

# Master's Degree Program In Software Engineering

## EngX Bootcamp

October 2024

# EngX Bootcamp

How to be agile and deliver high-quality code

1. Introduction
2. Foundations
  1. Introduction
  2. Agile fundamentals
  3. Engineering excellence culture
  4. EngX practices in agile-driven SD
3. Development
  1. Introduction
  2. Code quality
  3. Unit testing
  4. Code review
  5. Knowledge sharing
  6. Technical debt
  7. VCS and branching strategy
3. Quality assurance
  3. Introduction
  4. Fundamentals and overview
  5. Test case management
  6. Defect management
  7. Testing non-functional requirements
  8. QA metrics
  9. Automated testing
5. Systems Engineering
  5. Introduction and fundamentals
  6. CI/CD
  7. Infrastructure management
  8. Striving for engineering excellence
6. Secure SDLC
  5. Intro
  6. Secure SDLC
7. Summary and Discussion

# Introduction



“In today’s fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won’t get the job done.”

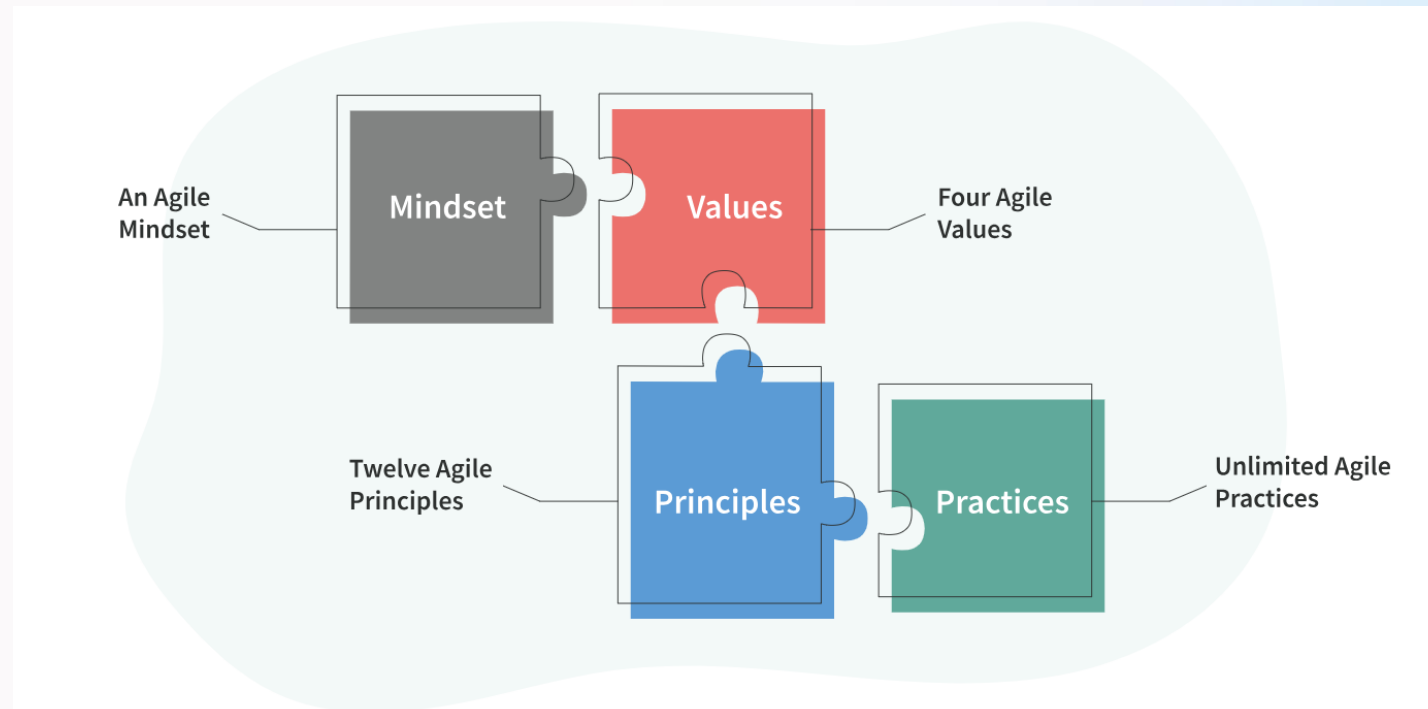
—Hiroataka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game,” *Harvard Business Review*, 1986

evolving from a **command-and-control** development approach to the **agile** one, software development teams become **more flexible**, **autonomous**, **self-organized**, and **prepared to face changing requirements**

# Agility

Agile software development is not about delivering the software  
it is about delivering the value

It is focused on people, adaptability, and providing maximum value to customers



There is no prescriptive set of actions or processes that define Agile development.  
It rather begins with a mindset and encompasses the Agile values and principles

# Agile Mindset

## Put Customer First

Examining and understanding the customer perspective enables everyone in a team to concentrate on one goal and understand why they do the work they do

## Trust in the Team

People can work independently and get their work done with no direct supervision and micromanagement

## Emphasize experimentation

With an Agile mindset, people value creativity and are eager to experiment and innovate even if it means taking a risk

# Agile Values

Agile Manifesto highlights 4 core values

We prefer, focus and value

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer value over contract negotiation

Responding to change over following the plan

# Agile Values

Individuals and interactions **over** processes and tools

Success is determined by the team you work with and how well you work together, **regardless** of how well-researched your method is or how high-tech your tools are

Working software **over** comprehensive documentation

Deliver **working software** – not documentation – into the customer's hands **as quickly as possible**

Customer value **over** contract negotiation

Even the most detailed contracts leave lots of room for misunderstandings about deliverables and overcomplicate the change-management process then **focus on customer-centric product development** where **customer feedback loop is built into the SDLC not on writing the contract**

Responding to change **over** following the plan

The product roadmap is not considered a static document but a dynamic strategy that allows the team to adjust its priorities and plans.

# Agile principles

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

**Welcome** changing requirements, **even late** in development

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

Businesspeople and developers **must work together daily** throughout the project

Build projects around motivated individuals and **trust them** to get the job done

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



# Agile principles

Working software is the **primary measure** of progress

Agile processes **promote sustainable** development

**Continuous attention** to **technical excellence** and **good design** enhances agility

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

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

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

# Agile ways of working

Work in **short iterations** and **focus on** the **what**, **how**, and **why**

**What are you doing**

Your goal is to support the customer's business

**How are you doing it**

Agile itself **is not** a **set of practices and techniques**, Agile **helps to identify** the **practices** and **techniques** that will help you to work **collaboratively** and **iteratively with the business** on the common goal

**Why are you doing it**

Knowing the goal helps your team move in the right direction and **continuously adjust** to **ensure that your direction is still valid**, and you are doing the right things.

# Waterfall vs Agile SDLC

# Waterfall SDLC

## Sequential work from start to finish

Clients typically see results **at the end** of the project life cycle.



# Agile SDLC

Iterations are usually **a few weeks long** and include the same phases as the Waterfall methodology, but clients see results **at the end of each iteration**

