Agile Project Management

Lecture 2

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://about.softuni.bg

Have a Question?

Software University

sli.do

#Agile

Table of Contents

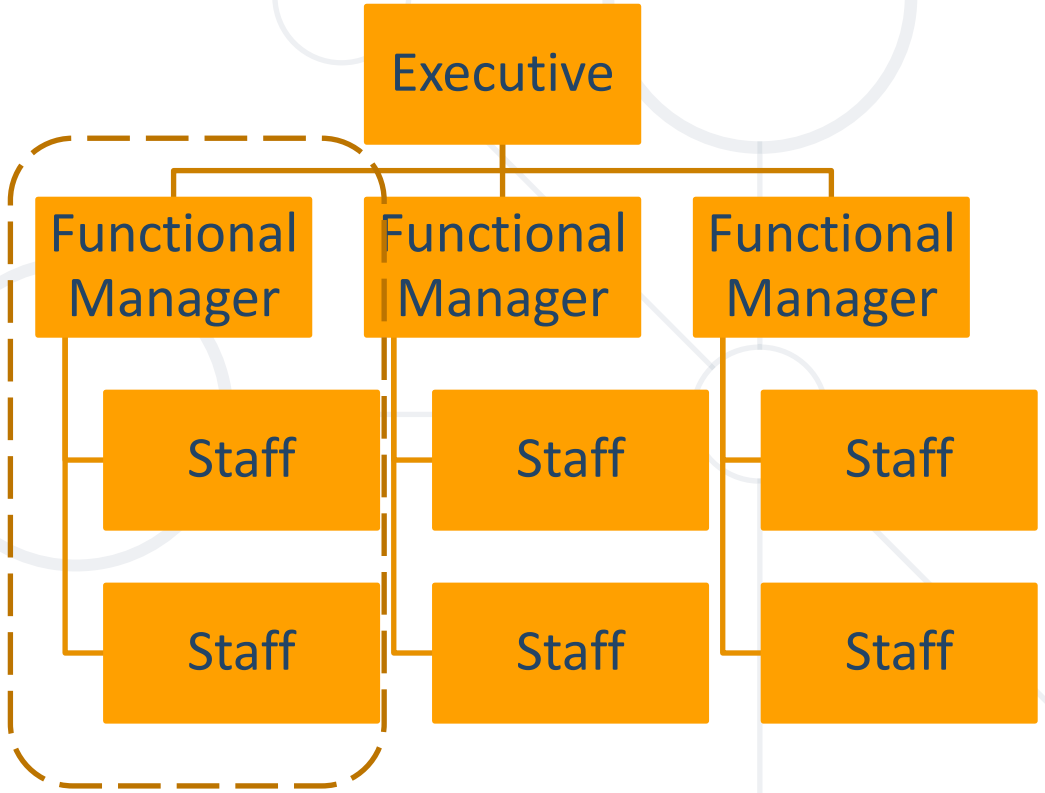
Organizational Structure Types

- Functional
- Projectized
- Matrix

Introduction to Classical Project Management

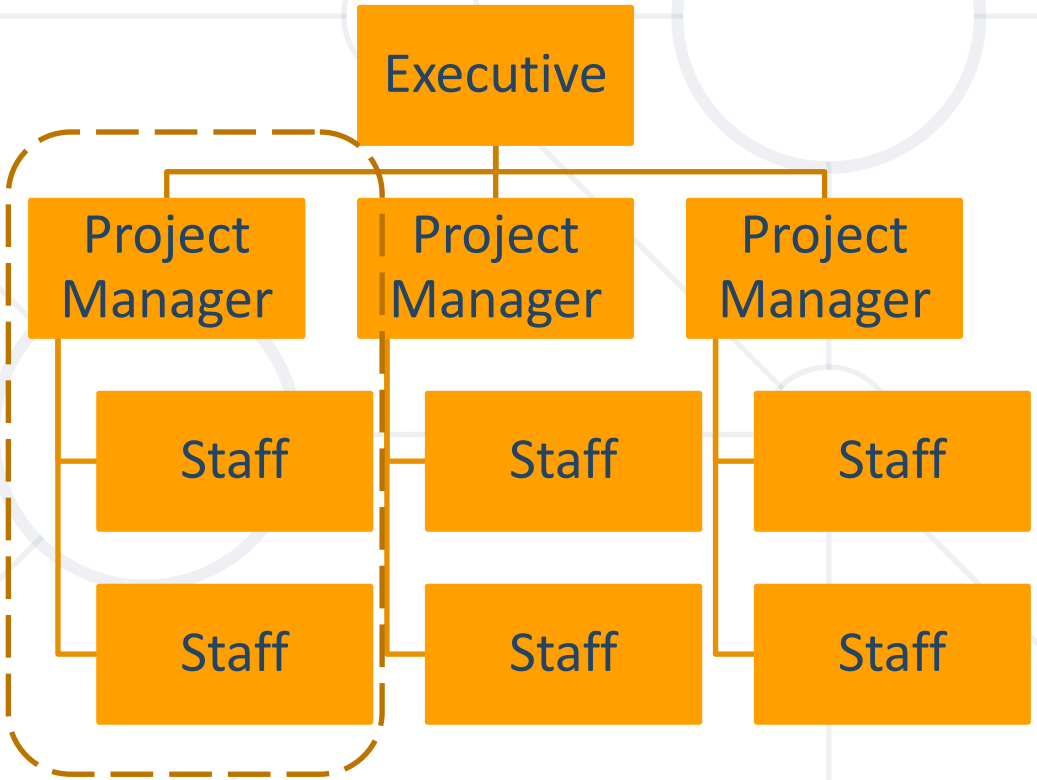
Functional Organization

- Employees are grouped by specialization
  - Accounting, marketing, etc.
- Governed by Functional manager
- Projects within a single department
- Project work in addition to normal work



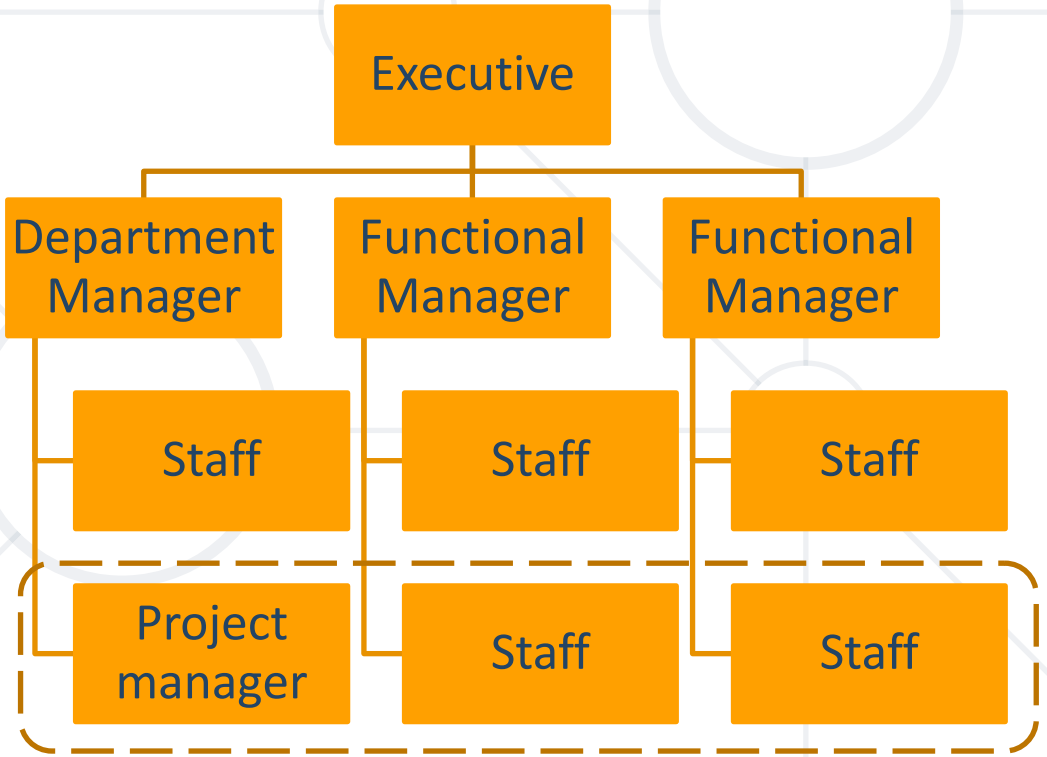
Projectized Organization

- The company is organized by projects
- The PM has full authority
- Employees report to the PM
- Only project work is done



