# F11018: INGEGNERIA DEL SOFTWARE

## *Software Engineering: Process - Agile*

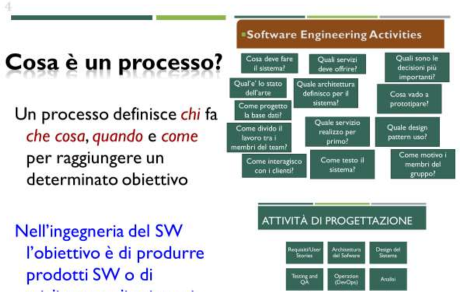
Lecturer: Prof. Henry Muccini
Università degli Studi dell'Aquila

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila

## Copyright Notice

- Un ringraziamento ai colleghi Daniele Di Pompeo e Patrizio Pelliccione per il materiale fornito per realizzare parte di queste slide.

Henry Muccini

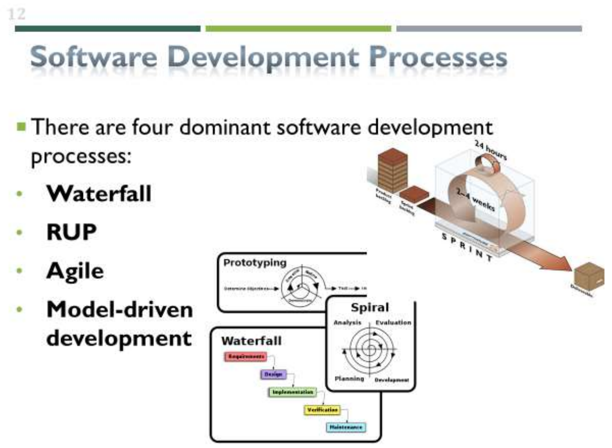


## Cosa è un processo?

Un processo definisce *chi* fa *che cosa*, *quando* e *come* per raggiungere un determinato obiettivo

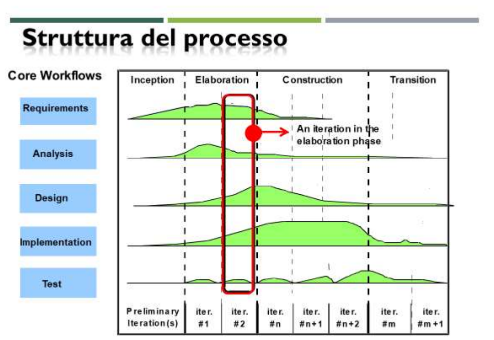
Nell'ingegneria del SW l'obiettivo è di produrre prodotti SW o di

Software Engineering Activities
Cosa deve fare il sistema?
Quali servizi deve offrire?
Quali sono le decisioni più importanti?
Qual'e' lo stato dell'arte
Quale architettura definisco per il sistema?
Cosa vado a prototipare?
Come progetto la base dati?
Come divido il lavoro tra i membri del team?
Quale servizio realizzo per primo?
Quale design pattern uso?
Come interagisco con i clienti?
Come testo il sistema?
Come motivo i membri del gruppo?
ATTIVITÀ DI PROGETTAZIONE
Requisiti/User Stories
Architettura del Software
Design del Sistema
Testing and QA
Operation (DevOps)
Analisi



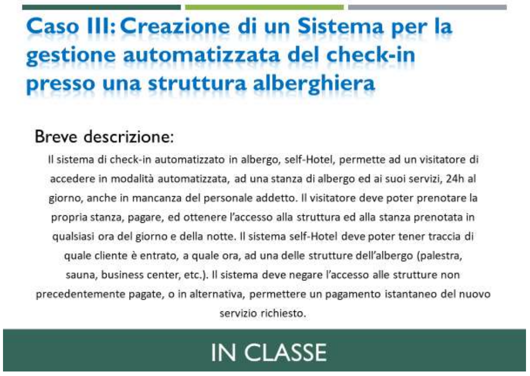
## Software Development Processes

- There are four dominant software development processes:
  - **Waterfall**
  - **RUP**
  - **Agile**
  - **Model-driven development**



