Agile Community Torino

AGILE

IL MANIFESTO DEI VALORI E DEI PRINCIPI

Conosciamoci

Alma Maria del Campo Vara
almadcv@yahoo.es
https://www.linkedin.com/in/alma-maria-del-campo-vara-1497781a/

Digital Platform - Reale ITES

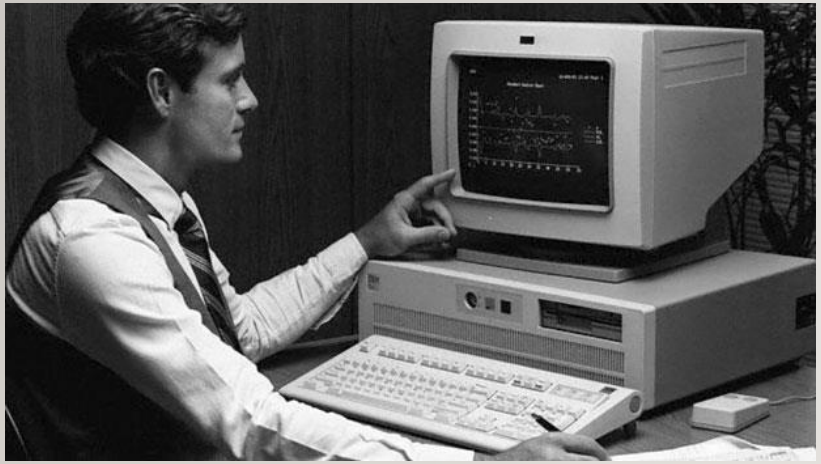
Chi di voi copre un ruolo IT?
Per chi di voi oggi è la prima esperienza con Agile?
Chi di voi necessita Agile per il lavoro?
Scrivi in un foglio di carta e tieni per te:
Una parola per descrivere Agile.
Un obbiettivo da portarsi a casa.

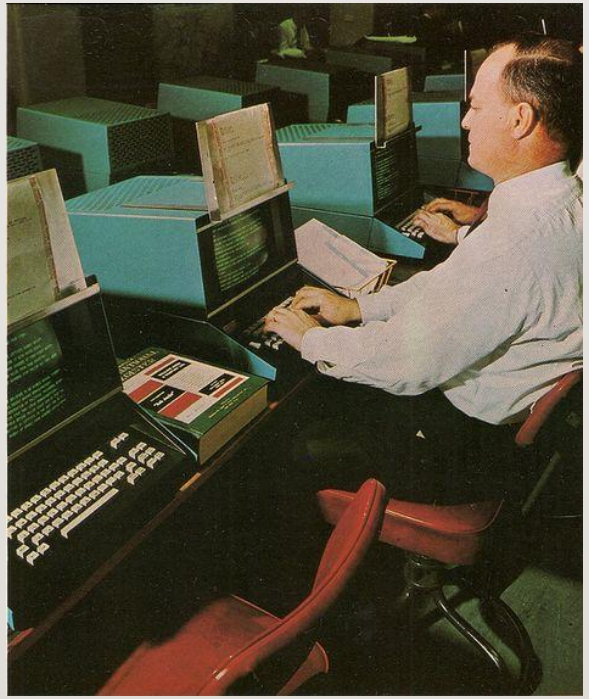
Cos’è Agile?

Attualmente esistono molte definizioni discordanti di "Cosa è Agile?"
La maggior parte si bassano sugli stessi concetti di base ma declinati in maniera diversa generando molteplici "scuole di pensiero".

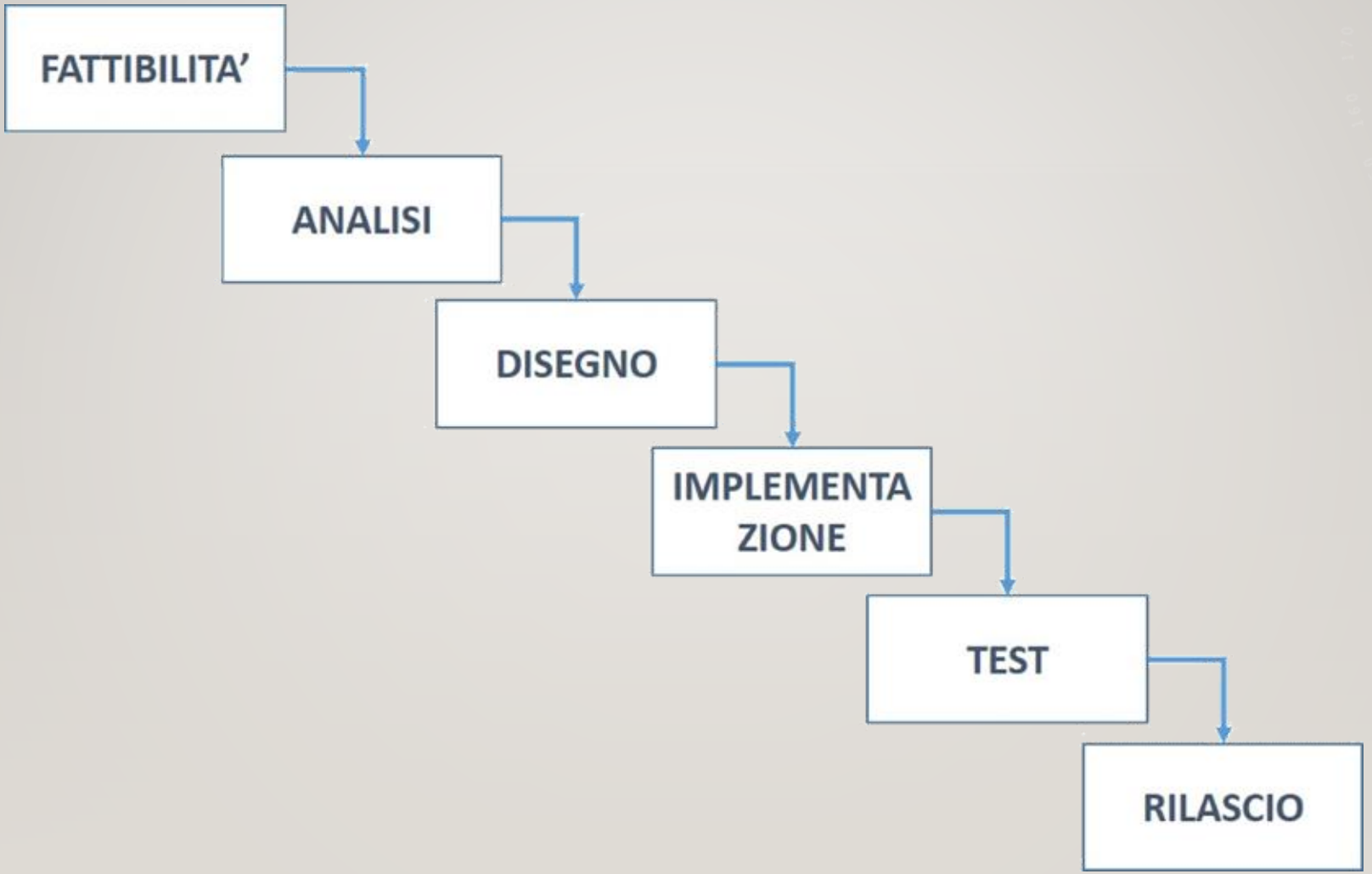
Per capire cosa è Agile veramente, bisogna capire quali sono state le necessità che hanno portato alla sua apparizione, e per questo dovremmo trasportarci nel tempo agli anni 70…

