

Introduzione al corso

[DT-0540] Metodi di sviluppo agile

Daniele Di Pompeo

daniele.dipompeo@univaq.it



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

Studenti con problematiche certificate e che necessitano di supporto sono invitati a segnalare in privato la propria condizione a:

- **Docente del corso:** Daniele Di Pompeo (daniele.dipompeo@univaq.it)
- **Referente del DISIM per la disabilità/Disturbi dell'apprendimento (DSA):** Giovanna Melideo (giovanna.melideo@univaq.it)



About myself

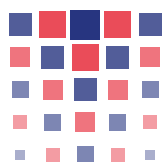
I'm an assistant professor at the University of L'Aquila previously post-doc researcher at the EMERGE (2020-2023) DEWS (2019-2020) Centre of Excellences.

My research interests mainly focus in the area of Software Performance Engineering Software Refactoring Model-Driven Engineering.

I have been involved in European and National projects:

- RECHARGE [PRIN] (started on 2023)
- Sobigdata++ [EU] (started on 2023)
- MegaM@RT2 [EU] (finished on 2020)
- Emerge [ITA] (started on 2019)
- AIDOaRT [EU] (started on 2021)





SOBIGDATA^{.it}

ITALIAN RESEARCH INFRASTRUCTURE

<https://plusplus.sobigdata.eu/>

Deliver a distributed, Pan-European, multi-disciplinary research infrastructure for big social data analytics, coupled with the consolidation of a cross-disciplinary European research community, aimed at using social mining and big data to understand the complexity of our contemporary, globally-interconnected society.



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025



A framework that leverages static analysis and search-based algorithms for automating performance testing in CI/CD pipelines. This project proposes a novel framework for automating performance testing in CI/CD pipelines by leveraging static analysis and search-based algorithms.



Comunicazioni

Potete comunicare con il docente tramite

- Discussione in classe
- sul canale teams del corso
- via email usando l'oggetto *[Agile24-25]*

Eventuali ricevimenti saranno concordati



Anatomia del corso

Codice Team

h1gwx5z

URL

<https://teams.microsoft.com/l/team/19%3ASQbBk8NF9j3YJmfIN4fey16DDkC0yj-nuX3ixYX2aws1%40thread.tacv2/conversations?groupId=93eb4def-1e65-4309-881f-9e47b3110ada&tenantId=9df08a7c-31d7-4024-9ba6-5ed5efac1a01>

Example

- Lezioni frontali
- Laboratori
- [opt] Seminari
- Revisioni del progetto



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

Calendario Lezioni

- TBD
- TBD



Obiettivi del corso

- Conoscenza delle diverse metodologie “Agile”
- Competenze nel gestire un lavoro di team
- Organizzare il lavoro rispettando le scadenze
- Competenze base/medio di GIT
- Dependency Management (Maven, Composer)
- Test Driven Development



Modalità d'esame

- Progetto di gruppo
 - ▶ 3 / 4 (*massimo*) componenti per gruppo
- Ogni sprint deve essere:
 - ▶ consegnato al docente e discusso durante le revisioni
 - ▶ “congelato” in una release sul sistema di versioning (e.g., GitHub)
- Orale

Verrà acquisito l'esame solamente con un voto sufficiente nel lavoro di progetto e orale



Primo Assignment: Formare il gruppo di lavoro

- Creazione del canale privato nel team del corso
- Assegnare un nome al gruppo
- Aggiungere tutti i membri del gruppo e il docente
- Comunicare il gruppo anche via e-mail
 - ▶ Oggetto dell'email *[Agile24-24] Gruppo <Nome del gruppo>*

Chi ha problemi nel trovare un gruppo lo comunichi al docente

DEADLINE

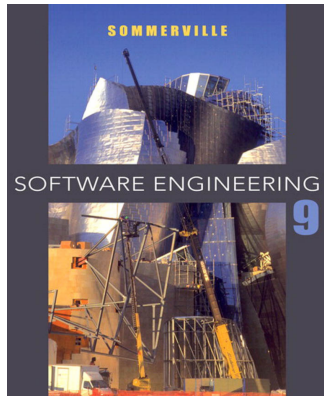
TBD



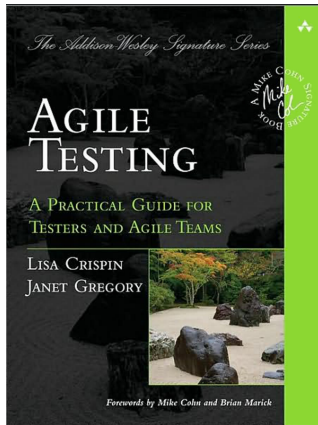
UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