## Struttura del processo

i Ingegneria e Scienze
ne e Matematica
di dell'Aquila



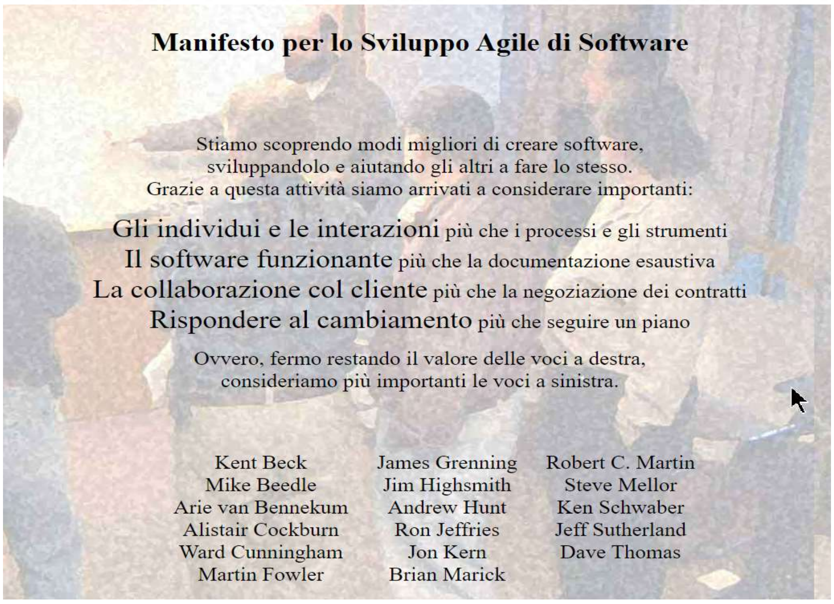
## Caso III: Creazione di un Sistema per la gestione automatizzata del check-in presso una struttura alberghiera

Breve descrizione:

Il sistema di check-in automatizzato in albergo, self-Hotel, permette ad un visitatore di accedere in modalità automatizzata, ad una stanza di albergo ed ai suoi servizi, 24h al giorno, anche in mancanza del personale addetto. Il visitatore deve poter prenotare la propria stanza, pagare, ed ottenere l'accesso alla struttura ed alla stanza prenotata in qualsiasi ora del giorno e della notte. Il sistema self-Hotel deve poter tener traccia di quale cliente è entrato, a quale ora, ad una delle strutture dell'albergo (palestra, sauna, business center, etc.). Il sistema deve negare l'accesso alle strutture non precedentemente pagate, o in alternativa, permettere un pagamento istantaneo del nuovo servizio richiesto.

IN CLASSE

# https://agilemanifesto.org/iso/it/manifesto.html



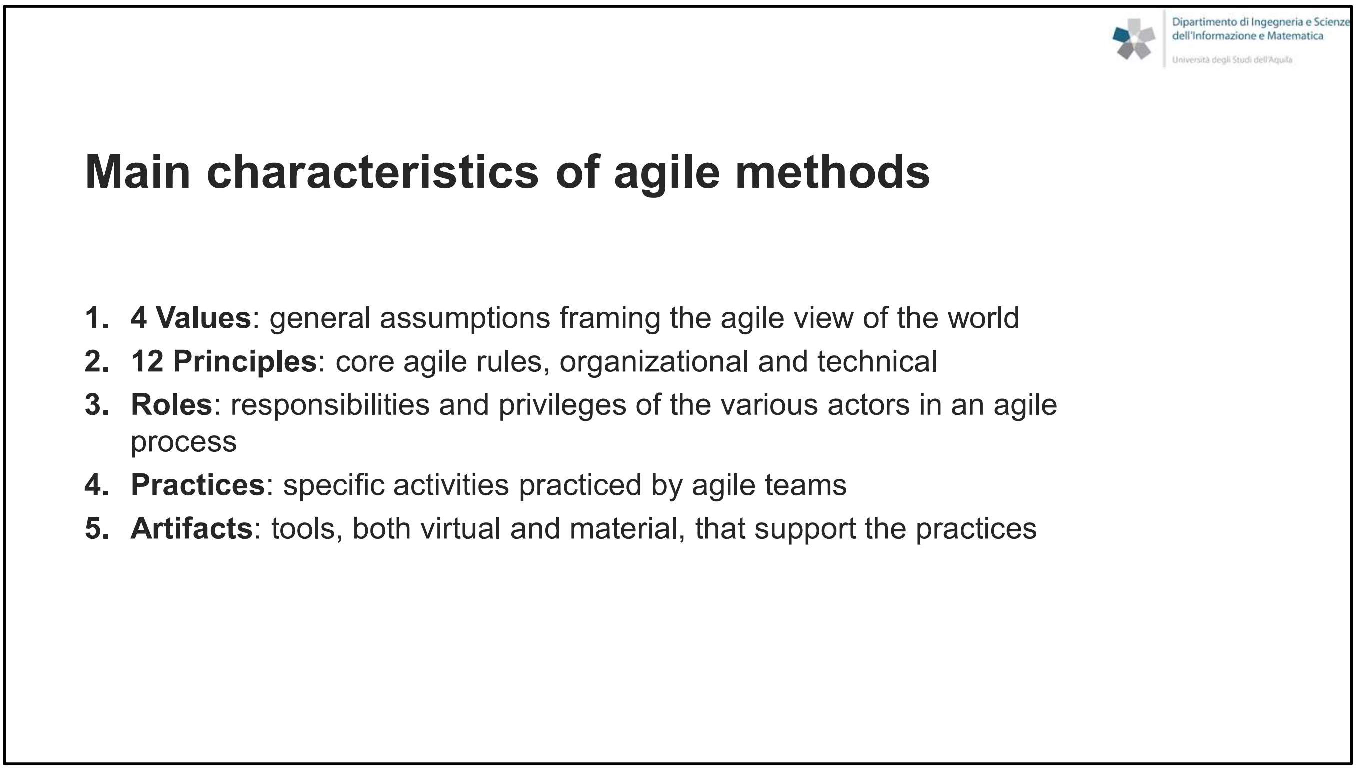
## Manifesto per lo Sviluppo Agile di Software

