



Introduzione al corso

Andrea Polini

Laboratorio di Ingegneria del Software
Corso di Laurea in Informatica – L31
Università di Camerino

Corso e Docente

- Andrea Polini
 - e-mail: andrea.polini@unicam.it
 - web: <http://www.cs.unicam.it/polini>
 - ricevimento: Martedì dalle ore 15:00 – 16:00
- Laboratorio di Ingegneria del Software
 - orario:
 - Martedì dalle 11 alle 13
 - Mercoledì dalle 11 alle 13
 - web: consultare il wiki della didattica – <http://didattica.cs.unicam.it/>
- Date di esame:
 - Martedì, 6 e 27 Febbraio 2018 - Ore 11
 - Mercoledì, 04 Aprile 2018 - Ore 11
 - Martedì, 19 Giugno 2018 - Ore 11
 - Martedì, 10 Luglio 2018 - Ore 11
 - Martedì, 11 e 25 Settembre 2018 - Ore 11
 - Martedì, 5 e 26 Febbraio 2019 - Ore 11

Obiettivi Formativi

- Al termine del corso il vostro bagaglio di competenze dovrebbe essersi arricchito con:
 - Metodologie e tecniche per la modellazione e lo sviluppo di **sistemi software complessi**
- Al termine del corso dovrete esser capaci di:
 - Comprendere le problematiche che influenzano il progetto di un sistema software complesso
 - Saper progettare un sistema software complesso attraverso fasi di modellazione e codifica

Materiale di studio

● Testo di riferimento:



Jim Arlow, *Ila Neustadt*

UML2 e Unified Process - Analisi e Progettazione Object-Oriented, 2^a Ed. Italiana

McGraw-Hill, 2006.

● Testi di consultazione:



Craig Larman

Applicare UML e i pattern. Analisi e progettazione orientata agli oggetti

Pearson, 2016



Ian Sommerville

Ingegneria del Software, 10^a Ed. Italiana

Addison-Wesley, 2017.



Bertrand Meyer

Object-Oriented Software Construction, 2^a Ed.

Prentice Hall, 1997.

● Materiale fornito dal docente

Esame!!!

- I corsi di Ingegneria del Software (IdS-T) e di Laboratorio di Ingegneria del Software (IdS-L) danno luogo ad una sola registrazione da 12 CFU.
- **Modalità di Esame:**
 - Progetto
 - Gruppi di 2 o 3 persone
 - I membri del gruppo presentano il progetto insieme nella stessa sessione
 - Tipologia e strumenti – UML (Visual Paradigm), IDE (Eclipse), Linguaggio (Java)
 - Discussione del progetto e prova orale nelle date di esame
(consegna venerdì precedente la data di esame)

Visual Paradigm



Visual Paradigm supports Università di Camerino with UML tools and ERD tools under the Visual Paradigm Academic Training Partner Program.

In quanto Academic partner di UNICAM è possibile usare la versione enterprise di Visual Paradigm che potete scaricare da:

- <https://ap.visual-paradigm.com/universita-di-camerino>

I progetti e qualsiasi documento utile alla realizzazione del progetto possono essere ottenuti all'interno del CASE

Contenuto del Corso

- Introduzione e caratterizzazione dell'Ingegneria del Software (IdS)
- Processi di Sviluppo Software (una panoramica)
 - Processo a cascata
 - Processi Evolutivi
 - Processi Iterativi
 - Processo a Spirale
 - Approcci Agili
 - ...
- Processo Unificato
- UML e suo uso nello sviluppo di un sistema software complesso
- Definizione di scenari e casi d'uso

Contenuto del Corso

- Specificità ed uso dei principali diagrammi UML
 - diagramma delle classi
 - diagramma degli oggetti
 - diagramma di sequenza
 - diagramma delle attività
 - diagramma dei package
 - diagramma dei componenti
 - diagramma di deployment
 - diagramma degli stati

Cos'è il software?

- Il termine software 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 software stesso. Includiamo nella definizione di software anche le sue caratteristiche strutturali, funzionali, extra-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?

- Il termine software 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 software stesso. Includiamo nella definizione di software anche le sue caratteristiche strutturali, funzionali, extra-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?

- Il termine software 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 software stesso. Includiamo nella definizione di software anche le sue caratteristiche strutturali, funzionali, extra-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

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 migliaia 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 appararono 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).

● Welfare Management System (2004)

- Un sistema di gestione della previdenza pubblica canadese costato diverse centinaia di milioni di dollari fu incapace di adattarsi a modifiche nei tassi di calcolo. Il sistema fu sottoposto a sole sei settimane di test di accettazione.

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 appararono 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).

- **Welfare Management System (2004)**

- Un sistema di gestione della previdenza pubblica canadese costato diverse centinaia di milioni di dollari fu incapace di adattarsi a modifiche nei tassi di calcolo. Il sistema fu sottoposto a sole sei settimane di test di accettazione.

Un po' di storia

...continua

- **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.

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.

Un po' di storia

...continua

- **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.

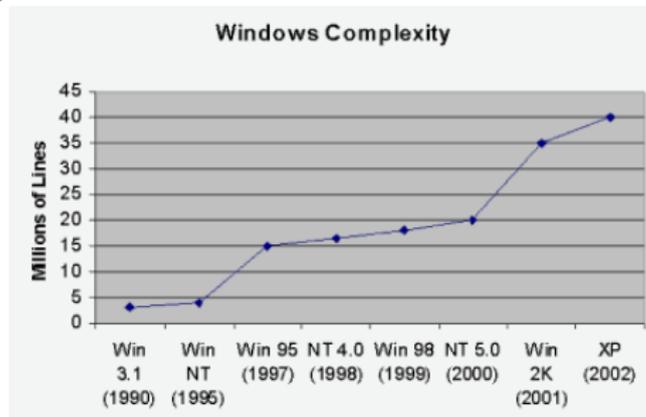
Standish Chaos Report 2009:

- 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
- ...

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 **extra-funzionali** 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



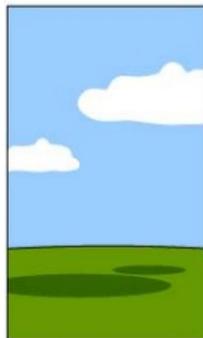
How the Analyst designed it



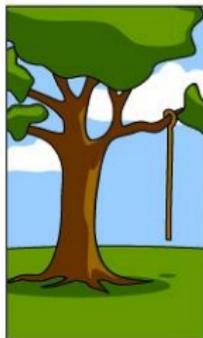
How the Programmer wrote it



How the Business Consultant described it



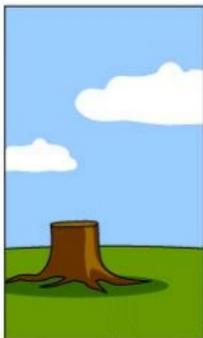
How the project was documented



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, 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

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**
- 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

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 risulti "massima". (e.g. introduzione delle tecniche di programmazione orientata agli oggetti). **Obiettivo dell'ingegneria del software non è quello di produrre documentazione!**

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.