Negli anni 70…





Il modello classico di sviluppo software



Le prime idee "agili" - 70s & 80s

- 1968: “Conway’s Law” - **Enterprise Organization & Comunication can influence IT performance -** "Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure«
- 1970. Winston W. Royce "Managing the Development of Large Software Systems",, prima cirtica al modello cascata
- 1970s: Barry Boehm – **Estimation depends on people** - Proposes “Wideband Delphi”, a forerunner of Planning Poker
- 1976: D. Panzl – **Automation Unit Test** - describing tools with features closely resembling those of JUnit
- 1976: Glenford Myers - **“A developer should never test their own code” -** Software Reliability book
- 1984: Barry Boehm – **Prototyping** - develops an early empirical study of projects using prototyping,
- 1984: Leo Brodie - **(re)factoring, modularity, bottom-up and incremental design** – book “Thinking Forth”,
- 1985:Tom Gilb – **Incremental software release** - Evolutionary Delivery Model, nicknamed “Evo”.
- 1986: Barry Boehm – **Incremental & Validation -**”A Spiral model of software development and enhancement”, an iterative model geared to identifying and reducing risks

Le prime idee "agili" - late 80s & early 90s

- 1986: Takeuchi and Nonaka – self-organizing teams, flexibility, multilearning - The New New Product Development Game". The inspiration for the Scrum framework
- 1988: Scott Schultz – Timebox - Rapid Iterative Production Prototyping
- 1988: Dijsktra - Object Oriented Programming
- 1990: Bill Opdyke - Coins the term "refactoring"
- 1991: James Martin - RAD, possibly the first approach in which timeboxing and "iterations" are used
- 1991: Taligent - testing framework SUnit (source)
- 1992: Larry Constantine - "Dynamic Duo" o Pair Programing
- 1993:Jim Cop lien - StandUpMeeting pattern

Le prime idee "agili" - late 90s

- 1995: The earliest writings on Scrum introduce the notion of the “sprint” as iteration
- 1995: Andrew Koenig - coins the term antipattern
- 1995: Ken Schwaber and Jeff Sutherland - co-present Scrum at the OOPSLA Conference.
- 1996: Automated tests are a practice of Extreme Programming, without much emphasis on the distinction between unit and acceptance testing
- 1997: Ken Schwaber - describes the “daily scrum”
- 1997: Beck and Gamma - The testing tool JUnit is written
- 1998: Continuous integration and the “daily stand-up” are listed among the core practices of Extreme Programming.
- 1998: The earliest article about Extreme Programming, “Chrysler goes to Extremes”, describes several XP practices
- 1999: Alan Cooper - Describes "Personas" in one chapter of “The Inmates are Running the Asylum”

Le prime idee "agili" - 00s

- 1999: Ron Jeffries - Mentions the unit "Gummi Bears", an alternative to "story points" for estimating user stories,
- 2000,ca: The "three questions" of Scrum's daily meeting format are largely adopted by Extreme Programming teams.
- 2000: Freeman - Describes the "mock objects" testing technique
- 2000: Ken Schwaber – Describes the burndown chart
- 2000: The term "velocity" is a relatively late addition to Extreme Programming, replacing a previous notion of "load factor" deemed overly complex.

February 11-13 2001

17 people who develop software and help others do it met at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah to find common ground among their different approaches to software development. The outcome of this meeting is the Manifesto for Agile Software Development. Several members of that discussion went on to found the Agile Alliance.

https://www.agilealliance.org/agile101/practices-timeline/

Cos’è Agile?