Stiamo scoprendo modi migliori di creare software,
sviluppandolo e aiutando gli altri a fare lo stesso.
Grazie a questa attività siamo arrivati a considerare importanti:

Gli individui e le interazioni più che i processi e gli strumenti
Il software funzionante più che la documentazione esaustiva
La collaborazione col cliente più che la negoziazione dei contratti
Rispondere al cambiamento più che seguire un piano

Ovvero, fermo restando il valore delle voci a destra,
consideriamo più importanti le voci a sinistra.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

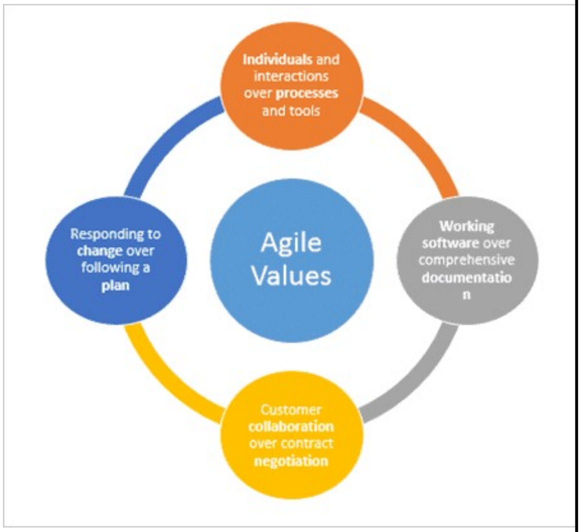


## Main characteristics of agile methods

1. **4 Values**: general assumptions framing the agile view of the world
2. **12 Principles**: core agile rules, organizational and technical
3. **Roles**: responsibilities and privileges of the various actors in an agile process
4. **Practices**: specific activities practiced by agile teams
5. **Artifacts**: tools, both virtual and material, that support the practices

# 1. Agile values

- **Redefined roles** for developers, managers and customers.
- **No “big Upfront”** steps.
- **Iterative** development.
- Limited, negotiated **functionality**.
- Focus on **quality**, understood as achieved through testing.


