

Agile Software Engineering



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

Version 2.0.0
© Marco Torchiano, Antonio Vetrò, 2023



Licensing Note



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-nd/4.0/>.

You are free: to copy, distribute, display, and perform the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.



Non-commercial. You may not use this work for commercial purposes.



No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

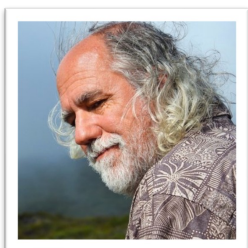
Your fair use and other rights are in no way affected by the above.

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

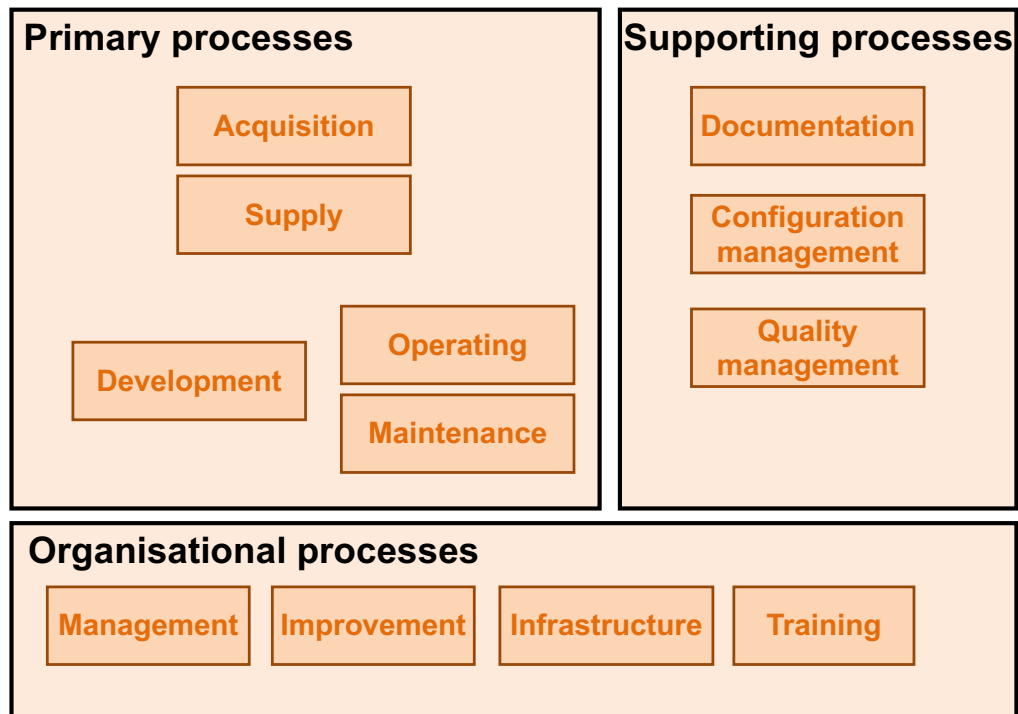
- **Design**: the intentional solution to a problem within a set of constraints
- **Construction**: planning, monitoring, controlling the activities to achieve a solution, + tools and techniques
- **Operation**: conduction of the solution and adaptation, within its limits

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 SOFTWARE DEVELOPMENT

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.