Matrix Organization

- A blend of functional and projectized
- High level of coordination is required
- Communication to both PM and Functional Manager



Project Life Cycles – Predictive

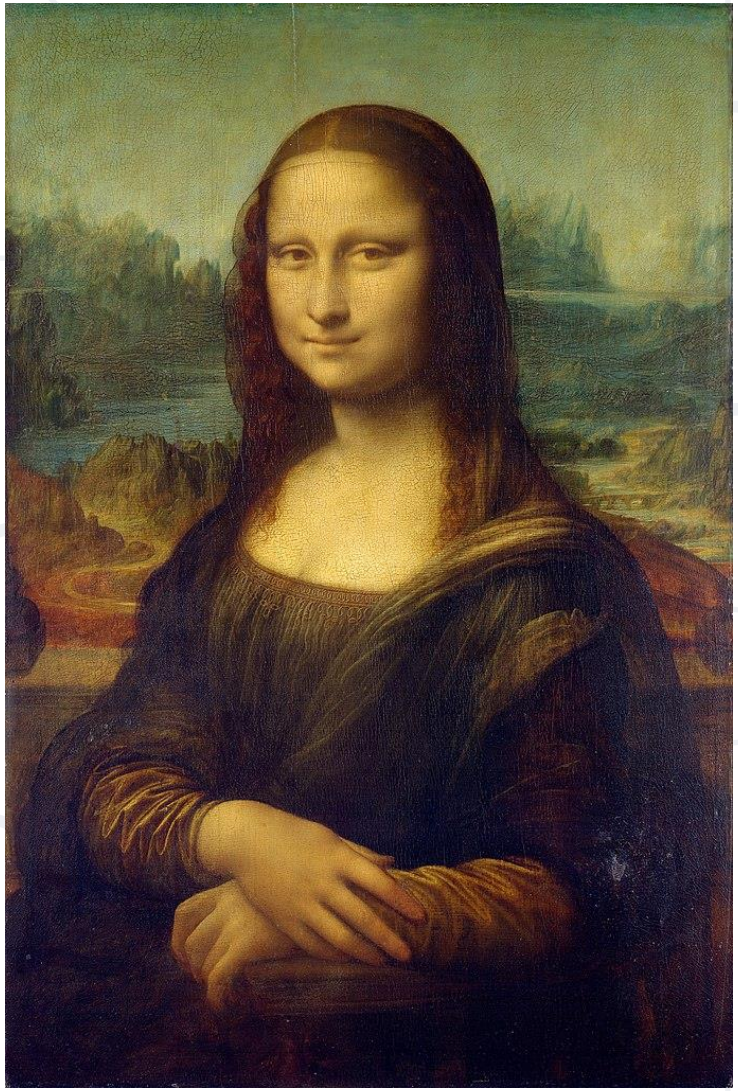
- Predictive (also knowns as Plan-Driven, Waterfall, Traditional)
- Used when it is possible to understand the deliverable well or when there is a need to do so
- Three major constraints are developed in the beginning
- Any changes are carefully managed

Predictive Life Cycles

- Clear vision of what the end results should look like

- Details about the woman – face, clothes, colors etc.

- Details about the scenery – forest, water, colors, etc.



Project Life Cycles – Iterative

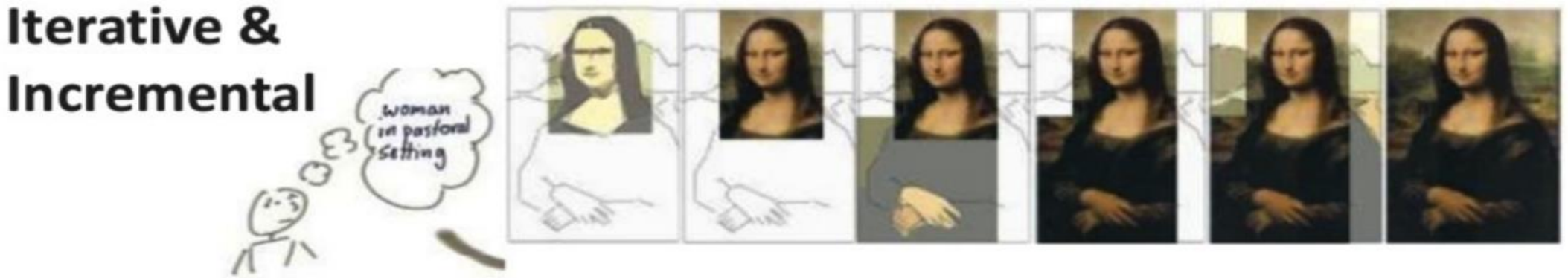
- Development of product happens through a series of repeated cycles
- Scope is determined early, but time and cost are updated as understanding of the product increases
- Higher customer involvement (periodic)

Project Life Cycles – Incremental

- Deliverables divided in fully functional increments
- The increments' timeframe might be specified upfront
- Higher customer involvement (periodic)

Project Life Cycles – Adaptive

- Adaptive (also known as Change-driven or **Agile**)
- Scope is determined for each phase or iteration
- Scope is driven by requirements and business value
- Rapid iterations with continuous customer involvement
- Used when deliverable is not well understood, for complex project with high level of change