Individuals and interactions over processes and tools
Working software over comprehensive documentatio n
Customer collaboration over contract negotiation
Responding to change over following a plan
Agile Values

# 2. 12 Agile Principles

1. highest priority: **satisfy the customer**
2. even **late change of requirements** is welcomed
3. Frequent **delivery of** working **software**
4. daily **work together**
5. motivated individuals is given environment and support they need, and trust them to get the job done
6. conveying information: **face-to-face conversation**
7. primary measure of progress: working software
8. agile processes promote **sustainable development**, stakeholders should be able to maintain a constant pace **indefinitely**
9. continuous attention to technical excellence and good design enhances agility.
10. simplicity - the art of maximising the amount of work not done - is essential
11. **self-organising teams ➔** best architectures, requirements, and designs
12. team regularly reflects on how to become more effective

# 2. Agile principles - organizational

- Put the **customer** at the center.
- Let the team **self-organize**.
- Work at a **sustainable** pace.
- Develop **minimal software**
  - Produce minimal functionality.
  - Produce only the product requested.
  - Develop only code and tests.
- Accept **change**.

Values Principles Roles Practices Artifacts

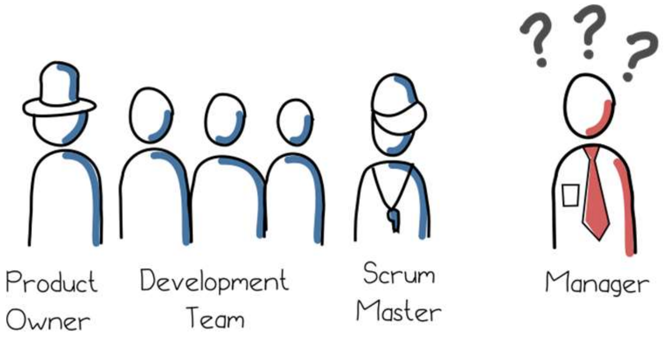
# 2. Agile principles - technical

- Develop **Iteratively**
  - Produce frequent working iterations.
  - Freeze requirements during iterations.
- Treat **tests** as a key resource.
  - Do not start any new development until all tests pass.
  - Test first.
- Express requirements through **scenarios**.

Values Principles Roles Practices Artifacts

# 3. Agile roles

- Team
- Product owner
- Scrum master
- Customer



https://www.agile-school.com/blog/quali-sono-i-tre-ruoli-definiti-allinterno-dello-scrum-team
(= https://bit.ly/3F9QMtX)
https://www.agileway.it/il-team-scrum-e-i-ruoli/
https://www.biancolavoro.it/la-metodologia-agile-in-breve-ruoli-artefatti-ed-eventi/

Values Principles Roles Practices Artifacts

# 4. Key agile practices - organizational

1. Daily meeting.
2. Planning game (XP), planning poker (scrum).
3. Continuous integration.
4. Retrospective.
5. Shared code ownership.

# 4. Key agile practices - organizational

1. Daily meeting.
2. Planning game (XP), planning poker (scrum).
3. Continuous integration.
4. Retrospective.
5. Shared code ownership.

What did I do in the previous working day?

What do I plan to do today?

What impediments am I facing?

Values Principles Roles Practices Artifacts

# 4. Key agile practices - organizational

1. Daily meeting.
2. Planning game (XP), planning poker (scrum).
3. Continuous integration.
4. Retrospective.
5. Shared code ownership.

Estimation techniques: give space to individuals and try to reach consensus

## 4. Key agile practices - organizational

1. Daily meeting.
2. Planning game (XP), planning poker (scrum).
3. Continuous integration.
4. Retrospective.
5. Shared code ownership.

Values Principles Roles Practices Artifacts

# 4. Key agile practices - organizational

1. Daily meeting.
2. Planning game (XP), planning poker (scrum).
3. Continuous integr
4. Retrospective.
5. Shared code ownership.

Reflect on the experience and the lessons learned with the goal of improving the development process

Values Principles Roles Practices Artifacts

# 4. Key agile practices - organizational

1. Daily meeting.
2. Planning game (XP), planning poker (scrum).
3. Continuous integration.
4. Retrospective.
5. Shared code ownership.

Values Principles Roles Practices Artifacts

# 4. Key agile practices - technical

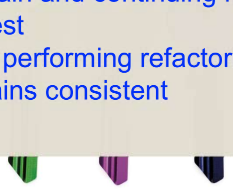
1. Test-driven development.
2. Refactoring.
3. Pair programming.
4. Simplest solution that can possibly work.
5. Coding standards.

