

Gestione dei Progetti Software

Docenti: Filomena Ferrucci, Fabio Palomba

Gestione dei Progetti Software: Perché?

Facciamo un passo indietro, cercando di mettere insieme le conoscenze da voi acquisite nel corso dei vostri anni di università.

Gestione dei Progetti Software: Perché?

Facciamo un passo indietro, cercando di mettere insieme le conoscenze da voi acquisite nel corso dei vostri anni di università.

Progettazione e Programmazione

- *Programmazione I.* Introduzione alla programmazione, oltre che a concetti quali specifica, precondizione, postcondizione e progettazione di soluzioni di programmazione.
- *Programmazione e strutture dati.* Complementi di programmazione, oltre che introduzione alle strutture dati e alla loro gestione.
- *Programmazione Object-Oriented.* Introduzione alla programmazione Object-Oriented, con introduzione a concetti quali astrazione dei dati, encapsulamento dell'informazione, coesione ed accoppiamento, riuso del codice.
- *Basi di dati.* Introduzione alla modellazione, oltre che ai metodi di gestione di basi di dati.
- *Tecnologie SW per il Web.* Introduzione alle architetture software e allo sviluppo web.
- *Progettazione di Algoritmi.* Divide-et-impera, programmazione dinamica e algoritmi su grafi.
- *Mobile Programming.*
- *Programmazione Distribuita.*
- *Ingegneria del SW.*

Gestione dei Progetti Software: Perché?

Dovreste essere già dei buoni **ingegneri del software** che sanno analizzare, progettare, realizzare e testare SW per la soluzione di problemi lavorando in team.
E allora perché un corso di GPS?

Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.

Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2026. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2026. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Deadline per la prima release: 31 Giugno 2026



Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Deadline per la prima release: 31 Giugno 2026

Costo stimato dall'azienda: 375.000 €



Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Deadline per la prima release: 31 Giugno 2026

Costo stimato dall'azienda: 375.000 €

Unità a disposizione per lo sviluppo: 13



Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Deadline per la prima release: 31 Giugno 2026

Costo stimato dall'azienda: 375.000 €

Unità a disposizione per lo sviluppo: 13

Penale per il mancato raggiungimento dell'obiettivo: 150.000 €



Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Deadline per la prima release: 31 Giugno 2026

Costo stimato dall'azienda: 375.000 €

Unità a disposizione per lo sviluppo: 13

Penale per il mancato raggiungimento dell'obiettivo: 150.000 €

Danni di immagine per eventuali difetti emersi durante l'esercizio: 1.872.000 €



Gestione dei Progetti Software: Perché?

Potenzialmente, tra meno di un anno potreste essere assunti da un'azienda. Lavoriamo di immaginazione, proiettandoci nel futuro.



E' il 3 Marzo 2025. Siete laureati, il futuro vi sorride! Avete da poco trovato lavoro in un'azienda di informatica, la quale vi ha proposto un buon contratto ed una prospettiva sul futuro.

Dopo un periodo di integrazione all'interno di un team di sviluppo, il vostro project manager vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.

In particolare, il progetto ha l'obiettivo di supportare i pagamenti della pubblica amministrazione tramite un'applicazione web-based che dovrà essere resiliente ad attacchi esterni ed effettuare predizioni sul grado di affidabilità creditizia degli utenti.

Deadline per la prima release: 31 Giugno 2026

Costo stimato dall'azienda: 375.000 €

Unità a disposizione per lo sviluppo: 13

Penale per il mancato raggiungimento dell'obiettivo: 150.000 €

Danni di immagine per eventuali difetti emersi durante l'esercizio: 1.872.000 €

Un incubo? No, è solo il mondo reale! Per arrivare al raggiungimento dell'obiettivo non basta conoscere gli algoritmi, non basta conoscere le strutture dati o un linguaggio di programmazione. C'è bisogno di *progettazione, comunicazione e collaborazione!*

Gestione dei Progetti Software: Perché?

Analizziamo meglio lo scenario proposto...

*Dopo un periodo di integrazione all'interno di un **team di sviluppo**, il vostro **project manager** vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.*

Questo implica l'esistenza di una struttura complessa, molto spesso gerarchica, composta da un project manager ed una serie di sviluppatori (con diverse competenze ed esperienze):

Gestione dei Progetti Software: Perché?

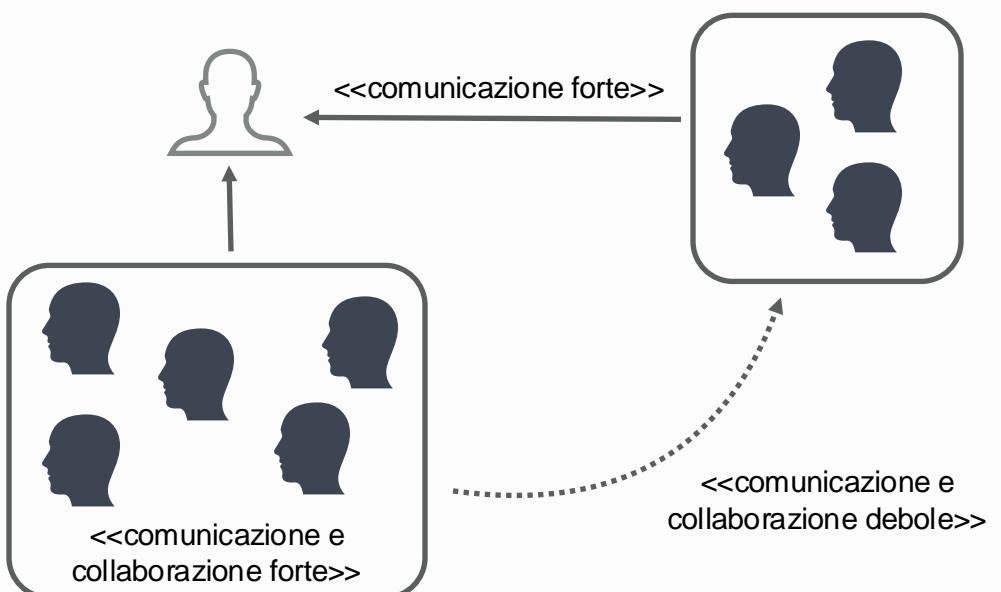
Analizziamo meglio lo scenario proposto...

*Dopo un periodo di integrazione all'interno di un **team di sviluppo**, il vostro **project manager** vi chiede di far parte di un nuovo progetto commissionato da un'importante multinazionale.*

Questo implica l'esistenza di una struttura complessa, molto spesso gerarchica, composta da un project manager ed una serie di sviluppatori (con diverse competenze ed esperienze):

Per avere successo rispettando le scadenze, è spesso necessario lavorare come un team, ovvero suddividendo i compiti ed integrando il risultato

Quindi, lo sviluppo potrebbe tradursi graficamente in qualcosa del genere.



Gestione dei Progetti Software: Perché?

In altri termini, lo sviluppo software è principalmente uno *sforzo collaborativo e comunicativo*, all'interno del quale la *progettazione* consente di raggiungere gli obiettivi prestabiliti con il committente.

I team NON sono piccoli → forte necessità di coordinamento

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente “controllato” in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina e richiede management!

Capire come il PM ragiona, cosa si aspetta dai membri del team è indispensabile per avere successo e quindi fare carriera!

SOFTWARE ENGINEERING AND IT MANAGEMENT

Obiettivo

Formare figure professionali con competenze nella progettazione, organizzazione, gestione, sviluppo, manutenzione, misurazione e collaudo di impianti, sistemi e servizi IT complessi e innovativi, nonché nella pianificazione, stima, direzione e gestione di progetti IT.



SOFTWARE ENGINEERING AND IT MANAGEMENT

Gestione dei Progetti Software

Lo sviluppo, la manutenzione e la gestione di sistemi software e IT complessi avviene nell'ambito di progetti

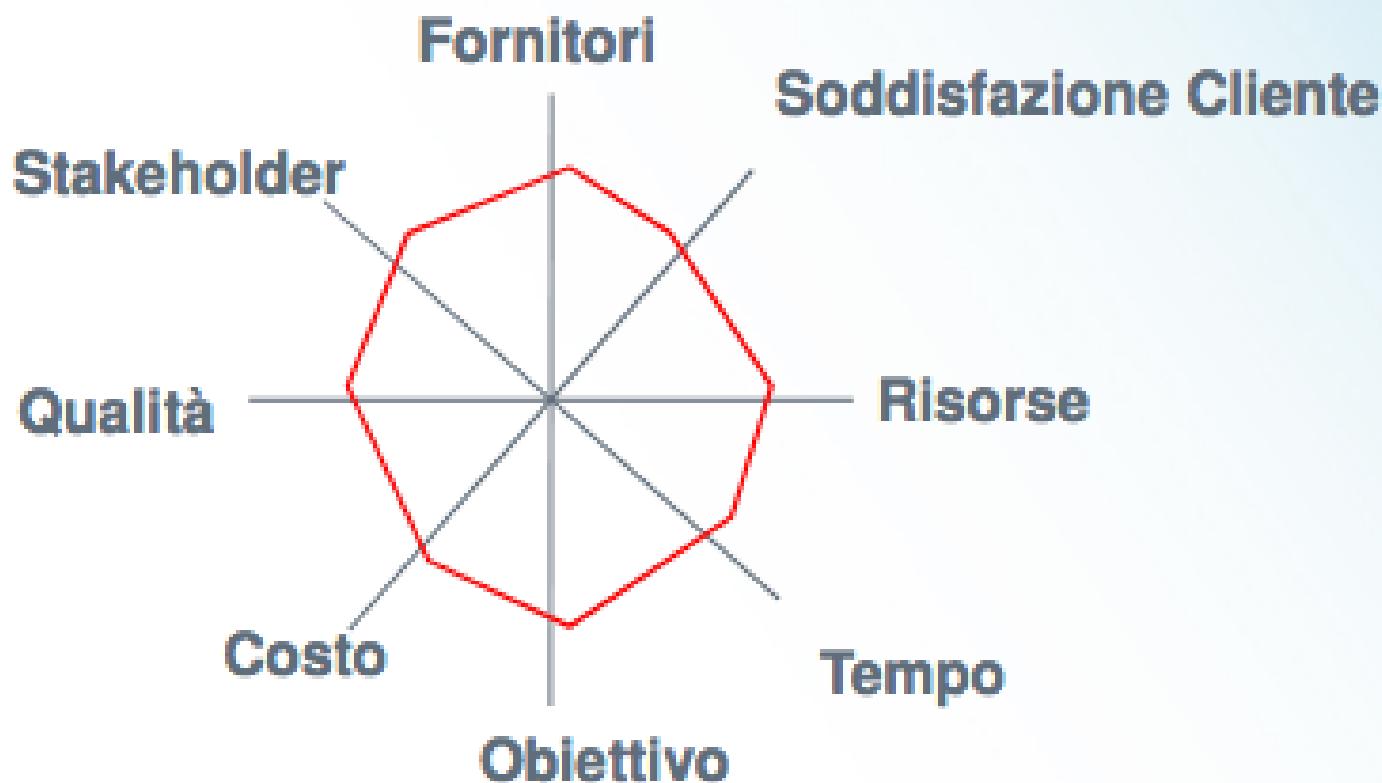
Stima di tempi e costi delle attività, allocazione di risorse, definizione dello schedule, gestione del team, ...



Indispensabile la conoscenza di tecniche di project management e la loro applicazione a progetti software e IT

COS'È IL PROJECT MANAGEMENT

È l'applicazione di conoscenze, skill, strumenti e tecniche alle attività di progetto per soddisfarne i requisiti (PMBOK)



L'equilibrio è una scelta del Project Manager

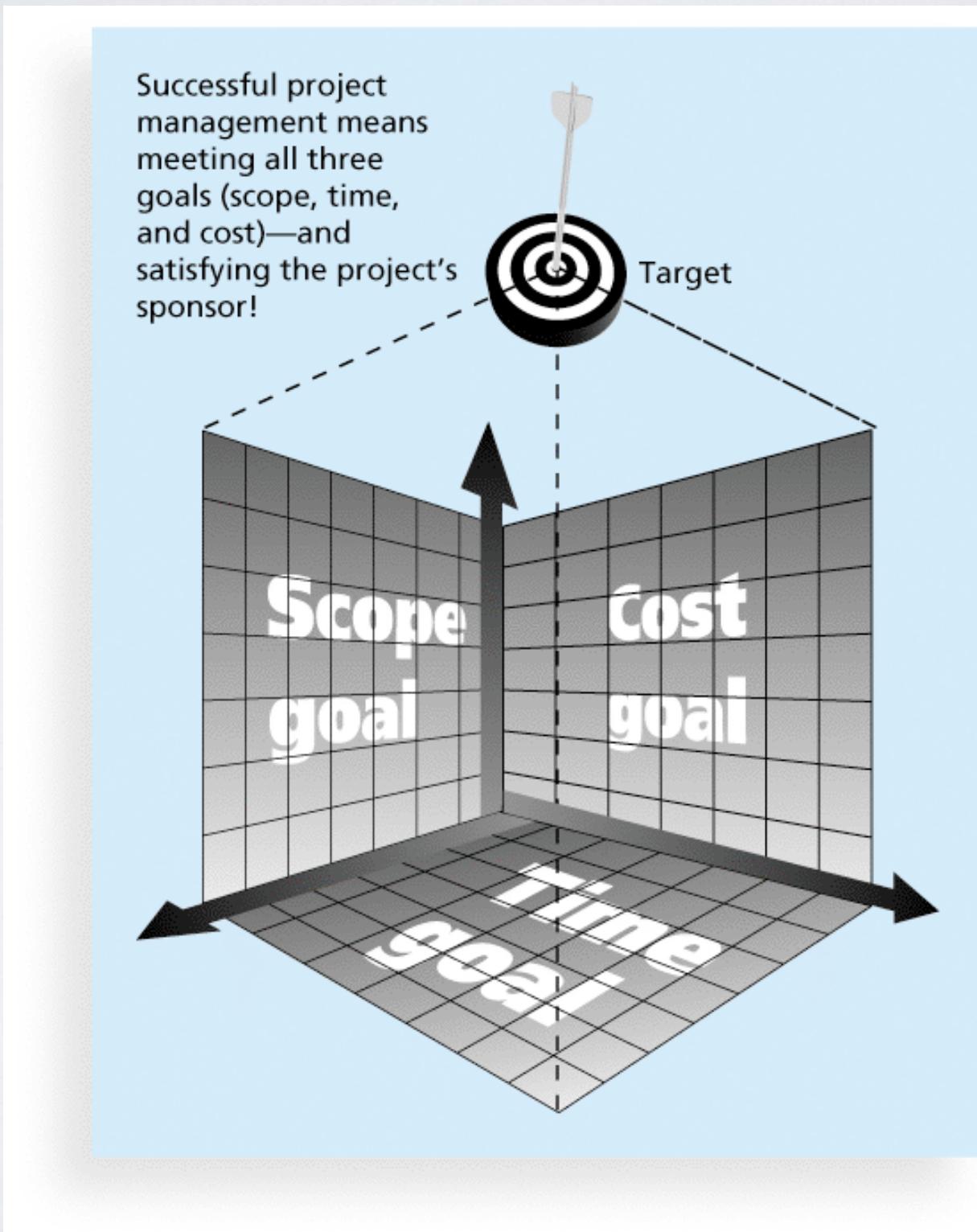
CHI È IL PM?



