Ingegneria del Software

(2022-2023)

Laurea Triennale in Informatica

Modulo A: prof. Paolo Ciancarini

Modulo B: prof. Giancarlo Succi

Esercitazioni: prof. Marcello Missiroli

Software powers the world

(il software fa girare il mondo)

https://www.youtube.com/watch?v=F01JmJGJ9n8

«Sessanta anni dopo la rivoluzione informatica, 40 anni dopo l'invenzione del microprocessore, e 20 anni dopo la diffusione mondiale di Internet, finalmente funziona tutta la tecnologia software richiesta per trasformare le aziende; tale tecnologia può essere consegnata su scala globale.
Dal lato degli utenti (*front end*), la disponibilità mondiale di reti a larga banda e telefoni smart a basso costo permette a quasi ogni persona accesso istantaneo a tutto il potere di Internet, ogni momento del giorno.
Dal lato delle aziende (*back end*), gli strumenti di programmazione del software e i servizi basati su Internet semplificano il lancio in molti contesti industriali di nuove startup «software-powered»—senza bisogno di investire in nuove infrastrutture e addestrare nuovi impiegati."

Marc Andreessen, WSJ-8/20/2011

Creare software è difficile

- È difficile progettarlo
- È difficile scriverlo
- È difficile leggerlo o descriverlo
- È difficile modificarlo
- È difficile misurarlo
- È difficile decidere se è fatto bene

Tutti i programmatori sono ottimisti

Frederick Brooks, The mythical man-month, pag. 14

Optimism
All programmers are optimists. Perhaps this modern sorcery especially attracts those who believe in happy endings and fairy godmothers. Perhaps the hundreds of nitty frustrations drive away all but those who habitually focus on the end goal. Perhaps it is merely that computers are young, programmers are younger, and the young are always optimists. But however the selection process works, the result is indisputable: 'This time it will surely run," or "I just found the last bug."

Legge di Eagleson

Qualsiasi codice che non hai rivisto
negli ultimi sei mesi o più
ti sembrerà estraneo,
come se l'avesse scritto qualcun altro

Legge di Wirth

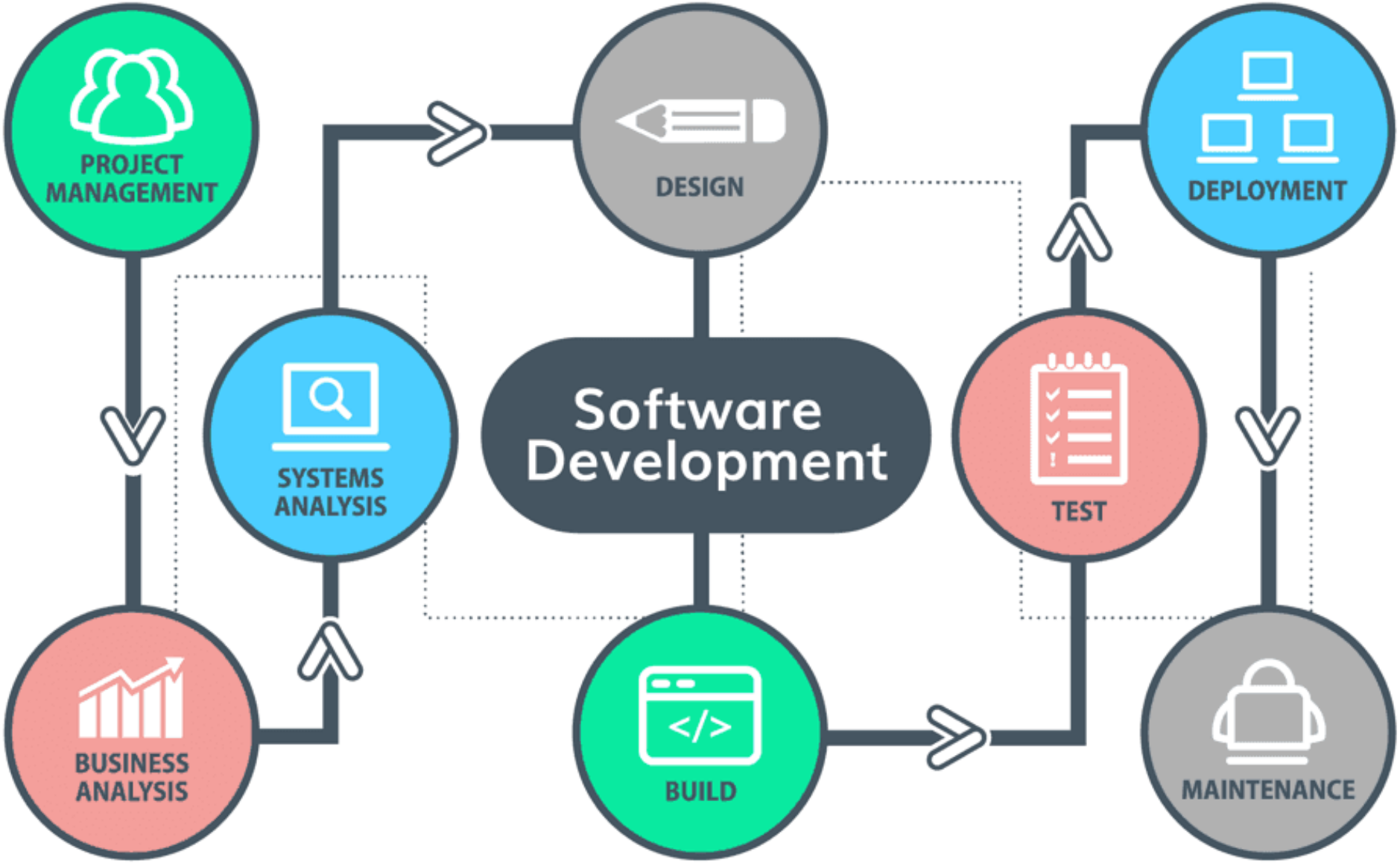
«*Il software diventa lento più rapidamente di quanto l'hardware diventi più veloce*»