Values Principles Roles Practices Artifacts

# 4. Key agile practices technical

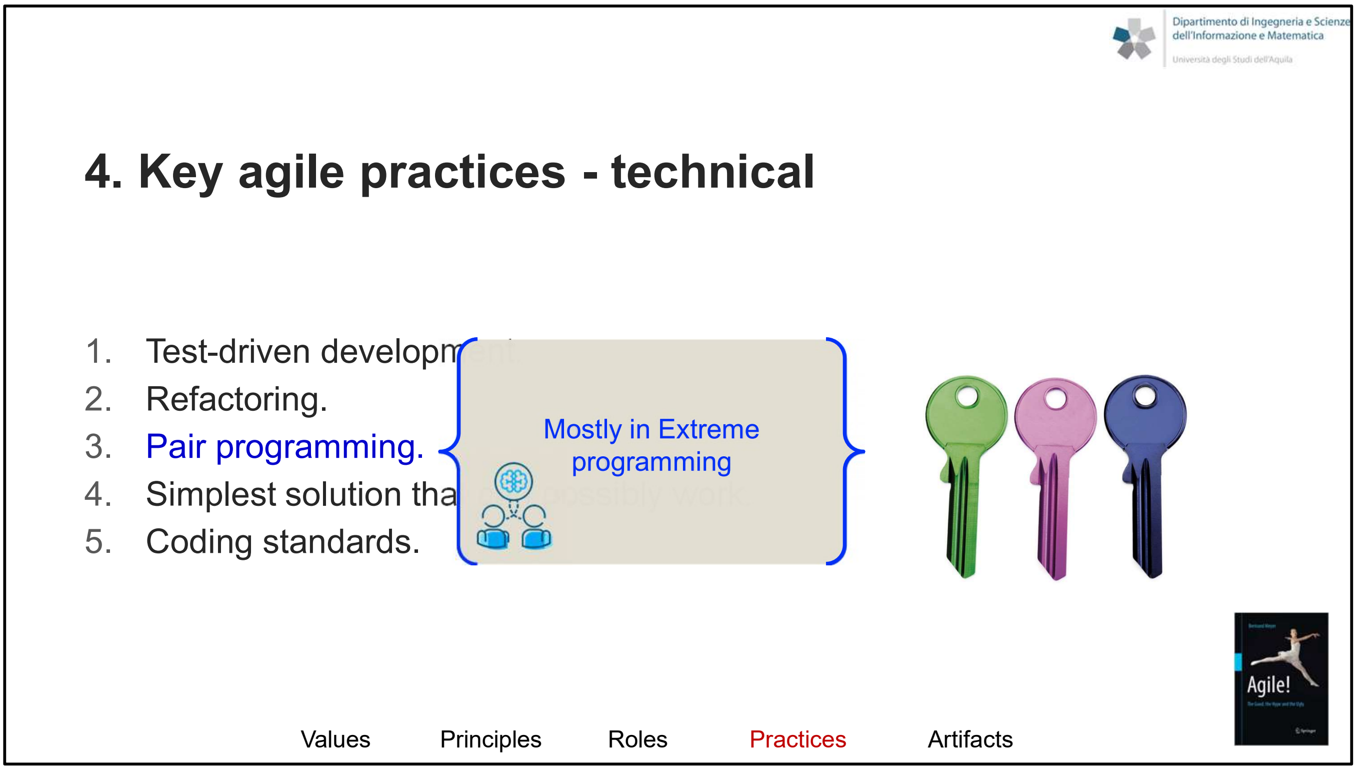
1. Test-driven development.
2. Refactoring.
3. Pair programming.
4. Simplest solution that can possibly
5. Coding standards.

1. Writing a test corresponding to a new functionality
2. Running the program, which should not pass the test since the functionality is new
3. Fixing the program
4. Running the program again and continuing fixing it until it does pass the test
5. Examining the code and performing refactoring to make the design remains consistent



# 4. Key agile practices - technical

1. Test-driven development.
2. Refactoring.
3. Pair programming.
4. Simplest solution that can possibly work.
5. Coding standards.



Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila

# 4. Key agile practices - technical

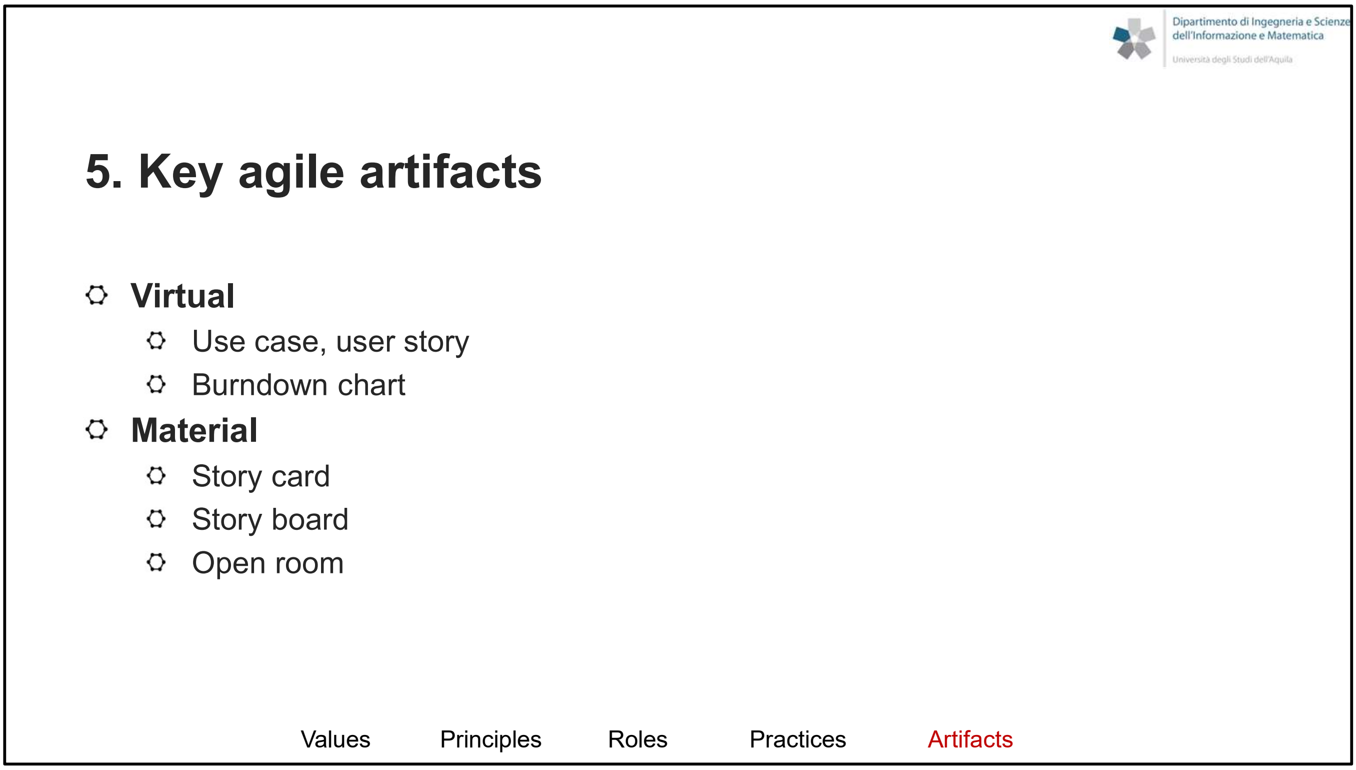
1. Test-driven development.
2. Refactoring.
3. Pair programming.
4. Simplest solution tha[illegible] possibly work.
5. Coding standards.

Mostly in Extreme programming

Values Principles Roles Practices Artifacts

# 4. Key agile practices - technical

1. Test-driven development.
2. Refactoring.
3. Pair programming.
4. Simplest solution that can possibly work.
5. Coding standards.

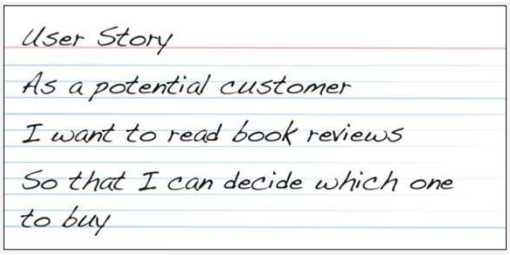


# 5. Key agile artifacts

- **Virtual**
  - Use case, user story
  - Burndown chart
- **Material**
  - Story card
  - Story board
  - Open room

## 5. Key agile artifacts

- **Virtual**
  - Use case, user story
  - Burndown chart
- **Material**
  - Story card
  - Story board
  - Open room



