Lean Software Development
Toward Lean Software Development
Lecture 12

Artem Kruglov, Giancarlo Succi

Innopolis University

a.kruglov@innopolis.ru, *g.succi@innopolis.ru*

24th February 2021

Lean Software Development
Toward Lean

Outline

- **Existing Proposals to Create "Lean Software Development"**
- Share a Common Vision
- Deprive Gurus of Their Power
- Agile extremists
- Experience factory

Existing Proposals to Create "Lean Software Development"

Seven principles of Lean software development (Mary and Tom Poppendieck):

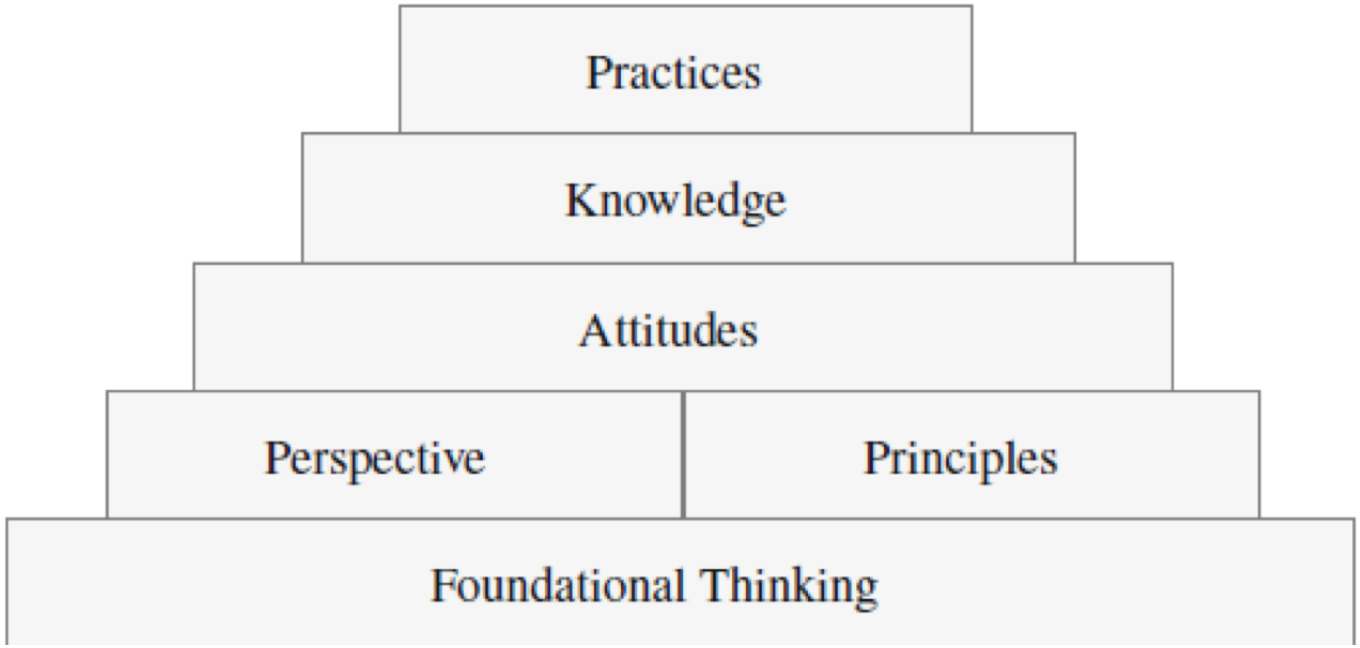
- Eliminate waste;
- Build quality - we used the terms "autonomation" and "standardization";
- Create knowledge;
- Defer commitment - we used the term "just-in-time";
- Deliver fast - get frequent feedback from the customer and increase learning through frequent deployments;
- Respect people - we used the term "worker involvement";
- Optimize the whole - we used the term "constant improvement."

Existing Proposals to Create "Lean Software Development"

Curt Hibbs propose the following practices:

- Source Code Management and Scripted Builds;
- Automated Testing;
- Continuous Integration;
- Less Code;
- Short Iterations; and
- Customer Participation.