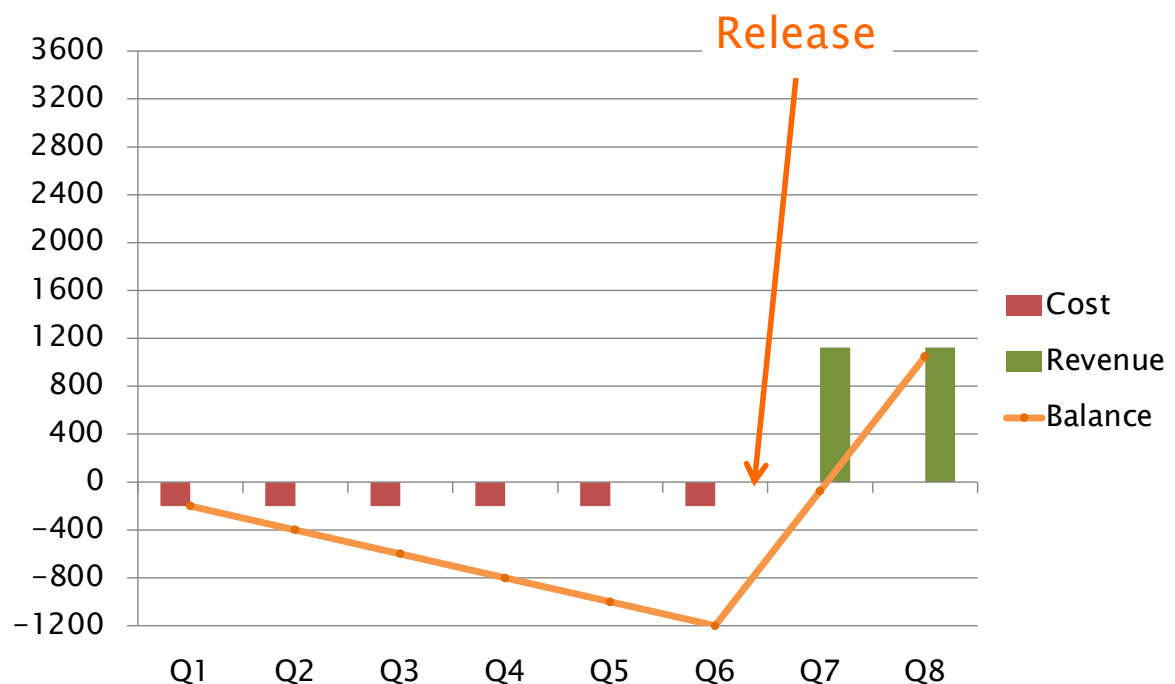
Agile Manifesto – Principles

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

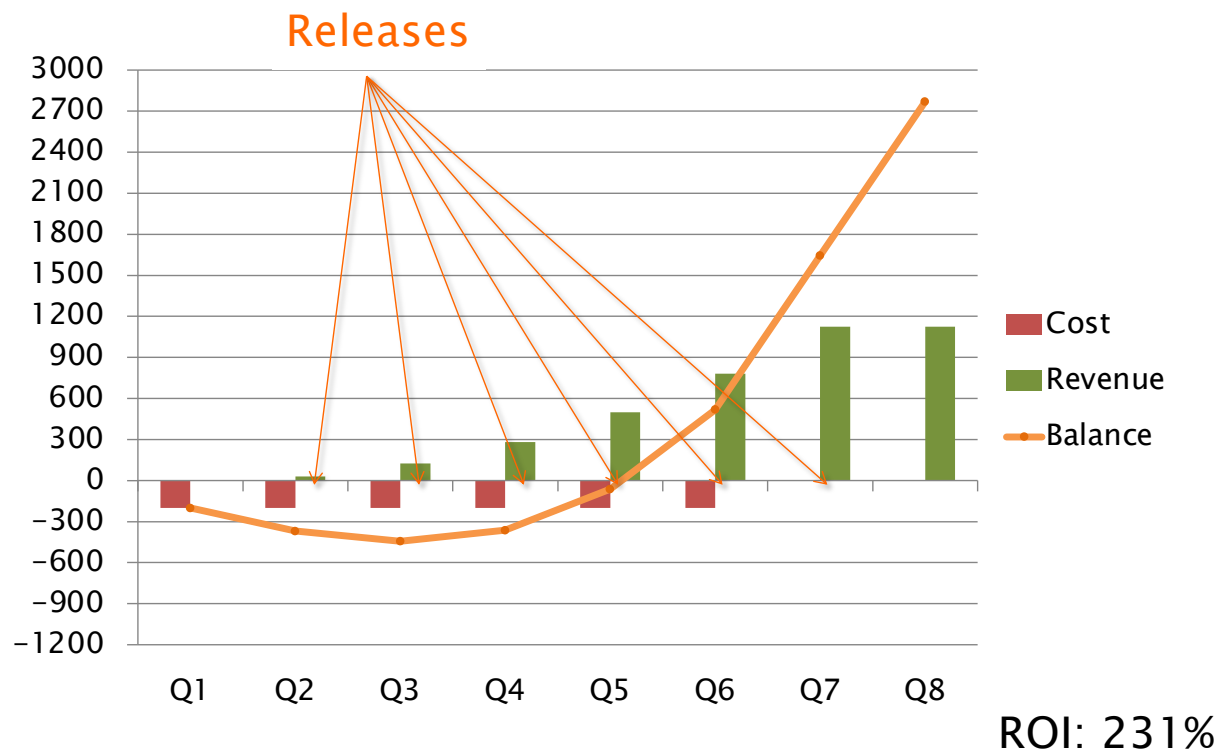
- Project characteristics
 - ♦ Duration: 1.5 years
 - ♦ Budget: 800 K€/year
 - ♦ Revenue: 4.5 M€/year
- Approaches
 - ♦ Waterfall
 - One release, at the end
 - ♦ Agile
 - 6 releases, each quarter