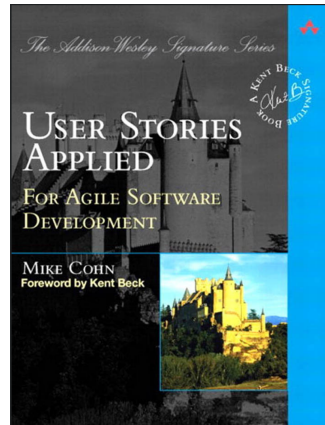
Testi di riferimento



Sommerville



Agile Testing



User Stories Applied



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

Introduzione ai processi di sviluppo software



Development methodologies

- Successful projects are managed well.
- To manage a project efficiently, the manager or development team must choose the software development methodology that will work best for the project at hand.
- All methodologies have different **strengths** and **weaknesses** and exist for different reasons.



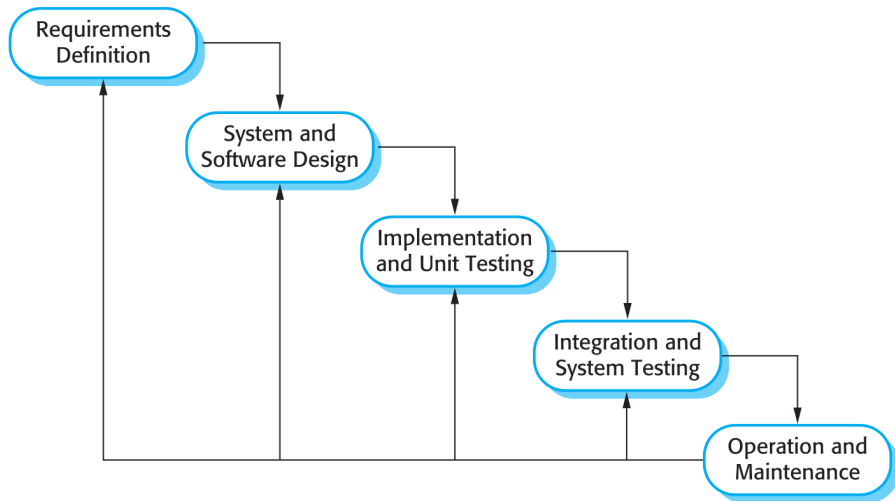
Development methodologies

- A methodology can help make the process efficient and reliable by guiding users on which method to employ at each step.
- A method is a structured procedure for bringing about a certain goal.
- The study of methods involves a detailed description and analysis of these processes. It includes evaluative aspects by comparing different methods to assess their advantages and disadvantages relative to different research goals and situations.

<https://en.wikipedia.org/wiki/Methodology>



Waterfall



Waterfall

- Requirements definition: The system's services, constraints, and goals are established by consultation with system users.
- System and software design: The systems design process allocates the requirements to either hardware or software systems by establishing an overall system architecture.
- Implementation and unit testing: The software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.
- Integration and system testing: The individual program units or programs are integrated and tested as a complete system. After testing, the software system is delivered to the customer.
- Operation and maintenance: Normally (although not necessarily), this is the longest life cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle.



