

Agile software development



SoftEng
<http://softeng.polito.it>

Version 1.0.0
© Marco Torchiano, Antonio Vetrò, 2021

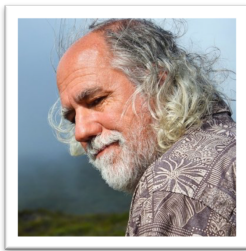


SOFTWARE + ENGINEERING

Software

..is the invisible language that
whispers stories of possibility
to the hardware

– Grady Booch



Engineering

The creative application of scientific principles to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them singly or in combination; or to construct or operate the same with full cognizance of their design; or to forecast their behavior under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property.

– ECPD

Engineering

- **Design**: the intentional solution to a problem within a set of constraints
- **Construction**: planning, monitoring, controlling the activities to achieve a solution, + tools and techniques
- **Operation**: conduction of the solution and adaptation, within its limits

Engineering

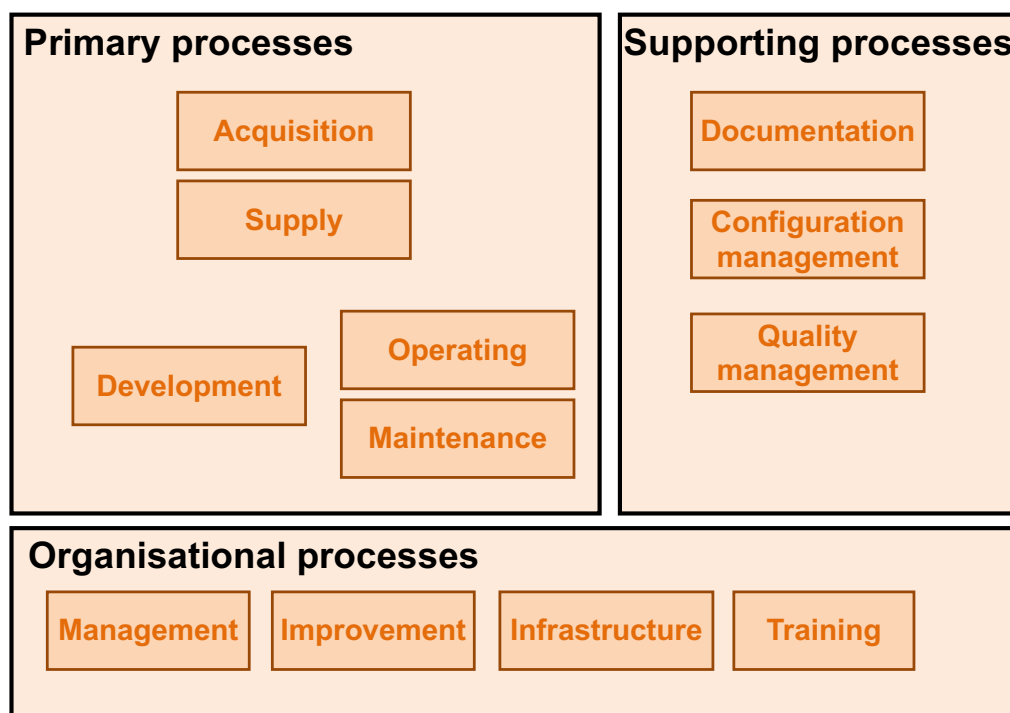
- **Design**: the intentional solution to a problem within a set of constraints
- **Construction**: planning, monitoring, controlling the activities to achieve a solution, + tools and techniques
- **Operation**: conduction of the solution and adaptation, within its limits

Construction

- How does Software Development work in practice?
 - ◆ Effort estimation
 - ◆ Planning development and release
 - ◆ Day-by-day practices
 - ◆ Customer relationship
 - ◆ Quality assurance
 - ◆ Acquisition
 - ◆ ...