Il project manager è come un regista.

Il regista può non essere il miglior attore, costumista, designer, truccatore, redattore o promoter, ma ha l'innata capacità di mettere insieme tutti questi contributi per creare un'opera d'arte

THE TRIPLE CONSTRAINT OF PROJECT MANAGEMENT



PROJECT MANAGEMENT FRAMEWORK – PMBOK (PMI)

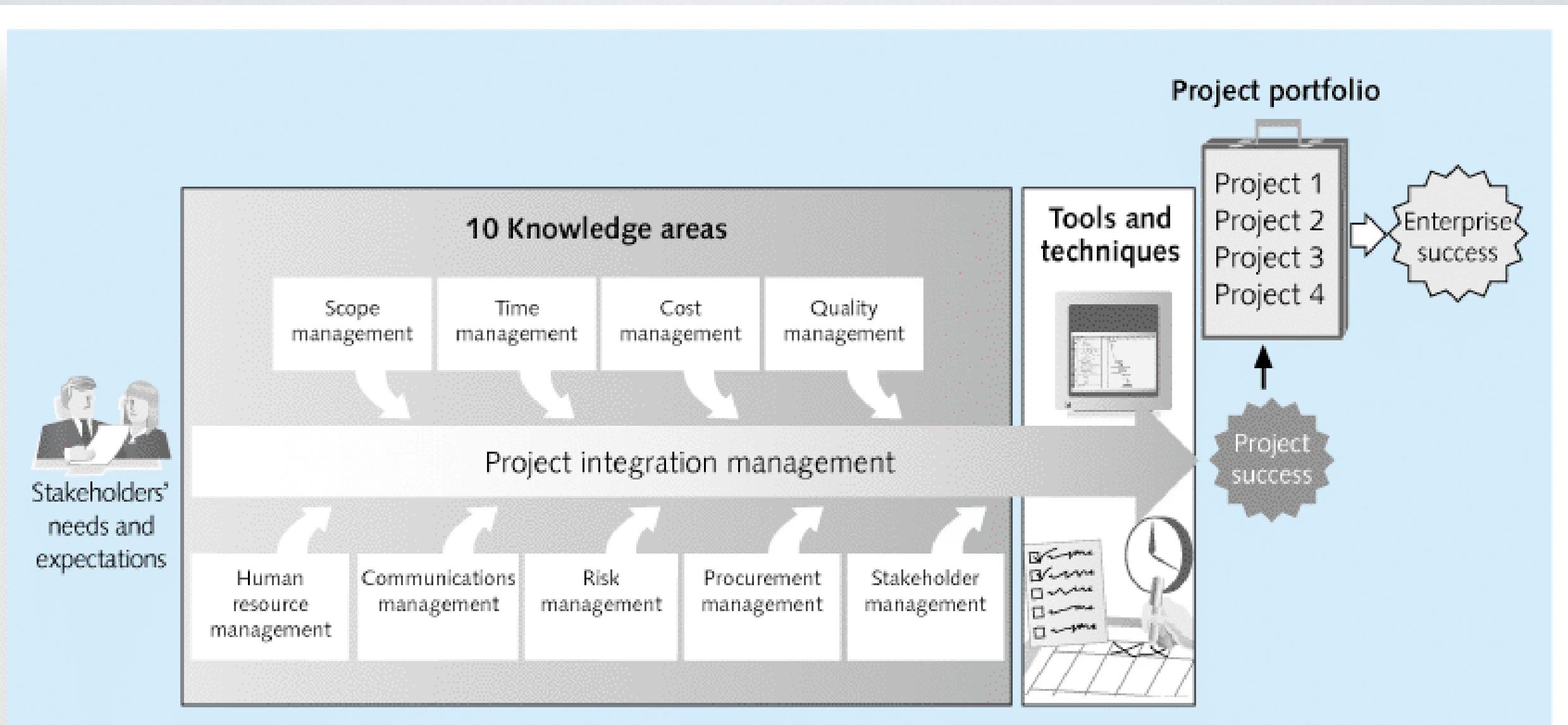


TABLE 3-2 JWD Consulting's business case

1.0 Introduction/Background

JWD Consulting's core business goal is to provide world-class project management consulting services to various organizations. The CEO, Joe Fleming, believes the firm can streamline operations and increase business by providing information related to project management on its intranet site, making some information and services accessible to current and potential clients.

2.0 Business Objective

JWD Consulting's strategic goals include continuing growth and profitability. The project management intranet site project will support these goals by increasing visibility of the firm's expertise to current and potential clients by allowing client and public access to some sections of the intranet. The project will also improve profitability by reducing internal costs by providing standard tools, techniques, templates, and project management knowledge to all internal consultants. Because JWD Consulting focuses on identifying profitable projects and measuring their value after completion, this project must meet those criteria.

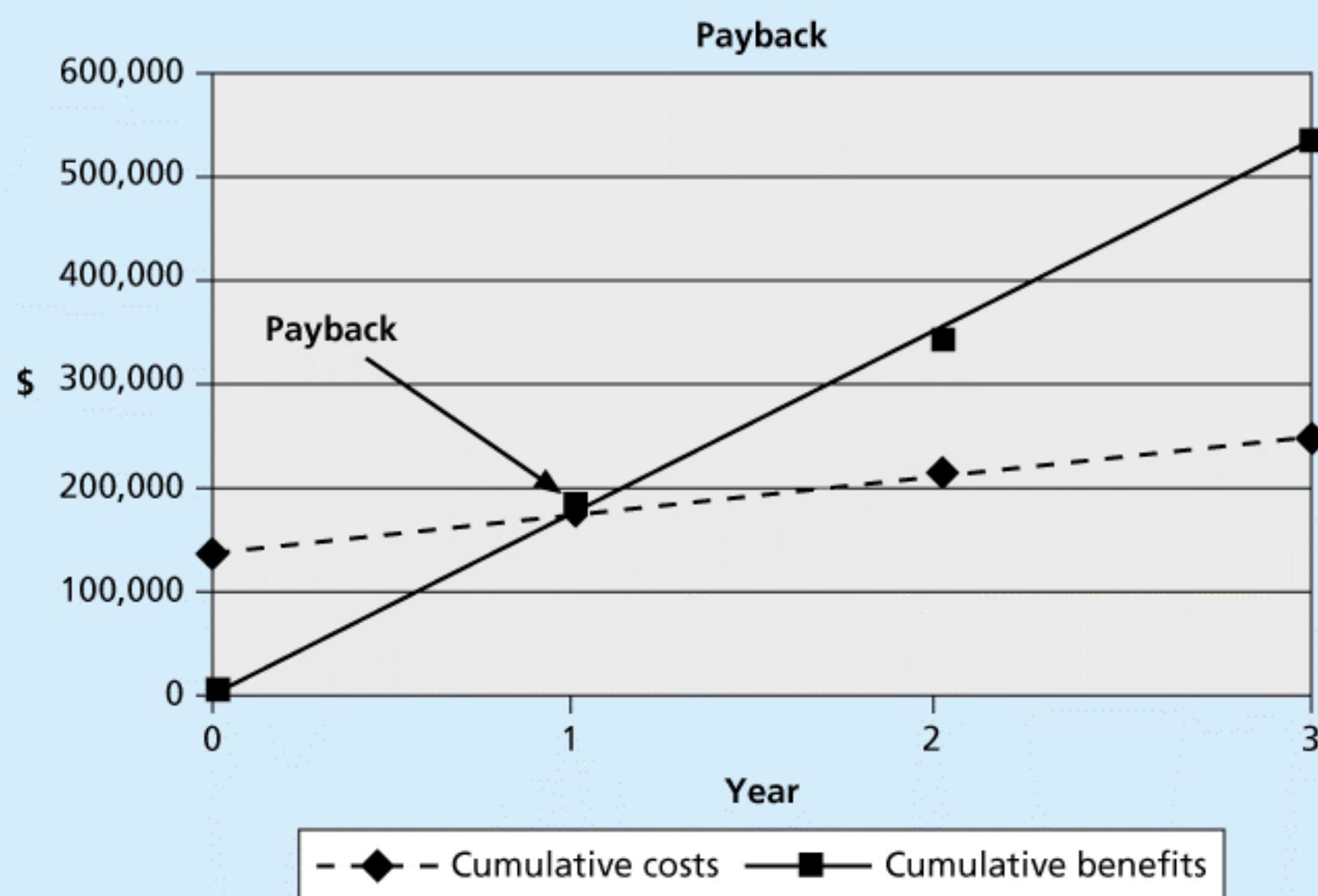
3.0 Current Situation and Problem/Opportunity Statement

JWD Consulting has a corporate Web site as well as an intranet. The firm currently uses the Web site for marketing information. The primary use of the intranet is for human resource information, such as where consultants enter their hours on various projects, change and view their benefits information, and access an online directory and Web-based e-mail system. The firm also uses an enterprise-wide project management system to track all project information, focusing on the status of deliverables and meeting scope, time, and cost goals. There is an opportunity to provide a new section on the intranet dedicated to sharing consultants' project management knowledge across the organization. JWD Consulting only hires experienced consultants and gives them freedom to manage projects as they see fit. However, as the business grows and projects become more complex, even experienced project managers are looking for suggestions on how to work more effectively.

10.0 Exhibits

Exhibit A: Financial Analysis for Project Management Intranet Site Project

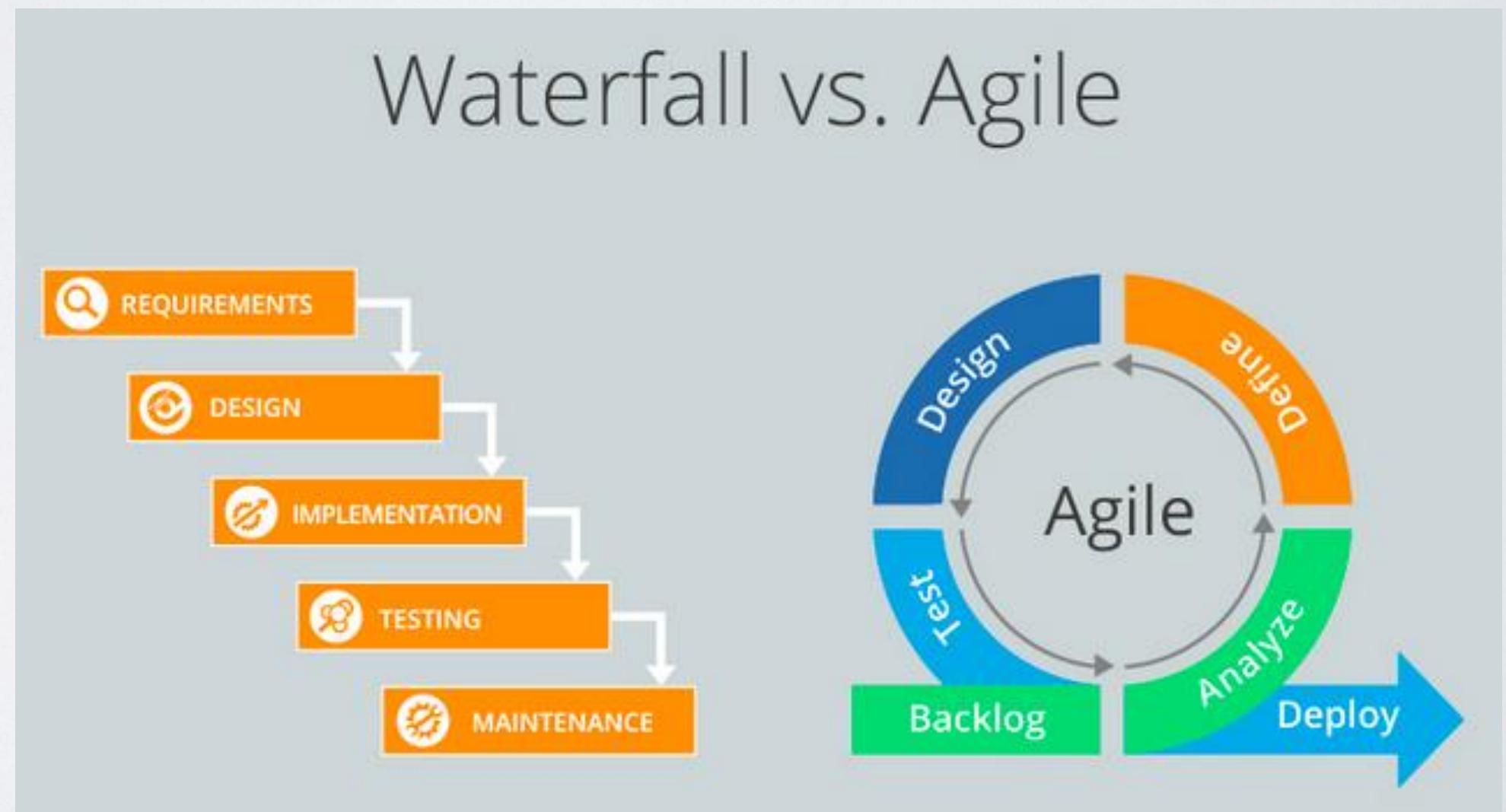
Discount rate	8%				
Assume the project is done in about 6 months		Year			
	0	1	2	3	Total
Costs	140,000	40,000	40,000	40,000	
Discount factor	1	0.93	0.86	0.79	
Discounted costs	140,000	37,037	34,294	31,753	243,084
Benefits	0	200,000	200,000	200,000	
Discount factor	1	0.93	0.86	0.79	
Discounted benefits	0	186,185	171,468	158,766	515,419
Discounted benefits - costs	(140,000)	148,148	137,174	127,013	
Cumulative benefits - costs	(140,000)	8,148	145,322	272,336	← NPV
Payback in Year 1					
Discounted life cycle ROI----->	112%				
Assumptions					
Costs	# hours				
PM (500 hours, \$50/hour)	25,000				
Staff (1500 hours, \$70/hour)	105,000				
Outsourced software and services	10,000				
Total project costs (all applied in year 0)	140,000				
Benefits					
# consultants	400				
Hours saved	40				
\$/hour profit	10				
Benefits from saving time	160,000				
Benefits from 1% increase in profits	40,000				
Total annual projected benefits	200,000				