Waterfall

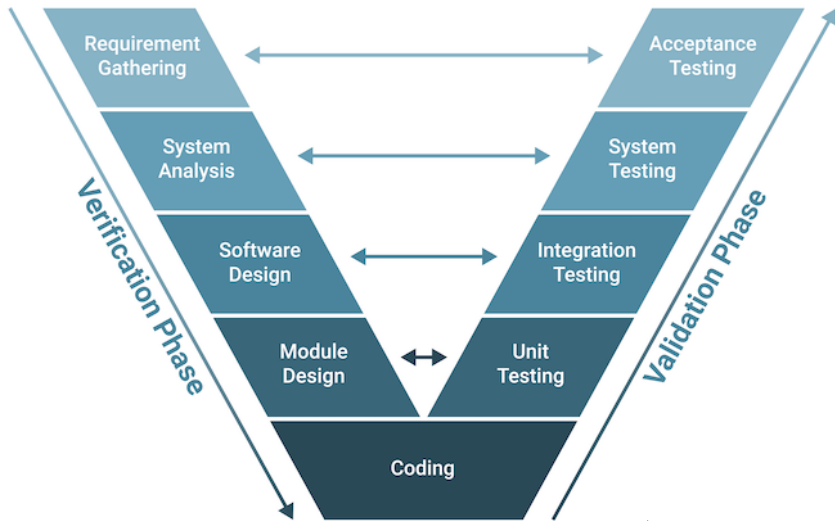
A software process model that involves discrete development stages: specification, design, implementation, testing, and maintenance.

In principle, one stage must be complete before progress to the next stage is possible.

In practice, there is significant iteration between stages.



V-Model



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

- **Verification and Validation** The V-Model is an extension of the waterfall model that emphasizes verification and validation. It represents the process of software development in a V-shape, where each development phase has a corresponding testing phase.
- **Development Phases** On the left side of the V, the development phases include requirements analysis, system design, architectural design, and module design. Each phase ends with a deliverable that is tested in the corresponding phase on the right side of the V.
- **Testing Phases** The right side of the V represents the testing phases, including unit testing, integration testing, system testing, and acceptance testing, which are used to validate the deliverables from the development phases.

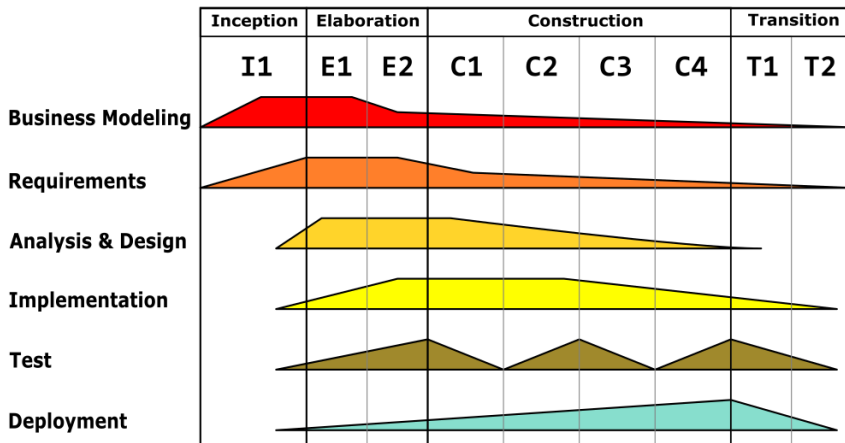


V-Model

- **Requirements Analysis** Capturing detailed requirements which are used as the basis for acceptance testing.
- **System Design** Defining the overall system architecture and how components will interact.
- **Module Design** Breaking down the system into modules, with each module being designed in detail.
- **Unit Testing** Testing individual modules for correctness.
- **Integration Testing** Ensuring that modules interact correctly and that data flows smoothly between them.
- **System Testing** Testing the entire system to verify that all requirements have been met.
- **Acceptance Testing** Validating the system against the requirements, often with the involvement of the end user.



Rational Unified Process



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

Rational Unified Process

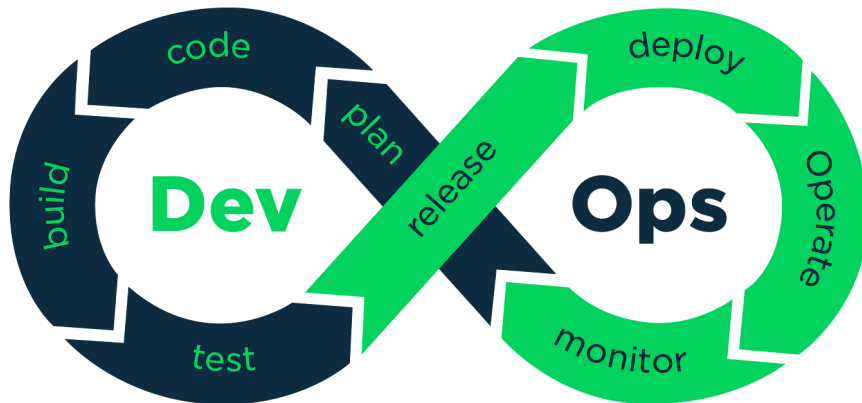
The RUP is a phased model that identifies four discrete phases in the software process. However, unlike the waterfall model where phases are equated with process activities, the phases in the RUP are more closely related to business rather than technical concerns.