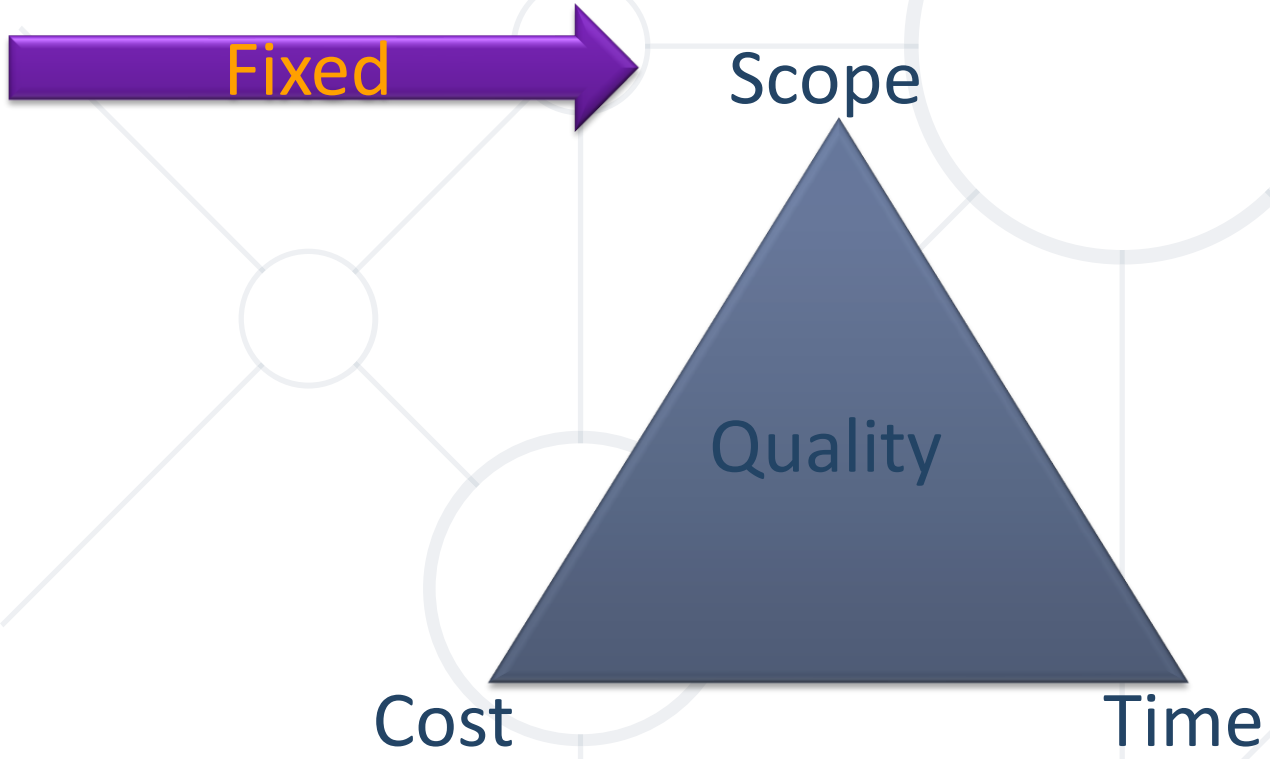


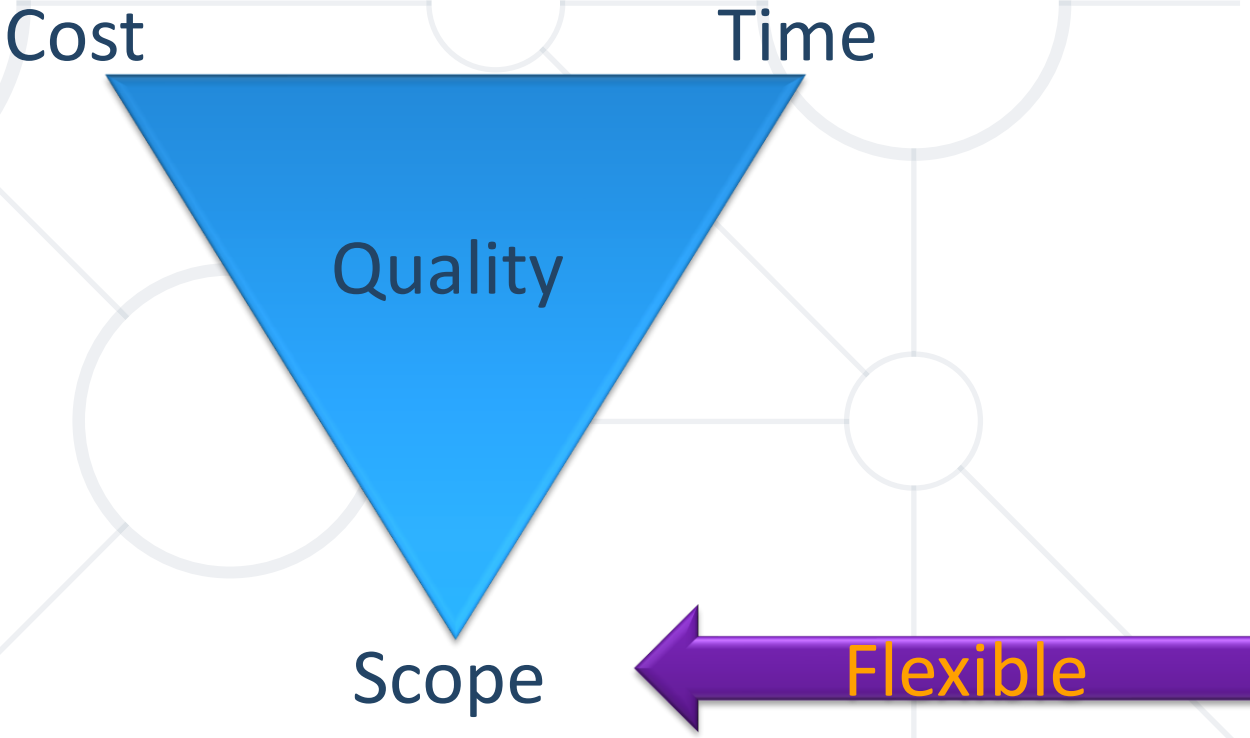
Project Life Cycles – Hybrid

- Combination of predictive and adaptive life cycles
- Well-know elements follow predictive life cycle
- Evolving elements follow adaptive life cycle