Scopo del corso

Presentare e sperimentare *metodi* e *strumenti* di

- analisi,
- modellazione,
- progettazione, e
- misura

ovvero utili per lo **sviluppo (development)**
di prodotti, componenti e sistemi software
…con particolare riguardo ai metodi **agili**


PROJECT MANAGEMENT
DESIGN
DEPLOYMENT
SYSTEMS ANALYSIS
Software Development
TEST
BUSINESS ANALYSIS
BUILD
MAINTENANCE

Una storia vera

Il mio volo era in attesa di decollare quando il comandante fece un annuncio.

"*Abbiamo un problema con l'aria condizionata. Su questo aereo il condizionatore controlla i livelli di ossigeno e dobbiamo essere sicuri che funzioni prima del decollo. Abbiamo provato a fare restart del condizionatore ma non funziona. Proviamo adesso a fare restart dell'intero aereo. Questi aerei moderni sono tutti controllati a software, e dunque non sono molto affidabili*"

Il pilota spense tutto l'aereo e poi lo riaccese – ovvero fece reboot dell'aereo. Partimmo e tutto andò bene.

Fui molto contento di scendere da questo particolare volo

Saliresti su un aereo di cui hai scritto il software?

GIZMODO

Search

Researcher Can Hack Airplanes Through In-Flight Entertainment Systems

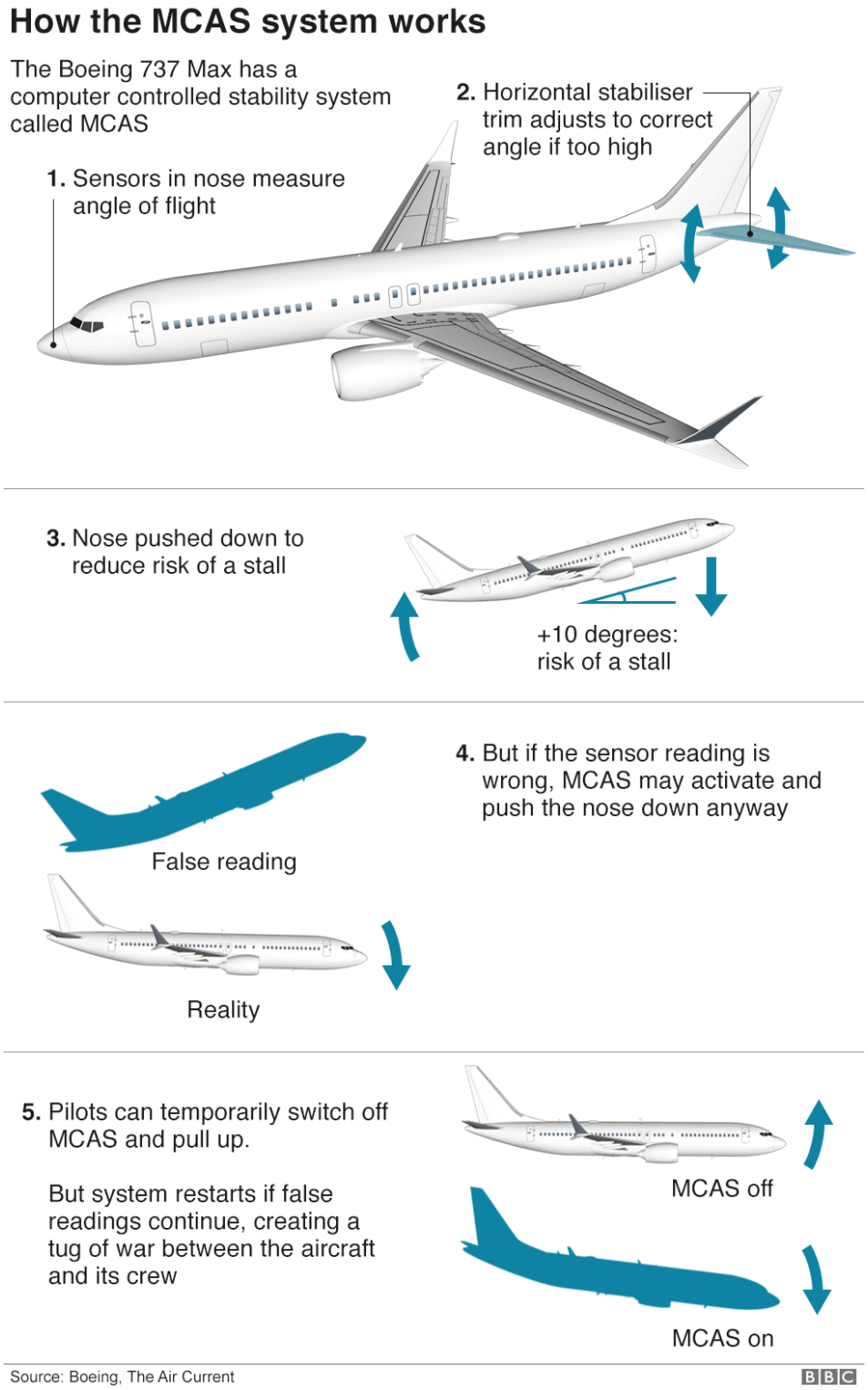
Adam Clark Estes

Filed to: HACKERS 8/04/14 12:32pm

96,405 6

If you're about to get on an airplane, you might want to wait until you land before you read this post. Because cyber security whiz Ruben Santamarta says he has devised a method that can give hackers access to a passenger jet's satellite communications equipment through the passenger Wi-Fi and in-flight entertainment systems*.