Rational Unified Process

- **Inception** The goal of the inception phase is to establish a business case for the system. You should identify all external entities (people and systems) that will interact with the system and define these interactions.
- **Elaboration** The goals of the elaboration phase are to develop an understanding of the problem domain, establish an architectural framework for the system, develop the project plan, and identify key project risks.
- **Construction** The construction phase involves system design, programming, and testing. Parts of the system are developed in parallel and integrated during this phase.
- **Transition** The final phase of the RUP is concerned with moving the system from the development community to the user community and making it work in a real environment.

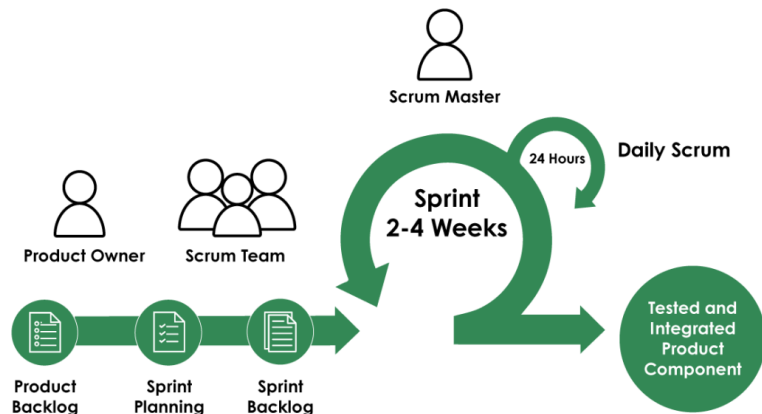




- **Collaboration** DevOps emphasizes close collaboration between development and operations teams, breaking down traditional silos to ensure continuous delivery and integration.
- **Automation** The automation of processes, from code integration to deployment and infrastructure management, is central to DevOps. Tools like Jenkins, Docker, and Kubernetes are commonly used.
- **Continuous Integration/Continuous Deployment (CI/CD)** DevOps practices integrate continuous testing, integration, and deployment pipelines to ensure that code changes are tested and deployed rapidly and safely.
- **Monitoring and Feedback** Continuous monitoring of applications in production, along with rapid feedback loops, allows for quick detection and resolution of issues, leading to a more reliable system.



Agile development methodology



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025

Agile Principles

Agile methodology prioritizes individuals and interactions over processes and tools, focusing on delivering working software frequently and efficiently. Agile advocates for embracing change, even in later stages of development, to ensure the product remains relevant and useful to the customer. Close collaboration between the development team and stakeholders ensures that the product evolves according to user needs. The ultimate goal of Agile is to deliver valuable, functional software that meets the users' needs while fostering a collaborative and responsive development environment.



Agile Methodology Overview

- **Customer Collaboration:** Agile emphasizes working closely with the customer to ensure that the product meets their needs. Regular feedback from the customer is used to adjust the product throughout the development process.
- **Incremental Delivery:** Instead of delivering the product as a whole at the end of the project, Agile focuses on delivering small, workable pieces of the product incrementally, allowing for quick adjustments based on feedback.
- **Adaptive Planning:** Agile planning is flexible and allows for changes in project requirements, which are common in dynamic and fast-changing environments.
- **Cross-functional Teams:** Agile teams are typically composed of members with various skills, enabling them to handle different aspects of the project collaboratively and efficiently.
- **Continuous Improvement:** Agile encourages teams to reflect on their processes and make improvements continuously, often through retrospectives after each iteration or sprint.



Introduzione al corso

[DT-0540] Metodi di sviluppo agile

Daniele Di Pompeo

daniele.dipompeo@univaq.it



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025