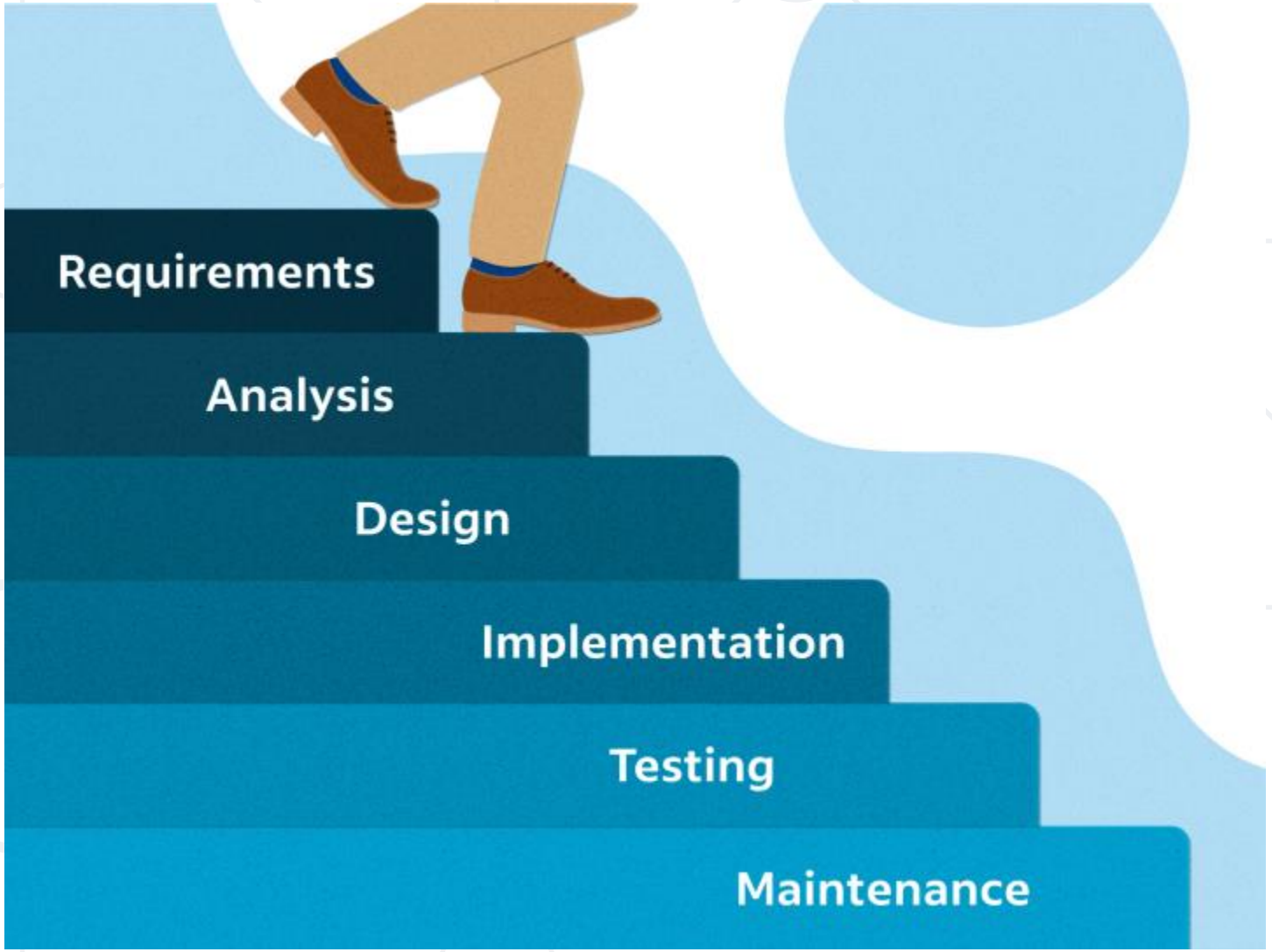
Project Life Cycles – Comparison

Triple Constraints – Traditional & Agile





Waterfall

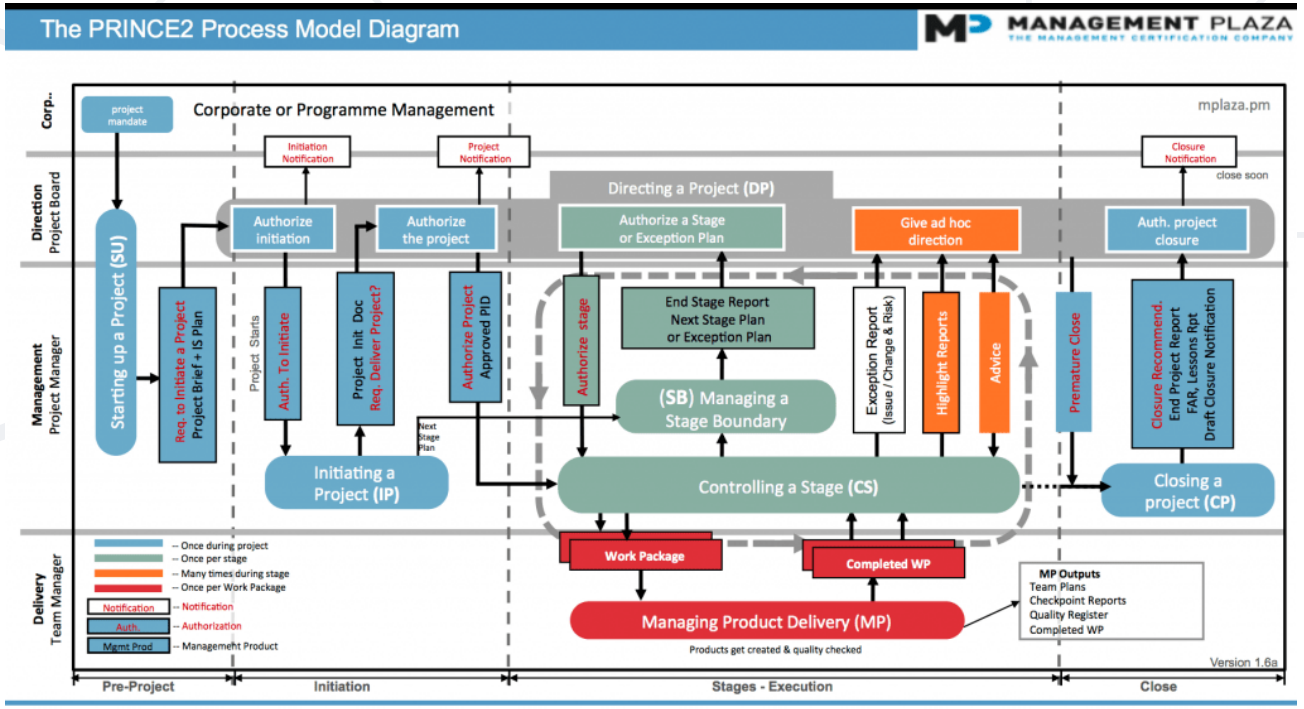

Requirements
Analysis
Design
Implementation
Testing
Maintenance

Most Common Traditional Project Approaches

Software University

PMBOK Guide 2021 Update

	PMBOK 6th Edition	PMBOK 7th Edition	
01	Focus on process	Focus on value based delivery	01
02	10 Knowledge areas	8 Performance domains	02
03	Emphasis on phases of project management	Emphasis on project delivery principles	03
04	Follows traditional methodology of PM	Introduction to various PM approaches	04
05	Well understood by project managers	Understood by anyone directly involved with project	05



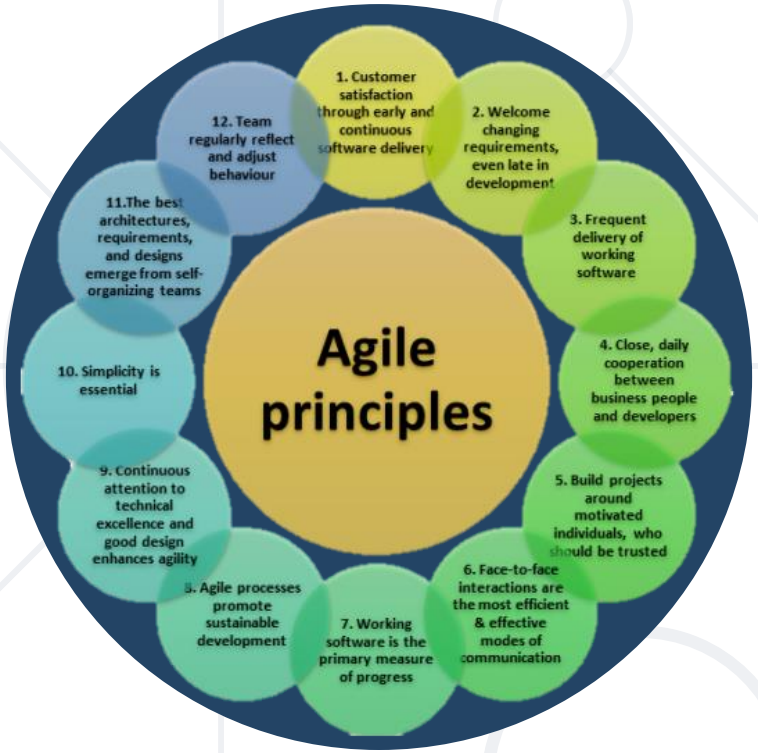
Software University

Traditional PM: A Defined Process

Software University

- Traditional waterfall development is a "defined process." A plan is **defined at the beginning** and precisely followed to the end.
- This assembly line approach **requires minimizing deviations** from plan to be successful.
- On average 65% of requirements change during software development causing **waterfall projects to have an 11% worldwide success rate** during 2011-2015.

(Jim Johnson, Standish Group, 2015)



Agile Principles

Agile Manifesto

12 AGILE PRINCIPLES

01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	02 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
04 Business people and developers must work together daily throughout the project.	05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
07 Working software is the primary measure of progress.	08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	09 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.

Principle #1

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Customer Satisfaction

- We work for a customer – internal or external one
- **Fail Early**, Fail often
  - Agile is not a way to prevent us from making errors – instead, it's a way of reducing the cost of those mistakes
    - Agile process is based on **iterations**
    - Continuous delivery and short feedback loops
    - Implement feedback and delight the **Customer**

Principle #2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Welcome Change

- Why is change an important factor?
  - Objectives are often a moving target
  - Constantly increasing speed across industries
  - It's (almost) never too late for change
  - The work cycle is also a learning curve – you can't foresee it all
  - Change might be expensive, or prevent you from getting anything done

Principle #3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Working Increment After Each Iteration

Software University

- Iterations no longer than 4 weeks
- Be aware
  - Agile maturity is key – company & stakeholders
  - Set realistic expectations
  - Good sizing and planning support the process

Principle #4

Business people and developers must work together daily throughout the project.

Communication and Collaboration

Software University

- Creates a common knowledge
- Developers get to know the business goal
- Business sees more about the technical background
- It means a better product at the end of the cycle
- It is often the best way to drive **innovation**
- **Team Collaboration**

Principle #5

Builds projects around motivated individuals.
Give them the environment and the support they need, and trust them to get the job done.