Il sistema MCAS degli aerei Boeing 737 è un software che corregge l'assetto dell'aereo e lo stabilizza.
Il suo (mal)funzionamento potrebbe aver causato gli incidenti in Etiopia (2019) e Indonesia (2018) – oltre 300 morti.
Sia le autorità federali statunitensi che il Congresso stanno indagando per capire se Boeing o la FAA avessero sottovalutato i rischi collegati all'uso di questo sistema.
In particolare, vogliono capire se la FAA abbia valutato il MCAS rigorosamente, basandosi su comprovati standard ingegneristici e di design, oppure se, per favorire Boeing, abbia seguito scorciatoie che non hanno garantito adeguati livelli di sicurezza di volo o una sufficiente formazione ai piloti.
All'aereo Boeing di Ethiopian Airlines, così come a quello di Lion Air precipitato in Indonesia, mancava un sistema di sicurezza in grado di avvisare i piloti di eventuali problemi con il sistema MCAS.
Il software di sicurezza, ritenuto non essenziale, **era venduto da Boeing come extra** e non era stato comprato né da Ethiopian Airlines né da Lion Air.



Come sviluppi il tuo sw?

- Come un falegname? Come un esploratore? Come uno scienziato? Come un archeologo? Come un architetto? Come un mercante?
- *"During software design, I'm an architect. When I'm designing the user interface, I'm an artist. During construction, I'm a craftsman. And during unit testing, I'm one mean son of a bitch!"*

O come …

- Roberto Bolle
- Uma Thurman and John Travolta
- The Rockettes

Come ti piace programmare?