Waterfall



ROI: 88%

Agile



From manifesto to practice

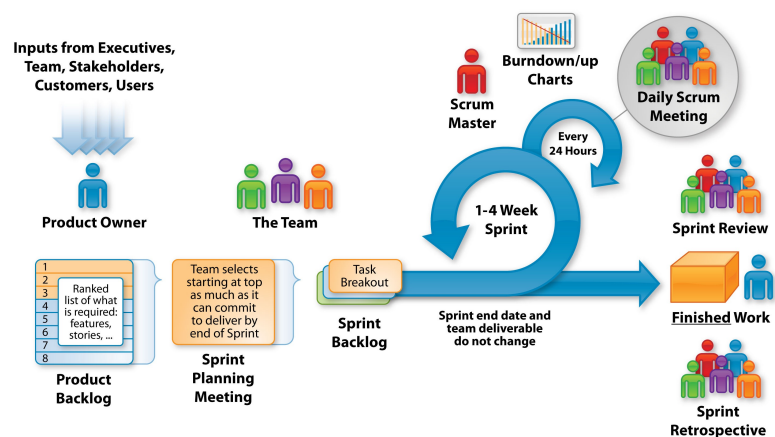
- Tight collaboration between developers and stakeholder
 - ◆ over the entire course of the project
- Self-organizing teams
- Software SCRUM is the most used methodology
 - ◆ Other methodologies like Kanban and eXtreme Programming are less used

Scrum



Software SCRUM

- User stories
- Estimation
- Planning
- Tight Feedback Loops
- Reflection



Software Quality

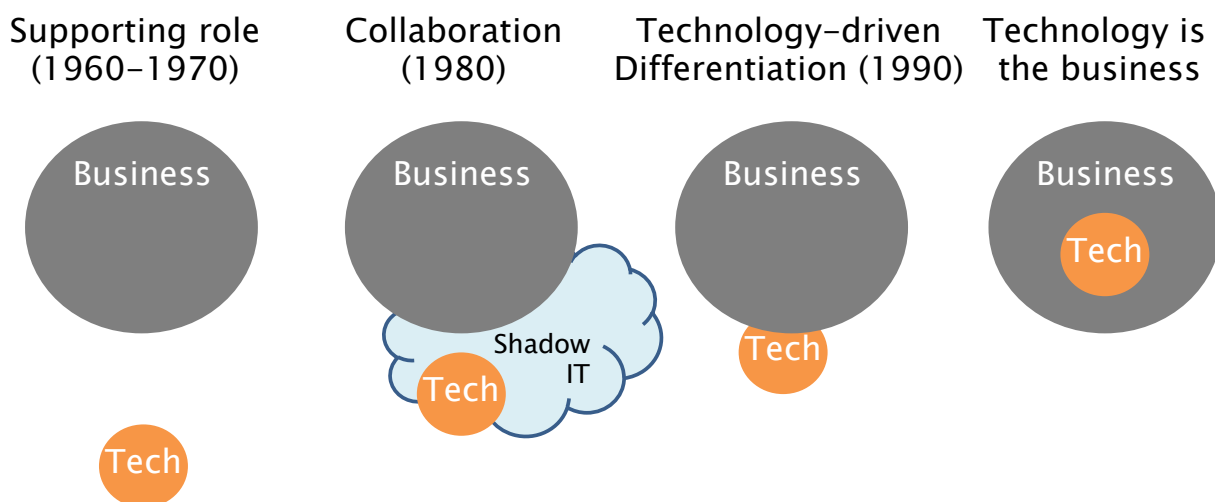
- Quality is a key component in any engineering practice
- How does it work in Sw engineering?
 - ♦ Sw analytics allow measuring and controlling Sw projects
 - ♦ Code smells indicate possible problems
 - ♦ Technical Debt

DEVOPS

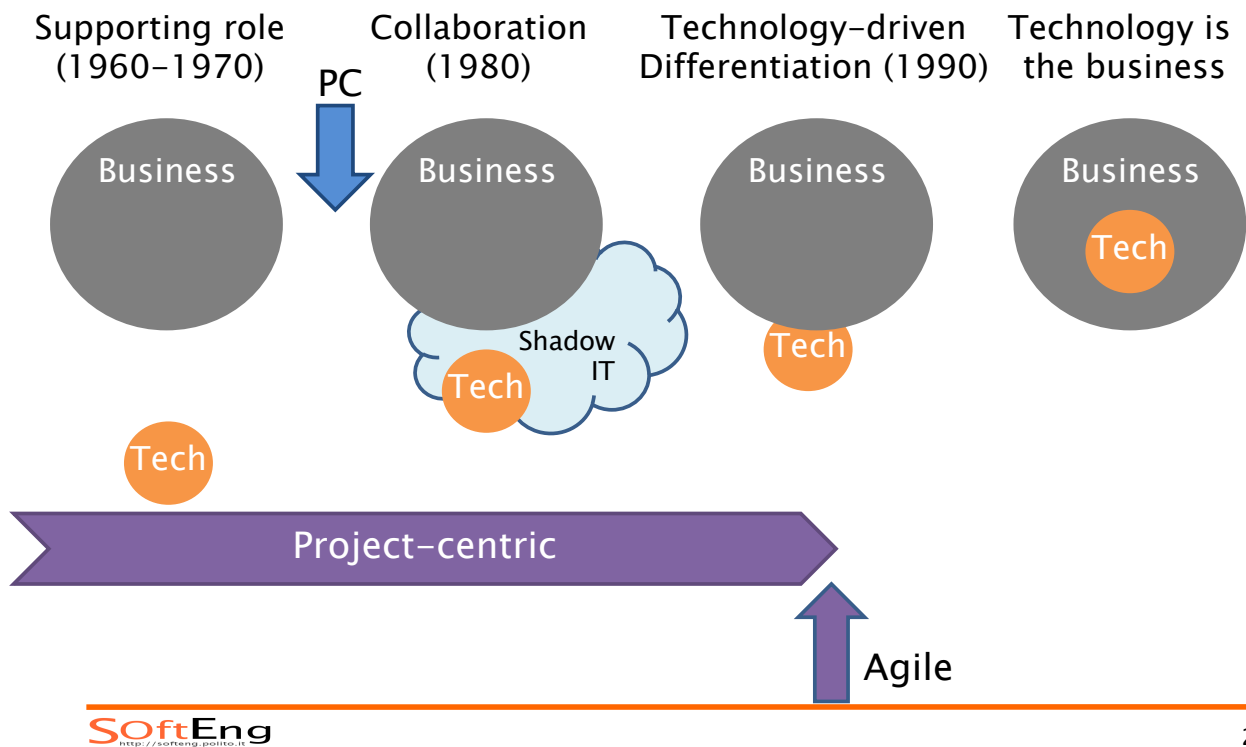
Engineering

- **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

IT Technology w.r.t. Business



IT Technology w.r.t. Business



27

Collaboration mindset

«Never underestimate the bandwidth of a station wagon loaded with tape»