User Story

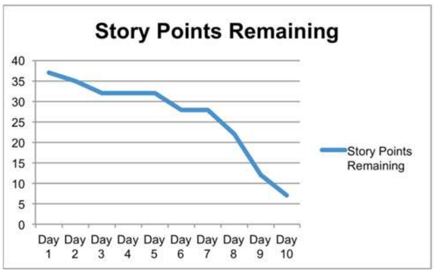
As a potential customer

I want to read book reviews

So that I can decide which one to buy

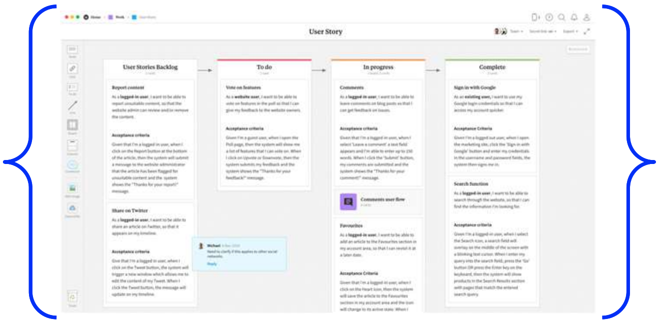
# 5. Key agile artifacts

- **Virtual**
  - Use case, user story
  - Burndown chart
- **Material**
  - Story card
  - Story board
  - Open room



# 5. Key agile artifacts

- **Virtual**
  - Use case, user story
  - Burndown chart
- **Material**
  - Story card
  - Story board
  - Open room



User Story

User Stories Backlog

**Report content**

As a **logged-in user**, I want to be able to report unsuitable content, so that the website admin can review and/or remove the content.

**Acceptance criteria**

Given that I'm a logged in user, when I click on the Report button at the bottom of the article, then the system will submit a message to the website administrator that the article has been flagged for unsuitable content and the system shows the "Thanks for your report!" message.

**Share on Twitter**

As a **logged-in user**, I want to be able to share an article on Twitter, so that it appears on my timeline.

**Acceptance criteria**

Give that I'm a logged in user, when I click on the Tweet button, the system will trigger a new window which allows me to edit the content of my Tweet. When I click the Tweet button, the message will update on my timeline.

To do

**Vote on features**

As a **website user**, I want to be able to vote on features in the poll so that I can give my feedback to the website owners.

**Acceptance criteria**

Given I'm a guest user, when I open the Poll page, then the system will show me a list of features that I can vote on. When I click on Upvote or Downvote, then the system submits my feedback and the system shows the "Thanks for your feedback!" message.

Michael

Need to clarify if this applies to other social networks.

Reply

In progress

**Comments**

As a **logged-in user**, I want to be able to leave comments on blog posts so that I can get feedback on issues.

**Acceptance criteria**

Given that I'm a logged in user, when I select 'Leave a comment' a text field appears and I'm able to enter up to 150 words. When I click the 'Submit' button, my comments are submitted and the system shows the "Thanks for your comment!" message.

**Comments user flow**

**Favourites**

As a **logged-in user**, I want to be able to add an article to the Favourites section in my account area, so that I can revisit it at a later date.

**Acceptance Criteria**

Given that I'm a logged in user, when I click on the Heart icon, then the system will save the article to the Favourites section in my account area and the icon will change to its active state. When I

Complete

**Sign in with Google**

As an **existing user,** I want to use my Google login credentials so that I can access my account quicker.

**Acceptance Criteria**

Given I'm a logged out user, when I open the marketing site, click the 'Sign in with Google' button and enter my credentials in the username and password fields, the system then signs me in.

**Search function**

As a **logged-in user**, I want to be able to search through the website, so that I can find the information I'm looking for.

**Acceptance criteria**

Given I'm a logged-in user, when I select the Search icon, a search field will overlay on the middle of the screen with a blinking text cursor. When I enter my query into the search field, press the 'Go' button OR press the Enter key on the keyboard, then the system will show products in the Search Results section with pages that match the entered search query.