- Da solo?
- In coppia?
- In team?

Alcune domande

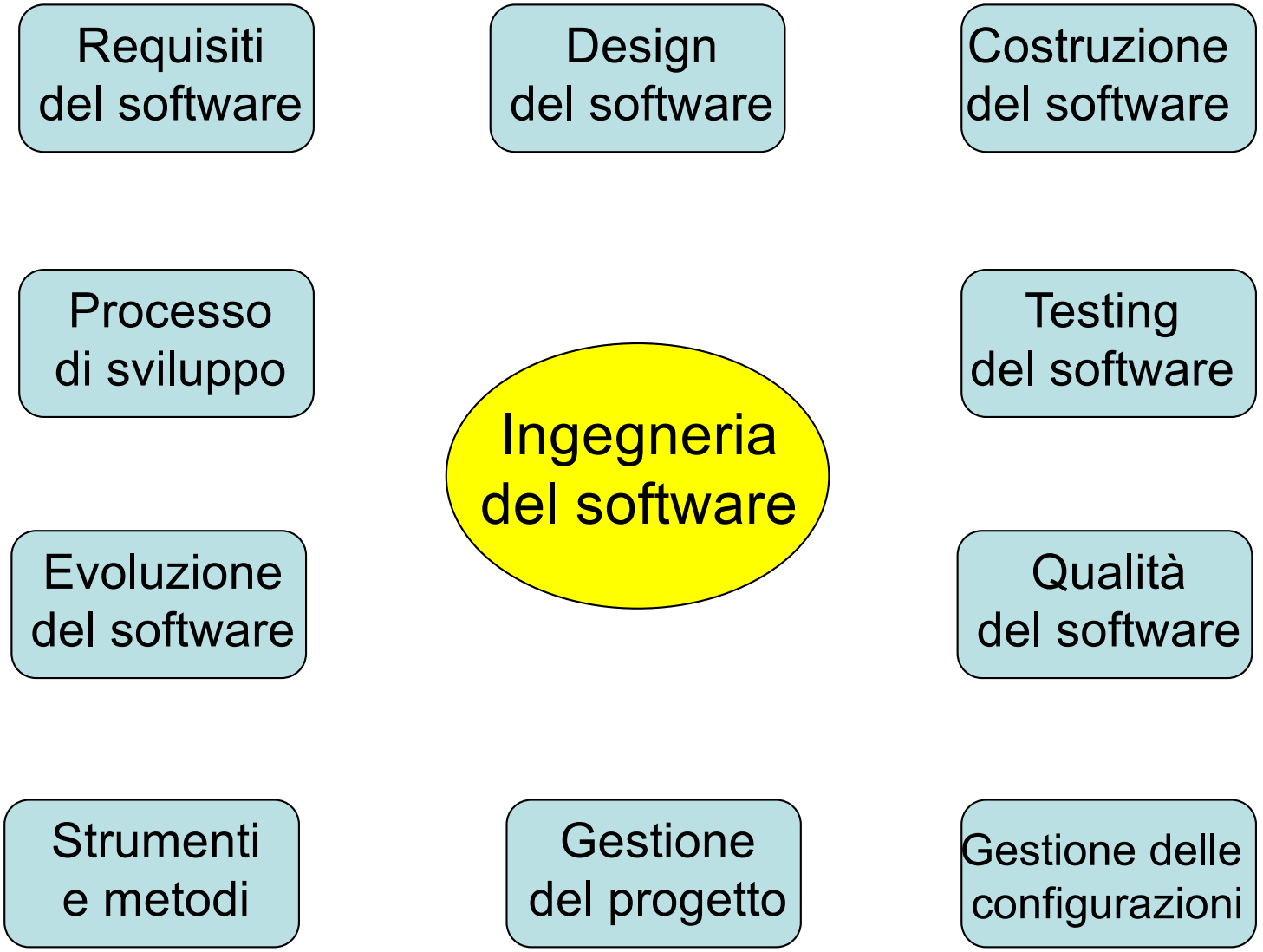
- Come si *progetta* un prodotto software quando si è in team?
- Quali *strumenti* sono disponibili per chi costruisce prodotti software?
- Quanto *costa* costruire il software?
- Come si valuta la *qualità* del software?