Cycle time

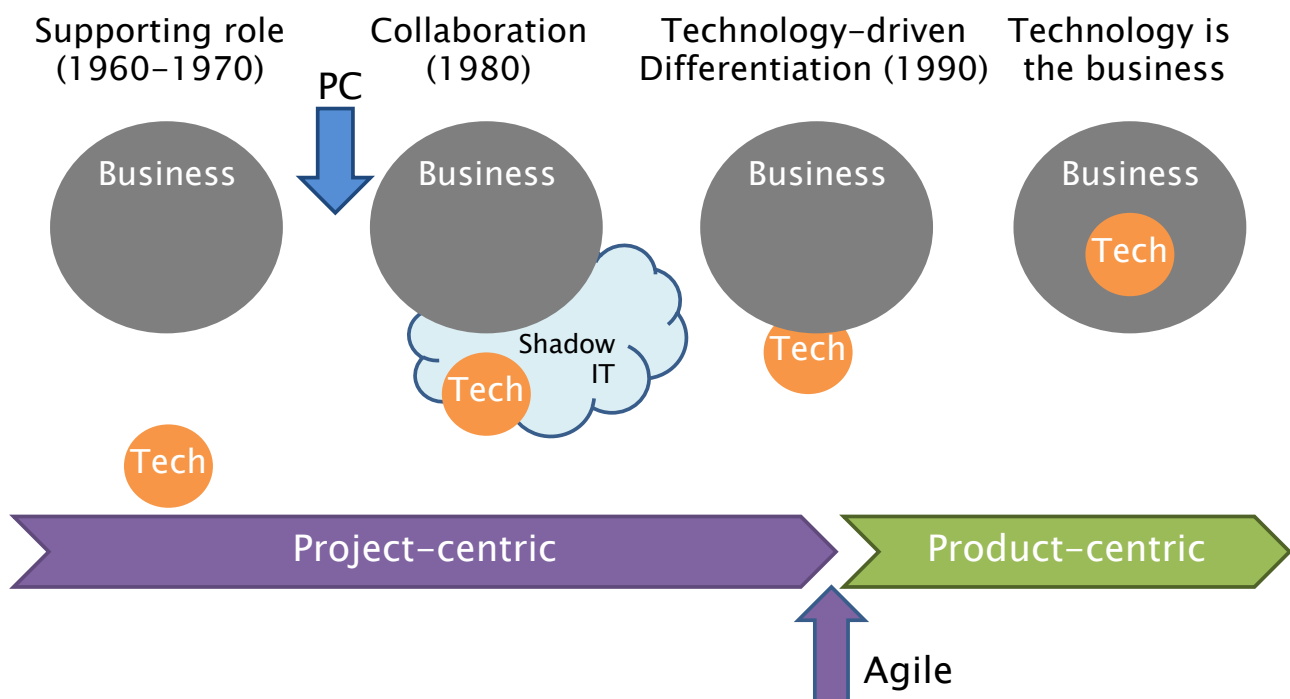
Agile



- Throughput
- Reliability
- Predictability

- **Cycle time**
- Throughput
- Reliability
- ~~Predictability~~

IT Technology w.r.t. Business



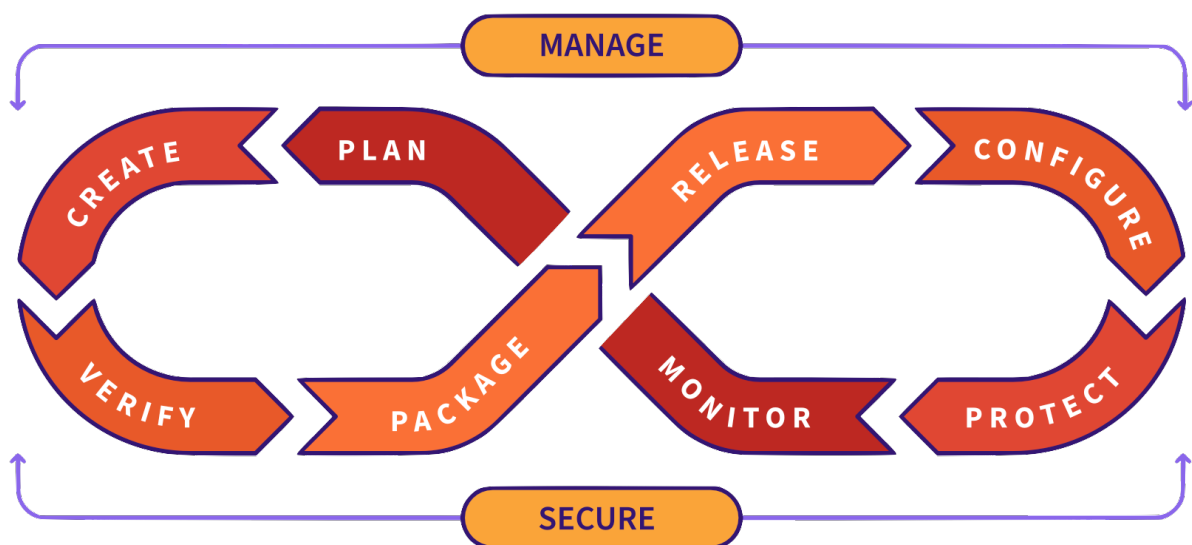
DevOps

Development + Operations

«You build it you run it»