(Alan Shalloway et al.) approach organizes the transition of Lean Thinking to Lean Software Development into the following layered model:



Existing Proposals to Create "Lean Software Development"

- **Foundational Thinking.** The underlying belief system of Lean Thinking, based on the work of Deming.
- **Perspective and Principles.** The Perspective is the choice of what is considered important to observe in the process. The Principles are the rules of behavior that adhere to the Foundational Thinking and are taken from the work of Mary and Tom Poppendieck.

- **Attitudes.** The choice of what is considered important and what is not.
- **Knowledge.** "Know-how" based on experience or, in other words, "lessons learned."
- **Practices.** Recommendations on what to do, based on the knowledge acquired.

Existing Proposals to Create "Lean Software Development"

The three examples presented above show that Lean Thinking can be translated in different ways into software engineering.

However, all these three approaches lack an essential component of Lean Management, its concrete use of real measurements supporting the process.

They are more faith-based, while Lean advocates a constant and concrete analysis of the process to produce value and eliminate waste.

Existing Proposals to Create "Lean Software Development"

Our approach is to develop a Lean software development process that avoids the three issues:

- the problem of communicating the goals and methods of Agile methods to stakeholders, which generates skepticism since Agile methods seem to ignore "well-known" best practices;
- the guru approach that has dominated the way Agile ideas became known among practitioners; and
- agile extremists that promote the dark side of Agile.

Outline

- Existing Proposals to Create “Lean Software Development”
- **Share a Common Vision**
- Deprive Gurus of Their Power
- Agile extremists
- Experience factory

Share a Common Vision

Agile methods heavily rely on the collaboration with the customers. But it is hard!!! Van Deursen has identified three major causes why it is hard:

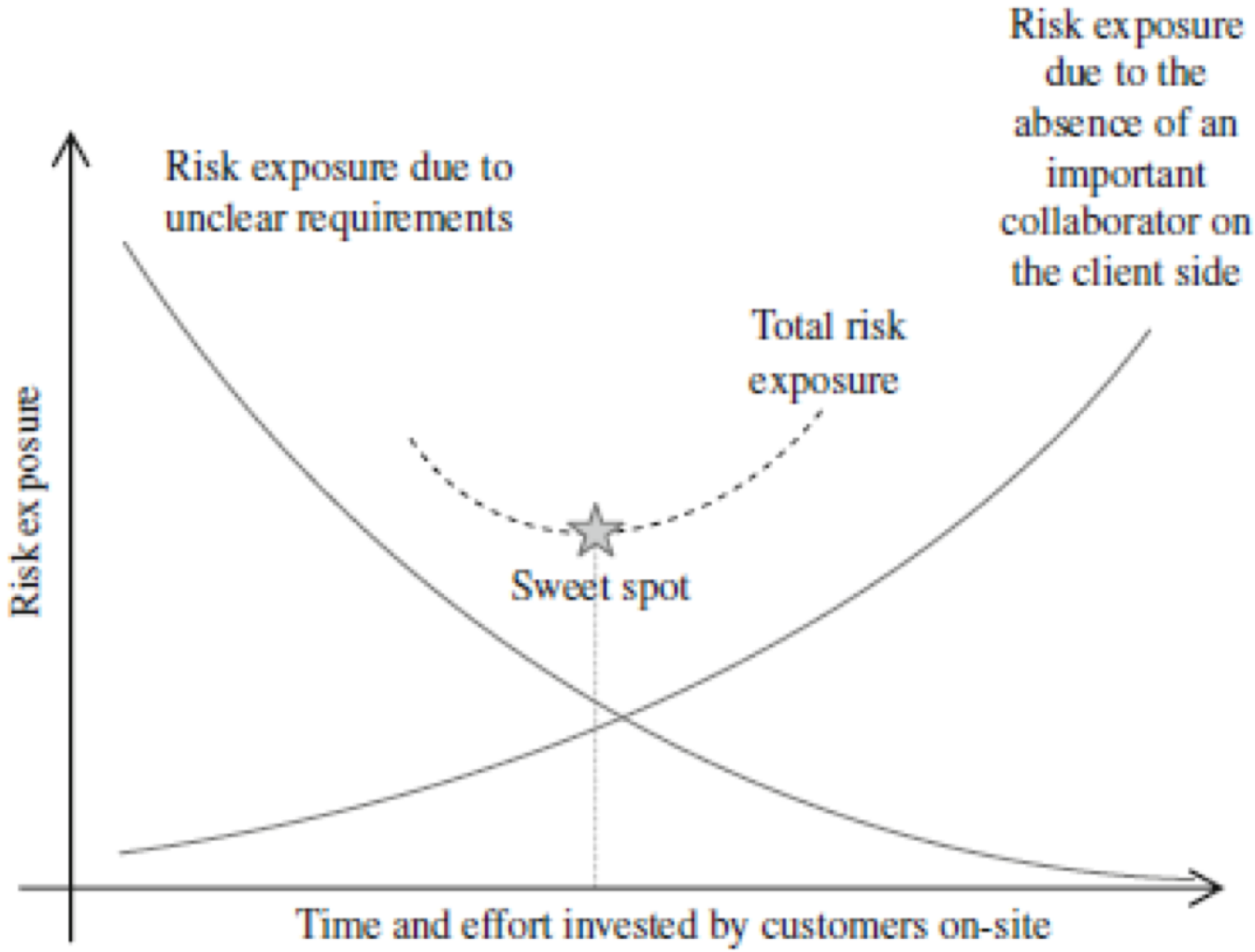
- customers have to do their regular work and be on-site, which is not always possible;
- the customer usually wants to buy a "whole solution," and not to run a customization project requiring his involvement; and
- the best customers from a programmer's perspective are also often best in other aspects, which makes them busy, and it is unlikely to allocate to the project all the required time.