Motivation and Knowledge Workers

- At its most simple definition, a **knowledge worker** is someone whose job requires them to think for a living
  - Try not to micromanage
  - Motivation is different for different people
  - In Agile, the manager is a servant leader
    - Support the employees
    - Protects the employees
    - Communicates the company's goals
    - Is a caretaker

Principle #6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

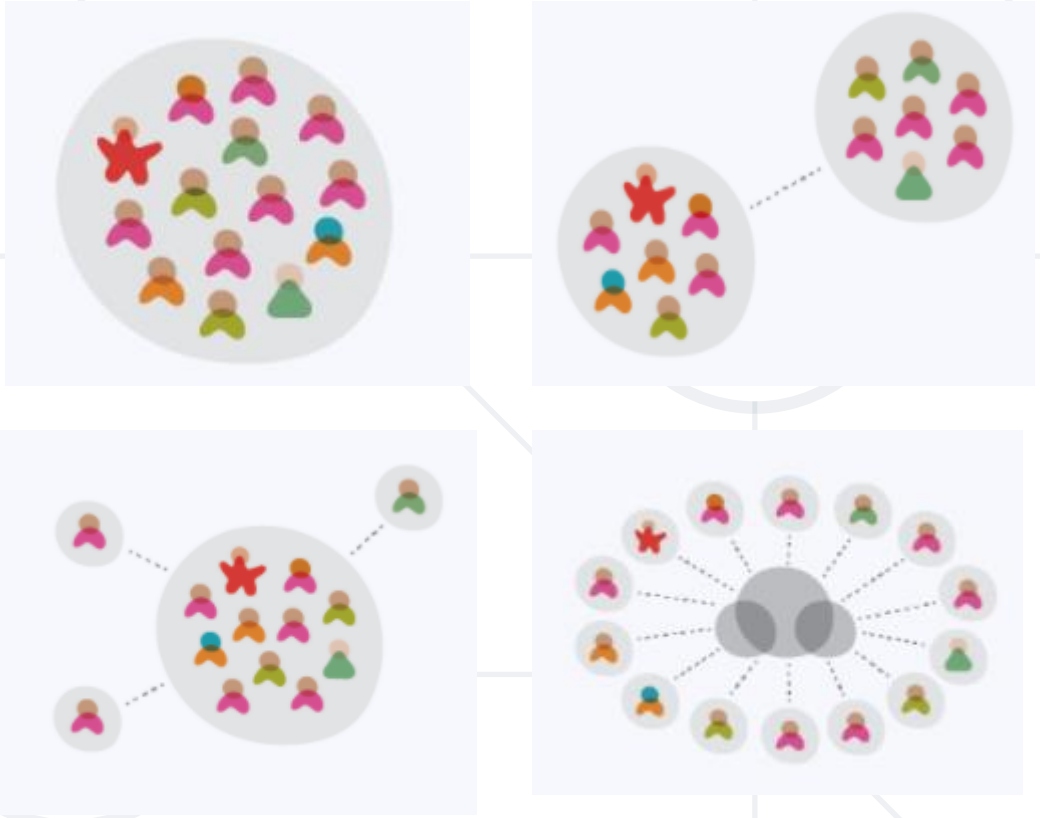
Agile Teams and Colocation

Software University

- Ideally the team should be collocated
- Team Size – 2 pizzas team
- Communication in Agile vs communication in traditional PM
- Remote Vs Onsite presence
  - Is "face to face" possible when the team is distributed?
  - Can we have "pair programming" in a distributed team?

Distributed Teams

- Types of distributed teams
  - Co-located – a single sited team
  - Multi-site team
  - Satellite workers
  - A remote-first
- Pros and Cons of distributed teams
- How to overcome the challenge?



Principle #7

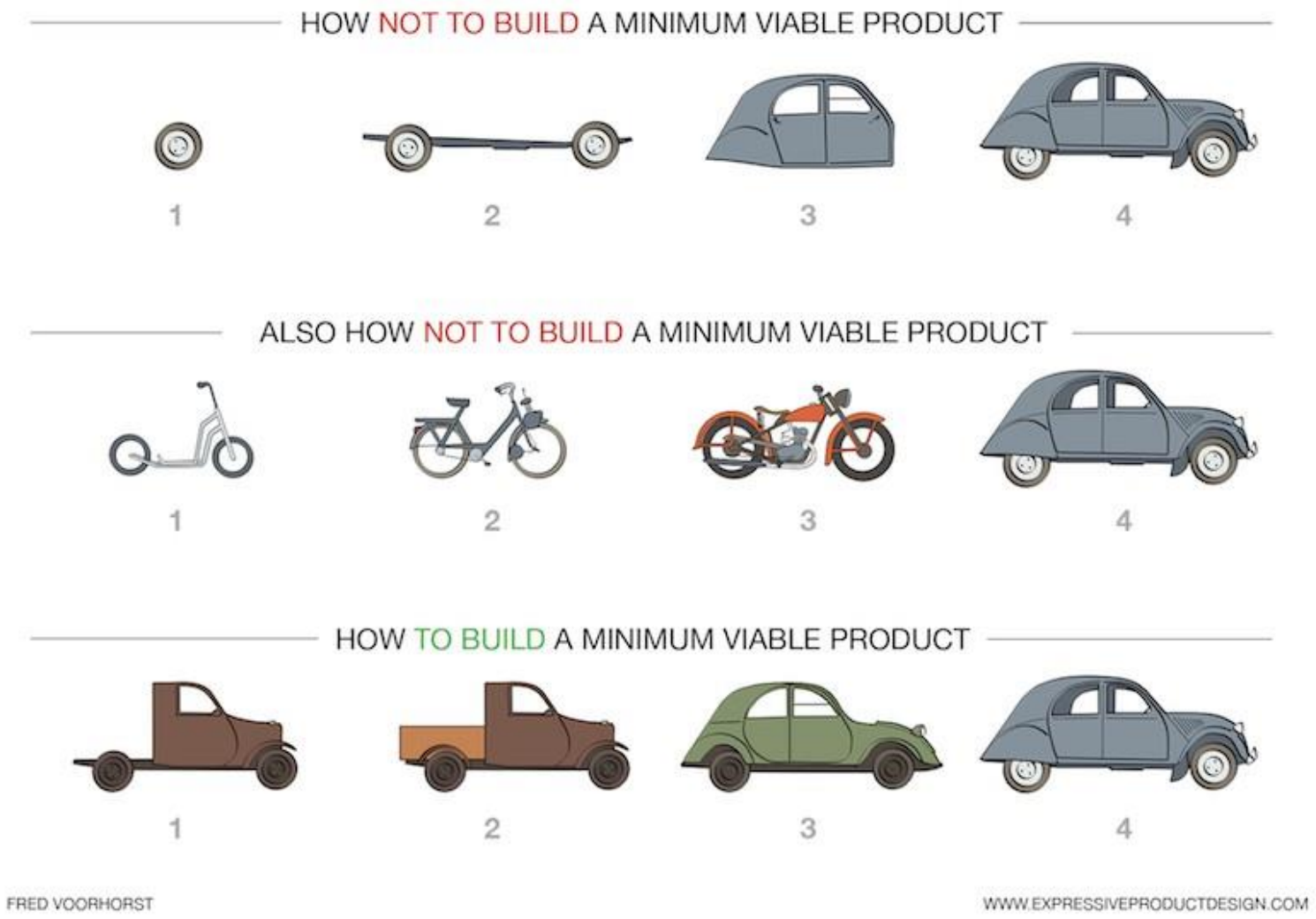
Working software is the primary measure of progress.

Working Software and MVP (1)

Software University

- Software is finished when it is tested and accepted by the (key) end-user
- Now the question is simple: the feature is either working or it's not!
- Every developer has a compulsion to write the most perfect software, but progress has to be made
- What's the easiest way to implement requirements to meet the goal?
- MVP = Minimum Viable Product

Working Software and MVP (2)



Working Software and MVP (3)

Principle #8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Sustainability

- Sustainable pace helps to plan and predict delivery
- Difference between negative and positive stress
- Trust the team to get the job done

Principle #9

Continuous attention to technical excellence and good design enhances agility.

Agile Principles

- In SW there is a difference between "good" design, and "complex" design
- The same is applicable for "Technical Excellence" and "Complex Technique"
- Use the right tools and frameworks
- If something does not work for the project, change it

Principle #10

Simplicity – the art of maximizing the amount of work not done – is essential.