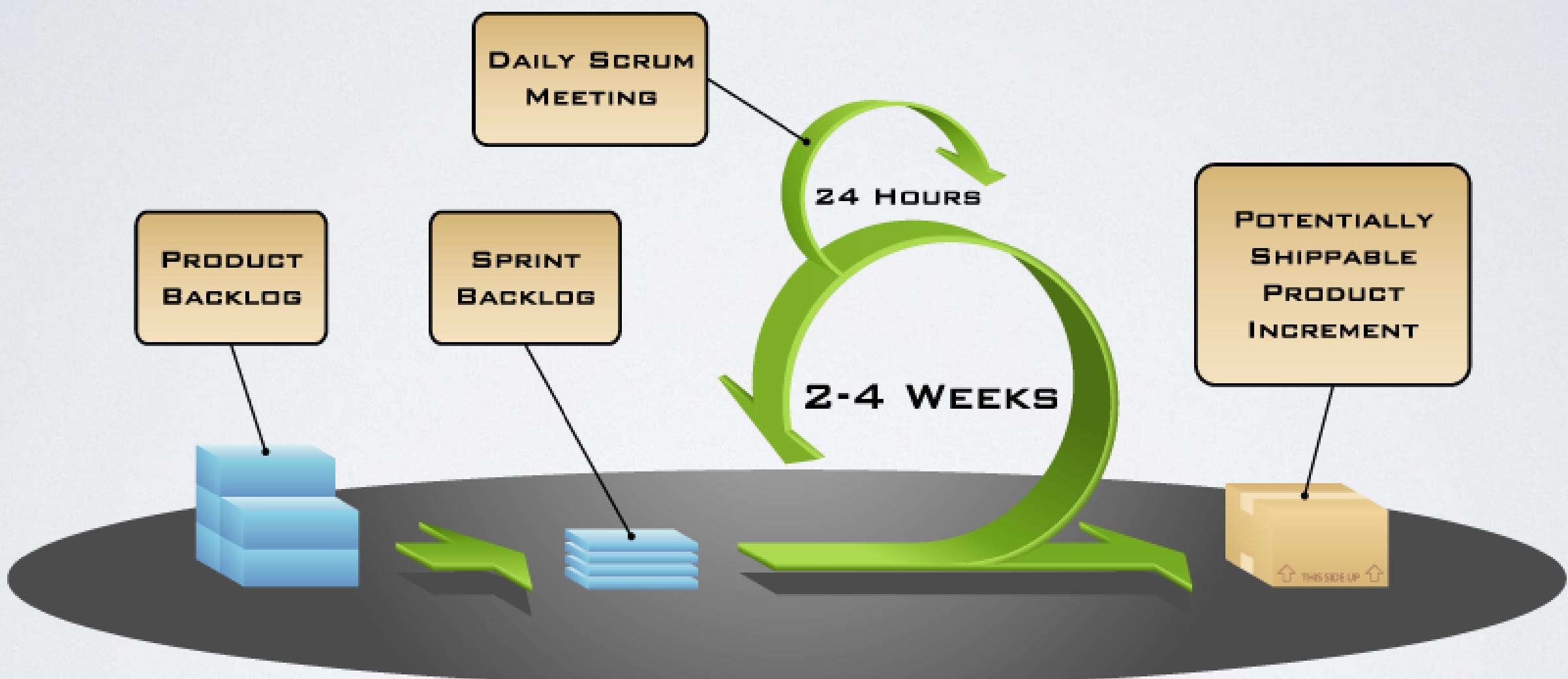


APPROCCI ALLO SVILUPPO

- Tradizionale
- Agile
- Ibrido



SCRUM



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

DIFFERENZE

How product managers are slowly becoming mini CEOs

by BARRON ERNST — 9 weeks ago in CONTRIBUTORS

A Product Manager is a Mini CEO

Published on May 26, 2016

 Scott Gordon | + Follow
Business Owner, Marketing Consultant and Advocate for Leadership Dev...
2 articles

(27 □ 2 ⏪ 7)

- Project Manager vs Scrum Master
- Project Manager vs Product Manager

Gestione dei Progetti Software: Overview del Corso

- Definizioni fondamentali di management: prodotto, progetto, project management, portfolio e programma. Il ruolo del project manager. IT project management e software project management. Il ruolo del product manager. Similitudini e differenze tra product manager e project manager. Gestione dell'innovazione.
- Il ciclo di vita del software. Modelli del ciclo di vita del software; metodologie agili, scrum, ibride. Agile project management.
- I gruppi di processi fondamentali del project management: inizio, pianificazione, esecuzione, monitoraggio e controllo, chiusura.
- La gestione delle aree di conoscenza: l'integrazione, lo scope, il tempo, il costo, la qualità, le risorse umane, la comunicazione, il rischio, gli acquisti, gli stakeholder.
- Attività di gestione in un progetto software; la work breakdown structure (WBS); il piano di gestione del progetto software. Retrospettiva di progetto, root cause analysis, gestione dei conflitti. Misurazione del software; paradigma goal-question-metric; classificazione delle metriche del software; misure di size funzionali (FPA e COSMIC), stima dell'effort.
- Sistemi di gestione qualità e modelli, standard per la qualità del prodotto software e relative caratteristiche. Software process improvement, requisiti di misurazione del CMMI.

Gestione dei Progetti Software: Overview del Corso

Il “solito” corso di Gestione dei Progetti Software, no?

Il “solito” corso di Gestione dei Progetti Software, no?

NO

Il “solito” corso di Gestione dei Progetti Software, no?

NO

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

Il “solito” corso di Ingegneria del Software, no?

NO

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;

Il “solito” corso di Ingegneria del Software, no?

NO

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;
- I vincoli e la gestione dei progetti subiranno cambiamenti sostanziali;

Il “solito” corso di Ingegneria del Software, no?

NO

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;
- I vincoli e la gestione dei progetti subiranno cambiamenti sostanziali;
- Il materiale didattico sarà rimodulato in modo da dare maggior peso ad alcuni aspetti.

Il “solito” corso di Ingegneria del Software, no?

NO

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;
- I vincoli e la gestione dei progetti subiranno cambiamenti sostanziali;
- Il materiale didattico sarà rimodulato in modo da dare maggior peso ad alcuni aspetti.

Pertanto, dimenticate ciò che fu.

Gestione dei Progetti Software: Overview del Corso

Il “solito” corso di Ingegneria del Software, no?

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;
- I vincoli e la gestione dei progetti subiranno cambiamenti sostanziali;
- Il materiale didattico sarà rimodulato in modo da dare maggior peso ad alcuni aspetti.

Pertanto, dimenticate ciò che fu.

Già guardando l'overview, sembra ci saranno tante scartoffie da preparare o, in alcuni casi, tanti documenti degli scorsi anni da copiare!

Gestione dei Progetti Software: Overview del Corso

Il “solito” corso di Ingegneria del Software, no?

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;
- I vincoli e la gestione dei progetti subiranno cambiamenti sostanziali;
- Il materiale didattico sarà rimodulato in modo da dare maggior peso ad alcuni aspetti.

Pertanto, dimenticate ciò che fu.

Già guardando l'overview, sembra ci saranno tante scartoffie da preparare o, in alcuni casi, tanti documenti degli scorsi anni da copiare!

Decisamente no.

- La documentazione **non** è vostra nemica, ma un supporto concreto allo sviluppo di sistemi software reali! Se seguirete con attenzione, lo capirete.

Gestione dei Progetti Software: Overview del Corso

Il “solito” corso di Ingegneria del Software, no?

Non proprio - e non lasciatevi fuorviare dai commenti dei vostri predecessori! Ci saranno diverse novità che renderanno il corso profondamente diverso rispetto agli anni passati.

- La parte di laboratorio sarà profondamente cambiata;
- I vincoli e la gestione dei progetti subiranno cambiamenti sostanziali;
- Il materiale didattico sarà rimodulato in modo da dare maggior peso ad alcuni aspetti.

Pertanto, dimenticate ciò che fu.

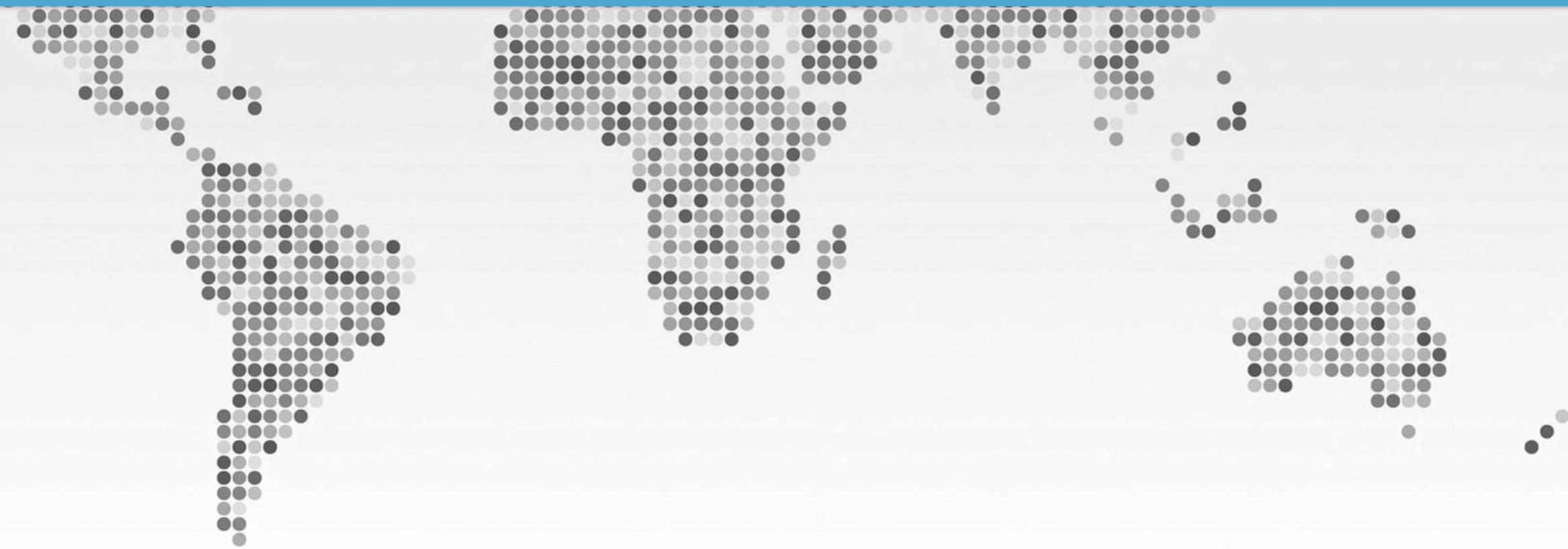
Già guardando l'overview, sembra ci saranno tante scartoffie da preparare o, in alcuni casi, tanti documenti degli scorsi anni da copiare!

Decisamente no.

- La documentazione **non** è vostra nemica, ma un supporto concreto allo sviluppo di sistemi software reali! Se seguirete con attenzione, lo capirete.
- Copiare significa sbagliare nel 99.9% dei casi. Inoltre, un documento copiato è visibile già dalle prime righe. I docenti se ne accorgono sempre, non siate ingenui.

Filomena Ferrucci, Fabio Palomba
SeSa Lab - University of Salerno

Ingegneria del Software



Ingegneria del Software: I Docenti



Filomena Ferrucci

Docente di Ingegneria del Software,
Gestione dei Progetti Software e
Didattica dell'Informatica

E-mail: fferrucci@unisa.it
Sito web: shorturl.at/BNQR3
Google Scholar: shorturl.at/dKRVY



Fabio Palomba

Docente di Ingegneria del Software,
Fondamenti di Intelligenza Artificiale, Software
Engineering for Artificial Intelligence, Scientific
Writing and Publishing

E-mail: fpalomba+is@unisa.it
Sito web: <http://fpalomba.github.io/>
Google Scholar: <https://goo.gl/dorFrh>
Twitter: fabiopalomba3

SeSa Lab Software Engineering @ Salerno

sesalab@unisa.it

sesa_lab



<https://sesalabunisa.github.io/>



Sentitevi liberi di seguire il SeSa Lab sui social media. Questi sono spesso aggiornati con news, informazioni utili e altro!