Software life cycle processes

ISO/IEC 12207



Agile Development



SoftEng
<http://softeng.polito.it>

History

- Through the 90s several consultant experienced the limitation of “hard” waterfall-like processes
- Each developed his own flavor of agile methodology

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

SoftEng
<http://softeng.polito.it>

Agilemanifesto.org

Individuals and interactions *over*
processes and tools

Working software *over*
comprehensive documentation

Customer collaboration *over*
contract negotiation

Responding to change *over*
following a plan

SoftEng
<http://softeng.polito.it>

Agile Manifesto – Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

SoftEng
<http://softeng.polito.it>

Agile Manifesto – Principles

5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

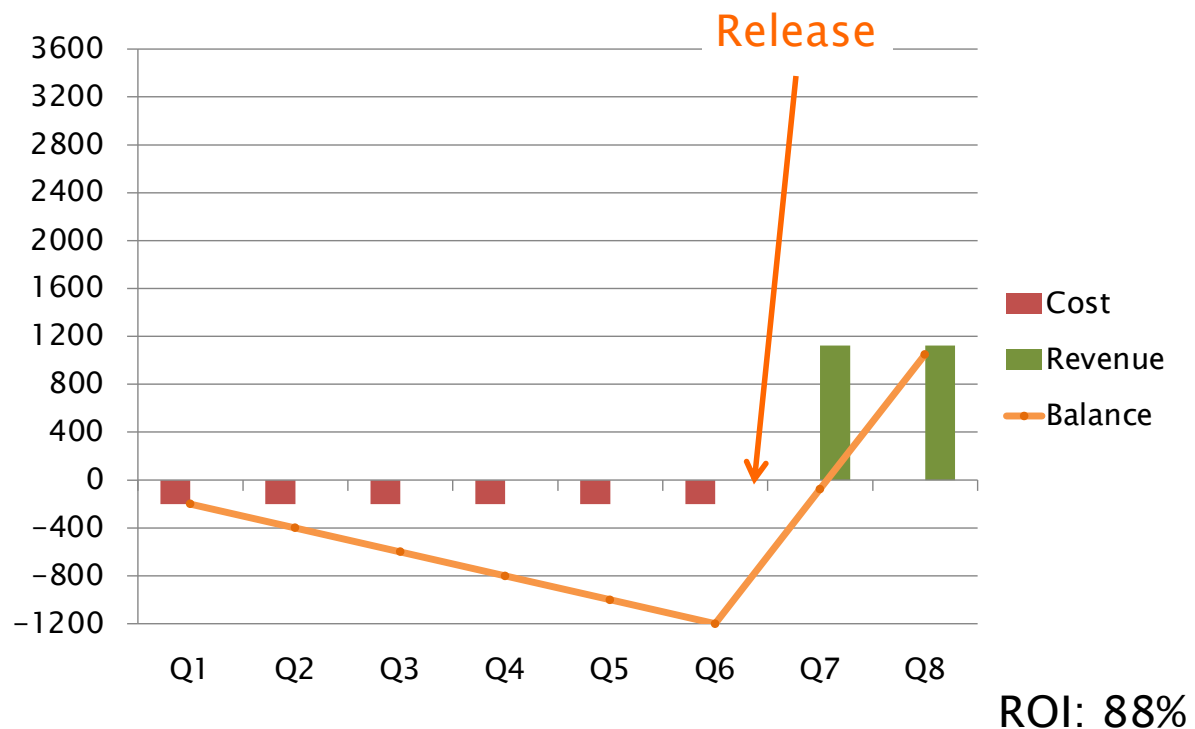
Agile Manifesto – Principles

9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

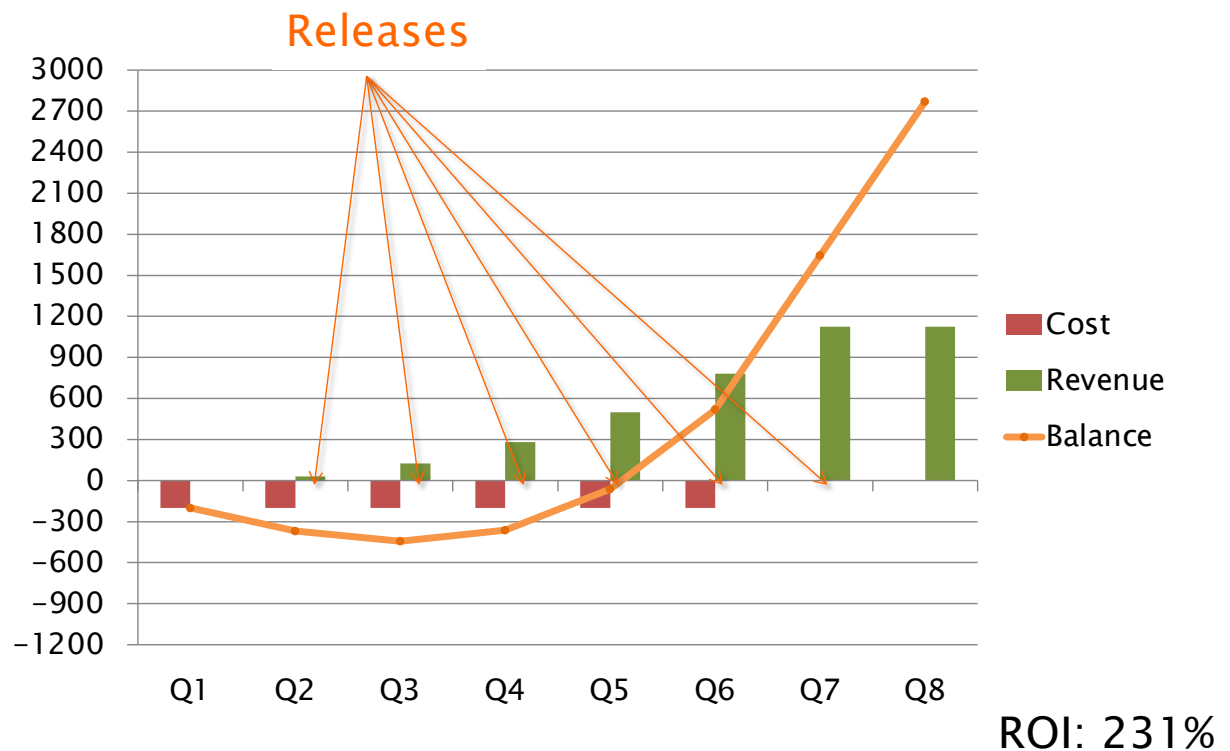
Fictional Case

- Project characteristics
 - ◆ Duration: 1.5 years
 - ◆ Budget: 800 K€/year
 - ◆ Revenue: 4.5 M€/year
- Approaches
 - ◆ Waterfall
 - One release, at the end
 - ◆ Agile
 - 6 releases, each quarter

Waterfall

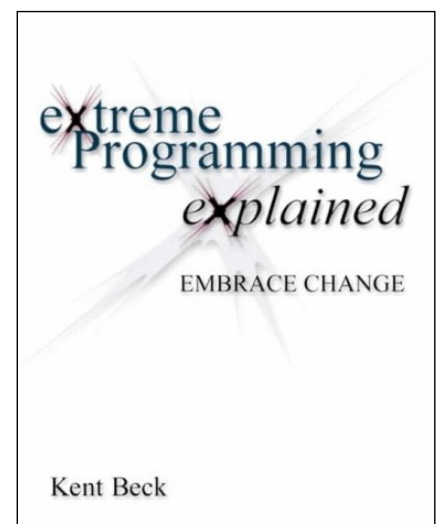


Agile



Extreme programming

- Kent Beck: Extreme Programming Explained Addison-Wesley, 2000, ISBN 0-201-61641-6



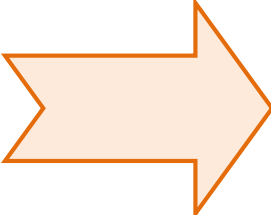
Fundamentals of XP

- Distinguish between decisions made by business stakeholders and developers
- Simplistic – keep design as simple as possible
 - “design for today not for tomorrow”
- Write automated test code before writing production code and keep all tests running
- Pair programming
- Very short iterations with fast delivery

Four values

- Communication
 - ♦ “problems with projects can invariably be traced to somebody not talking to somebody else about something important”
- Simplicity
 - ♦ “what is the simplest thing that could possibly work?”
- Feedback
 - ♦ Put system in production ASAP
 - ♦ “Have you written a test case for that yet?”
- Courage
 - ♦ Big jumps take courage

Four project variables

- Cost
 - ♦ “Adding people to a late project just makes it later”
 - Scope
 - Time
 - Quality
- 
- Time-boxed