- *Mindset applicabile nel portare avanti progetti/prodotti/servizi in contesti complessi che enfatizza l’approccio:*

  - *Value driven*
  - *Gestione del cambiamento*
  - *Cultura del miglioramento continuo*
  - *Cultura della collaborazione*

Cynefin framework

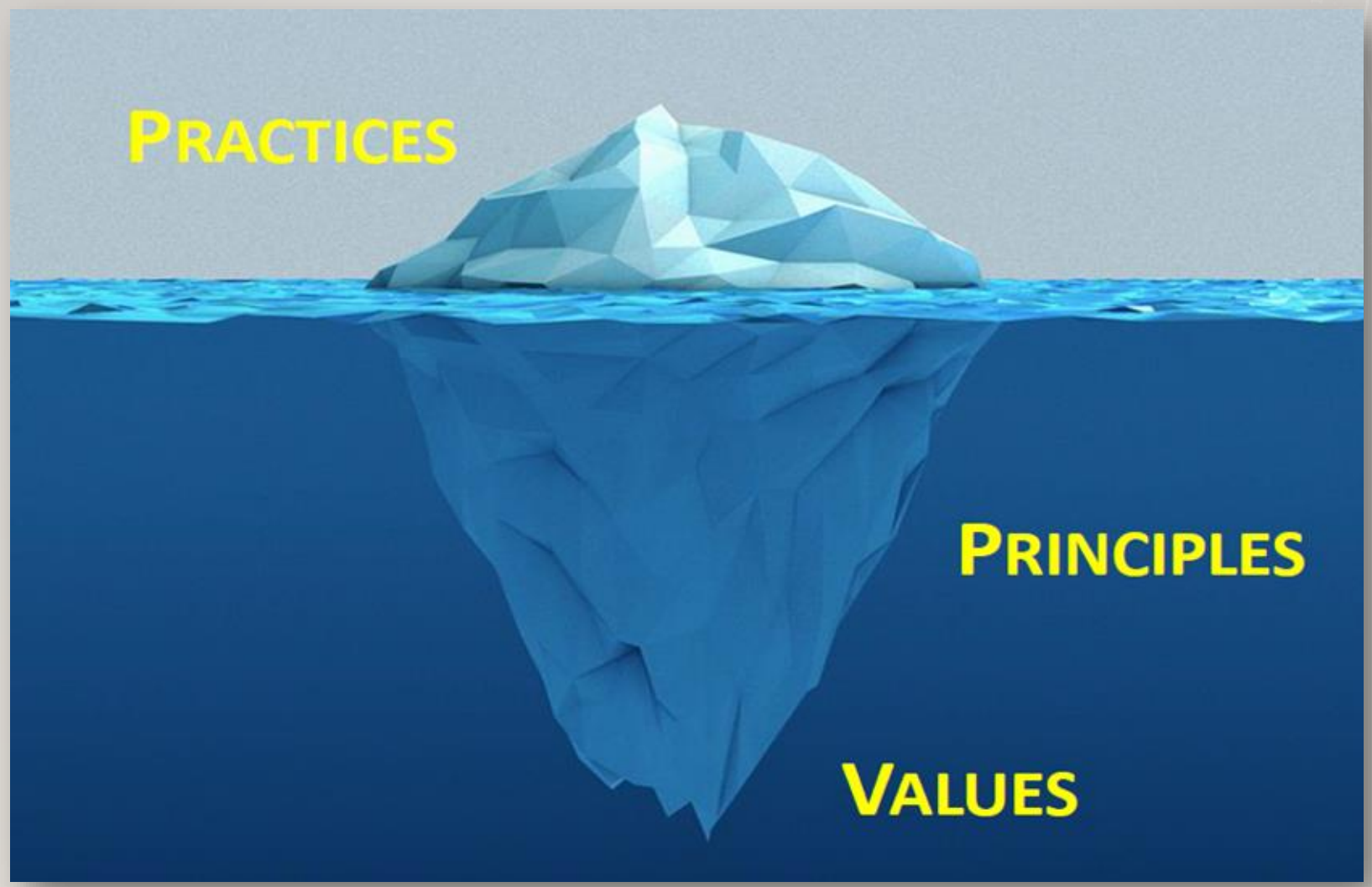
Quando e dove si applica?

<u>Esempi di Ambito di utilizzo di Agile:</u>

Progetti Variabili:
Information Technology
Marketing & Comunication - Agile Marketing
Managing Events
Education
Innovation & Digital

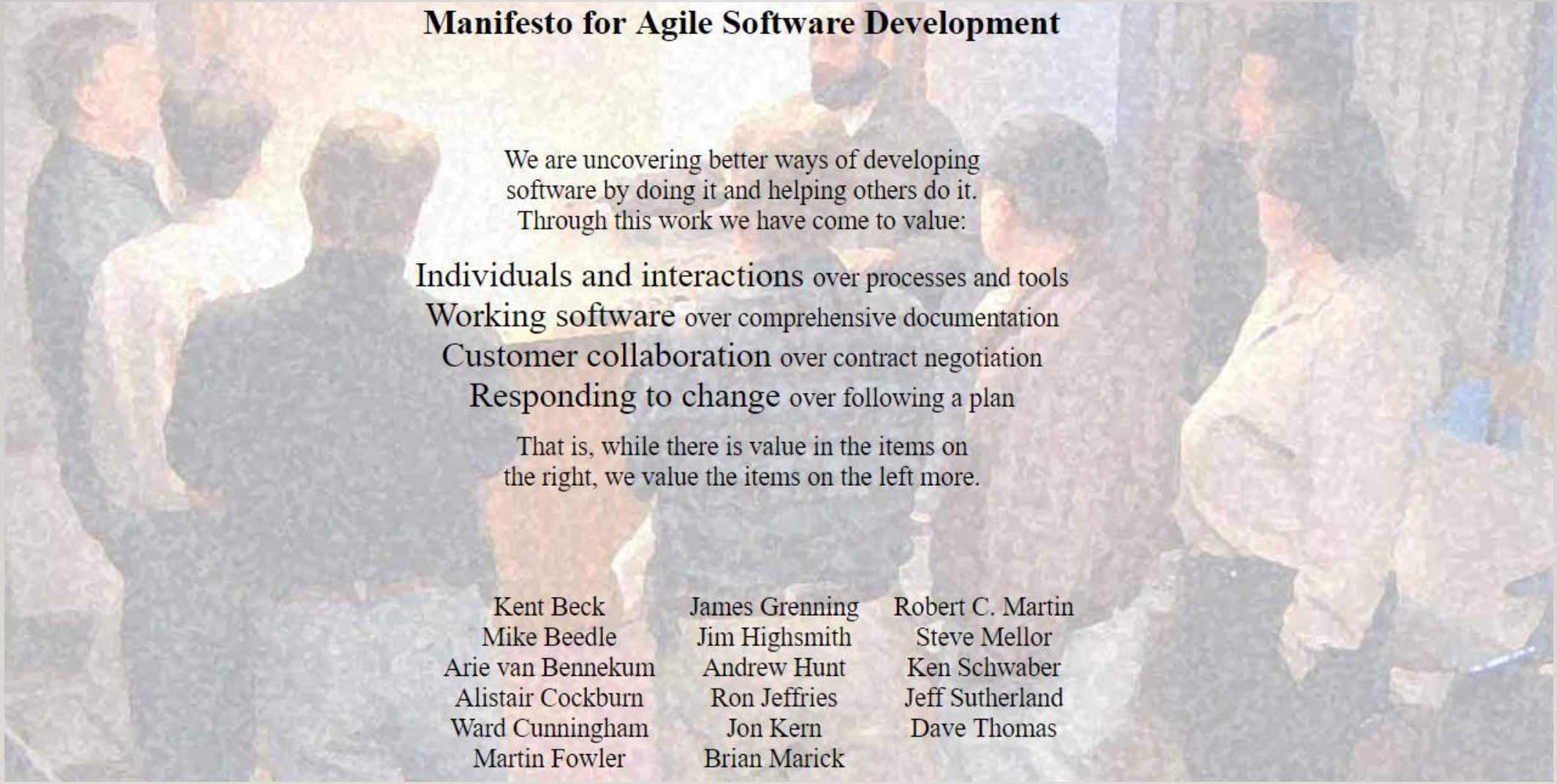
Progetti Complessi:
Agile Construction Management
Agile Manufacturing

Pratiche, valori e principi



Manifesto agile

Il Manifesto: gli autori



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

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

http://agilemanifesto.org

Il Manifesto: gli autori

Dave Thomas

- Adjunct Professor at Southern Methodist University
- https://pragdave.me/
- Keyword : Pragmatic Programmer, Code Kata, Ruby

Il Manifesto: i Valori

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

- **A individui e interazioni** più che a processi e strumenti
- **A software funzionante** più che a documentazione esaustiva
- **Alla collaborazione col cliente** più che alla negoziazione dei contratti
- **A rispondere al cambiamento** più che a seguire un piano

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

https://www.slideshare.net/mobile/JaeHeeLee2/agile-manifesto-17-authors-who-made-it-what-do-they-do

http://agilemanifesto.org

Il Manifesto: i Principi

1. La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua.
2. Accogliamo i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.
3. Consegnammo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.
4. Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.
5. Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.
6. Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team ed all'interno del team.
7. Il software funzionante è il principale metro di misura di progresso.
8. I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante.
9. La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità.
10. La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.
11. Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano.
12. A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

I Principi - Valore

1.- La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua.

2.- Accogliamo i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.

3.- Consegnammo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.

Individui e interazioni *più che a processi e strumenti*
Software funzionante *più che a documentazione esaustiva*
Collaborazione col cliente *più che alla negoziazione dei contratti*
Rispondere al cambiamento *più che a seguire un piano*

http://agilemanifesto.org

I Principi - Comunicazione

4.- Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.

5.- Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.

6.- Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team ed all'interno del team.

Individui e interazioni *più che a processi e strumenti*
Software funzionante *più che a documentazione esaustiva*
Collaborazione col cliente *più che alla negoziazione dei contratti*
Rispondere al cambiamento *più che a seguire un piano*

http://agilemanifesto.org

I Principi - Esecuzione

Individui e interazioni *più che a processi e strumenti*
Software funzionante *più che a documentazione esaustiva*
Collaborazione col cliente *più che alla negoziazione dei contratti*
Rispondere al cambiamento *più che a seguire un piano*

7.- Il software funzionante è il principale metro di misura di progresso.

8.- I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante.

9.- La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità.

http://agilemanifesto.org

I Principi - Miglioramento

10.- La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.

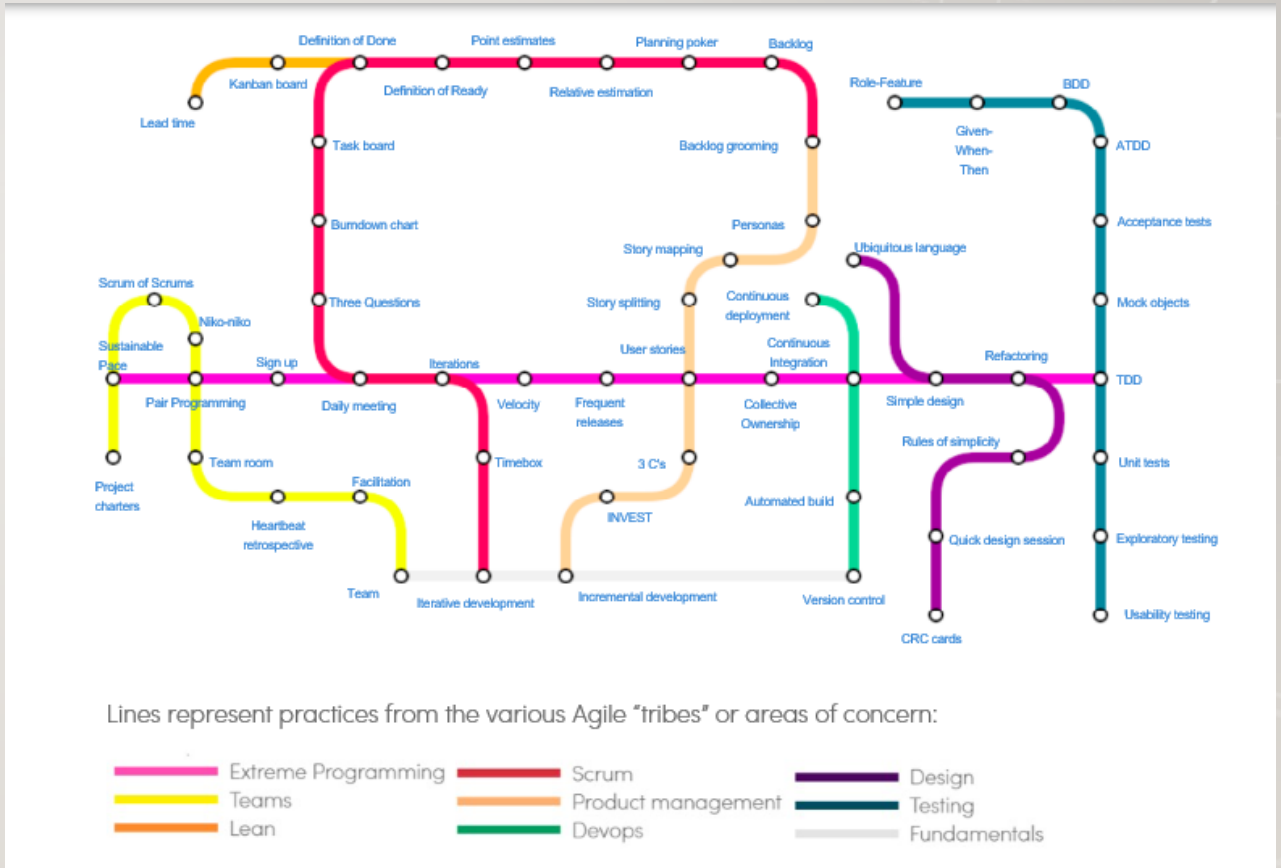
11.- Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano.

12.- A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

http://agilemanifesto.org

Le pratiche

- eXtreme programming (XP)
- Scrum
- Kanban
- DevOps
- Framework di scaling



Agile Subway Map from Agile Alliance

Fundamentals:

- Teams

Gli individui e le interazioni più che i processi e gli strumenti

- Iterative development

Il software funzionante più che la documentazione esaustiva

- Incremental development. Incremental Value.

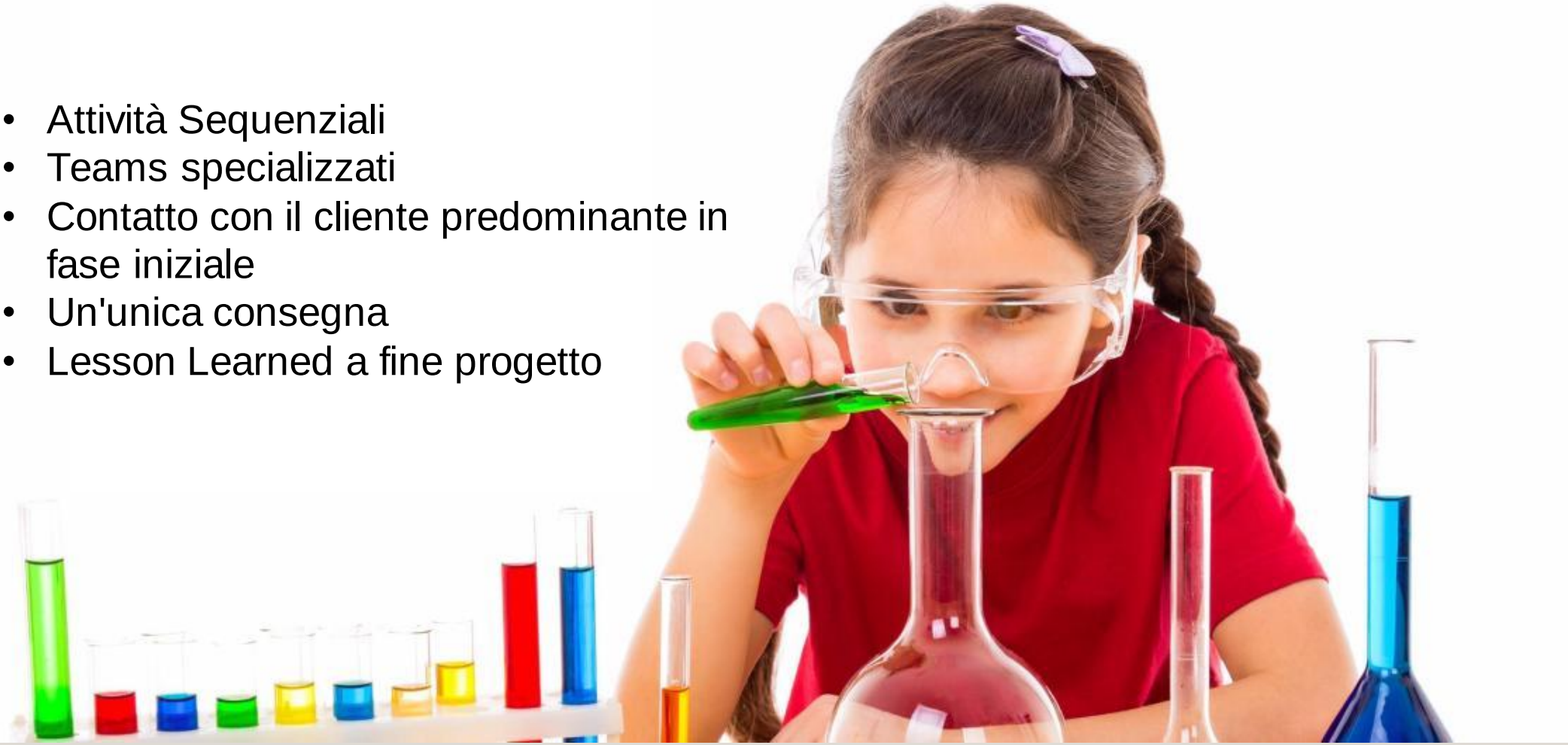
La collaborazione col cliente più che la negoziazione dei contratti

- Version Control

Rispondere al cambiamento più che seguire un piano


Giochiamo?

- Attività Sequenziali
- Teams specializzati
- Contatto con il cliente predominante in fase iniziale
- Un'unica consegna
- Lesson Learned a fine progetto



Il Manifesto: i Valori – Cosa posso fare per...

Individui e interazioni *vs processi e strumenti*

Collaborazione col utente *vs negoziazione dei contratti*

Software funzionante vs documentazione esaustiva

Rispondere al cambiamento più che seguire un piano

http://agilemanifesto.org

- Attività Incrementale con rilascio di valore
- Teams autogestiti e multifunzionali
- Contatto con il cliente accessibile durante tutta la esecuzione del progetto
- Consegna funzionante ad ogni iterazione
- Lesson Learned (Retrospettiva) ad ogni iterazione

Cosa abbiamo imparato?

All'inizio del meetup avete scritto due biglietti. Prendete un momento per rivedere quello che avete scritto e capire:

- C'è stata differenza tra la conoscenza che pensavate di ottenere da questo meetup e quella che avete acquisito finora?
- Gli obbiettivi del corso, sono stati raggiunti?

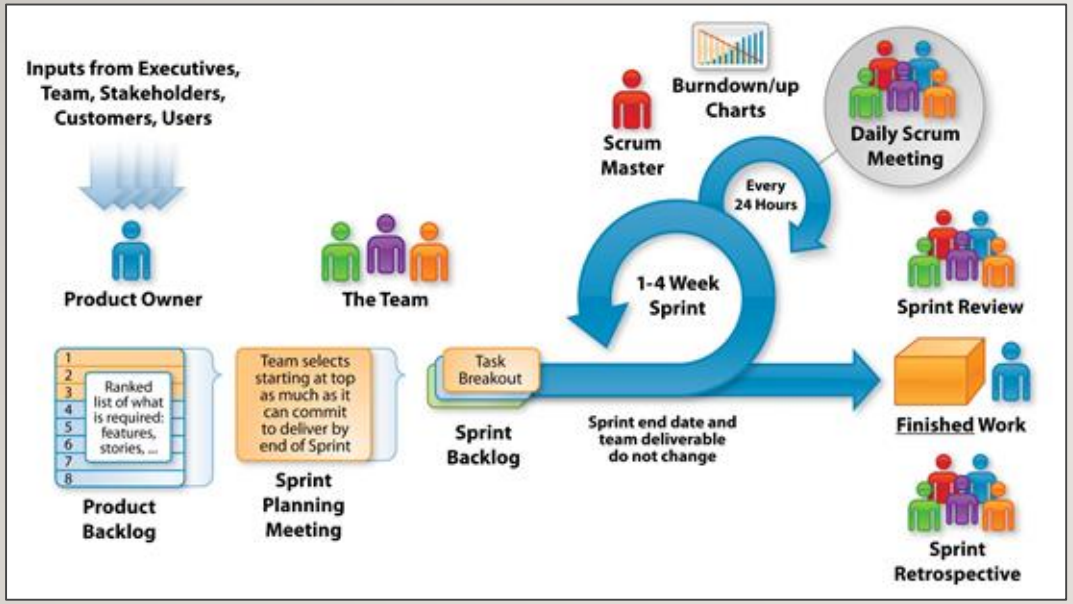
Fammi un regalo!

Lascia il tuo feedback, così potrò migliorare!

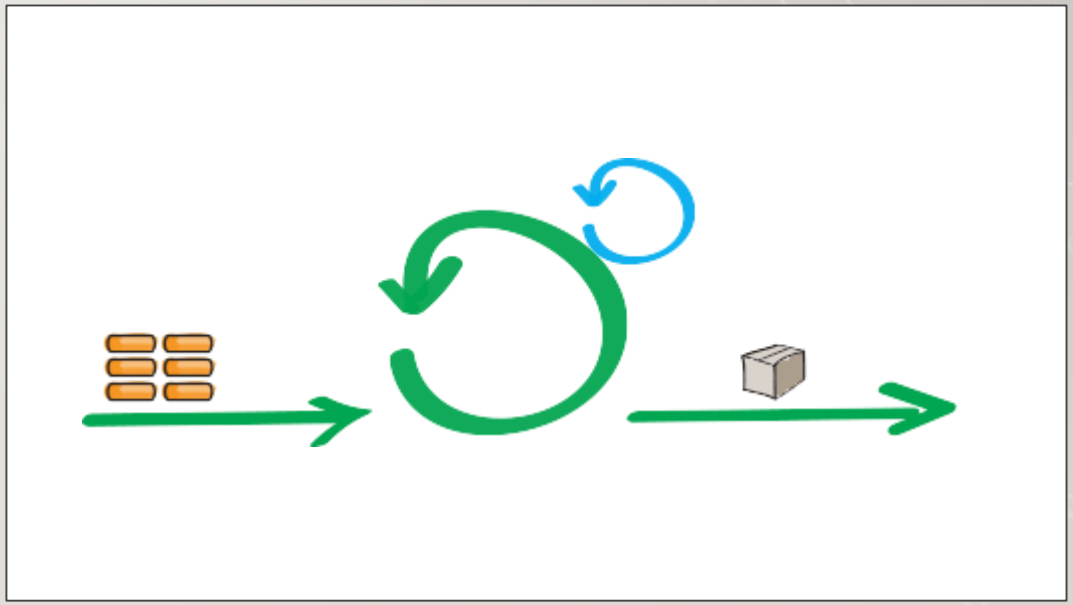
Grazie!

Adozione Agile

Come implementare Agile all'interno delle organizzazioni?