Share a Common Vision

Establishing a fruitful communication with the customer is also challenging because:

- Technologists and end users have a high "semantic gap," which makes communication complicated.
- Neither developers nor customers consider talking to each other a useful task, but rather a waste of time.
- End users may resist changes in their way of working, making it very hard to involve them in a constructive way in the customization of the product.
- Developers might be against an on-site customer.

Share a Common Vision



Trade-off between an on-site and off-site customer

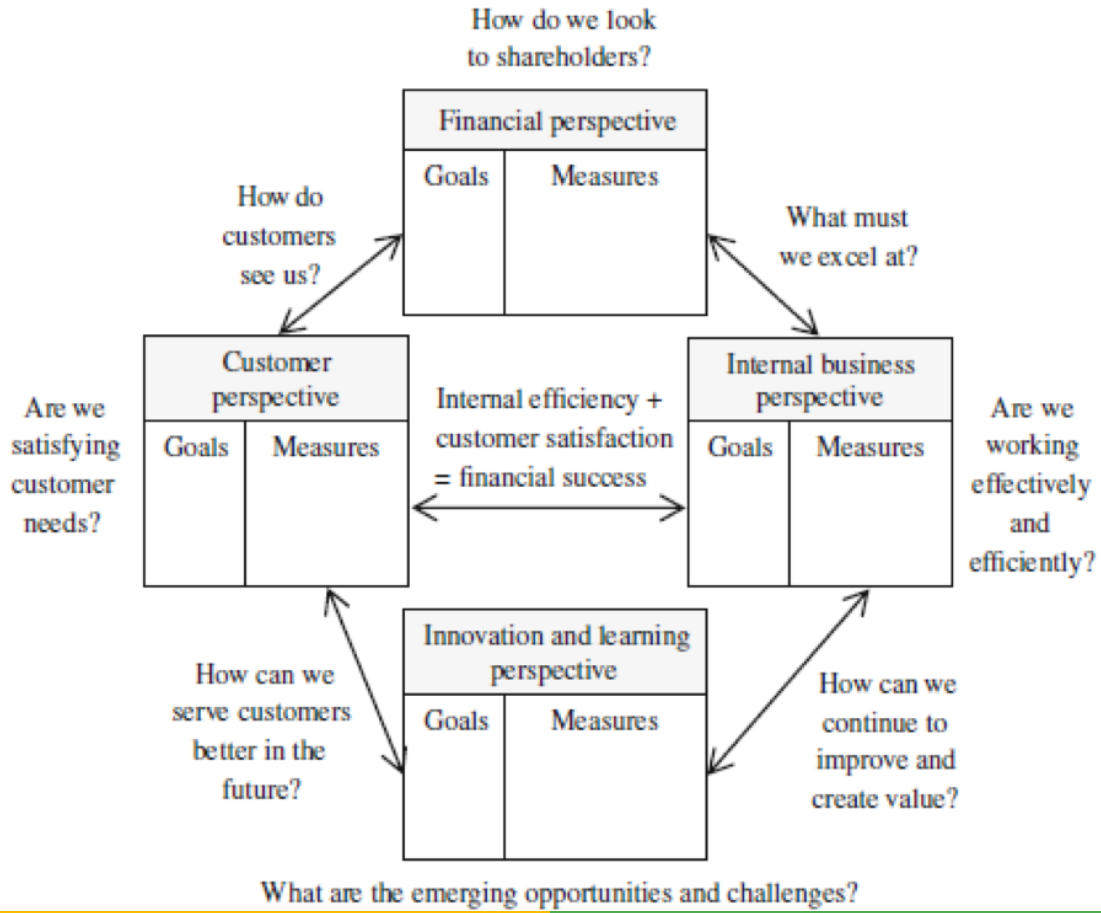
Share a Common Vision

There are several approaches to communicating a strategy to a client. One possibility is a "mission statement": it states the vision and describes the chosen means to achieve it.

An example of a "mission statement" with measurements is the Balanced Scorecard

Share a Common Vision

The Balanced Scorecard(BSC) is a strategy performance management tool. The goal of the BSC is to provide a balanced view of the performance of the company.



Share a Common Vision

- **Customer perspective:** measures the ability of the company to provide value to the customers. This perspective includes performance, quality, and service measurements.
- **Internal business perspective:** measures the ability of the company to adapt the internal processes to satisfy customer needs.
