
FII018: INGEGNERIA DEL SOFTWARE

Introduction to Software Engineering

Lecturer: Prof. Henry Muccini



Dipartimento di Ingegneria e Scienze
dell'Informazione e Matematica

Università degli Studi dell'Aquila

Copyright Notice

The material in these slides may be freely reproduced and distributed, partially or totally, as far as an explicit reference or acknowledge to the material author is preserved.

Henry Muccini

Agenda

Welcome to the Software Engineering Course

Why Software Engineering?

Software Engineering: what is that?

- how it relates to previous courses?
- What «Software Engineering» means?

Course Objective

Course plan



WELCOME



Agenda

Welcome to the Software Engineering Course

Software Engineering: what is that?

- how it relates to previous courses?
- What «Software Engineering» means?

Course Objective

Course plan



MY PROFILE

- Professor in Computer Science
 - Software Engineering area
- Topics of interest:
 - Software Engineering
 - Software Architettura
 - Model Driven Engineering
 - Software Engineering ML systems
- Teaching:
 - Software Engineering (2006-today)
 - Software Architettura (2006-today)
 - Design of web/mobile Applications @Master in Web Technology (2004-today) ⁵
 - Advanced Software Architecture (2010-today)

On developing methods and tools for the analysis and design of software architectures



- Machine Learning a supporto delle Architetture Adattive
- Microservizi, IoT, e Modernization
- Green AI
- Architetture dei dati

Course Administration



The screenshot shows a course administration page for 'Ingegneria del Software A.A. 2024-25'. On the left is a sidebar with a menu containing: 'Pagina iniziale', 'Class Notebook', 'Il lavoro in classe', 'Attività', 'Voti', 'Reflect', 'Insights', 'Canali principali', and 'Generale'. The main content area features a header with 'Dettagli pagina' and 'Analisi' on the left, and 'Pubblicata in data 23/09/2024', 'Condividi', and 'Modifica' on the right. Below the header is a large banner image of a stack of books with a dark overlay containing the text 'Benvenuti in classe!'. Under the banner, the text reads: 'Benvenuti al corso di Ingegneria del Software', followed by 'Buongiorno. Il corso di Ingegneria del Software si terrà nei giorni lunedì (14:30 - 16:30) in aula A1.7 (edificio Alan Turing) e mercoledì (11:30 - 13:30) sempre in aula A1.7.'

Codice del Team: 8n5mbah

Course Contract

- Lectures start at 2:45 on Monday and 11:45 on Wednesday
- Take notes!
 - Slides are NOT enough.
- Registration for the exam is “mandatory”
 - No registration, no exam
- Ask questions!



SOFTWARE ENGINEERING: WHY?



Fault management
Complexity management
Composizionalità/riuso

Why Software Engineering?

DISASTRO CROWDSTRIKE

Blocco globale dei PC Windows, cause e conseguenze: attenzione alle truffe

Home > Attacchi Hacker E Malware: Le Ultime News In Tempo Reale E Gli Approfondimenti

[f](#) [in](#) [X](#) [✉](#) [🔗](#) [🖨](#)

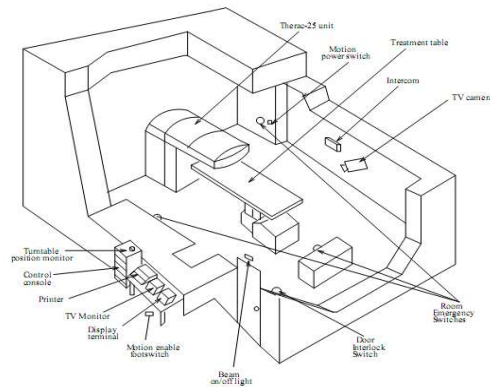
Il grande cyber caos che ha impattato su diversi milioni di computer Windows, ha rimarcato l'importanza di prestare attenzione a eventi di questa portata, proprio per le attività malevole che immediatamente si innescano

Pubblicato il 22 lug 2024

Examples of «Badly» Engineered Software

Therac-25 safety failure:

- approximately 100 times the intended dose of radiation
- 3 people died, and 6 got injured



see article at: <http://sunnyday.mit.edu/papers/therac.pdf>

Factors:

- Overconfidence in Software
- Confusing reliability with safety
- Lack of defensive Design
- Failure to eliminate fault causes
- Inadequate software engineering practices
- ...

<http://www.devtopics.com/20-famous-software-disasters/>

The screenshot shows a web browser displaying the DevTopics website. The page title is "20 Famous Software Disasters" and it is categorized under "Humor, Quality, Software". The article is written by "C# 411 Launched!".