Agile Principles

Software University

"Maximizing the amount of work not done includes minimizing the overhead for ensuring things that are necessary, but are not specifically what our customers want to pay for."

Principle #11

The best architectures, requirements, and designs emerge from self-organizing teams.

Self-Organized Teams

Software University

- What is self-organization?
- Micromanagement kills innovation
- Allow employees to show their full potential
- Reduced number of approvals
- Give space for learning new things
- Coach the team – self-organization doesn't happen overnight

Tuckman's Model

Principle #12

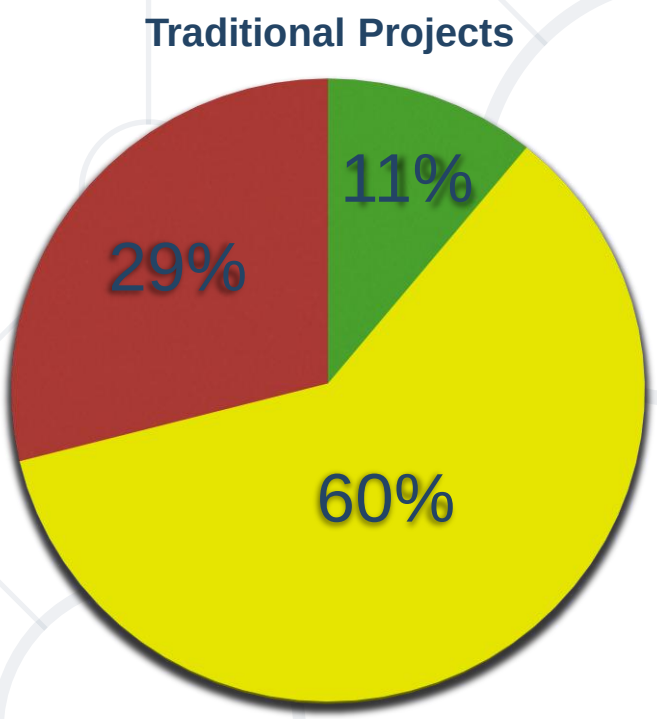
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

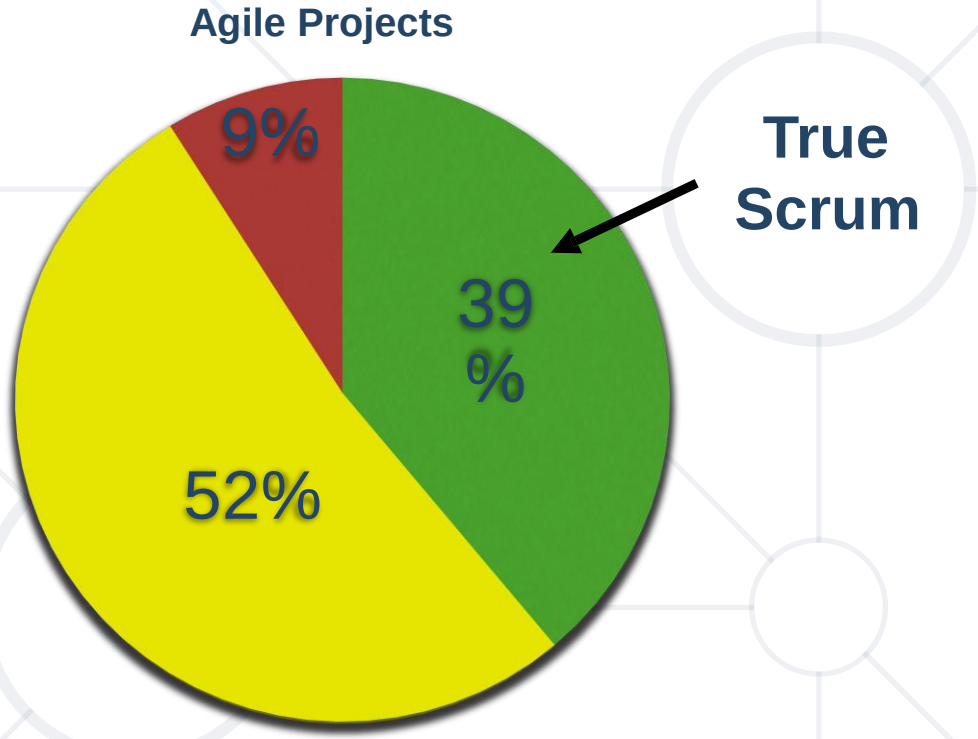
Reflect

- Reflect as a team
- Do it after each iteration
- Do it together with the customer
- It is the only way to become better
- Be honest…have courage…drive change if such is needed

Identifying the Gap

On Time Delivery

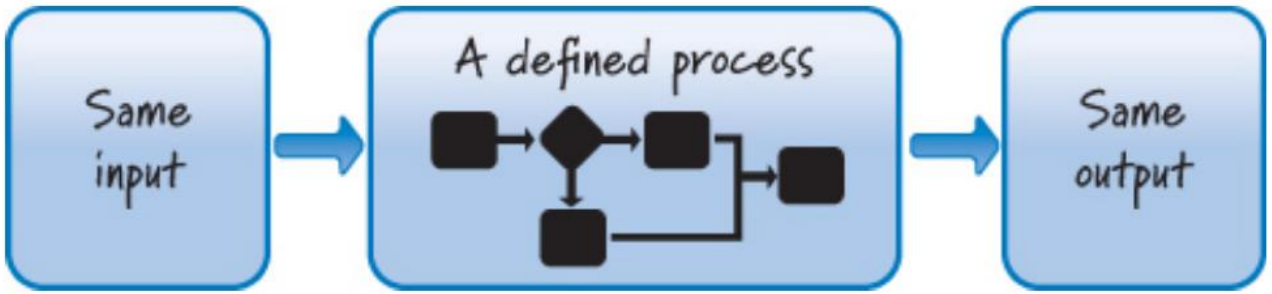


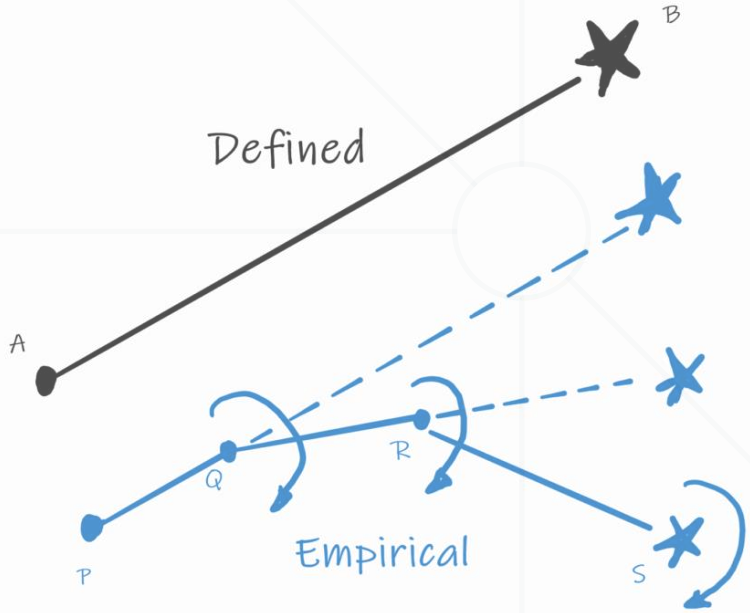


During the past decade, the worldwide success rate of software projects developed with empirical processes has been triple the success rate of defined projects. *(Jim Johnson, Standish Group, 2015)*

Defined vs Empirical Processes

- **Defined process**: Defines the constituent steps in advance, as in construction projects
- **Empirical process**: Iterative and incremental, as with new or uncertain projects