SeSa Lab Software Engineering @ Salerno

sesalab@unisa.it

sesa_lab



<https://sesalabunisa.github.io/>

Education

- Ingegneria del Software (x3);
- Fondamenti di Intelligenza Artificiale;
- Programmazione Object-Oriented.
- Gestione dei Progetti Software;
- Ingegneria, Gestione ed Evoluzione del Software;
- Metriche e Qualità del Software;
- Software Engineering for AI;
- Software Dependability;
- Software Engineering for Secure Cloud Computing
- Didattica dell'Informatica.
- 100+ tesisti all'anno;
- Occasioni di tirocini esterni in azienda.
- Più info:
<https://sesalabunisa.github.io/it/didattica.html>.



SeSa Lab Software Engineering @ Salerno

sesalab@unisa.it

sesa_lab



<https://sesalabunisa.github.io/>

Education

- Ingegneria del Software (x3);
- Fondamenti di Intelligenza Artificiale;
- Programmazione Object-Oriented.
- Gestione dei Progetti Software;
- Ingegneria, Gestione ed Evoluzione del Software;
- Metriche e Qualità del Software;
- Software Engineering for AI;
- Software Dependability;
- Software Engineering for Secure Cloud Computing
- Didattica dell'Informatica.
- 100+ tesisti all'anno;
- Occasioni di tirocini esterni in azienda.
- Più info:
<https://sesalabunisa.github.io/it/didattica.html>.

Research

- Ingegneria dei requisiti;
- Manutenzione ed evoluzione del software;
- Fondamenti di Data Science e Machine Learning per l'Ingegneria del Software;
- Sicurezza software;
- Verifica e convalida del software;
- Aspetti umani e gestione dei progetti software, inclusi sistemi ML e IoT.
- Quantum Software Engineering;
- Metaverse Engineering;
- Accessibilità e usabilità del software;
- Sviluppo del software sostenibile.
- Più info:
<https://sesalabunisa.github.io/it/index.html>.

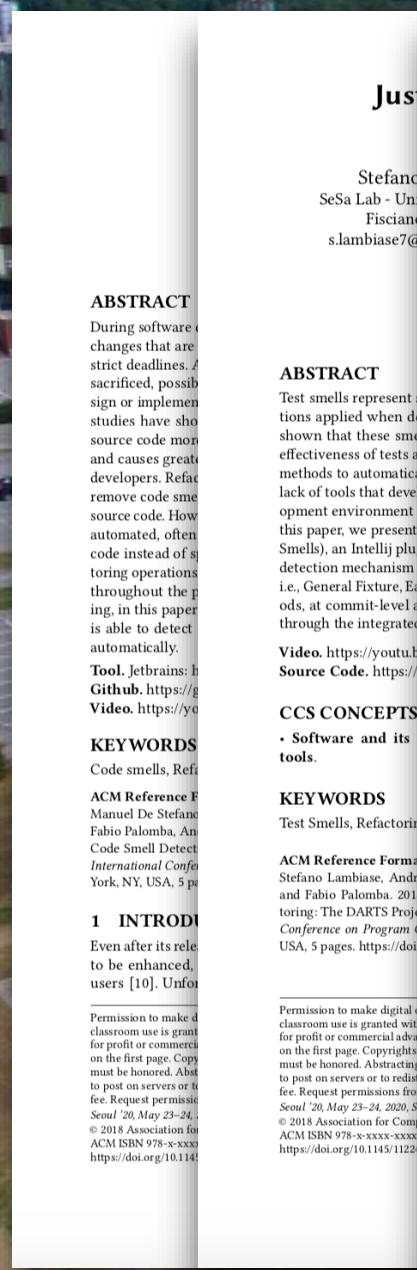
SeSa Lab Software Engineering @ Salerno

sesalab@unisa.it

sesa_lab



<https://sesalabunisa.github.io/>



Just another page

Stefano Lambiase, Emanuele Iannone, Fabiano Pecorelli, Dario Di Nucci, Andrea De Lucia

ABSTRACT
During software development, changes that are often made under strict deadlines. A developer may sacrifice, possibly unintentionally, design or implementation decisions. Studies have shown that source code modification can cause greater problems than it solves and causes greater problems for developers. Refactoring is a process that can remove code smells from the source code. However, automated refactoring tools instead of static analysis, monitoring operations throughout the process, in this paper we present a tool that is able to detect code smells automatically.

Tool: Jetbrains: <https://www.jetbrains.com/toolbox/>
Source Code: <https://github.com/se-sa-lab/ADoCTOR>
Video: <https://youtu.be/1c2EhVXiKis>

KEYWORDS
Code smells, Refactoring, Android Studio

ACM Reference Format:
Manuel De Stefanis, Stefano Lambiase, Andre De Lucia, and Fabio Palomba. 2018. Refactoring: The DARTS Project. In *Proceedings of the International Conference on Program Comprehension (Seoul '20)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting or reusing portions of this work in whole or in part is permitted. Request permission from permissions@acm.org.
Seoul '20, May 23–24, 2020, Seoul, South Korea
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/20/05...\$15.00
<https://doi.org/10.1145/1122445.1122456>

Refactoring Android-specific Energy Smells: A Plugin for Android Studio

Emanuele Iannone
SeSa Lab - University of Salerno
Fisciano (SA), Italy
e.iannone16@studenti.unisa.it

Fabiano Pecorelli
SeSa Lab - University of Salerno
Fisciano (SA), Italy
fpecorelli@unisa.it

Dario Di Nucci
JADE Lab - University of Tilburg/JADS
's-Hertogenbosch, The Netherlands
d.dinucci@uvt.nl

Fabio Palomba
SeSa Lab - University of Salerno
Fisciano (SA), Italy
fpalomba@unisa.it

Andrea De Lucia
SeSa Lab - University of Salerno
Fisciano (SA), Italy
adelucia@unisa.it

ABSTRACT

Mobile applications are major means to perform daily actions, including social and emergency connectivity. However, their usability is threatened by energy consumption that may be impacted by code smells, i.e., symptoms of bad implementation and design practices. In particular, researchers derived a set of mobile-specific code smells resulting in increased energy consumption of mobile apps and removing such smells through refactoring can mitigate the problem. In this paper, we extend and revise ADOCTOR, a tool that we previously implemented to identify energy-related smells. On the one hand, we present and implement automated refactoring solutions to those smells. On the other hand, we make the tool completely open-source and available in ANDROID STUDIO as a plugin published in the official store. The video showing the tool in action is available at: <https://www.youtube.com/watch?v=1c2EhVXiKis>

CCS CONCEPTS

- Software and its engineering → Software maintenance tools.

KEYWORDS

- Code smells, Refactoring

CCS CONCEPTS

- Software and its engineering → Software maintenance tools.

KEYWORDS

- Code smells, Refactoring, Energy Consumption.

ACM Reference Format:

Emanuele Iannone, Fabiano Pecorelli, Dario Di Nucci, Fabio Palomba, and Andrea De Lucia. 2018. Refactoring Android-specific Energy Smells: A Plugin for Android Studio. In *Proceedings of Seoul '20: ICPC International Conference on Program Comprehension (Seoul '20)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting or reusing portions of this work in whole or in part is permitted. Request permission from permissions@acm.org.
Seoul '20, May 23–24, 2020, Seoul, South Korea
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/20/05...\$15.00
<https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Mobile applications (*a.k.a.* apps) have gained great popularity in recent years. Developing a successful app requires to take into account different contrasting constraints such as user experience, performance, privacy, and energy consumption. One of the leading causes of a quick battery drain is the hardware setup, *i.e.*, the battery capacity, and the quality of the components. However, recent studies have shown that even wrong source code implementation decisions may decrease energy efficiency. For instance, Hasan *et al.* [5] stated that choosing a wrong data structure can lead to up to 300% energy waste. Palomba *et al.* [2, 10] analyzed the impact of some Android-specific code smells and discovered that their presence has a notable impact on battery usage, while Hecht *et al.* [6] later confirmed these findings and report that those smells also affect apps performance. These works based their analyses on a particular subset of the Android-specific code smells defined by Reimann's *et al.* [11] called "energy-smells" (*i.e.*, smells that have an impact on energy consumption) and were supported by their own energy smell detection tools: PAPRIKA and ADOCTOR, respectively.

On the one hand, ADOCTOR (AnDrOid Code smell dETECTOR) [9] identifies 15 Android-specific code smells. It syntactically analyzes the source code by extracting the Abstract Syntax Trees from the classes (*i.e.*, Java files) and then runs the detection algorithms which are based on the Visitor pattern. On the other hand, PAPRIKA [6] works at byte-code level and currently detects 16 code smells (both Object-Oriented and Android-specific). The empirical experimentation conducted to assess the accuracy of these tools showed that both can reach a high precision and recall; nevertheless, none of them is still able to provide developers with mechanisms that can (1) recommend a refactoring opportunity and (2) automatically refactor the code to remove a code smell instance.

Indeed, while most of these energy smells can be removed by applying simple program transformations, *e.g.*, adding a single statement or a keyword into a method signature, practitioners still tend not to refactor their code because (i) they are not aware of the existence of certain code smells and/or cannot estimate their impact [8]; (ii) they perceive the refactoring as a fault-prone activity [1, 4, 7]. For these reasons, an automated energy smell refactoring tool has the potential to be used in practice to help developers dealing with energy consumption.



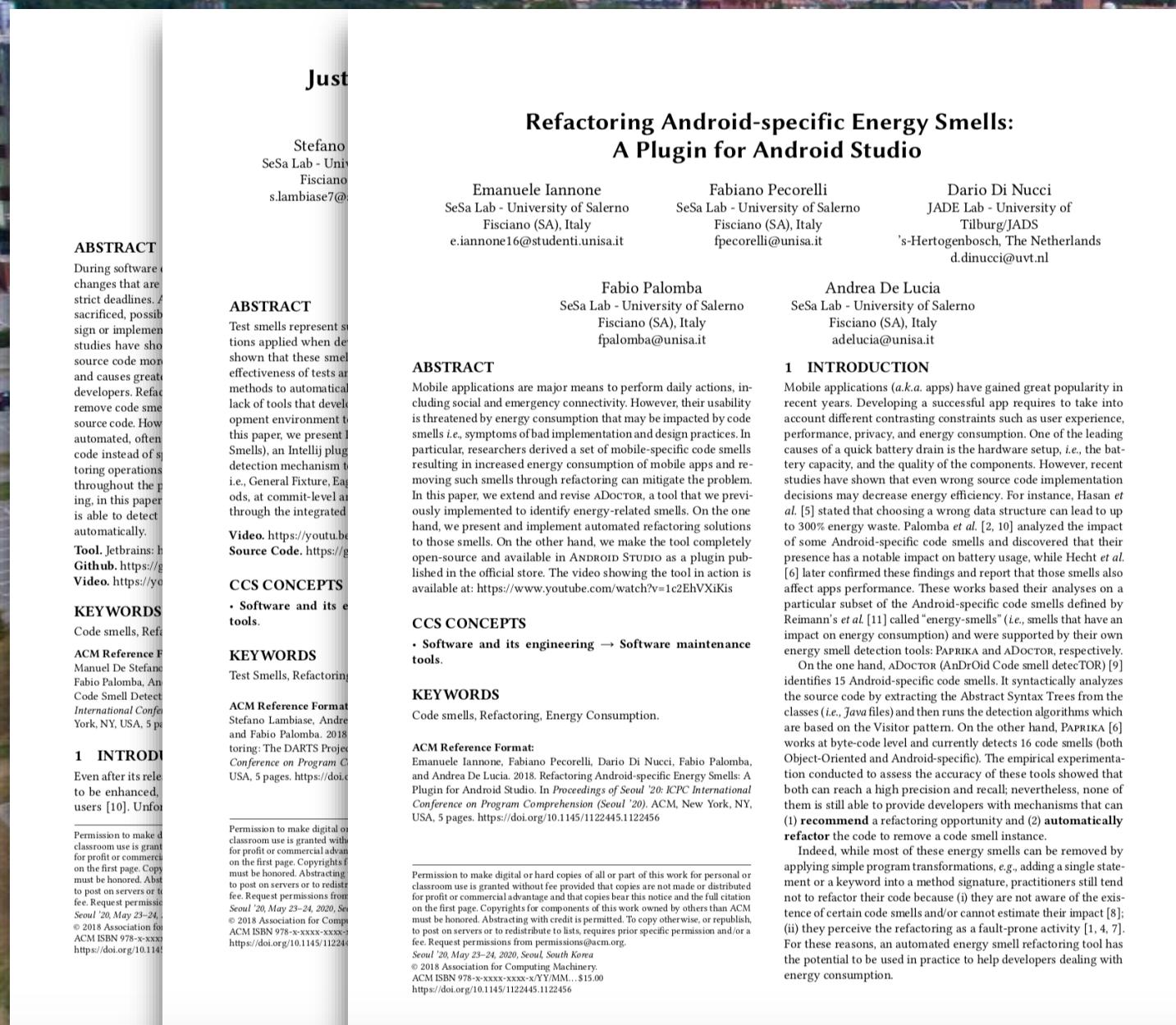
SeSa Lab Software Engineering @ Salerno

sesalab@unisa.it

sesa_lab



<https://sesalabunisa.github.io/>



Just another WordPress site

Stefano Lambiase
SeSa Lab - University of Salerno
Fisciano (SA), Italy
slambiasi7@unisa.it

ABSTRACT
During software development, changes that are made to the code often have strict deadlines. As a result, developers may sacrifice, possibly unintentionally, the design and implementation of the system. Studies have shown that this leads to source code smells, which can cause many problems and causes greater difficulties for future maintenance. Refactoring is a process that can remove code smells from the source code. However, it is often manual and time-consuming, especially if it is done at the end of the development cycle. In this paper, we present **ADoCTOR**, an IntelliJ plugin that automatically detects and refactors Android-specific code smells throughout the entire development cycle. The tool is able to detect and remove code smells automatically.