DevTopics Software Development Topics

Information:

- About
- Advertise
- Contact
- Copyright
- Sample Page
- Search

Subscribe DevTopics

Subscribe by email:

Delivered by FeedBurner

Links:

- C# 411
- .NET News
- Is .NET Dead?
- Elbow Pain
- Mini-Tools Software

20 Famous Software Disasters

"To err is human, but to really foul things up you need a computer." -Paul Ehrlich

Software errors cost the U.S. economy \$60 billion annually in rework, lost productivity and actual damages. We all know software bugs can be annoying, but faulty software can also be expensive, embarrassing, destructive and deadly. Following are 20 famous software "disasters" in chronological order:

- 1. Mariner Bugs Out (1962)**
Cost: \$18.5 million
Disaster: The Mariner 1 rocket with a space probe headed for Venus diverted from its intended flight path shortly after launch. Mission Control destroyed the rocket 293 seconds after liftoff.
Cause: A programmer incorrectly transcribed a handwritten formula into computer code, missing a single superscript bar. Without the smoothing function indicated by the bar, the software treated normal variations of velocity as if they were serious, causing faulty corrections that sent the rocket off course. (more)
- 2. Hartford Coliseum Collapse (1978)**
Cost: \$70 million, plus another \$20 million damage to the local economy
Disaster: Just hours after thousands of fans had left the Hartford Coliseum, the steel-latticed roof collapsed under the weight of wet snow.

Recent Posts:

- Print a Directory Tree in Windows Explorer
- Mobile App Trends 2011
- Clear DNS Cache to See DNS Changes Immediately
- Programmer's 12 Days of Christmas
- Great Geek Gifts for 2012
- Why I Don't Own a Computer
- Happy 11/11/11 11:11:11
- Google Unveils Laptop with its Head in the Cloud
- New Camera Focuses After You Shoot
- Google+ Hopes to be the Privacy-Friendly Facebook

Featured Articles:

- Programmer Productivity: The "Trenfry Factor"
- How to Tell if You're a Programmer Geek
- Top 10 Software Innovators of All Time
- Top 10 Advances in Software Development
- 20 Famous Software Disasters



Discussione sui tipi di complessità

Agenda

Welcome to the Software Engineering Course

Software Engineering: what is that?

- how it relates to previous courses?
- What «Software Engineering» means?

Course Objective

Course plan



SOFTWARE ENGINEERING: WHAT IS THAT?

15

Engineering

Engineering is the use of scientific principles to design and build machines, structures, and other items, including bridges, tunnels, roads, vehicles, and buildings.^[1] The discipline of engineering encompasses a broad range of more specialized fields of engineering, each with a more specific emphasis on particular areas of applied mathematics, applied science, and types of application. See glossary of engineering.

Civil Engineering

Civil engineering is a professional engineering discipline that deals with the design, construction, and maintenance of the physical and naturally built environment, including public works such as roads, bridges, canals, dams, airports, sewage systems, pipelines, structural components of buildings, and railways.^{[1][2]}

[https://en.wikipedia.org/wiki/Civil_engineering]

My Software Engineering working Def.

Set of **automated** methods
to **systematically** develop **quality** software
that fulfils **customer needs**
while satisfying **existing constraints**

Software Engineering @ NATO conference – 1968 -

https://en.wikipedia.org/wiki/Software_engineering

https://en.wikipedia.org/wiki/Software_engineering

o AppsTesting Apps Funds GiochiBimbi vacanza Firefox EUROWEB Social Azienda Easyline BBQ Blog dei Genito... Video2MP3_Yo... KeepVid: Downl... Altri Pr


Main article: History of software engineering

When the first digital computers appeared in the early 1940s,^[9] the instructions to make them operate were wired into the machine. Practitioners quickly realized that this design was not flexible and came up with the "stored program architecture" or von Neumann architecture. Thus the division between "hardware" and "software" began with abstraction being used to deal with the complexity of computing.

Programming languages started to appear in the 1950s and this was also another major step in abstraction. Major languages such as Fortran, ALGOL, and COBOL were released in the late 1950s to deal with scientific, algorithmic, and business problems respectively. E.W. Dijkstra wrote his seminal paper, "Go To Statement Considered Harmful",^[10] in 1968 and David Parnas introduced the key concept of modularity and information hiding in 1972^[11] to help programmers deal with the ever increasing complexity of software systems.

The term "software engineering", coined by Margaret Hamilton,^{[12][13]} was first used in 1968 as a title for the world's first conference on software engineering, sponsored and facilitated by NATO. The conference was attended by international experts on software who agreed on defining best practices for software grounded in the application of engineering. The result of the conference is a report that defines how software should be developed [i.e., software engineering foundations]. The original report is publicly available.^[14]

The discipline of software engineering was created to address poor quality of software, get projects exceeding time and budget under control and ensure that software is built systematically, rigorously, measurably, on time, on budget, and within specification. Engineering already addresses all these issues, hence the same principles used in engineering can be applied to software. The widespread lack of best practices for software at the time was perceived as a "software crisis".^{[15][16][17]}



Margaret Hamilton, credited with coining the term software engineering, standing with the data results from the simulation of the code^[clarification needed] her team designed for Apollo 11.

On the relationship between Sw Engineering and Computer Science



Perche' tanta gente lavora su TikTok

■ TikTok è una delle applicazioni più avanzate...

- per l'**analisi dei video**,
- nell'uso dell'**intelligenza artificiale** applicata contemporaneamente ai video ed alla community,
- una delle **maggiori librerie musicali**,
- un sistema di **streaming audio e video ad alta qualità** accessibile da tutto il pianeta,
- uno dei **più grossi repository di dati** esistente in ambito social networks.

<https://www.agendadigitale.eu/cultura-digitale/tik-tok-una-finestra-sul-futuro-dei-social-network-e-di-noi-stessi/>



Algoritmi per la scelta di video da mostrare, data analytics, mobile/web, data storage, ...

Programming is NOT enough!

It is not enough to do your best: you must
Know what to do, and THEN do your best.
-- *W. Edwards Deming*



https://it.wikipedia.org/wiki/William_Edwards_Deming

A Software System is... not only software... not only programming



Programming is NOT enough!

It is not enough to do your best: you must
Know what to do, and THEN do your best.
-- W. Edwards Deming



Sw Process: come metto insieme i mattoncini?

Vincoli

Terreno

etc



Quand'e' che un sistema software
è ben ingegnerizzato?

Quali sono i tipici problemi da
evitare?



Quand'e' fondamentale ingegnerizzare un sistema software





COURSE ORGANIZATION



30

Course Organization

- Lecture
- Lab
- Project
- Quizz
- Oral Exam

Il Programma del Corso

Requirements Engineering

Software Design

Software Development Process

Project Management

Sw Testing

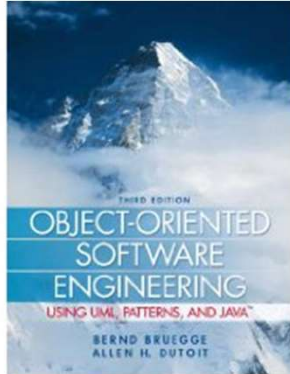
Course Administration



The screenshot shows a course administration page. On the left is a sidebar with a menu for 'Ingegneria del Software A.A. 2024-25' containing links like 'Pagina iniziale', 'Class Notebook', 'Il lavoro in classe', 'Attività', 'Voti', 'Reflect', 'Insights', and 'Canali principali'. The main content area has a header with 'Dettagli pagina' and 'Analisi'. Below this is a banner image of stacked books with a dark box containing the text 'Benvenuti in classe!'. Under the banner, the text reads 'Benvenuti al corso di Ingegneria del Software'. A small note below says 'Buongiorno. Il corso di Ingegneria del Software si terrà nei giorni lunedì (14:30 - 16:30) in aula A1.7 (edificio Alan Turing) e mercoledì (11:30 - 13:30) sempre in aula A1.7.' The top right corner shows 'Pubblicata in data 23/09/2024' and icons for 'Condividi' and 'Modifica'.

Codice del Team: 8n5mbah

TextBook



Bernd Bruegge, Allen H. Dutoit

Object-Oriented Software
Engineering: Using UML,
Patterns and Java, 3rd Edition

Publisher: **Prentice Hall**, Upper
Saddle River, NJ, 2009;

ISBN-10: 0136061257

ISBN-13: 978-0136061250

**Additional readings will be
added during the lectures.**

Verifica dell'apprendimento

- L'esame prevede un progetto, una prova orale, e dei quiz.
 - progetto: volto all'applicazione pratica della teoria presentata durante il corso. Il progetto viene sviluppato in maniera incrementale, seguendo un processo di sviluppo RUP. Il progetto viene discusso dal team che lo ha realizzato, e conta per il 60-80% del voto finale (a seconda della complessità dello stesso);
 - orale: volto a comprendere la preparazione teorica dello studente, la sua capacità di applicare la teoria in contesti più ampi del progetto, e la sua capacità espositiva verbale. L'esame orale conta per il 20/40% del voto finale;
 - quiz tematici: volti ad attivare subito l'interesse dello studente alla materia, coprono un tema specifico trattato a lezione. Tipicamente riconoscono 0,5 punti extra. Tali quiz non vengono assegnati tutti gli anni.

[Course catalogue](#) (link)

Verifica dell'apprendimento

- L'attribuzione del voto rifletterà i seguenti livelli di preparazione:
 - sufficiente: progetto minimale e conoscenza base degli argomenti trattati durante il corso;
 - adeguata: progetto adeguato ed acquisizione di una basilare abilità di progettazione dei sistemi software;
 - buona: buon progetto e capacità di ragionamento rigoroso;
 - approfondita: ottimo progetto e capacità di applicare la teoria su progetti complessi di diversa natura.

TODO IN CLASS (by you)

Max 10 PUNTI IMPORTANTI PRESENTATI IN CLASSE

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.