- **Innovation and learning perspective:** the customer and internal business perspective define what the company considers important for competitive success.
- **Financial perspective:** measures if the company's strategy, implementation, and execution are contributing to bottom-line improvement.

Outline

- Existing Proposals to Create “Lean Software Development”
- Share a Common Vision
- **Deprive Gurus of Their Power**
- Agile extremists
- Experience factory

Deprive Gurus of Their Power

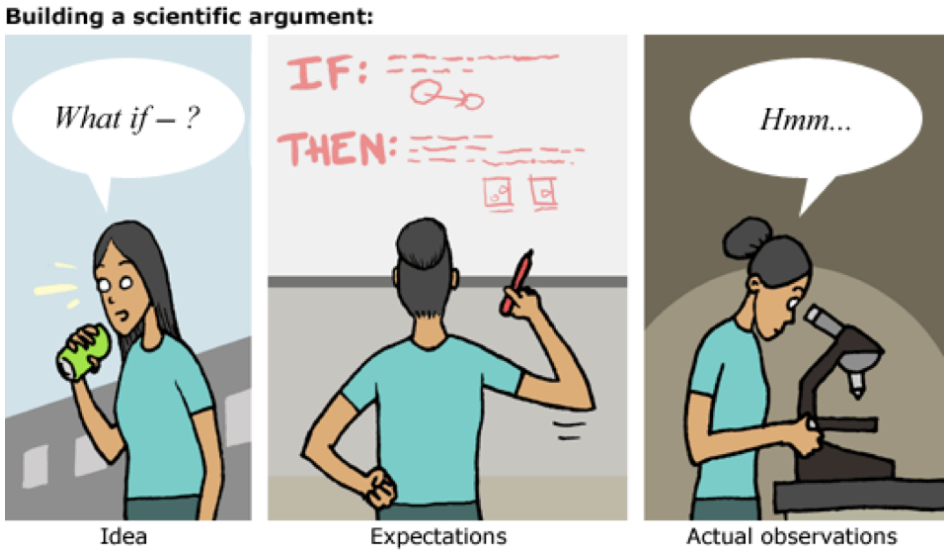
If we need to find out how good a certain technology can work for us and cannot find anyone that can tell us, we have to develop the experience ourselves.

We need to use the scientific method to systematically find the knowledge we seek continued

Deprive Gurus of Their Power

We need to:

- formulate a problem in form of hypotheses;
- identify what we want to study;
- apply research methods to obtain data;
- analyze the data; and
- use the results to confirm or falsify the hypotheses;



[1] http://www.understandingscience.org

Outline

- Existing Proposals to Create “Lean Software Development”
- Share a Common Vision
- Deprive Gurus of Their Power
- **Agile extremists**
- Experience factory

Agile extremists

- They are risk neutral, optimistic, and idealistic people.
- They are willing to accept risk to introduce radical changes.
- Their idealism makes them see the whole world as Agile, Lean, etc.
- Every problem is framed in their "believe system," in their view of the world.