Tool: JetBrains IntelliJ IDEA
Github: <https://github.com/SeSaLab/ADoCTOR>
Source Code: <https://github.com/SeSaLab/ADoCTOR/tree/main>

Video: <https://youtu.be/v1c2EhVXiKis>

KEYWORDS
Code smells, Refactoring, Android, Smells, Tools

ACM Reference Format:
Manuel De Stefanis, Stefano Lambiase, Andre De Lucia, Fabio Palomba, and Emanuele Iannone. 2018. Refactoring: The DARTS Project. In *Proceedings of the International Conference on Program Comprehension (Seoul '20)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting or reusing portions of this work in whole or in part without permission is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Emanuele Iannone, Fabiano Pecorelli, Dario Di Nucci, Fabio Palomba, and Andrea De Lucia. 2018. Refactoring Android-specific Energy Smells: A Plugin for Android Studio. In *Proceedings of Seoul '20: ICPC International Conference on Program Comprehension (Seoul '20)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting or reusing portions of this work in whole or in part without permission is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Seoul '20, May 23–24, 2020, Seoul, South Korea
© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/Y/M/, \$15.00
<https://doi.org/10.1145/1122445.1122456>

Refactoring Android-specific Energy Smells:
A Plugin for Android Studio

Emanuele Iannone
SeSa Lab - University of Salerno
Fisciano (SA), Italy
e.iannone16@studenti.unisa.it

Fabiano Pecorelli
SeSa Lab - University of Salerno
Fisciano (SA), Italy
fpecorelli@unisa.it

Dario Di Nucci
JADE Lab - University of Tilburg/JADS
Tilburg, The Netherlands
d.dinucci@uvt.nl

Fabio Palomba
SeSa Lab - University of Salerno
Fisciano (SA), Italy
fpalomba@unisa.it

Andrea De Lucia
SeSa Lab - University of Salerno
Fisciano (SA), Italy
adelucia@unisa.it

ABSTRACT
Mobile applications are major means to perform daily actions, including social and emergency connectivity. However, their usability is threatened by energy consumption that may be impacted by code smells, i.e., symptoms of bad implementation and design practices. In particular, researchers derived a set of mobile-specific code smells resulting in increased energy consumption of mobile apps and removing such smells through refactoring can mitigate the problem. In this paper, we extend and revise **ADoCTOR**, a tool that we previously implemented to identify energy-related smells. On the one hand, we present and implement automated refactoring solutions to those smells. On the other hand, we make the tool completely open-source and available in ANDROID STUDIO as a plugin published in the official store. The video showing the tool in action is available at: <https://www.youtube.com/watch?v=1c2EhVXiKis>

CCS CONCEPTS
• Software and its engineering → Software maintenance tools.

KEYWORDS
Test Smells, Refactoring

ACM Reference Format:
Stefano Lambiase, Andre De Lucia, and Fabio Palomba. 2018. Refactoring: The DARTS Project. In *Proceedings of the International Conference on Program Comprehension (Seoul '20)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting or reusing portions of this work in whole or in part without permission is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Emanuele Iannone, Fabiano Pecorelli, Dario Di Nucci, Fabio Palomba, and Andrea De Lucia. 2018. Refactoring Android-specific Energy Smells: A Plugin for Android Studio. In *Proceedings of Seoul '20: ICPC International Conference on Program Comprehension (Seoul '20)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting or reusing portions of this work in whole or in part without permission is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Seoul '20, May 23–24, 2020, Seoul, South Korea
© 2018 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/Y/M/, \$15.00
<https://doi.org/10.1145/1122445.1122456>

3 Best Paper Award

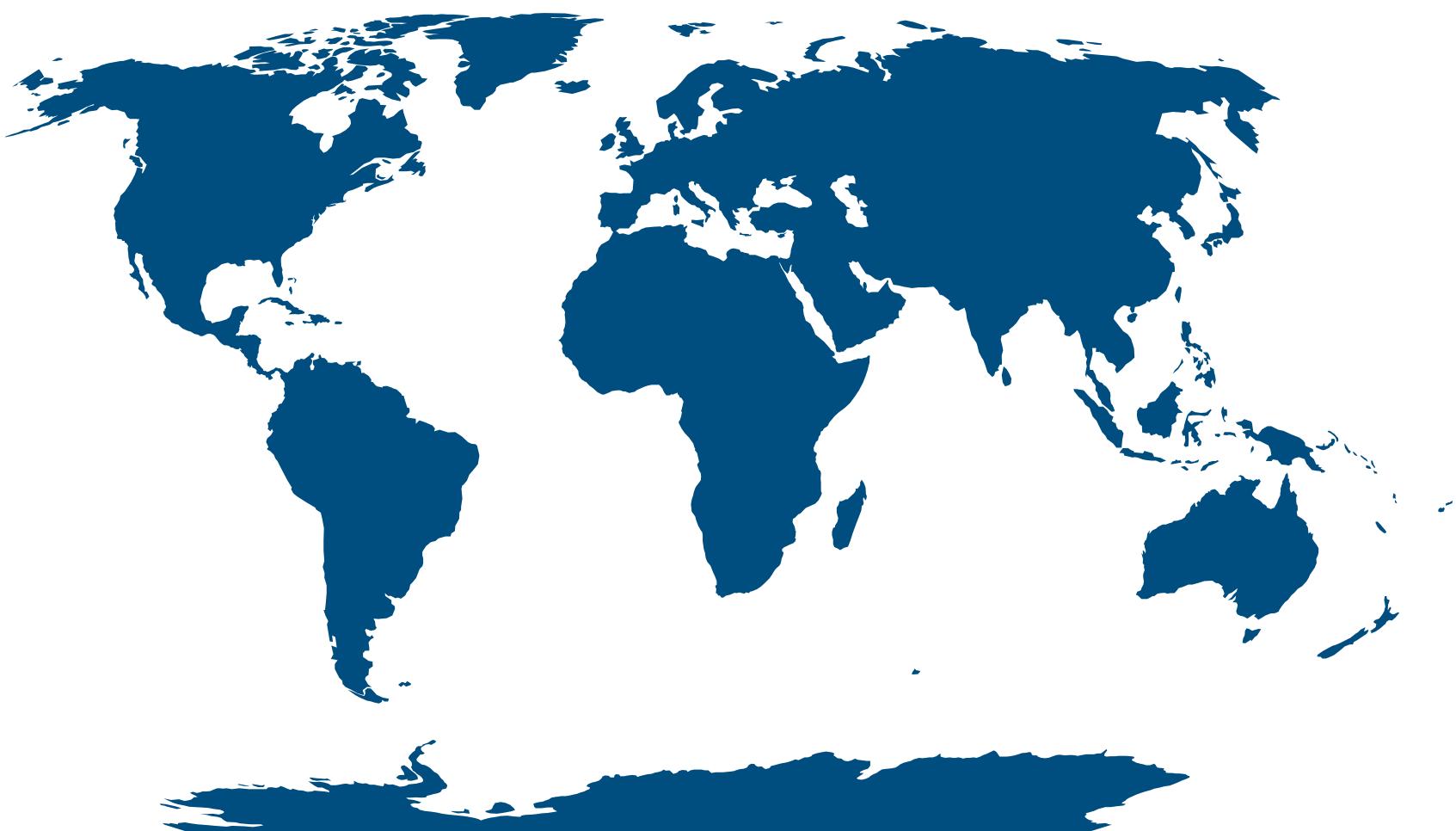
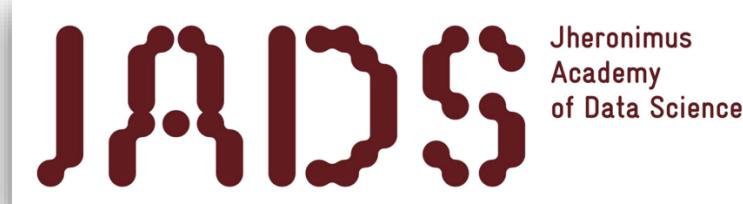
1 ACM/SIGSOFT Distinguished Paper Award
International Conference on Automated Software Engineering

30+ articoli scientifici con studenti in conferenze e riviste internazionali

14 nuovi docenti negli ultimi 10 anni di attività

<https://sesalabunisa.github.io/it/index.html>

Il Software Engineering (SeSa) Lab - Collaborazioni Recenti



E altri...

Informazioni amministrative generali

Docenti:

Filomena Ferrucci, Fabio Palomba

Tutor:

Da definire

Importunateli solo se
necessario!

Orario Lezioni:

Lunedì: 11:00 - 14:00 - Lab Hopper;
- 11:00 - Aula P4; Giovedì: 09:00
Venerdì: 11:00 - 13:00 - Aula P4.

Piattaforma E-learning:

<https://elearning.informatica.unisa.it/el-platform/course/view.php?id=1089>

Email:

fferrucci@unisa.it;
fpalomba+is@unisa.it.

Orario di risposta alle
email:

Lunedì—Venerdì 12:30 - 13:30

Appuntamento sempre
richiesto.

Orario di ricevimento:

Ferrucci. Martedì 16:00 - 18:00; Mercoledì 12:30 -
13:30. Palomba. Martedì 15:30 - 17:30; Giovedì 15:30 -
16:30.

Informazioni amministrative - Comunicazione

Destinatario

fferrucci@unisa.it e/o fpalomba+is@unisa.it

Invia l'email solo al docente che ha discusso a lezione dell'argomento per il quale chiedi spiegazioni e/o chiarimenti. Se la domanda è di carattere organizzativo, scrivi ad entrambi.

Informazioni amministrative - Comunicazione

Destinatario

fferrucci@unisa.it e/o fpalomba+is@unisa.it

Oggetto

[GPS] <Oggetto esplicito, breve e conciso>

Corpo

<Saluto>,

<presentazione del mittente>

<chiara e sintetica descrizione della richiesta/comunicazione>

<segnalazione di eventuali allegati>

<Saluto>

<Firma>

Altre semplici avvertenze e suggerimenti:

- (1) La punteggiatura è importante, usatela appropriatamente;
- (2) Usate le lettere maiuscole, il grassetto e l'italico appropriatamente;
- (3) Siate ordinati, ad esempio dividendo il corpo in paragrafi;
- (4) Prima di inviare l'email, verificate la presenza di errori ortografici e/o di battitura;
- (5) Date del Lei, non del Voi (il periodo fascista è finito 80 anni fa!)
- (6) Le e-mail che non seguiranno il formato riportato **NON verranno considerate!**

Informazioni amministrative - Comunicazione

Destinatario

fferrucci@unisa.it e/o **fpalomba+is@unisa.it**

Ricorda di rispondere a
tutti (funzione reply-all)

Oggetto

[GPS] <Oggetto esplicito, breve e conciso>

Corpo

<Saluto>,

<presentazione del mittente>

<chiara e sintetica descrizione della richiesta/comunicazione>

<segnalazione di eventuali allegati>

<Saluto>

<Firma>

Altre semplici avvertenze e suggerimenti:

- (1) La punteggiatura è importante, usatela appropriatamente;
- (2) Usate le lettere maiuscole, il grassetto e l'italico appropriatamente;
- (3) Siate ordinati, ad esempio dividendo il corpo in paragrafi;
- (4) Prima di inviare l'email, verificate la presenza di errori ortografici e/o di battitura;
- (5) Date del Lei, non del Voi (il periodo fascista è finito 80 anni fa!)
- (6) Le e-mail che non seguiranno il formato riportato **NON verranno considerate!**

Informazioni amministrative - Comunicazione

Destinatario

fferrucci@unisa.it e/o fpalomba+is@unisa.it

Ricorda di rispondere a tutti (funzione reply-all)

Oggetto

[IS] <Oggetto esplicito, breve e coinciso>

Corpo

<Saluto>,

<presentazione del mittente>

<chiara e sintetica descrizione della richiesta/comunicazione>

<segnalazione di eventuali allegati>

<Saluto>

<Firma>

Se non ricevi risposta entro 48h, invia nuovamente la mail!

Altre semplici avvertenze e suggerimenti:

- (1) La punteggiatura è importante, usatela appropriatamente;
- (2) Usate le lettere maiuscole, il grassetto e l'italico appropriatamente;
- (3) Siate ordinati, ad esempio dividendo il corpo in paragrafi;
- (4) Prima di inviare l'email, verificate la presenza di errori ortografici e/o di battitura;
- (5) Date del Lei, non del Voi (il periodo fascista è finito 80 anni fa!)
- (6) Le e-mail che non seguiranno il formato riportato **NON verranno considerate!**

Informazioni amministrative - Comunicazione

Destinatario

fferrucci@unisa.it e/o fpalomba+is@unisa.it

Se la tua richiesta è a nome del tuo gruppo, ricorda di mettere in copia tutti i membri del team (funzione cc)

Oggetto

[GPS] Richiesta di Esenzione da Prova Progettuale

Corpo

Gentili Proff. Ferrucci e Palomba,

Sono Mario Rossi, uno studente del corso di Ingegneria del Software.

Poiché sono uno studente lavoratore, iscritto in modalità part-time, non ho modo di partecipare attivamente alle lezioni e sviluppare un progetto per il superamento dell'esame. Pertanto, vorrei chiedere l'esenzione dalla prova progettuale. Invio in allegato la richiesta, documentata così come da indicazioni fornite sulla piattaforma e-learning.

Ringraziando per l'attenzione, saluto cordialmente,
Mario Rossi

Allegati



richiesta_esenzione_rossi.pdf

A meno di richieste esplicite, i file si inviano
SOLO ed ESCLUSIVAMENTE in formato pdf

Altre informazioni sulla cosiddetta “netiquette”: <https://www.linkedin.com/pulse/la-netiquette-il-galateo-di-internet-15-regole-per-rete-gianquinto/?originalSubdomain=it>

Informazioni amministrative - Il Ruolo dei Tutor per IS

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le lezioni frontali dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le lezioni frontali dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).
- (2) I tutor saranno spesso coinvolti durante le ore di laboratorio e presenteranno strumenti e/o nozioni utili per lo sviluppo del progetto. Quindi, **possono** essere contattati per eventuali dubbi e/o problemi nell'utilizzo degli strumenti.

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le lezioni frontali dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).
- (2) I tutor saranno spesso coinvolti durante le ore di laboratorio e presenteranno strumenti e/o nozioni utili per lo sviluppo del progetto. Quindi, **possono** essere contattati per eventuali dubbi e/o problemi nell'utilizzo degli strumenti.

NB: Tuttavia, non sono dei tecnici —> il supporto non riguarda l'installazione/configurazione di tool o errori di compilazione del codice sorgente, ma riguarda l'uso appropriato di tool.

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le lezioni frontali dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).
- (2) I tutor saranno spesso coinvolti durante le ore di laboratorio e presenteranno strumenti e/o nozioni utili per lo sviluppo del progetto. Quindi, **possono** essere contattati per eventuali dubbi e/o problemi nell'utilizzo degli strumenti.
NB: Tuttavia, non sono dei tecnici —> il supporto non riguarda l'installazione/configurazione di tool o errori di compilazione del codice sorgente, ma riguarda l'uso appropriato di tool.
- (3) I tutor saranno spesso coinvolti nel monitoraggio dei vostri progetti. **Possono** supportarvi nella definizione e sviluppo del vostro progetto.

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le lezioni frontali dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).
 - (2) I tutor saranno spesso coinvolti durante le ore di laboratorio e presenteranno strumenti e/o nozioni utili per lo sviluppo del progetto. Quindi, **possono** essere contattati per eventuali dubbi e/o problemi nell'utilizzo degli strumenti.
- NB:** Tuttavia, non sono dei tecnici —> il supporto non riguarda l'installazione/configurazione di tool o errori di compilazione del codice sorgente, ma riguarda l'uso appropriato di tool.
- (3) I tutor saranno spesso coinvolti nel monitoraggio dei vostri progetti. **Possono** supportarvi nella definizione e sviluppo del vostro progetto.

NB: Tuttavia, non sono dei membri aggiuntivi del vostro team. Da loro si va per chiedere spiegazioni sui giusti metodi da utilizzare, ma non per far correggere i vostri artefatti.

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le **lezioni frontali** dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).
- (2) I tutor saranno spesso coinvolti durante le ore di laboratorio e presenteranno strumenti e/o nozioni utili per lo sviluppo del progetto. Quindi, **possono** essere contattati per eventuali dubbi e/o problemi nell'utilizzo degli strumenti.

NB: Tuttavia, non sono dei tecnici —> il supporto non riguarda l'installazione/configurazione di tool o errori di compilazione del codice sorgente, ma riguarda l'uso appropriato di tool.

- (3) I tutor saranno spesso coinvolti nel monitoraggio dei vostri progetti. **Possono** supportarvi nella definizione e sviluppo del vostro progetto.

NB: Tuttavia, non sono dei membri aggiuntivi del vostro team. Da loro si va per chiedere spiegazioni sui giusti metodi da utilizzare, ma non per far correggere i vostri artefatti.

- (4) I tutor **non** sono tenuti a rispondere a domande su argomenti che non riguardano il corso di Ingegneria del Software - ad esempio, non possono colmare le lacune in Basi di Dati.

Informazioni amministrative - Il Ruolo dei Tutor

Come detto, i tutor ci assisteranno durante l'intero semestre. E' doveroso quindi spiegare meglio quali saranno i loro compiti ed in che modo è necessario interagire con loro.

- (1) I tutor **non** sono tenuti a dare spiegazioni su argomenti trattati durante le lezioni frontali dai docenti—per quello, ci sono i docenti! Quindi, in caso di dubbi sugli argomenti trattati, vanno contattati i docenti (secondo le modalità descritte prima).
- (2) I tutor saranno spesso coinvolti durante le ore di laboratorio e presenteranno strumenti e/o nozioni utili per lo sviluppo del progetto. Quindi, **possono** essere contattati per eventuali dubbi e/o problemi nell'utilizzo degli strumenti.

NB: Tuttavia, non sono dei tecnici —> il supporto non riguarda l'installazione/configurazione di tool o errori di compilazione del codice sorgente, ma riguarda l'uso appropriato di tool.

- (3) I tutor saranno spesso coinvolti nel monitoraggio dei vostri progetti. **Possono** supportarvi nella definizione e sviluppo del vostro progetto.

NB: Tuttavia, non sono dei membri aggiuntivi del vostro team. Da loro si va per chiedere spiegazioni sui giusti metodi da utilizzare, ma non per far correggere i vostri artefatti.

- (4) I tutor **non** sono tenuti a rispondere a domande su argomenti che non riguardano il corso di Ingegneria del Software - ad esempio, non possono colmare le lacune in Basi di Dati.

- (5) I tutor hanno impostato una e-mail dedicata alla quale richiedere informazioni. Per la nostra classe la mail è: tutoratois1@gmail.com. Utilizzate preferibilmente questa e-mail per interagire con loro, inviando l'eventuale materiale soggetto a chiarimento; ad esempio, se avete dubbi sulla scrittura di un sequence diagram, inviate ciò che avete prodotto prima di richiedere chiarimenti - in questo modo, la comunicazione sarà più proficua ed efficiente.

Informazioni amministrative - Materiale Didattico

- Kathy Schwalbe, "Information technology project management", International Edition 7e, Cengage Learning, © 2014
- Bernd Bruegge, Allen H. Dutoit "Object-oriented software engineering using UML, patterns and Java" - Third Ed. - Pearson, 2010
- I. Sommerville, Software engineering, Addison Wesley
- Harvard Business School Publishing: pack ad acquistare sul sito
- Pmbok® guide e software extension to the Pmbok® guide -5° Edition
- Pmbok® guide & software extension to the Pmbok® guide -5° Edition, Project management institute, 2013

Slide e materiale aggiuntivo

Disponibile sulla piattaforma e-learning del corso: <https://elearning.informatica.unisa.it/el-platform/course/view.php?id=1117>

Informazioni amministrative - Modalità di Esame

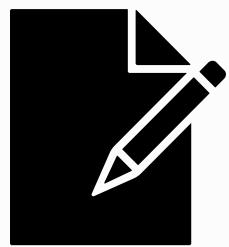
Modalità di Esame

Come ogni esame, ci sarà un accertamento individuale della conoscenza dei contenuti trattati durante il corso. Questa avverrà in due momenti.

Informazioni amministrative - Modalità di Esame

Modalità di Esame

Come ogni esame, ci sarà un accertamento individuale della conoscenza dei contenuti trattati durante il corso. Questa avverrà in due momenti.



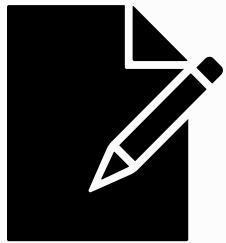
Prova scritta a fine corso.

La prova sarà composta da domande a risposta chiusa e aperta, la cui valutazione potrà arrivare ad un massimo di 30.

Informazioni amministrative - Modalità di Esame

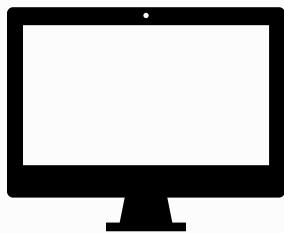
Modalità di Esame

Come ogni esame, ci sarà un accertamento individuale della conoscenza dei contenuti trattati durante il corso. Questa avverrà in due momenti.



Prova scritta a fine corso.

La prova sarà composta da domande a risposta chiusa, la cui valutazione potrà arrivare ad un massimo di 30.



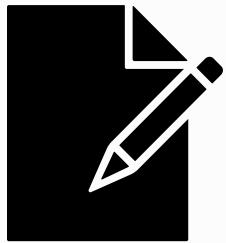
Progetto di gruppo + Discussione del progetto a fine corso.

Il progetto dovrà essere sviluppato con le metodologie/strumenti trattati a lezione, nonché rispettando i vincoli stabiliti - approfondimento a breve. La valutazione del progetto potrà arrivare ad un massimo di 30 e abiliterà l'eventuale lode.

Informazioni amministrative - Modalità di Esame

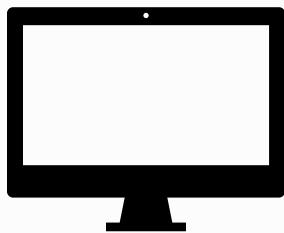
Modalità di Esame

Come ogni esame, ci sarà un accertamento individuale della conoscenza dei contenuti trattati durante il corso. Questa avverrà in due momenti.



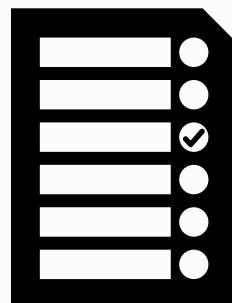
Prova scritta a fine corso.

La prova sarà composta da domande a risposta chiusa, la cui valutazione potrà arrivare ad un massimo di 30.



Progetto di gruppo + Discussione del progetto a fine corso.

Il progetto dovrà essere sviluppato con le metodologie/strumenti trattati a lezione, nonché rispettando i vincoli stabiliti - approfondimento a breve. La valutazione del progetto potrà arrivare ad un massimo di 30 e abiliterà l'eventuale lode.



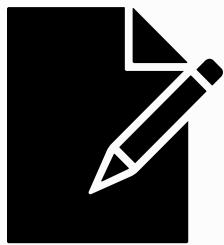
Votazione finale.

In linea generale, una media tra le valutazioni delle due prove. In casi eccezionali (ad esempio, adempimento di tutti i criteri di premialità nel progetto), il voto finale potrebbe essere incrementato fino a tre punti. La sufficienza sarà raggiunta a 18.

Informazioni amministrative - Modalità di Esame

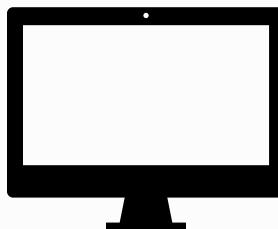
Modalità di Esame

Come ogni esame, ci sarà un accertamento individuale della conoscenza dei contenuti trattati durante il corso. Questa avverrà in due momenti.



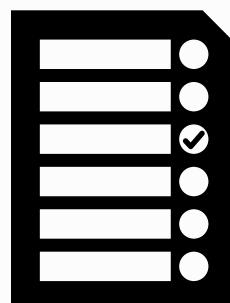
Prova scritta a fine corso.

La prova sarà composta da domande a risposta chiusa, la cui valutazione potrà arrivare ad un massimo di 30.



Progetto di gruppo + Discussione del progetto a fine corso.

Il progetto dovrà essere sviluppato con le metodologie/strumenti trattati a lezione, nonché rispettando i vincoli stabiliti - approfondimento a breve. La valutazione del progetto potrà arrivare ad un massimo di 30 e abiliterà l'eventuale lode.



Votazione finale.

In linea generale, una media tra le valutazioni delle due prove. In casi eccezionali (ad esempio, adempimento di tutti i criteri di premialità nel progetto), il voto finale potrebbe essere incrementato fino a tre punti. La sufficienza sarà raggiunta a 18.

Studenti impossibilitati a fare il progetto di gruppo.

Per essere esonerati è necessaria una motivazione valida (es: studenti lavoratori). Se esonerati, l'esame comprenderà solamente la prova scritta, ma la votazione massima sarà 24. Il modulo di esonero sarà disponibile sulla piattaforma e-learning del corso. La sufficienza sarà raggiunta a 18.

REGOLE PER IS

Che influenza anche GPS

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al coordinamento

Come da tradizione, il corso propone due tipologie di progetto che si differenziano in base al coordinamento necessario allo sviluppo.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al coordinamento

Come da tradizione, il corso propone due tipologie di progetto che si differenziano in base al coordinamento necessario allo sviluppo.

A

Intensa attività di coordinamento e gestione.

- Team di 6/8 persone coordinati da studenti senior;
- Proposta di progetto comunicata dai coordinatori;
- Individuazione dell'associazione team/coordinator sulla base di preferenze espresse dagli studenti di IS e la selezione da parte dei PM.



Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al coordinamento

Come da tradizione, il corso propone due tipologie di progetto che si differenziano in base al coordinamento necessario allo sviluppo.

A

Intensa attività di coordinamento e gestione.

- Team di 6/8 persone coordinati da studenti senior;
- Proposta di progetto comunicata dai coordinatori;
- Individuazione dell'associazione team/coordinator sulla base di preferenze espresse dagli studenti di IS e la selezione da parte dei PM.



B

Scarsa attività di coordinamento e gestione.

- Team di 3/4 persone senza coordinatore senior;
- Proposta di progetto presentata dal team;
- Scadenze identiche ai progetti di tipo A.



Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al coordinamento

Come da tradizione, il corso propone due tipologie di progetto che si differenziano in base al coordinamento necessario allo sviluppo.

A

Intensa attività di coordinamento e gestione.

- Team di 6/8 persone coordinati da studenti senior;
- Proposta di progetto comunicata dai coordinatori;
- Individuazione dell'associazione team/coordinator sulla base di preferenze espresse dagli studenti di IS e la selezione da parte dei PM.



B

Scarsa attività di coordinamento e gestione.

- Team di 3/4 persone senza coordinatore senior;
- Proposta di progetto presentata dal team;
- Scadenze identiche ai progetti di tipo A.



Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al focus del progetto

1

Greenfield Engineering

- Focus su **progettazione** e testing;
- Dalle esigenze, si definiscono i requisiti, la progettazione, lo sviluppo, il testing.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al coordinamento

Come da tradizione, il corso propone due tipologie di progetto che si differenziano in base al coordinamento necessario allo sviluppo.

A

Intensa attività di coordinamento e gestione.

- Team di 6/8 persone coordinati da studenti senior;
- Proposta di progetto comunicata dai coordinatori;
- Individuazione dell'associazione team/coordinator sulla base di preferenze espresse dagli studenti di IS e la selezione da parte dei PM.



B

Scarsa attività di coordinamento e gestione.