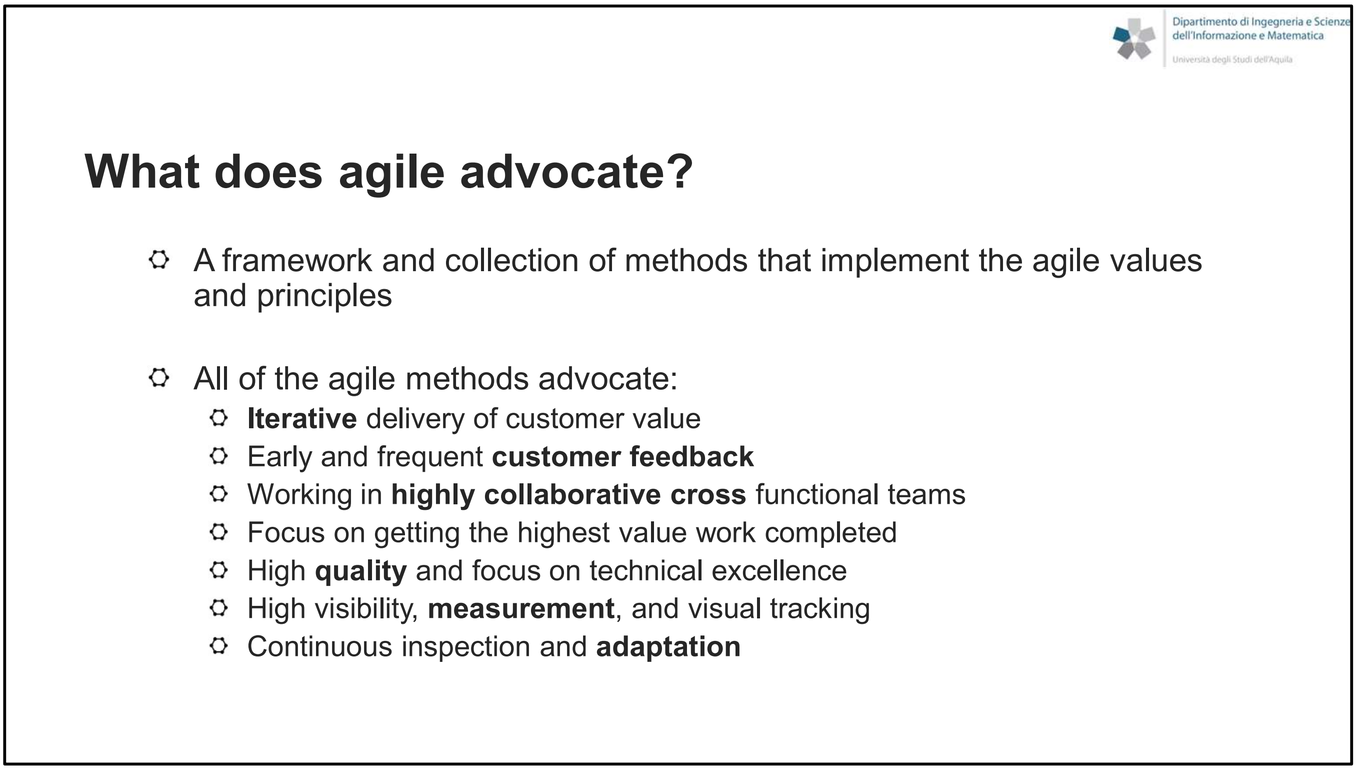
# Agile is not a Silver bullet

*“There is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity.”*

Frederick P. Brooks (Turing award) No Silver Bullet Essence and Accidents of Software Engineering. Computer 20, 4 (April 1987), 10-19. DOI: https://doi.org/10.1109/MC.1987.1663532

# What agile is not

- A silver bullet
- Easy to implement
- An excuse to discard documentation, design or good development practices
  - “We’re Agile now we don’t need to dot that!”
- About avoiding committing to deliver software
  - “We’re Agile now we don’t give dates!”



## What does agile advocate?

- A framework and collection of methods that implement the agile values and principles

- All of the agile methods advocate:
  - **Iterative** delivery of customer value
  - Early and frequent **customer feedback**
  - Working in **highly collaborative cross** functional teams
  - Focus on getting the highest value work completed
  - High **quality** and focus on technical excellence
  - High visibility, **measurement**, and visual tracking
  - Continuous inspection and **adaptation**

# TODO IN CLASS

Max 10 PUNTI IMPORTANTI PRESENTATI IN CLASSE

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