Software Development Team
PRODUCT MANAGER
DEVELOPER
UI DESIGNER

Legge di Conway

Le organizzazioni che progettano sw
sono indotte a generare design
che sono copie delle strutture di
comunicazione di tali organizzazioni

Gli argomenti principali del corso



Legge del versionamento

I prodotti software vengono SEMPRE consegnati a partire da un repository di versionamento, MAI direttamente dalla macchina di un programmatore

Conseguenza: usate Git!

(o una delle sue varianti)

Corsi propedeutici

Prequel (prerequisiti):

- Programmazione, Sistemi operativi

Agenda

- Gli standard di produzione del software
- Il ciclo di vita dei prodotti software
- I metodi agili
- L’analisi dei requisiti
- La progettazione del software
- La modellazione del software con UML
- Gestione di progetti software
- Controllare e misurare la qualità del software
- L’evoluzione del software
- Progetto in TEAM

Tempi del corso

Da settembre a dicembre

Lezioni settimanali:

- Lunedi ore 15 x3h aula Psicologia
- Martedi ore 13 x3h aula Psicologia
- Giovedi ore 9 x2h aula Tonelli

Alcuni strumenti

- Telegram https://t.me/joinchat/UN8CUvLWiSuJbIl3
- MS Teams
- Compositional Agile System aminsep.disi.unibo.it
  - GitLab, Jenkins, SonarQube, Mattermost, Taiga, logger-dashboard
- Spazi Web
- LaTeX, Powerpoint ecc.

Editor condivisi

- Repl.it
- https://teletype.atom.io/
- https://visualstudio.microsoft.com/services/live-share/
- Eclipse o IntelliJ o Atom with plugin
- https://glitch.com

Materiale didattico

Testi:
• Larman, *Applicare UML e i pattern*, 5ed., Pearson, 2020
• slide presentate a lezione

Testi aggiuntivi:
Jacobson, *The Essentials of Modern Software Engineering*, AW2019
Sommerville, *Ingegneria del software*, Pearson, 2021

Canali di conversazione

Gruppo Telegram: UniBoSWE

virtuale.unibo.it

Modalità di esame

Esame:

1. Midterm (fine corso): presentazione su argomento concordato
2. Progetto in team (+ orale a richiesta)

Voto:

75% progetto (voto al team),
25% presentazione (formato latex, pwp o equivalente),
Bonus per partecipazione attività in classe

Valore dell'esame: 9 CFU

Argomento, formato e consegna della presentazione midterm

- Argomento a scelta dello studente, previa approvazione
- Scegliere un articolo "recente" (= in stampa dal 2012 a oggi) tratto dalle riviste internazionali di ricerca su Ingegneria del Software
  - **IEEE Transactions on Software Engineering**
    - http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=32
  - **ACM Transactions on Software Engineering and Methodology**
    - https://dl.acm.org/citation.cfm?id=J790
  - **Journal of Systems and Software** https://www.journals.elsevier.com/journal-of-systems-and-software
  - **Empirical Software Engineering** https://www.springer.com/journal/10664

- Formato: latex, powerpoint o equivalente
- Consegna per email a me con subject [presentazione IdSw] in formato pdf e titolo CognomeMatricola entro due gg dalla data di discussione del progetto (entro febbraio)

Progetto

- Team: 4-6 persone
- Modalità agile – Scrum
- Uso di ambiente open source
- Inizio: durante il corso; termine: gennaio
- Istruzioni di processo
- Autovalutazione

Link utili

- CAS server aminsep.disi.unibo.it
- Scrumble scrumble.pyxis-tech.com
- Team forming: Trello

Attività in classe

- Test a risposte chiuse, senza voto ma con bonus finale
- Per ottenere il bonus bisogna partecipare a più della metà dei test (es: sette test, occorre partecipare almeno a quattro)
- La partecipazione ai test dà diritto ad un “bonus” da 1 a 4 punti che viene aggiunto al voto finale
- Il bonus è concesso solo se si fa l’esame nei primi due appelli, a gennaio/febbraio, e una volta sola

Importante

- Questo corso si supera **facilmente** seguendo le lezioni; è **difficile** da superare se non si frequentano le lezioni
- **Copiare** (le relazioni, il progetto) è **vietato** e quando si viene scoperti l'esame viene annullato e diventa **difficilissimo** da superare

Domande?

Legge di Paolo:

L'unica domanda stupida

è quella che non si fa