[1] http://baymard.com/blog/ab-testing-problem-and-hypothesis

Outline

- Existing Proposals to Create “Lean Software Development”
- Share a Common Vision
- Deprive Gurus of Their Power
- Agile extremists
- **Experience factory**

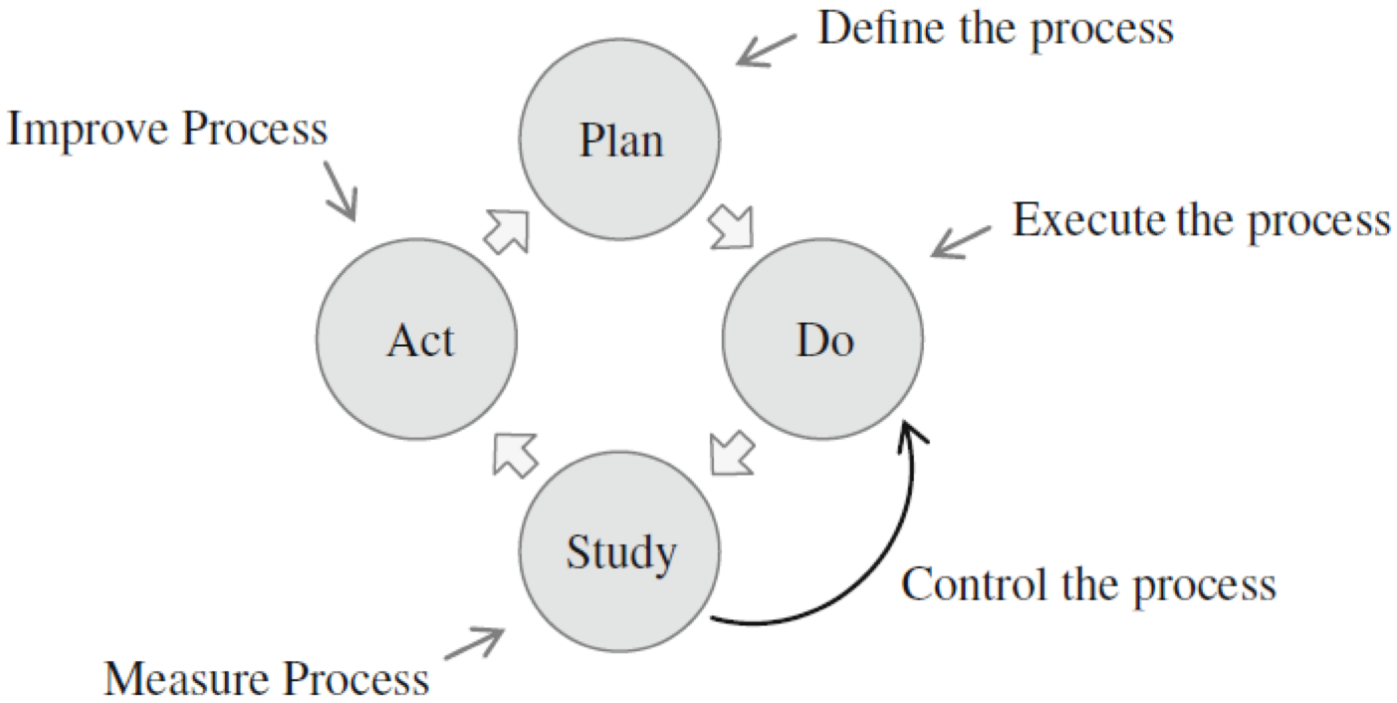
Warm up exercise

- Pair with a person who have not yet seat nearby in the course.
- In 5 minutes create a google doc where you list:
  - one situation each when you gained experience in a scholastic situation and
  - one situation each when you gained experience in a non-scholastic situation
- Share the google doc in Telegram
- Extract the common features
- Merge with another pair and in the newly formed group of 4 in 10 minutes:
  - find the common features in your gaining experience and
  - list the two most important aspects of gaining and sharing experience
  - list the two most difficult aspects in gaining and sharing experience