Big bang



Incrementale

Uno sguardo al futuro

Complessità

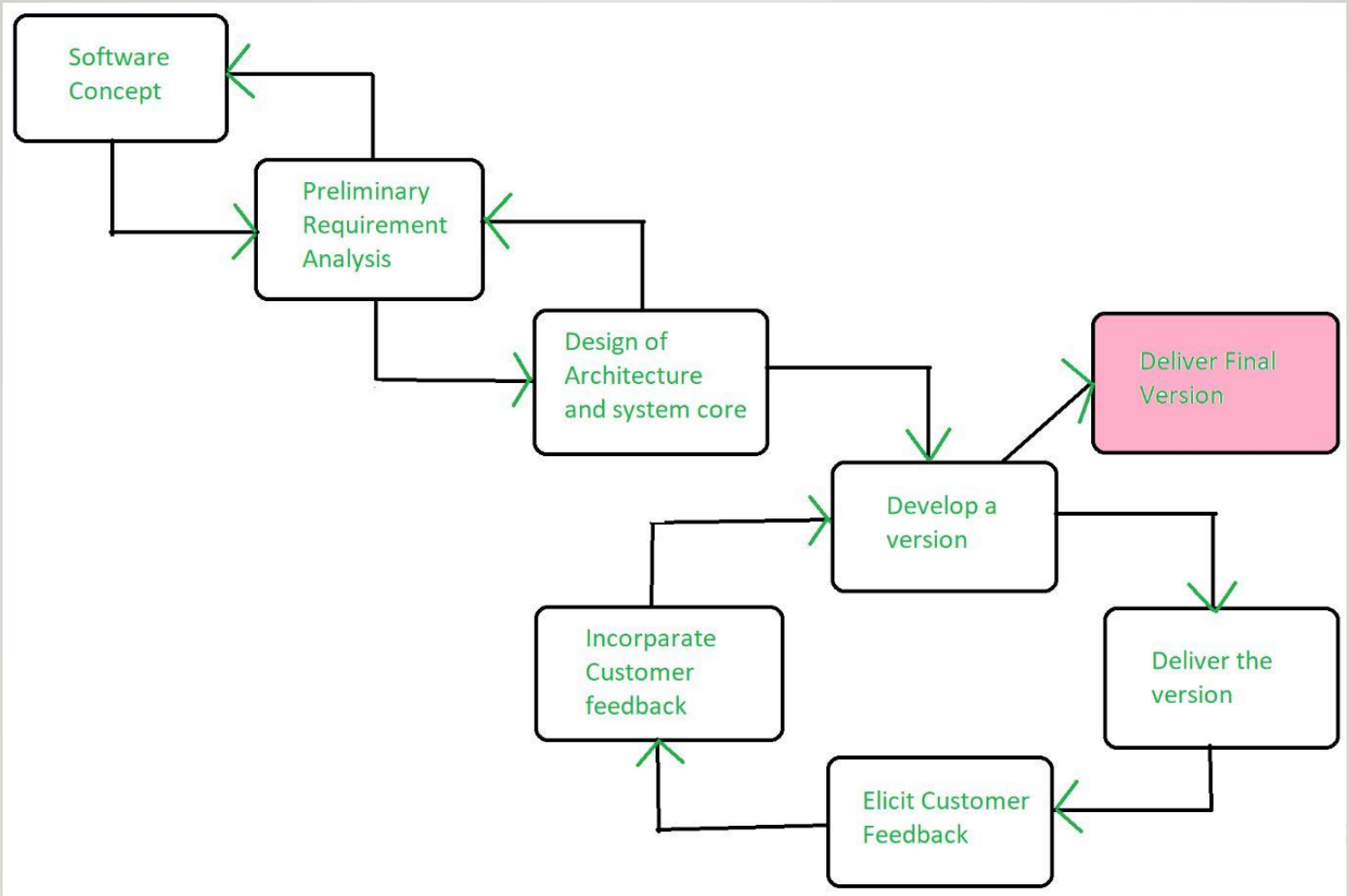
in continua crescita...

ANEXO -Wideband Delphi Process

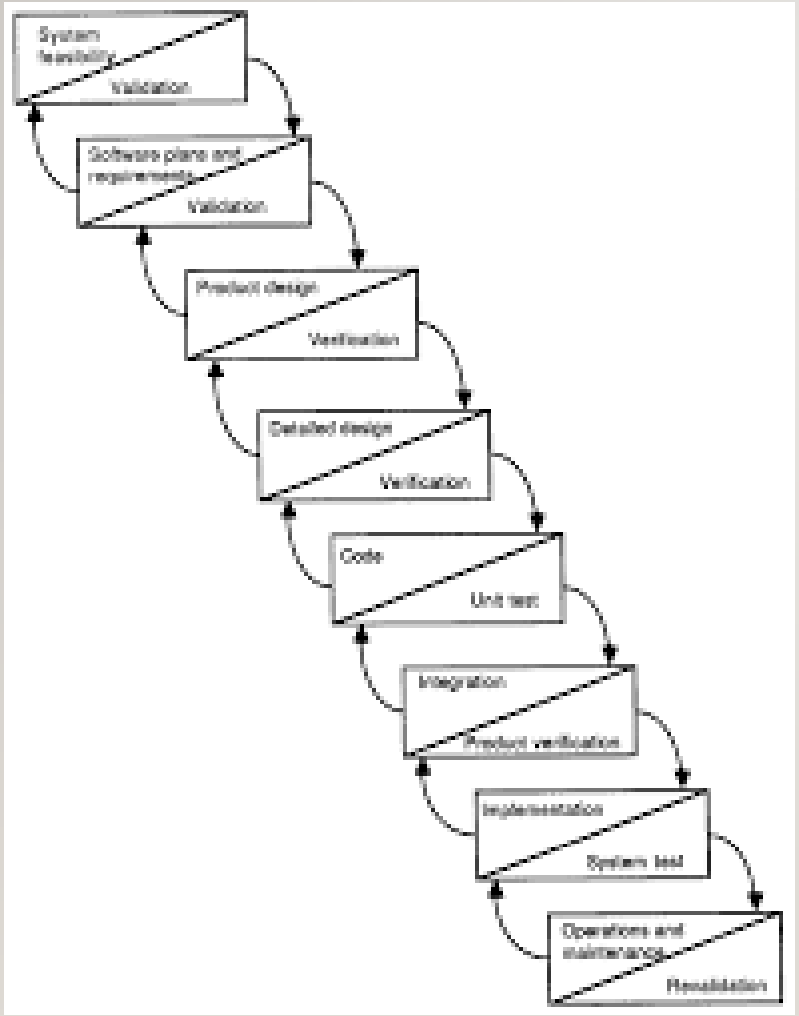
Barry Boehm and John A. Farquhar originated the Wideband variant of the Delphi method in the 1970s.