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

DevOps



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

Source: GitLab, A guide to getting started on DevOps

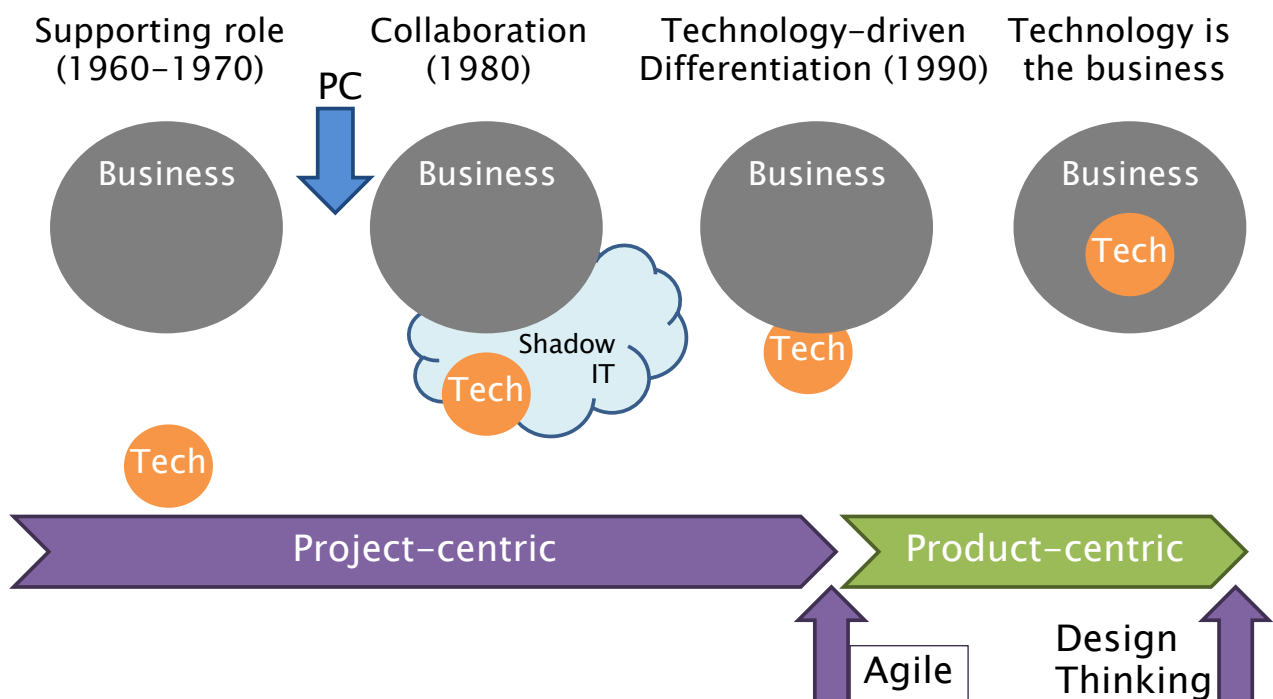
DesignOps

User eXperience **Design** +
Development + **Operations**



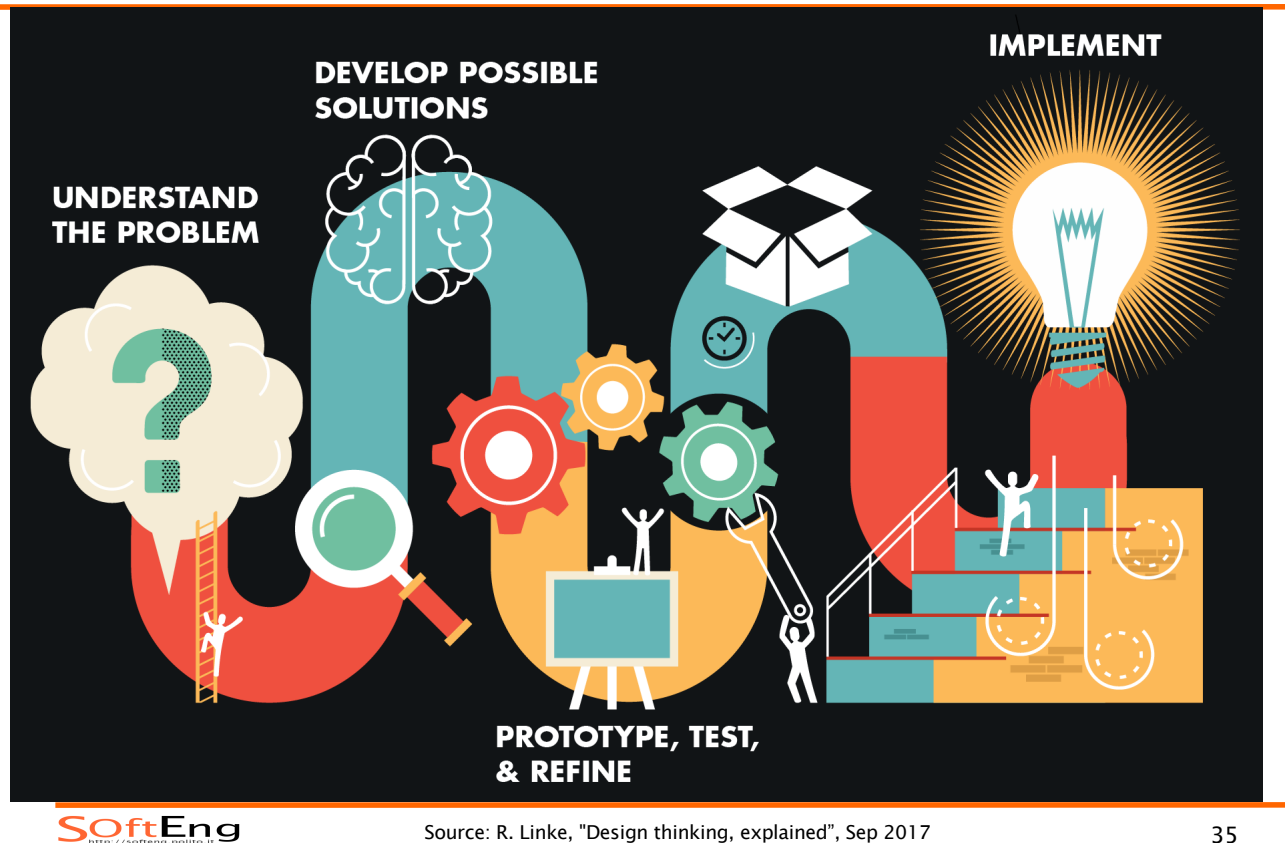
SoftEng
<http://softeng.polito.it>

IT Technology w.r.t. Business



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

Design Thinking



35

Suggested readings

- R.Jeffries "The Nature of Software Development" The Pragmatic Programmers
- K.Schwaber, M.Beedle. "Agile Software Development with Scrum", Prentice-Hall, 2001
- M.Cohn. "Succeeding with Agile", Addison-Wesley, 2010.
- C.Sims, H.L.Johnson. "The Elements of Scrum" DYNAMICON, 2011.
- K.Schwaber and J.Sutherland "The Scrum Guide" (<http://www.scrumguides.org>)
- D.Spinellis. Effective Debugging Addison-Wesley, 2016.
- P.Grubb, A.Takang "Software Maintenance: concepts and practice" World Scientific Publishing, 2003

Suggested readings

- E. Dornenburg, «The Path to DevOps», IEEE Software, Sep/Oct 2018
 - ♦ <https://ieeexplore.ieee.org/document/8409919>
- R. Linke, "Design thinking, explained", Sep 2017
 - ♦ <https://mitsloan.mit.edu/ideas-made-to-matter/design-thinking-explained>