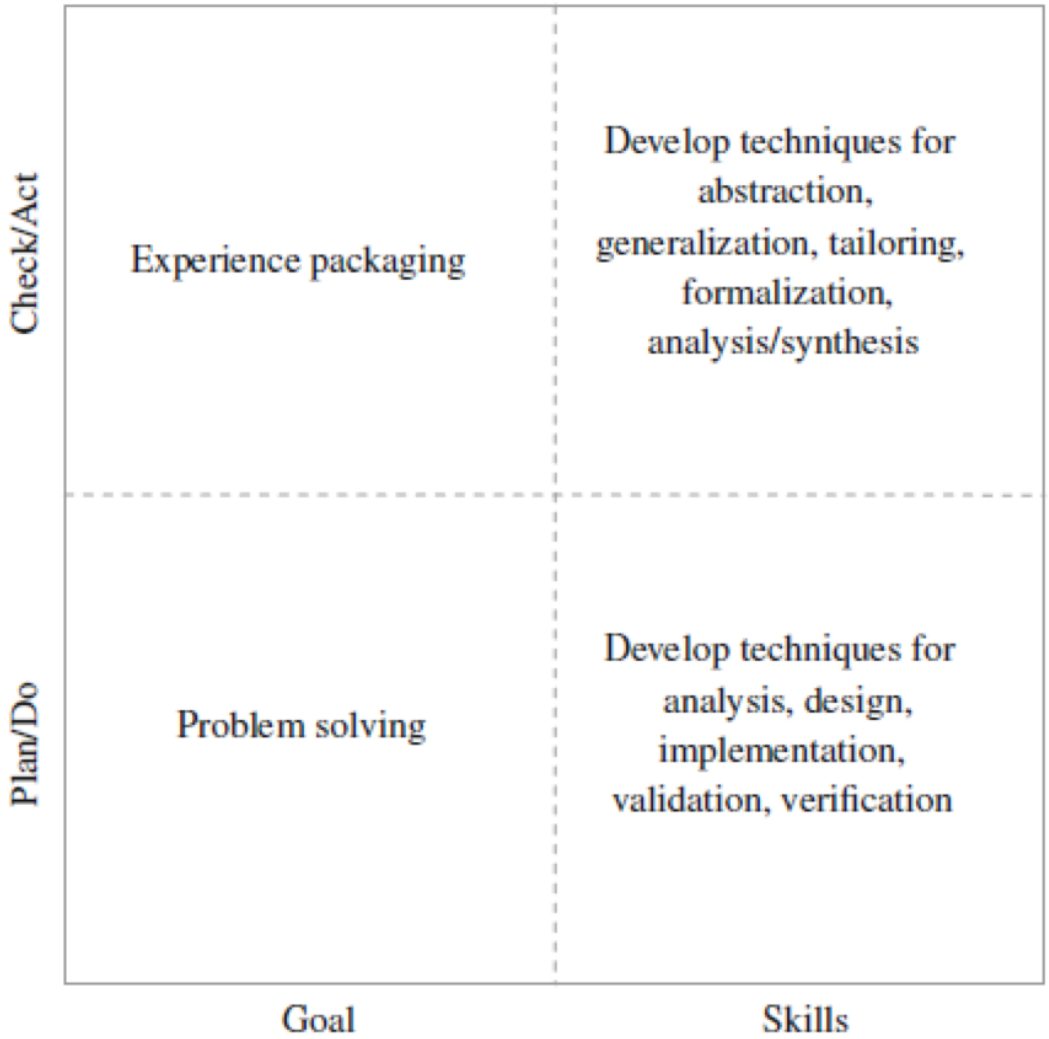
The Experience Factory

- The Experience Factory approach defines a framework to continuously improve the quality of the software development process. This is accomplished through the systematic collection, creation, and reuse of experience.
- Experience management can be considered a variant of knowledge management that manages data and information and transforms it into knowledge and wisdom to increase the understanding of the underlying principles of the analyzed phenomena.

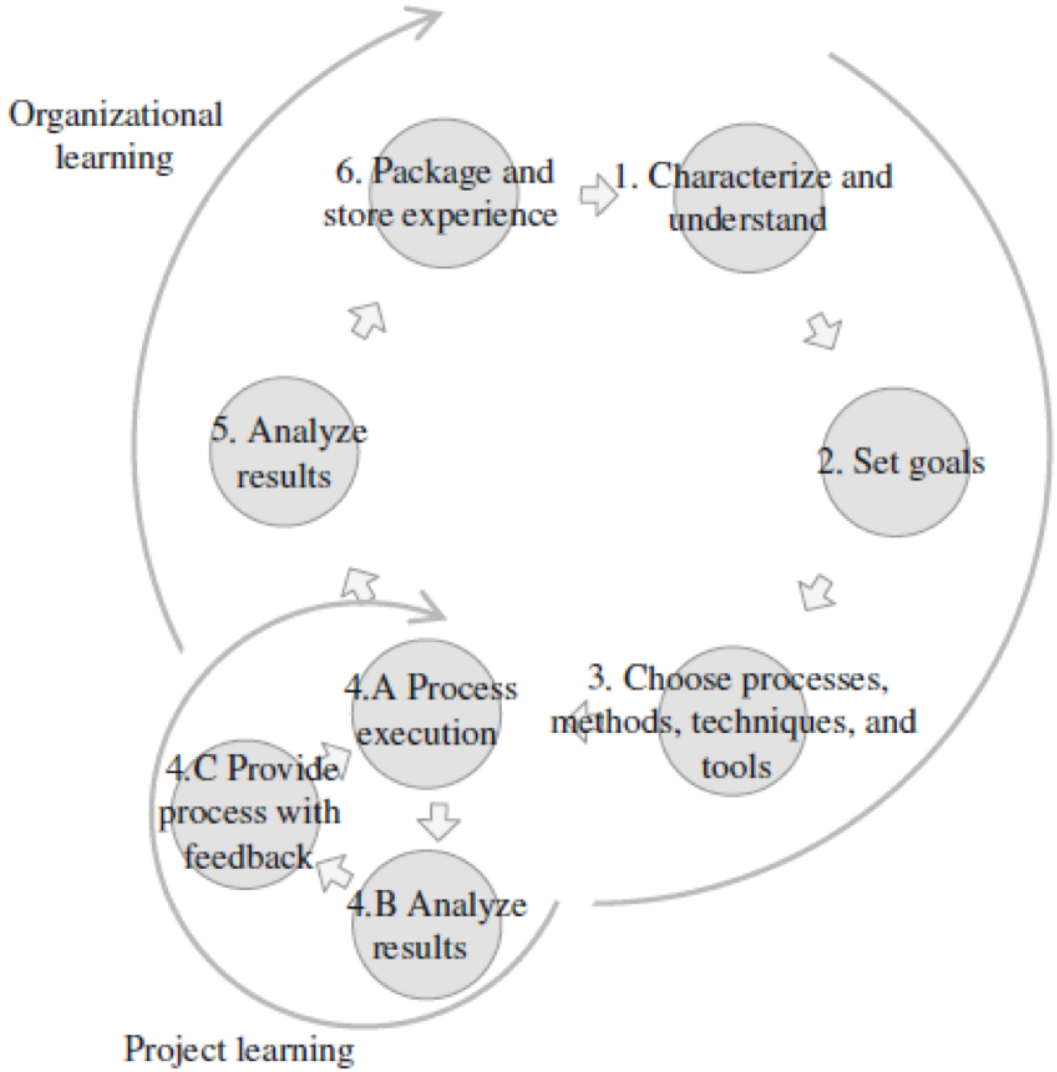
The Plan-Do-Study-Act cycle



Goals and skills in the Experience Factory



The Quality Improvement Paradigm cycle



The Quality Improvement Paradigm cycle

The main cycle consists of the following six steps:

1. The characterize and understand

Analyzes the current project with respect to different characteristics to find a similar set of projects.

2. Set goals

Identifies the goals of the process execution. It formalizes all aspects that are important and therefore should be observed during the project development continued

The Quality Improvement Paradigm cycle

3. Choose processes After settings the goals, the appropriate means to achieve them have to be selected

4. Execute

- Process execution
- Analyze results
- Provide process with feedback

The Quality Improvement Paradigm cycle

5. **Analyze results**
6. **Package and store experience** The aim is to Package Experience so that it can be used for future projects. continued

The Quality Improvement Paradigm cycle