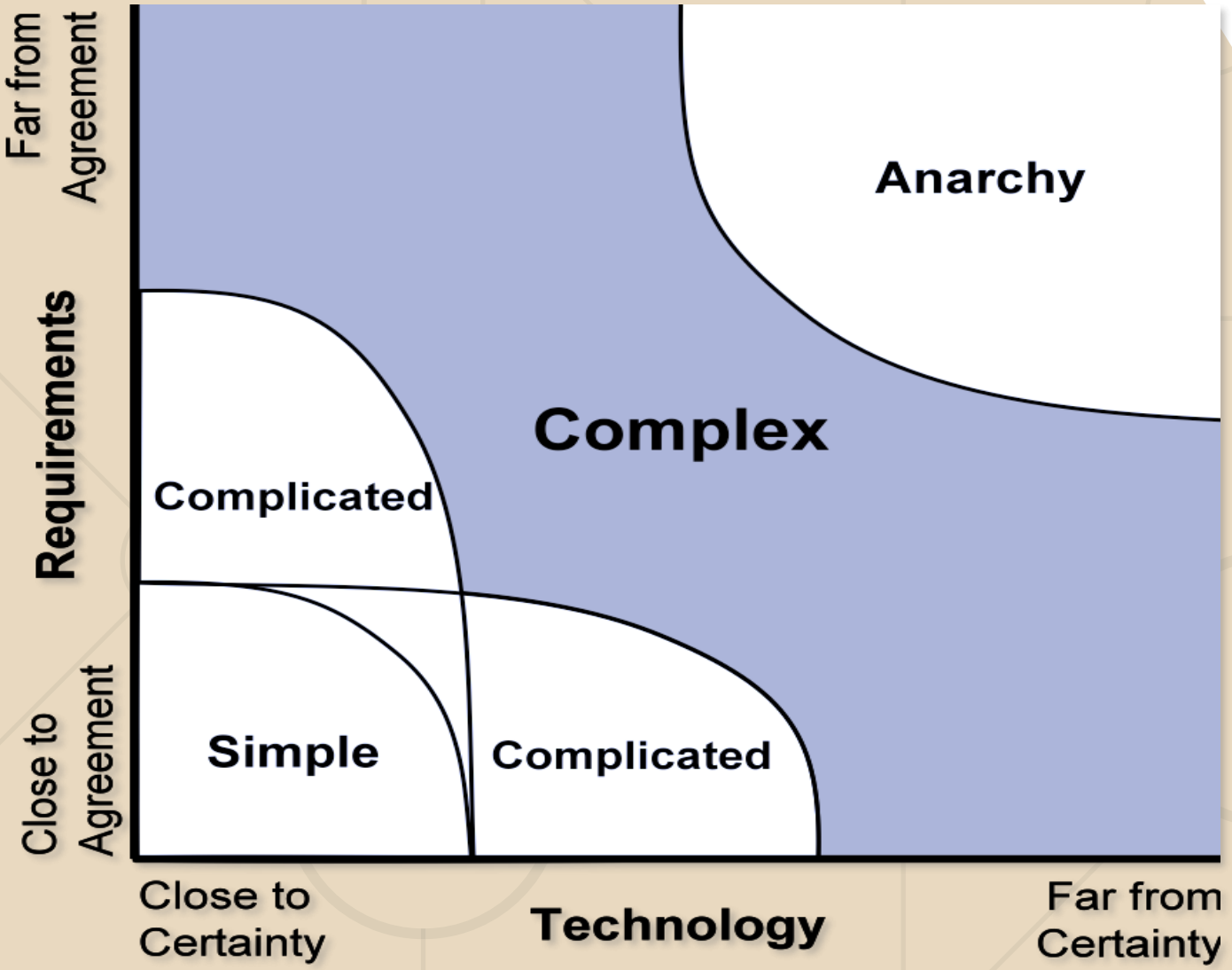


Management vs Leadership

- **Management**: Mechanical focus concerned with tasks, control, and speed
- **Leadership**: Humanistic focus on people and purpose, concerned with empowerment and effectiveness

All Work Falls Into This Spectrum

AGILE-EMPIRICAL PROCESS



- Now let's add people…
  - Culture
  - Time Zones
  - Language
  - Personalities
  - And there's more…

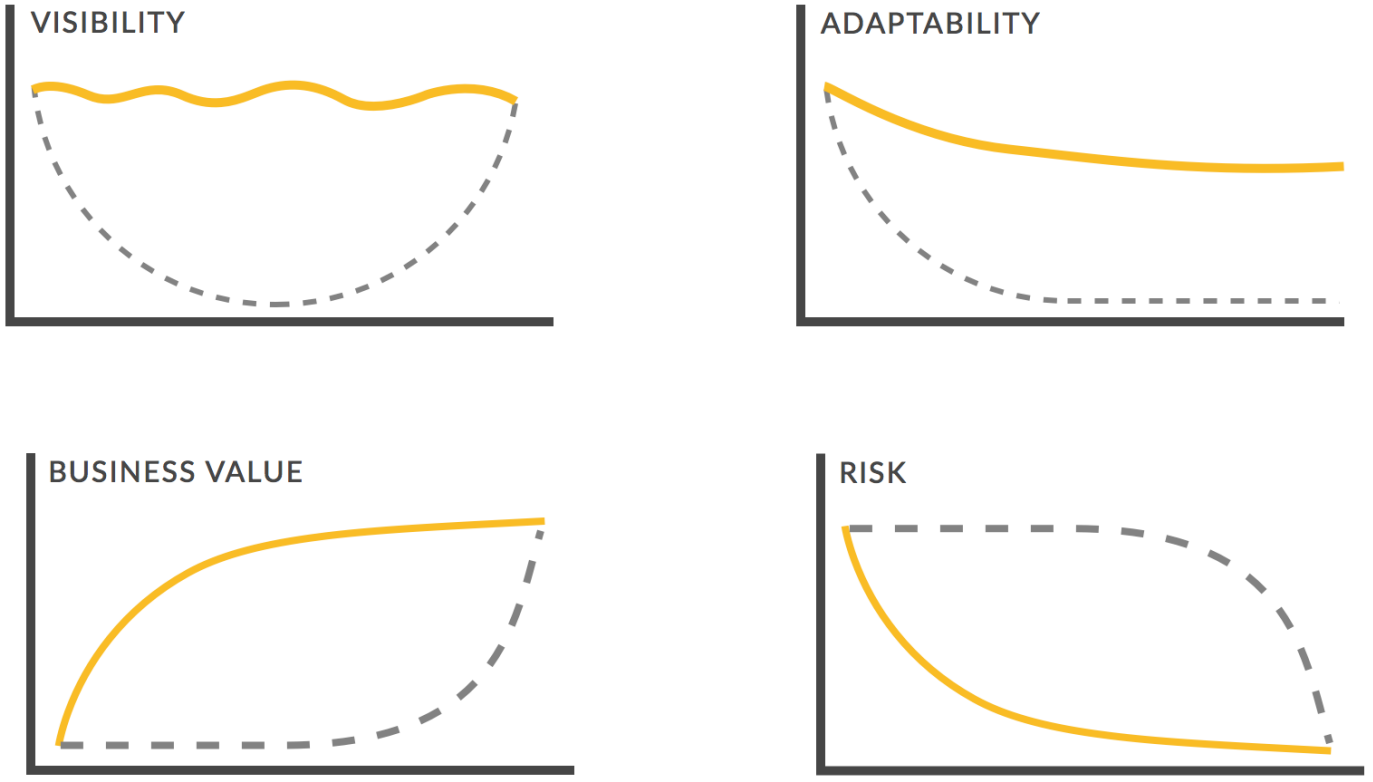
- Agile helps us take small pieces of upper-right and bring them towards lower-left