- Team di 3/4 persone senza coordinatore senior;
- Proposta di progetto presentata dal team;
- Scadenze identiche ai progetti di tipo A.



Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al focus del progetto

1

Greenfield Engineering

- Focus su **progettazione** e testing;
- Dalle esigenze, si definiscono i requisiti, la progettazione, lo sviluppo, il testing.

2

Open-Source

- Focus su **testing** e progettazione;
- Da un progetto open-source esistente, l'obiettivo è individuare failure.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Tipologie in base al coordinamento

Come da tradizione, il corso propone due tipologie di progetto che si differenziano in base al coordinamento necessario allo sviluppo.

A

Intensa attività di coordinamento e gestione.

- Team di 6/8 persone coordinati da studenti senior;
- Proposta di progetto comunicata dai coordinatori;
- Individuazione dell'associazione team/coordinator sulla base di preferenze espresse dagli studenti di IS e la selezione da parte dei PM.



B

Scarsa attività di coordinamento e gestione.

- Team di 3/4 persone senza coordinatore senior;
- Proposta di progetto presentata dal team;
- Scadenze identiche ai progetti di tipo A.



Modalità di Esame - Alcuni Chiarimenti sui Progetti

1

Greenfield Engineering

- Focus su **progettazione** e testing;
- Dalle esigenze, si definiscono i requisiti, la progettazione, lo sviluppo, il testing.

2

Sebbene il focus sia sul testing, sarà comunque necessario documentare parte del progetto. NB: Potreste fregiarvi del titolo di **sviluppatore open source!**

Open Source

- Focus su **testing** e progettazione;
- Da un progetto open-source esistente, l'obiettivo è individuare failure.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

In tal caso, il progetto dovrà integrare una componente “intelligente” come richiesto dall’insegnamento di Fondamenti di Intelligenza Artificiale. Maggiori dettagli saranno forniti durante la prima lezione del corso di FIA.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

In tal caso, il progetto dovrà integrare una componente “intelligente” come richiesto dall’insegnamento di Fondamenti di Intelligenza Artificiale. Maggiori dettagli saranno forniti durante la prima lezione del corso di FIA.

NB: **Non** è indispensabile che tutti i membri di un team di IS facciano FIA. Tuttavia, è chiaro che della componente intelligente dovranno occuparsene gli studenti di FIA.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

In tal caso, il progetto dovrà integrare una componente “intelligente” come richiesto dall’insegnamento di Fondamenti di Intelligenza Artificiale. Maggiori dettagli saranno forniti durante la prima lezione del corso di FIA.

NB: **Non** è indispensabile che tutti i membri di un team di IS facciano FIA. Tuttavia, è chiaro che della componente intelligente dovranno occuparsene gli studenti di FIA.

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Sintesi e scadenze

A

- A1f. Greenfield Engineering - FIA
- A1nf. Greenfield Engineering - No FIA
- A2. Open-Source

B

- B1f. Greenfield Engineering - FIA
- B1nf. Greenfield Engineering - No FIA
- B2. Open-Source

Step:

- I step: Consegna schede informative; schede disponibili sulla piattaforma.
- II step: Formazione gruppi e inizio proposte di progetto.
- III step:
 - Kick-off meeting;
 - Progetti tipo A: Coordinatori illustrano la proposta di progetto.
 - Progetti tipo B: Gruppi illustrano la proposta di progetto ai docenti.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

In tal caso, il progetto dovrà integrare una componente “intelligente” come richiesto dall’insegnamento di Fondamenti di Intelligenza Artificiale. Maggiori dettagli saranno forniti durante la prima lezione del corso di FIA.

NB: **Non** è indispensabile che tutti i membri di un team di IS facciano FIA. Tuttavia, è chiaro che della componente intelligente dovranno occuparsene gli studenti di FIA.

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Sintesi e scadenze

Step:

- I step: Consegna schede informative; schede disponibili sulla piattaforma.
- II step: Formazione gruppi e inizio proposte di progetto.
- III step:
 - Kick-off meeting;
 - Progetti tipo A: Coordinatori illustrano la proposta di progetto.
 - Progetti tipo B: Gruppi illustrano la proposta di progetto ai docenti.

Scadenze consegne intermedie/definitive:

- Metà Dicembre: Coloro che vogliono, potranno concludere l’intero progetto e discuterlo.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

In tal caso, il progetto dovrà integrare una componente “intelligente” come richiesto dall’insegnamento di Fondamenti di Intelligenza Artificiale. Maggiori dettagli saranno forniti durante la prima lezione del corso di FIA.

NB: **Non** è indispensabile che tutti i membri di un team di IS facciano FIA. Tuttavia, è chiaro che della componente intelligente dovranno occuparsene gli studenti di FIA.

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Sintesi e scadenze

Step:

- I step: Consegna schede informative; schede disponibili sulla piattaforma.
- II step: Formazione gruppi e inizio proposte di progetto.
- III step:
 - Kick-off meeting;
 - Progetti tipo A: Coordinatori illustrano la proposta di progetto.
 - Progetti tipo B: Gruppi illustrano la proposta di progetto ai docenti.

Scadenze consegne intermedie/definitive:

- Metà Dicembre: Coloro che vogliono, potranno concludere l’intero progetto e discuterlo.
- Le altre scadenze saranno in concomitanza con gli appelli di esame.

Informazioni amministrative - Progetti

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Ulteriore possibilità

Se la tipologia di progetto selezionata è di greenfield engineering, è possibile realizzare **combinare** i progetti di Ingegneria del Software con quelli di Fondamenti di Intelligenza Artificiale.

In tal caso, il progetto dovrà integrare una componente “intelligente” come richiesto dall’insegnamento di Fondamenti di Intelligenza Artificiale. Maggiori dettagli saranno forniti durante la prima lezione del corso di FIA.

NB: **Non** è indispensabile che tutti i membri di un team di IS facciano FIA. Tuttavia, è chiaro che della componente intelligente dovranno occuparsene gli studenti di FIA.

Modalità di Esame - Alcuni Chiarimenti sui Progetti; Sintesi e scadenze

Step:

- I step: Consegna schede informative; schede disponibili sulla piattaforma.
- II step: Formazione gruppi e inizio proposte di progetto.
- III step:
 - Kick-off meeting;
 - Progetti tipo A: Coordinatori illustrano la proposta di progetto.
 - Progetti tipo B: Gruppi illustrano la proposta di progetto ai docenti.

Scadenze consegne intermedie/definitive:

- Metà Dicembre: Coloro che vogliono, potranno concludere l’intero progetto e discuterlo.
- Le altre scadenze saranno in concomitanza con gli appelli di esame.

Studenti con DSA:

- Fanno pervenire prima possibile ai docenti le indicazioni da parte dell’ufficio diritto allo studio.
- Si individueranno le modalità più opportune.

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: “Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura” [Treccani]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente “controllato” in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonale ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: “Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura” [Treccani]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente “controllato” in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Quindi, l'effort sarà più orientato agli aspetti collaborativi e comunicativi piuttosto che all'aspetto tecnico.

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: “Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura” [Treccani]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente “controllato” in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Quindi, l'effort sarà più orientato agli aspetti collaborativi e comunicativi piuttosto che all'aspetto tecnico.

Questo implica nuovi vincoli - il ché rende i progetti completamente diversi dagli anni scorsi.

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: *"Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura"* [Treccani]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente "controllato" in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Quindi, l'effort sarà più orientato agli aspetti collaborativi e comunicativi piuttosto che all'aspetto tecnico.

Questo implica nuovi vincoli - il ché rende i progetti completamente diversi dagli anni scorsi.

Approfondiamo. Innanzitutto, avremo:

- **Vincoli collaborativi e comunicativi.** Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: *"Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura"* [Treccani]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente "controllato" in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Quindi, l'effort sarà più orientato agli aspetti collaborativi e comunicativi piuttosto che all'aspetto tecnico.

Questo implica nuovi vincoli - il ché rende i progetti completamente diversi dagli anni scorsi.

Approfondiamo. Innanzitutto, avremo:

- **Vincoli collaborativi e comunicativi.** Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.
- **Vincoli tecnici.** Limitazioni sugli artefatti da realizzare.

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: *"Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura"* [Trecanni]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente "controllato" in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Quindi, l'effort sarà più orientato agli aspetti collaborativi e comunicativi piuttosto che all'aspetto tecnico.

Questo implica nuovi vincoli - il ché rende i progetti completamente diversi dagli anni scorsi.

Approfondiamo. Innanzitutto, avremo:

- **Vincoli collaborativi e comunicativi.** Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.
- **Vincoli tecnici.** Limitazioni sugli artefatti da realizzare.
- **Criteri di accettazione.** Criteri che, se non rispettati, portano al fallimento del progetto.

Informazioni amministrative - Progetti

Vincoli progettuali

Come già detto in precedenza, i vincoli progettuali hanno subito cambiamenti sostanziali al fine di implementare un'idea di base: **è necessario fare meno, ma meglio!** O, in altri termini: **meno quantità ma più qualità!**

Ingegneria del Software: Perché?

In altri termini, lo sviluppo software è principalmente uno sforzo collaborativo e comunicativo, all'interno del quale la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Da qui il termine **Ingegneria**: *"Insieme di studi e tecniche che utilizzano le conoscenze delle varie branche delle scienze, unite a quelle tecnologiche, per risolvere problemi applicativi e per progettare e realizzare opere di diversa natura"* [Trecanni]

Già nel corso di questa lezione, approfondiremo come la definizione viene mappata al software e quali sono le principali problematiche che rendono un processo ingegneristico inevitabile.

In questo corso, apprenderete le implicazioni di queste problematiche e come gestirle - ma fortunatamente, lo apprenderete in un ambiente "controllato" in cui le conseguenze non sono catastrofiche né irreparabili!

Apprenderete che le capacità tecniche, da sole, contano ben poco e che lo sviluppo software ha componenti e dinamiche umane, sociali ed organizzative paragonabili alla costruzione di qualunque altro prodotto architettonico ed ingegneristico - pertanto, richiede disciplina.

Apprenderete infine che la progettazione di cui parleremo non è niente di più che la naturale evoluzione di ciò che avete già parlato nei precedenti corsi.

In che senso? Facciamo un esempio...

In che modo?

Simulando più fedelmente un caso reale. Ripartiamo dall'inizio: lo sviluppo software è **principalmente** uno sforzo collaborativo e comunicativo, **all'interno del quale** la progettazione consente di raggiungere gli obiettivi prestabiliti con il committente.

Quindi, l'effort sarà più orientato agli aspetti collaborativi e comunicativi piuttosto che all'aspetto tecnico.

Questo implica nuovi vincoli - il ché rende i progetti completamente diversi dagli anni scorsi.

Approfondiamo. Innanzitutto, avremo:

- **Vincoli collaborativi e comunicativi.** Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.
- **Vincoli tecnici.** Limitazioni sugli artefatti da realizzare.
- **Criteri di accettazione.** Criteri che, se non rispettati, portano al fallimento del progetto.
- **Criteri di premialità.** Criteri che, se rispettati, portano a punti aggiuntivi al progetto.

Vincoli progettuali

Vincoli collaborativi e comunicativi. Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.

- Rispetto delle scadenze intermedie/di fine progetto **definite dai project manager**, per i progetti di tipo A, e **definite nello statement of work**, per i progetti di tipo B.
- Uso di sistemi di versioning - **GitHub** in particolare.
- Uso di tool per la gestione di task e attività - **Trello** o simili.
- Uso di un tool di comunicazione tracciabile - **Slack, MS Teams** o **Discord**.

Già nei primi due laboratori, parleremo dei sistemi di versioning e di sistemi per la gestione di task/attività e per la comunicazione, in modo da agevolare il vostro lavoro.

Vincoli progettuali

Vincoli collaborativi e comunicativi. Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.

- Rispetto delle scadenze intermedie/di fine progetto **definite dai project manager**, per i progetti di tipo A, e **definite nello statement of work**, per i progetti di tipo B.
- Uso di sistemi di versioning - **GitHub** in particolare.
- Uso di tool per la gestione di task e attività - **Trello** o simili.
- Uso di un tool di comunicazione tracciabile - **Slack, MS Teams** o **Discord**.

Già nei primi due laboratori, parleremo dei sistemi di versioning e di sistemi per la gestione di task/attività e per la comunicazione, in modo da agevolare il vostro lavoro.

NB: Patti chiari, amicizia lunga! Il mancato uso di GitHub, Trello e/o Slack porta **automaticamente** al fallimento del progetto —> il progetto non verrà accettato né discusso; nessuna eccezione.

Informazioni amministrative - Progetti

Vincoli progettuali

Vincoli collaborativi e comunicativi. Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.

- Rispetto delle scadenze intermedie/di fine progetto **definite dai project manager**, per i progetti di tipo A, e **definite nello statement of work**, per i progetti di tipo B.
- Uso di sistemi di versioning - **GitHub** in particolare.
- Uso di tool per la gestione di task e attività - **Trello** o simili.
- Uso di un tool di comunicazione tracciabile - **Slack, MS Teams** o **Discord**.

Già nei primi due laboratori, parleremo dei sistemi di versioning e di sistemi per la gestione di task/attività e per la comunicazione, in modo da agevolare il vostro lavoro.

NB: Patti chiari, amicizia lunga! Il mancato uso di GitHub, Trello e/o Slack porta **automaticamente** al fallimento del progetto —> il progetto non verrà accettato né discusso; nessuna eccezione.

Chiaramente, non basta aver installato, ad esempio, Slack per considerare il vincolo rispettato, ma su questo ci arriveremo tra poco...

Informazioni amministrative - Progetti

Vincoli progettuali

Vincoli collaborativi e comunicativi. Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto.

- Rispetto delle scadenze intermedie/di fine progetto **definite dai project manager**, per i progetti di tipo A, e **definite nello statement of work**, per i progetti di tipo B.
- Uso di sistemi di versioning - **GitHub** in particolare.
- Uso di tool per la gestione di task e attività - **Trello** o simili.
- Uso di un tool di comunicazione tracciabile - **Slack, MS Teams** o **Discord**.

Già nei primi due laboratori, parleremo dei sistemi di versioning e di sistemi per la gestione di task/attività e per la comunicazione, in modo da agevolare il vostro lavoro.

NB: Patti chiari, amicizia lunga! Il mancato uso di GitHub, Trello e/o Slack porta **automaticamente** al fallimento del progetto —> il progetto non verrà accettato né discusso; nessuna eccezione.

Chiaramente, non basta aver installato, ad esempio, Slack per considerare il vincolo rispettato, ma su questo ci arriveremo tra poco...

NB: Già considerando questi vincoli, lo statement of work dovrà essere più dettagliato di quelli sviluppati in passato —> ancora una volta, copiare non porta a nulla!

Vincoli progettuali

Vincoli tecnici (1). Limitazioni sugli artefatti da realizzare.

Analisi e Specifica dei Requisiti.

- Specifica di **minimo** 2 e **massimo** 4 scenari per ogni membro del team;
- **Esattamente** uno use case per ogni membro del team - i casi d'uso aggiuntivi **non** saranno valutati;
- Specifica degli oggetti **boundary**, **control** e **entity** per gli use case specificati;
- **Esattamente** un sequence diagram ogni due membri del team - i sequence diagram aggiuntivi **non** saranno valutati;
- **Almeno un** activity diagram per sistema esistente o sistema proposto e **almeno uno** statechart diagram ogni due membri del team. La somma degli activity e statechart diagram **deve essere uguale al numero di membri del team diviso due**.

Vincoli progettuali

Vincoli tecnici (1). Limitazioni sugli artefatti da realizzare.

Analisi e Specifica dei Requisiti.

- Specifica di **minimo** 2 e **massimo** 4 scenari per ogni membro del team;
- **Esattamente** uno use case per ogni membro del team - i casi d'uso aggiuntivi **non** saranno valutati;
- Specifica degli oggetti **boundary**, **control** e **entity** per gli use case specificati;
- **Esattamente** un sequence diagram ogni due membri del team - i sequence diagram aggiuntivi **non** saranno valutati;
- **Almeno un** activity diagram per sistema esistente o sistema proposto e **almeno uno** statechart diagram ogni due membri del team. La somma degli activity e statechart diagram **deve essere uguale al numero di membri del team diviso due**.

NB: L'analisi dei requisiti è storicamente la parte in cui gli studenti hanno maggiore difficoltà, oltre ad essere il momento in cui gli artefatti vengono modificati più spesso. Quindi, **non** strafate ed **attenetevi** ai vincoli! Se fate di più, non rispetterete i vincoli e, pertanto, **questo porterà ad avere un punteggio inferiore**.

Informazioni amministrative - Progetti

Vincoli progettuali

Vincoli tecnici (1). Limitazioni sugli artefatti da realizzare.

Analisi e Specifica dei Requisiti.

- Specifica di **minimo** 2 e **massimo** 4 scenari per ogni membro del team;
- **Esattamente** uno use case per ogni membro del team - i casi d'uso aggiuntivi **non** saranno valutati;
- Specifica degli oggetti **boundary**, **control** e **entity** per gli use case specificati;
- **Esattamente** un sequence diagram ogni due membri del team - i sequence diagram aggiuntivi **non** saranno valutati;
- **Almeno un** activity diagram per sistema esistente o sistema proposto e **almeno uno** statechart diagram ogni due membri del team. La somma degli activity e statechart diagram **deve essere uguale al numero di membri del team diviso due.**

NB: L'analisi dei requisiti è storicamente la parte in cui gli studenti hanno maggiore difficoltà, oltre ad essere il momento in cui gli artefatti vengono modificati più spesso. Quindi, **non** strafate ed **attenetevi** ai vincoli! Se fate di più, non rispetterete i vincoli e, pertanto, **questo porterà ad avere un punteggio inferiore.**

NB: In virtù di questi vincoli, ancora una volta, “prendere spunto” dai progetti passati è un errore. Siate furbi: rispettate i vincoli.

Vincoli progettuali

Vincoli tecnici (2). Limitazioni sugli artefatti da realizzare.

System Design.

- Specifica dei **design goal** e analisi dei **trade-off relativi ad almeno due coppie di design goal**.
- Definizione dell'**architettura** del sistema.

Vincoli progettuali

Vincoli tecnici (2). Limitazioni sugli artefatti da realizzare.

System Design.

- Specifica dei **design goal** e analisi dei **trade-off relativi ad almeno due coppie di design goal**.
- Definizione dell'**architettura** del sistema.

Object Design.

- Individuazione di **dove poter utilizzare due design pattern**, indicandone l'obiettivo e come sarebbero implementati.

Vincoli progettuali

Vincoli tecnici (2). Limitazioni sugli artefatti da realizzare.

System Design.

- Specifica dei **design goal** e analisi dei **trade-off relativi ad almeno due coppie di design goal**.
- Definizione dell'**architettura** del sistema.

Object Design.

- Individuazione di **dove poter utilizzare due design pattern**, indicandone l'obiettivo e come sarebbero implementati.

Testing.

- Ogni studente dovrà effettuare il testing di unità, tramite category partition, **dei casi d'uso specificati**.

Vincoli progettuali

Vincoli tecnici (2). Limitazioni sugli artefatti da realizzare.

System Design.

- Specifica dei **design goal** e analisi dei **trade-off relativi ad almeno due coppie di design goal**.
- Definizione dell'**architettura** del sistema.

Object Design.

- Individuazione di **dove poter utilizzare due design pattern**, indicandone l'obiettivo e come sarebbero implementati.

Testing.

- Ogni studente dovrà effettuare il testing di unità, tramite category partition, **dei casi d'uso specificati**.

NB: Sebbene il carico di lavoro richiesto a valle dell'analisi e specifica dei requisiti è stato notevolmente ridotto, sarà necessario comunque sviluppare una documentazione **snella** delle attività condotte nel progetto.

Vincoli progettuali

Vincoli tecnici (2). Limitazioni sugli artefatti da realizzare.

System Design.

- Specifica dei **design goal** e analisi dei **trade-off relativi ad almeno due coppie di design goal**.
- Definizione dell'**architettura** del sistema.

Object Design.

- Individuazione di **dove poter utilizzare due design pattern**, indicandone l'obiettivo e come sarebbero implementati.

Testing.

- Ogni studente dovrà effettuare il testing di unità, tramite category partition, **dei casi d'uso specificati**.

NB: Sebbene il carico di lavoro richiesto a valle dell'analisi e specifica dei requisiti è stato notevolmente ridotto, sarà necessario comunque sviluppare una documentazione **snella** delle attività condotte nel progetto.

NB: Le attività di più basso livello, ovvero implementazione e testing, saranno supportate dall'utilizzo di strumenti di **smart programming**: già nei prossimi laboratori, discuteremo di come utilizzare Large Language Model come tutor di sviluppo.

Vincoli progettuali

Criteri di accettazione. Criteri che, se non rispettati, portano al fallimento del progetto.

- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab.** Chi non lo fa, **non sarà valutato al progetto.**
- Adeguato utilizzo del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del primo lab.**
- Adeguato utilizzo di Slack o simili, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab.**
- Adeguato utilizzo di Trello, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab.**
- Documentazione adeguata. Verranno usati tool di **plagiarism detection** per identificare casi in cui gli studenti hanno copiato da progetti di anni precedenti e/o da altre fonti.
- Utilizzo di **tool di controllo della qualità** per l'implementazione (CheckStyle)
- Appropriato test di unità di un metodo sviluppato, che preveda il **rispetto dei vincoli.**

Vincoli progettuali

Criteri di accettazione. Criteri che, se non rispettati, portano al fallimento del progetto.

- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab.** Chi non lo fa, **non sarà valutato al progetto.**
- Adeguato utilizzo del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del primo lab.**
- Adeguato utilizzo di Slack o simili, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab.**
- Adeguato utilizzo di Trello, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab.**
- Documentazione adeguata. Verranno usati tool di **plagiarism detection** per identificare casi in cui gli studenti hanno copiato da progetti di anni precedenti e/o da altre fonti.
- Utilizzo di **tool di controllo della qualità** per l'implementazione (CheckStyle)
- Appropriato test di unità di un metodo sviluppato, che preveda il **rispetto dei vincoli.**

NB: Sembrano criteri di difficile applicazione, ma non lo sono affatto. Bisogna diventare dei professionisti e sviluppare software in maniera adeguata: se seguirete il corso, imparerete presto che fare le cose per bene è sempre un guadagno!

Vincoli progettuali

Criteri di premialità. Criteri che, se rispettati, portano a punti aggiuntivi al progetto.

- Uso adeguato di **sistemi di build**;
- Uso adeguato di un processo di **continuous integration tramite Travis**;
- Adozione di **processi di code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).

Vincoli progettuali

Criteri di premialità. Criteri che, se rispettati, portano a punti aggiuntivi al progetto.

- Uso adeguato di **sistemi di build**;
- Uso adeguato di un processo di **continuous integration tramite Travis**;
- Adozione di **processi di code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).

NB: Visti così, anche questi sembrano criteri di difficile applicazione, ma ancora una volta: basta “solo” essere dei professionisti! Mettetevi alla prova!

Vincoli progettuali

Criteri di premialità. Criteri che, se rispettati, portano a punti aggiuntivi al progetto.

- Uso adeguato di **sistemi di build**;
- Uso adeguato di un processo di **continuous integration tramite Travis**;
- Adozione di **processi di code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).

NB: Visti così, anche questi sembrano criteri di difficile applicazione, ma ancora una volta: basta “solo” essere dei professionisti! Mettetevi alla prova!

Questionario. Entro il 27/09, si prega di compilare il seguente questionario:

L'indicazione della tipologia di progetto è al momento solo indicativa; potrà essere modificata nel prossimo form - solo allora la scelta sarà definitiva.

Vincoli progettuali

Criteri di premialità. Criteri che, se rispettati, portano a punti aggiuntivi al progetto.

- Uso adeguato di **sistemi di build**;
- Uso adeguato di un processo di **continuous integration tramite Travis**;
- Adozione di **processi di code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito, Cobertura**, etc.).

NB: Visti così, anche questi sembrano criteri di difficile applicazione, ma ancora una volta: basta “solo” essere dei professionisti! Mettetevi alla prova!

Questionario. Entro il 27/09, si prega di compilare il questionario presente sulla piattaforma

L'indicazione della tipologia di progetto è al momento solo indicativa; potrà essere modificata nel prossimo form - solo allora la scelta sarà definitiva.

Statement of Work. La prima scadenza progettuale da rispettare (04/10) è quella della presentazione dello statement of work, il documento che lista la vostra proposta di progetto e dettaglia i vincoli progettuali.

L'accreditamento professionale del PMI

Find direction.
Build credentials.



Il PMI ha sviluppato dal 1984 un programma di certificazione professionale, al fine di garantire un'adeguato riconoscimento della professionalità degli operatori nel campo del Project Management:

- ✓ **PMP - Project Management Professional**
- ✓ **CAPM - Certified Associate of Project Management**
- ✓ **PMI RMP - PMI Risk Management Professional**
- ✓ **PMI SP - PMI Scheduling Professional**
- ✓ **PgMP - Program Management Professional**



Certified Associate of Project Management (CAPM)



CAPM®	
Full Name	Certified Associate in Project Management
Project Role	Contributes to project team
Eligibility Requirements	Secondary diploma (high school diploma/global equivalent) AND 1,500 hours experience OR 23 hours pm education
Steps to Obtaining Credential	Application process + multiple-choice exam
Exam Information	3 hours; 150 questions
Fee	US\$225 PMI member (US\$300 nonmember)
Credential Maintenance Cycles and Requirements	5 years; re-exam

La CAPM è la certificazione base che si rivolge a chi si avvicina al mondo del project management, come i neolaureati e a chi entra per la prima volta nel mondo del lavoro.

- **Credenziale di livello base**
- **Lanciata nel 2007**
- **Si consegna a seguito di un esame più semplice di quello per la PMP**
- **Per accedere all'esame occorre avere come requisiti:**
 - Diploma
 - **35 ore di formazione sul PM**
 - Oppure
 - **1 anni di esperienza (1.500 ore) nel Project Management**
- **Validità 5 anni, non rinnovabile**





UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Attestato di partecipazione

A

Valeria Pontillo

Per

Corso di Gestione dei Progetti Software

*Il corso ha trattato tutte le aree del PMBOK® Guide v.6
(durata: 36 ore, periodo settembre/dicembre 2018)*

7 dicembre 2018

Nome e qualifica del docente

Giacomo Franco PMP® #207773

Docente

Prof.ssa Filomena Ferrucci

ULTERIORE INIZIATIVA CON PMI: “CHALLANGE INTER-ATENEO: PM4INNOVATION CHALLENGE ”

- *E' una competizione finalizzata allo sviluppo di conoscenze e competenze su principi e metodi del project management, attraverso una didattica esperienziale e project based che prevede l'ideazione e la redazione di progetti innovativi da parte di team di studenti sotto la supervisione dei docenti di riferimento.*
- *I progetti per l' A.A. 2024-2025 hanno come tema provvisorio “Innovazione Sociale”.*
- *Agli studenti è chiesto di ideare ed elaborare idee innovative soggette ad una serie di vincoli (di budget, di tempo e di disponibilità di risorse umane) per poter rispondere al tema della sfida. Le idee sono poi articolate in progetti attraverso la redazione del Project Management plan, il documento fondamentale contenente l'articolazione di tutti gli aspetti relativi ai tempi e risorse necessarie, rischi nella realizzazione, benefici attesi, secondo le linee guida del project management.*
- *Una giuria di esperti del Project Management Institute - Southern Italy Chapter valuterà i progetti finalisti attribuendo premi ai migliori progetti presentati.*

Filomena Ferrucci, Fabio Palomba
SeSa Lab - University of Salerno

Ingegneria del Software

