono e come organizzare le **attività** da mettere in atto nello sviluppo di un prodotto software. In questo senso fornisce una risposta alla domanda sul **come procedere nella produzione di un prodotto software**.
- l'Ingegneria del software ha affinato e definito molti diversi tipi di processi di sviluppo. Ognuno ha le sue peculiarità e qualità e si adatterà più o meno bene ai diversi ambiti di sviluppo. Un sistema safety critical non sarà sviluppato seguendo gli stessi passi attuati nello sviluppo di un applicazione Web.
- il processo non è di per se un fine. L'ipotesi "nascosta" è che adottando un disciplinato ed organizzato processo di sviluppo la qualità del prodotto e la sua economicità ne gioveranno.

# Processo

Come?

Un processo di sviluppo può essere osservato sotto diversi punti di vista. In generale si distingue tra:

- **Activity perspective** - definizione di cosa deve essere fatto
- **Workflow perspective** - sequenza ed organizzazione delle varie attività
- **Data-flow perspective** - flusso delle informazioni tra le varie attività
- **Role/action perspective** - Stabilisce chi fa cosa.

# Dicotomia o Dualità

- Processo e Prodotto non devono essere viste come due cose su piani ortogonali ma più come due facce della stessa medaglia.
- Ad esempio dato un prodotto le informazioni sul processo seguito permettono di contestualizzare meglio “l’oggetto” ed i suoi possibili ambiti di uso. Dunque il processo seguito fa in un certo qual modo parte delle caratteristiche del prodotto finale.

# Una nota sull'etica

ACM/IEEE Software Engineering Code of Ethics and Professional Practice – ([link](#))  
8 Principi fondamentali:

- **Public** – public interest
- **Client and Employer** – reconcile the interests of customers and public
- **Product** – highest professional standards
- **Judgement** – integrity and independence
- **Management** – ethical approach to management of dev and maintenance
- **Profession** – integrity and reputation of the profession
- **Colleagues** – be supportive of colleagues
- **Self** – improve and learn