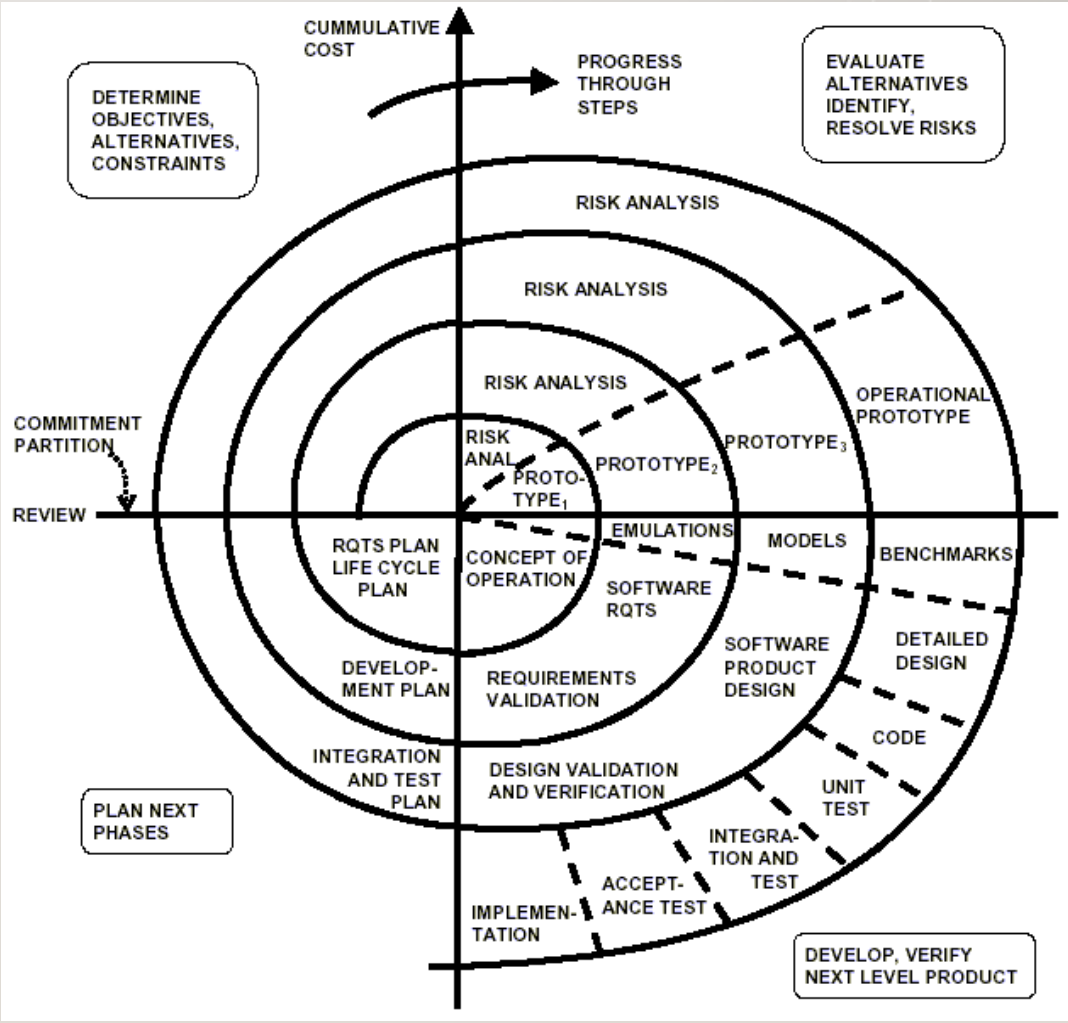
- Coordinator presents each expert with a specification and an estimation form.
- Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.
- Experts fill out forms anonymously.
- Coordinator prepares and distributes a summary of the estimates
- Coordinator calls a group meeting, specifically focusing on having the experts discuss points where their estimates vary widely
- Experts fill out forms, again anonymously, and steps 4 to 6 are iterated for as many rounds as appropriate.

ANEXO – Evolutionary Delivery Model - EVO



ANEXO – SPIRAL MODEL





MOVING THE SCRUM DOWNFIELD

From interviews with organization members from the CEO to young engineers, we learned that leading companies show six characteristics in managing their new product development processes:

1. Built-in instability
2. Self-organizing project teams
3. Overlapping development phases
4. "Multilearning"
5. Subtle control
6. Organizational transfer of learning

These characteristics are like pieces of a jigsaw puzzle. Each element, by itself, does not bring about speed and flexibility. But taken as a whole, the characteristics can produce a powerful new set of dynamics that will make a difference.

Bibliografia

Agile manifesto: http://agilemanifesto.org

Scrum guide: https://www.scrumguides.org/scrum-guide.html

Kanban Essential condensed: http://leankanban.com/guide

Succeeding with Agile: Software Development Using Scrum - Mike Cohn

eXtreme Programming Explained: Embrace Change - Kent Beck

Agile Estimating And Planning - Mike Cohn