AGILE-SUCCESS

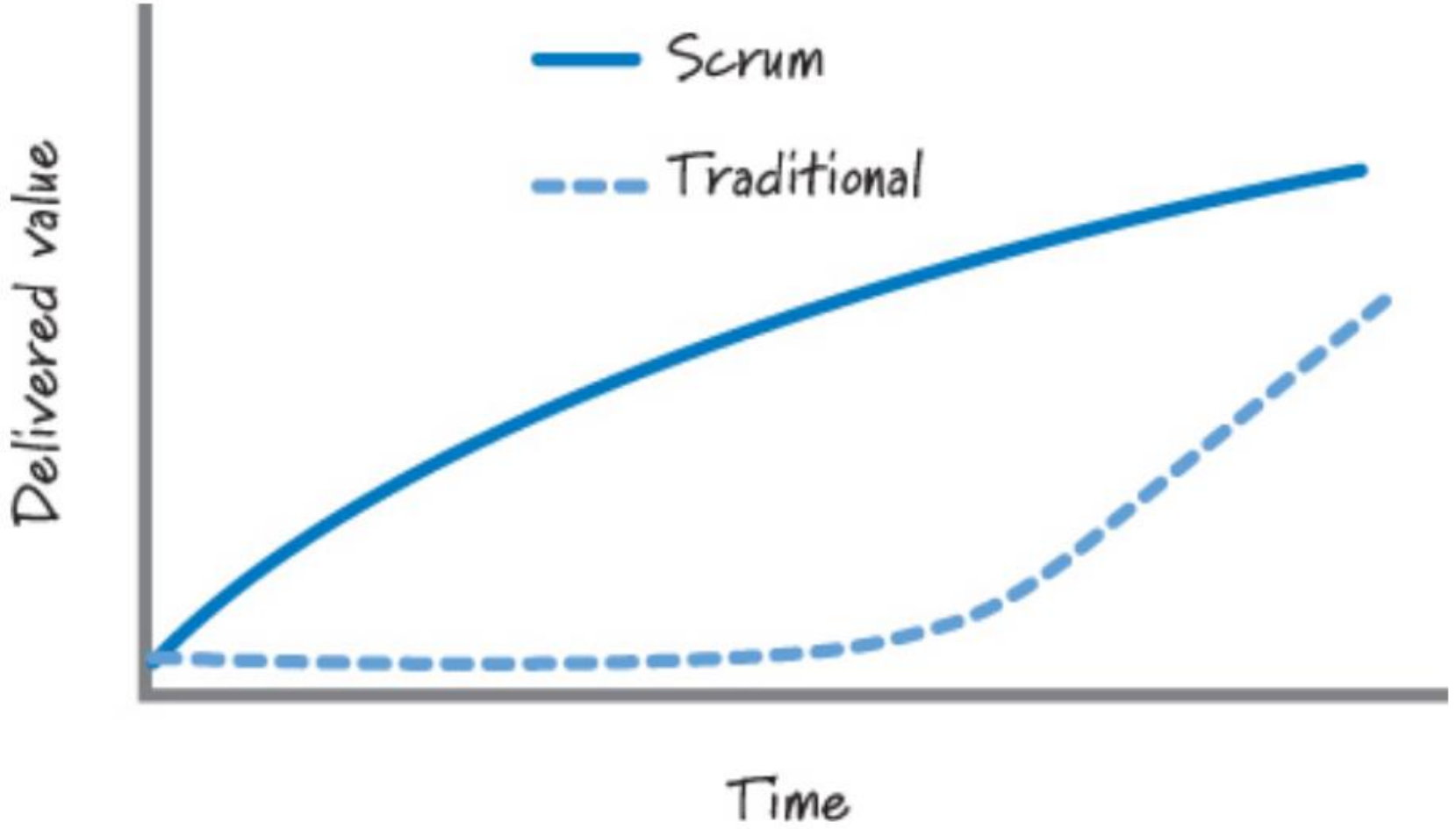
Software University

- Is that "success" where Agile started?
- Standish Group CHAOS
  - 15% Success
  - 170% Overrun

Agile – Value Proposition



Scrum Value Delivery



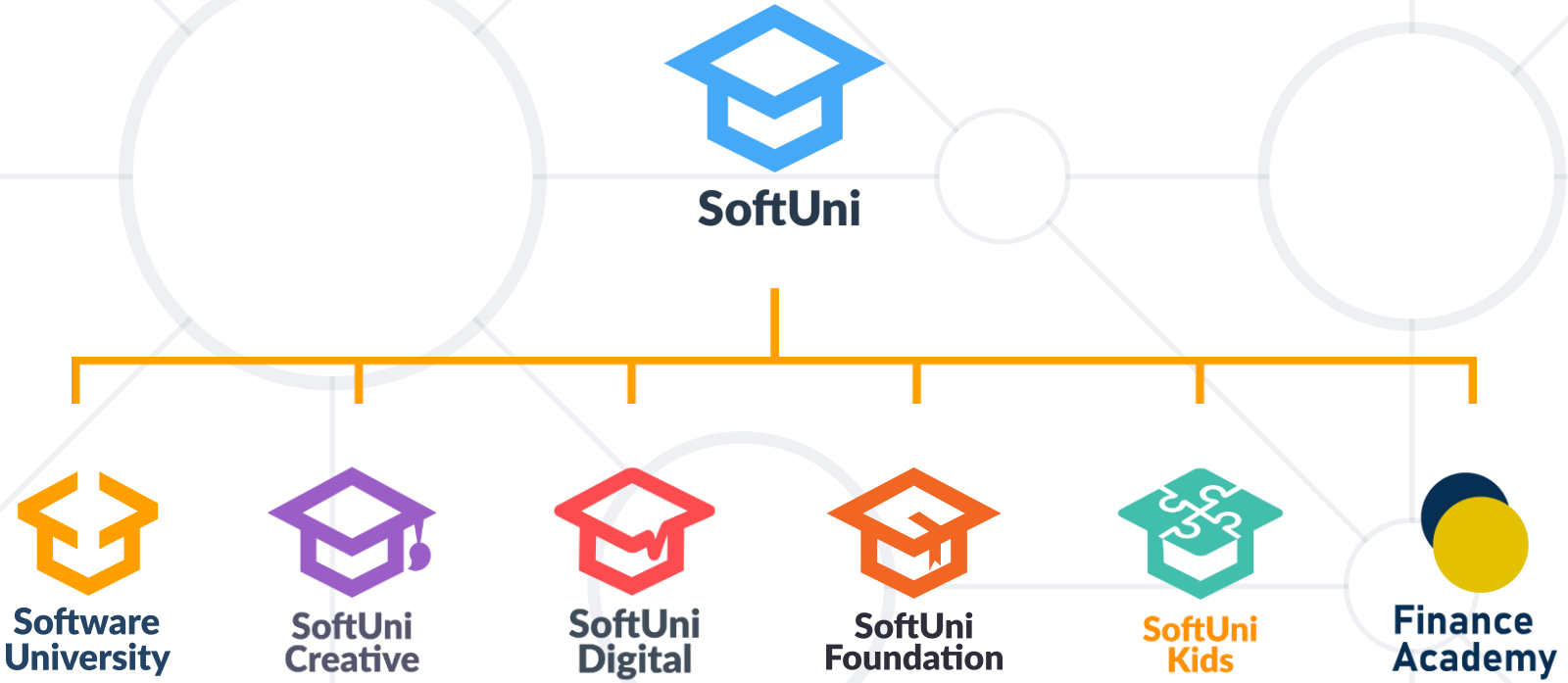
Homework and Additional Reading

- What is Agile?
- SCRUM Guide
- Minimum Viable Product (MVP) – Definition, Meaning and Examples
- Discussion topic – Have you seen any agile practices being used in traditional settings (i.e., non-agile environments)? Would they be useful?

Summary

- Agile is not a methodology that can be simply followed
- Whatever you do, keep the values and principles in mind
- Practices are there to support the values and principles of Agile
- Explore supporting practices and review resources online

Questions?



SoftUni Diamond Partners

Software University

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

Flutter™ International

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

BOSCH

Postbank

Решения за твоето утре

PHAR VISION

SmartIT

DXC TECHNOLOGY

createX

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg