

IT Project Management



by Sebastia Isaghulyan

About Me



- ❖ **Ex-software engineer, now Head of Lean-Agile Project Management Division at Softconstruct**
- ❖ **Bachelor of Engineering, Master of Business Administration**
- ❖ **18 years in IT**
- ❖ **10 years of IT Project management experience**
- ❖ **8 years Training experience**

by Sebastia Isaghulyan

Intro to Project Management

- ❖ What is a Project?
- ❖ Who is PM
- ❖ Project Management Lifecycle
- ❖ Software Development Lifecycle
- ❖ Leadership
- ❖ Era before Agile
- ❖ Waterfall
- ❖ Agile-Values and Principles
- ❖ Project/Product



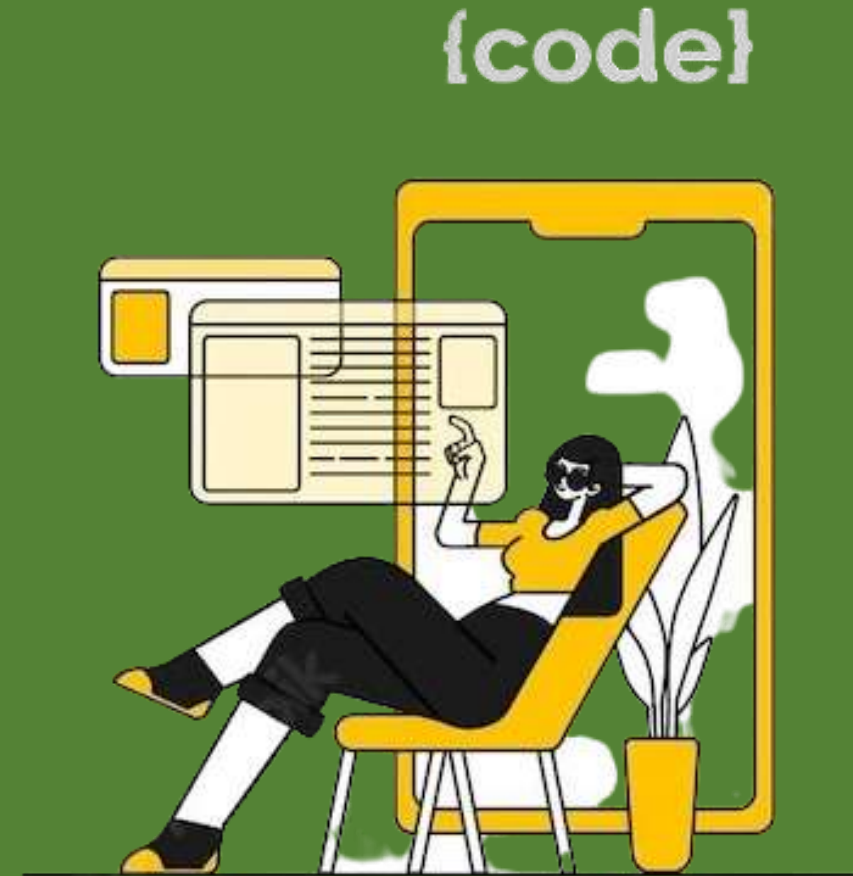
What is a Project?



by Sebastia Isaghulyan

Definition of project

A work or process to create a **unique** product, service, or result which has a **start** and **end**.



by Sebastia Isaghulyan

Example 1:Construction

`{code}`

➤ Townhouse Development

Description: Develop a series of attached townhouses, typically in a suburban or urban area.

by Sebastia Isaghulyan

Example 2: Education & Learning

`{code}`

➤ After-School Enrichment Program

Description: Create an after-school program that offers a range of educational activities, such as arts, music, to help students explore different subjects.

by Sebastia Isaghulyan

Example 3:IT

`{code}`

➤ E-commerce Website

Description: Develop a full-stack e-commerce site where users can browse products, add them to the cart, and proceed to checkout.

by Sebastia Isaghulyan

Example 4:IT

`{code}`

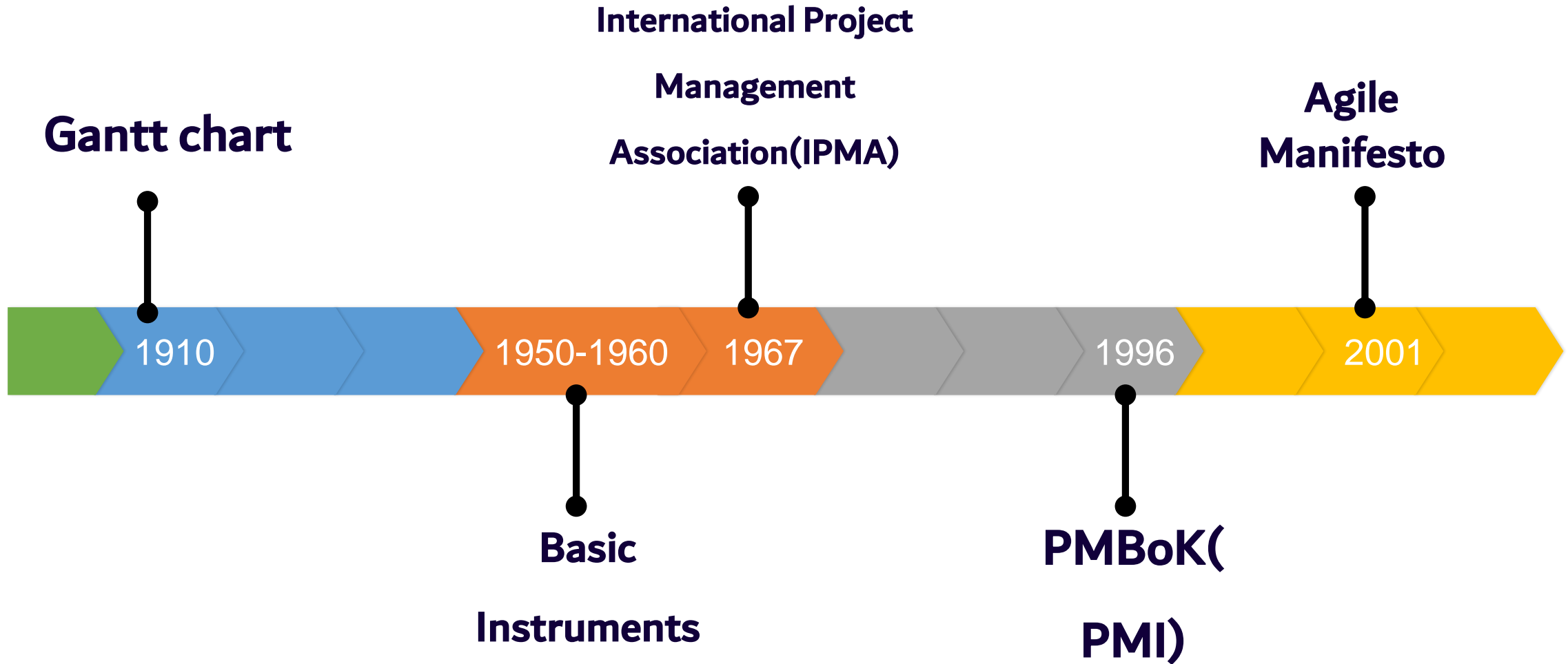
➤ AI-based Chatbot

Description: Develop a full-stack e-commerce site where users can browse products, add them to the cart, and proceed to checkout.

by Sebastia Isaghulyan

History

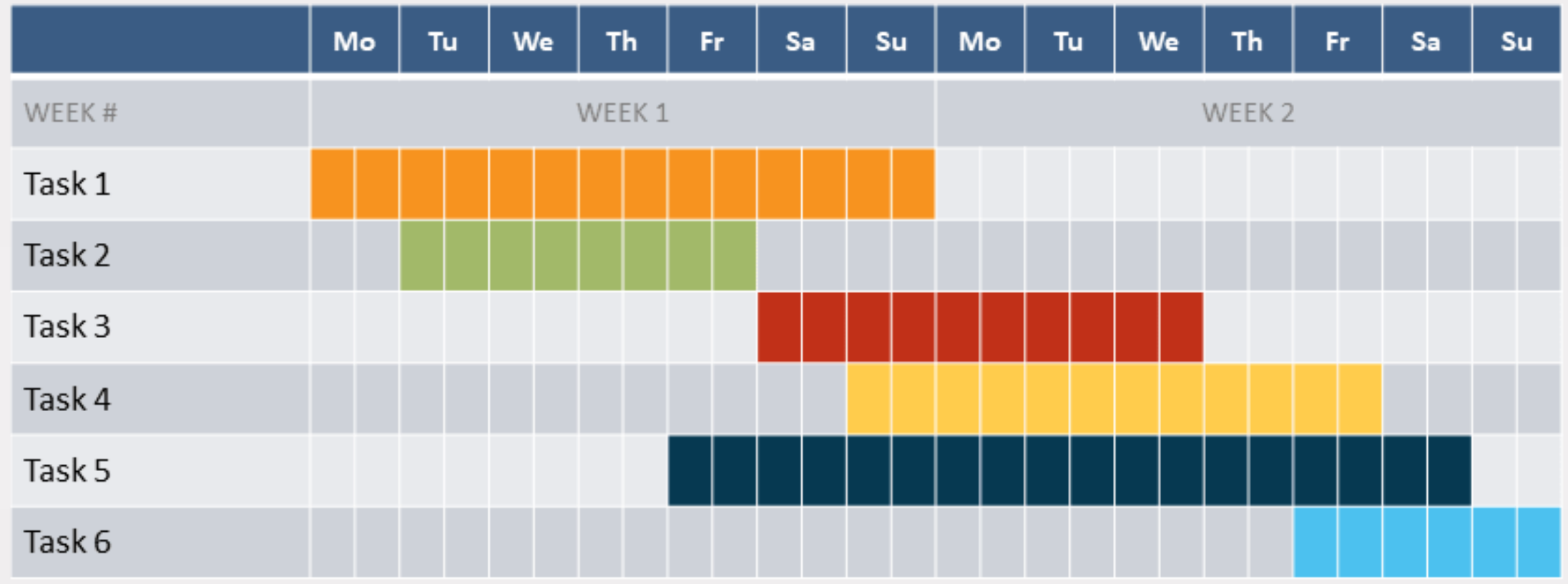
{code}



by Sebastia Isaghulyan

Gantt Chart

{code}



by Sebastia Isaghulyan

Why is Project Management relevant?



by Sebastia Isaghulyan

The product life cycle is too short. The time to develop a new product is limited by the market. Business has become limited in time.



by Sebastia Isaghulyan

Competition becomes global.



by Sebastia Isaghulyan

Quality requirements have increased.



by Sebastia Isaghulyan

**Products and technologies are getting
more complex.**



by Sebastia Isaghulyan

Multi-project environment.



by Sebastia Isaghulyan

Project Management is not an exact science.



You can experiment,experiment and experiment...



{code}

Definition of Empiricism

Empiricism is a theory that states that knowledge comes only or primarily from experience.



by Sebastia Isaghulyan

Project Limitations

`{code}`

- ❖ Cost
- ❖ Scope
- ❖ Time

- ❖ Risk
- ❖ Resource
- ❖ Quality

Who is PM?



by Sebastia Isaghulyan

PROJECT MANAGERS



What my friends think I do



What my coworkers think I do



What society thinks I do



What my boss thinks I do



What I think I do



What I really do

Who is PM?

A project manager is a professional responsible for planning, executing, and closing projects. Their primary goal is to ensure that a project is completed on time, within budget and meets its objectives. The role of a project manager can vary depending on the industry and the specific requirements of the project.



PM

Collaborations

- ❖ Project Team
- ❖ Stakeholders
- ❖ Business and Management
- ❖ External Teams

by Sebastia Isaghulyan

Stakeholder

An individual, group, or organization, who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project.



Hard Skills VS Soft Skills

- ❖ Hard skills are learned abilities acquired and enhanced through practice, repetition, and education. Hard skills are teachable abilities or skill sets that are easy to quantify.
- ❖ Soft skills are character traits and interpersonal skills that characterize a person's relationships with other people. Soft Skills are subjective skills that are much harder to quantify.

Hard Skills a PM should have

- ❖ Task Management tools
- ❖ Foreign languages
- ❖ Domain Skills

Soft Skills a PM should have

- ❖ Communication
- ❖ Problem Solving
- ❖ Time Management
- ❖ Decision Making
- ❖ Leadership
- ❖ Negotiation
- ❖ Organization
- ❖ Teamwork
- ❖ Creativity
- ❖ Flexibility

Why are not all IT projects successful?



by Sebastia Isaghulyan

Standish Group Statistic

	SUCCESSFUL	CHALLENGED	FAILED
2011	29%	49%	22%
2012	27%	56%	17%
2013	31%	50%	19%
2014	28%	55%	17%
2015	29%	52%	19%

by Sebastia Isaghulyan

IT projects failure reasons

- ❖ Business-IT Interaction Problems
- ❖ Project Management Problems

Business-IT Interaction Problems

- ❖ Inconsistency between IT strategy and business strategy or lack of IT strategy as such.
- ❖ Lack of understanding by specialists of the impact of their actions on the business.
- ❖ Business misunderstanding of information technology opportunities.
- ❖ The need to accelerate the pace of implementation

Project Management Lifecycle



by Sebastia Isaghulyan

Project Management Lifecycle



1. Project Initiation

The goal of the initiation phase is to define the project at a broad level in terms of what needs to be done and achieved in order for it to be successful.



by Sebastia Isaghulyan

2. Project

Planning

The Project Planning phase is most critical and requires complete diligence as it lays out the project's roadmap.



by Sebastia Isaghulyan

3. Project Execution

The execution phase involved actually carrying out the activities that are identified in the project plan.



by Sebastia Isaghulyan

4. Project Monitoring & Control

This phase involves keeping the project on track and ensuring that objectives and project deliverables are met.

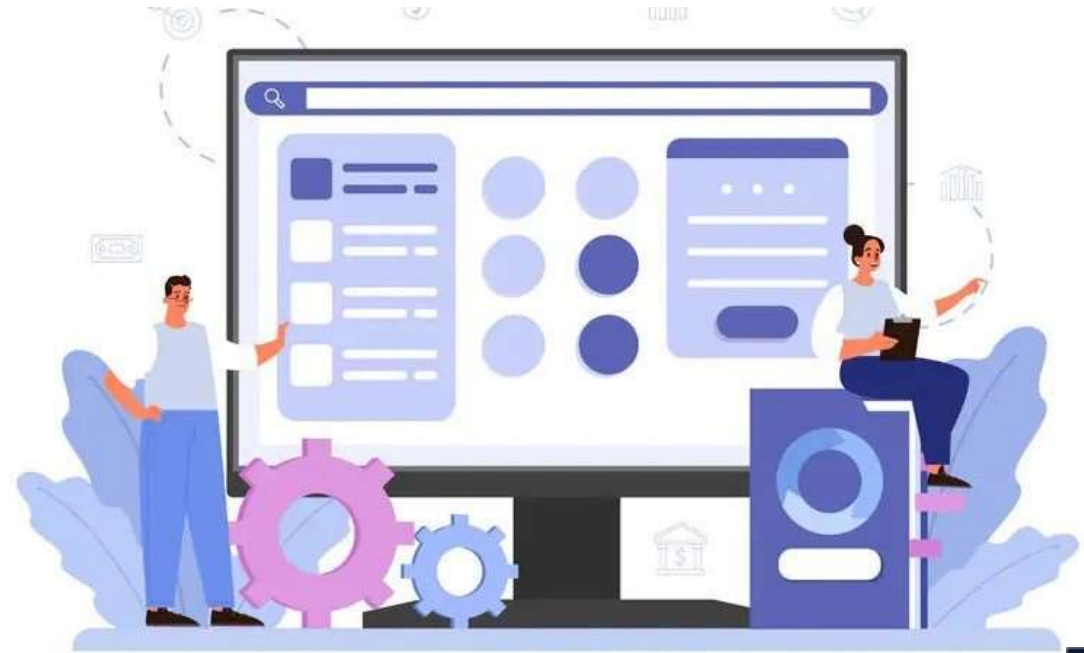


by Sebastia Isaghulyan

5. Project

Closing

Project Closing is the final phase of the project management life cycle, which indicates the end of the project and the final delivery of the project deliverables.



by Sebastia Isaghulyan

Project Management Process (Lifecycle)



Software Development Lifecycle



Definition of Leadership

Leadership is the ability to guide, inspire, and influence a group of individuals or an organization to achieve a common goal or purpose.



by Sebastia Isaghulyan

Key components of Leadership



by Sebastia Isaghulyan

Vision

Effective leaders often have a clear vision of where they want to take their team or organization. They can articulate this vision to inspire and motivate others.



by Sebastia Isaghulyan

Communication

Good communication skills are crucial for leaders. They must be able to convey their ideas, goals, and expectations clearly and effectively to their team.



by Sebastia Isaghulyan

Decision-Making

Leaders are often responsible for making important decisions that can impact the success of their group. They must be capable of making informed and timely choices.



by Sebastia Isaghulyan

Empowerment

Great leaders empower their team members by delegating tasks, trusting them to do their work, and providing the necessary support and resources.



by Sebastia Isaghulyan

Leading by Example

Leading through one's actions and behaviors is essential. Leaders should set a positive example for their team, demonstrating the values and work ethic they expect.



by Sebastia Isaghulyan

Adaptability

Being open to change and willing to adjust plans or strategies as circumstances evolve.



by Sebastia Isagulyan

Inspiration and Motivation

Leaders inspire and motivate their team members to work toward a common goal. This can involve recognizing and rewarding achievements, providing encouragement, and fostering a positive work environment.



by Sebastia Isaghulyan

Accountability

Leaders are accountable for the outcomes of their decisions and the performance of their team. They take responsibility for their actions and the results.



by Sebastia Isaghuayan

Who is a Leader?



by Sebastia Isaghulyan

21 OCTOBER 2020



MODRIC 54'
VINICIUS JR. 59'

29'
33'
43'

TETE
VARANE (OG)
SOLOMON

by Sebastia Isaghulyan



Zinedine Zidane
Real Madrid Coach

"I did not see the game as I wanted and when you do not see the game as you wanted it is your fault, I'm the coach and it's up to me to find the solution."

by Sebastia Isaghuilyan

BOSS

Says "I"

Takes Credit

Micromanages

Criticizes

Focuses on Weaknesses

Blames Others

Directs

Speaks More

Inspires Fear

Commands



LEADER

Says "We"

Gives Credit

Delegates

Encourages

Focuses on Strengths

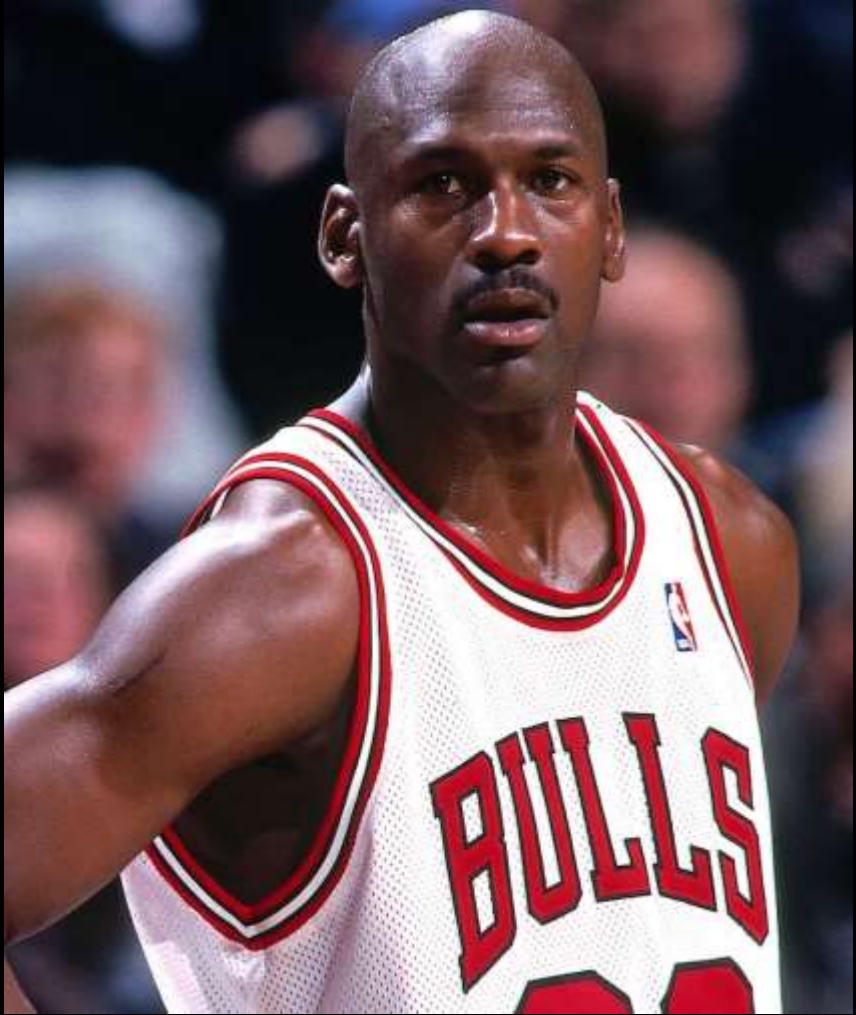
Takes Responsibility

Coaches

Listens More

Inspires Enthusiasm

Asks



*"Talent wins games,
teamwork win
championships."*

– Michael Jordan

by Sebastia Isaghulyan

`{code}`

Project management is a team sport!!!

by Sebastia Isaghulyan

Software Development Lifecycle

`{code}`



by Sebastia Isaghulyan

`{code}`

There are many ways to develop software.

by Sebastia Isaghulyan

Code & Fix Model

`{code}`

Code and Fix (often known as Cowboy coding) is considered the simplest development model. It is a cyclic process consisting of two major steps: Coding and Fixing.



by Sebastia Baghulyan

Code & Fix Model

`{code}`



by Sebastia Baghulyan

Code & Fix Model

`{code}`

The Model can Work Well for

- R&D activities.
- Experimentation, Observation, learning.
- Focusing on specific engineering challenges.
- Low-budget, easy short-term projects.
- Boosting engineering creativity.
- Individual contributors.
- Short-term outputs.

01

The Model has Problem with

- Dealing with product requirement analysis.
- Having a concrete definition of success.
- Quality assurance.
- Medium-sized and large engineering groups.
- Estimation of delivery work.
- Readability and maintainability of the code.
- Avoiding the upcoming technical debt.

02

by Sebastia Baghulyan

Waterfall Model

`{code}`

The waterfall model is a linear and sequential approach to the software development life cycle (SDLC). Waterfall is very easy to understand and use.



by Sebastia Baghulyan

Waterfall Model

`{code}`

- ❖ Each stage starts only when the previous one is completed.
- ❖ There is no overlap between different stages.



by Sebastia Baghulyan

The Process

`{code}`

1. REQUIREMENTS

by Sebastia Baghulyan

The Process

`{code}`

1. REQUIREMENTS

2. ANALYSIS

by Sebastia Baghulyan

The Process

`{code}`

1. REQUIREMENTS

2. ANALYSIS

3. DESIGN

by Sebastia Baghulyan

The Process

`{code}`

1. REQUIREMENTS

2. ANALYSIS

3. DESIGN

4. CODING

by Sebastia Baghulyan

The Process

`{code}`

1. REQUIREMENTS

2. ANALYSIS

3. DESIGN

4. CODING

5. TESTING

by Sebastia Baghulyan

The Process

`{code}`

1. REQUIREMENTS

2. ANALYSIS

3. DESIGN

4. CODING

5. TESTING

6. DEPLOYMENT

by Sebastia Baghulyan

When to use the waterfall model

`{code}`

- ❖ This model is used only when the requirements are very well known, clear and fixed.
- ❖ Product definition is stable.
- ❖ Technology is understood.
- ❖ There are no ambiguous requirements
- ❖ The project is short.

by Sebastia Isaghulyan

Disadvantages of waterfall model

{code}

- ❖ Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- ❖ No working software is produced until late during the life cycle.
- ❖ High amounts of risk and uncertainty.
- ❖ Not a good model for complex and object-oriented projects.
- ❖ Poor model for long and ongoing projects.
- ❖ Not suitable for the projects where requirements are at a moderate to high risk of changing

by Sebastia Isaghulyan

Do you have questions?



by Sebastia Baghulyan

Spiral Model

`{code}`

The spiral model is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the Waterfall model.



by Sebastia Baghulyan

Spiral Model

{code}



by Sebastia Baghulyan

Development Model Choosing Criteria.

- ❖ Business
- ❖ Project
- ❖ Team

Iterative Lifecycle

{code}

The iterative life cycle is a project management approach where work is divided into smaller, more manageable parts or iterations. Instead of completing the entire project in one go, it's broken down into stages or cycles that are revisited and refined over time.

Each iteration typically involves planning, execution, evaluation, and adjustment based on feedback received. This approach allows for flexibility and adaptation throughout the project's lifespan. It's commonly used in software development but can be applied to various other industries and projects.



by Sebastia Baghulyan

Benefits

{code}

- ❖ Continuous improvement
- ❖ Stakeholder involvement and feedback
- ❖ Respond to changes or new requirements

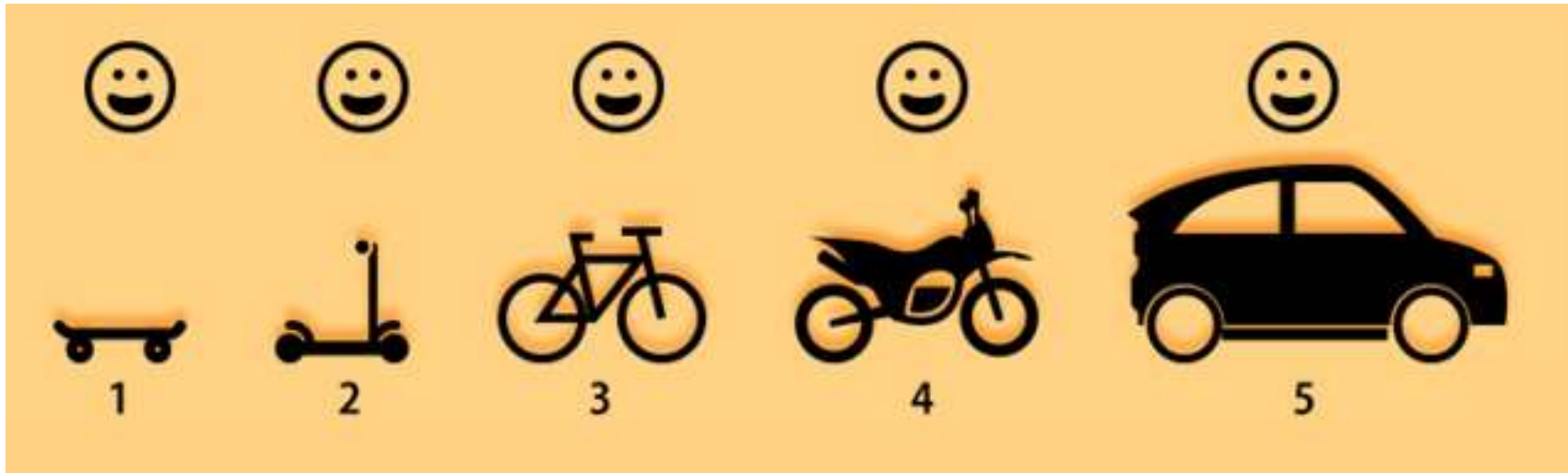


by Sebastia Baghulyan

Incremental Lifecycle

{code}

In an incremental life cycle, the project is divided into distinct, sequential increments or phases. Each phase represents a portion of the overall project scope and delivers a partial product or component. These increments are completed sequentially, with each subsequent phase building upon the previous one.

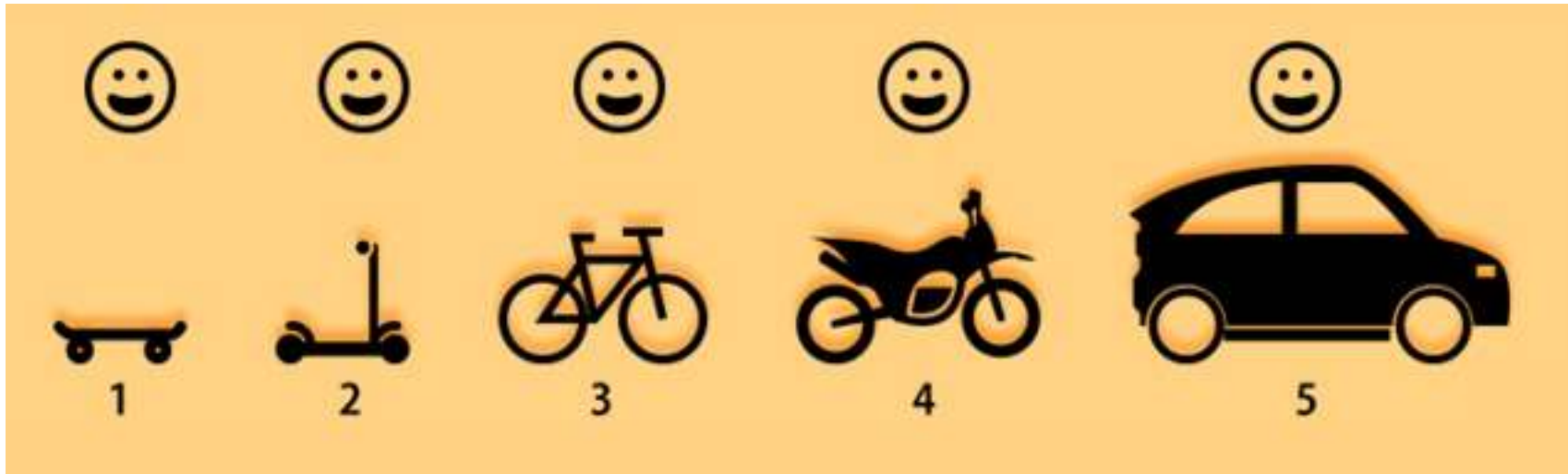


by Sebastia Baghulyan

Benefits

{code}

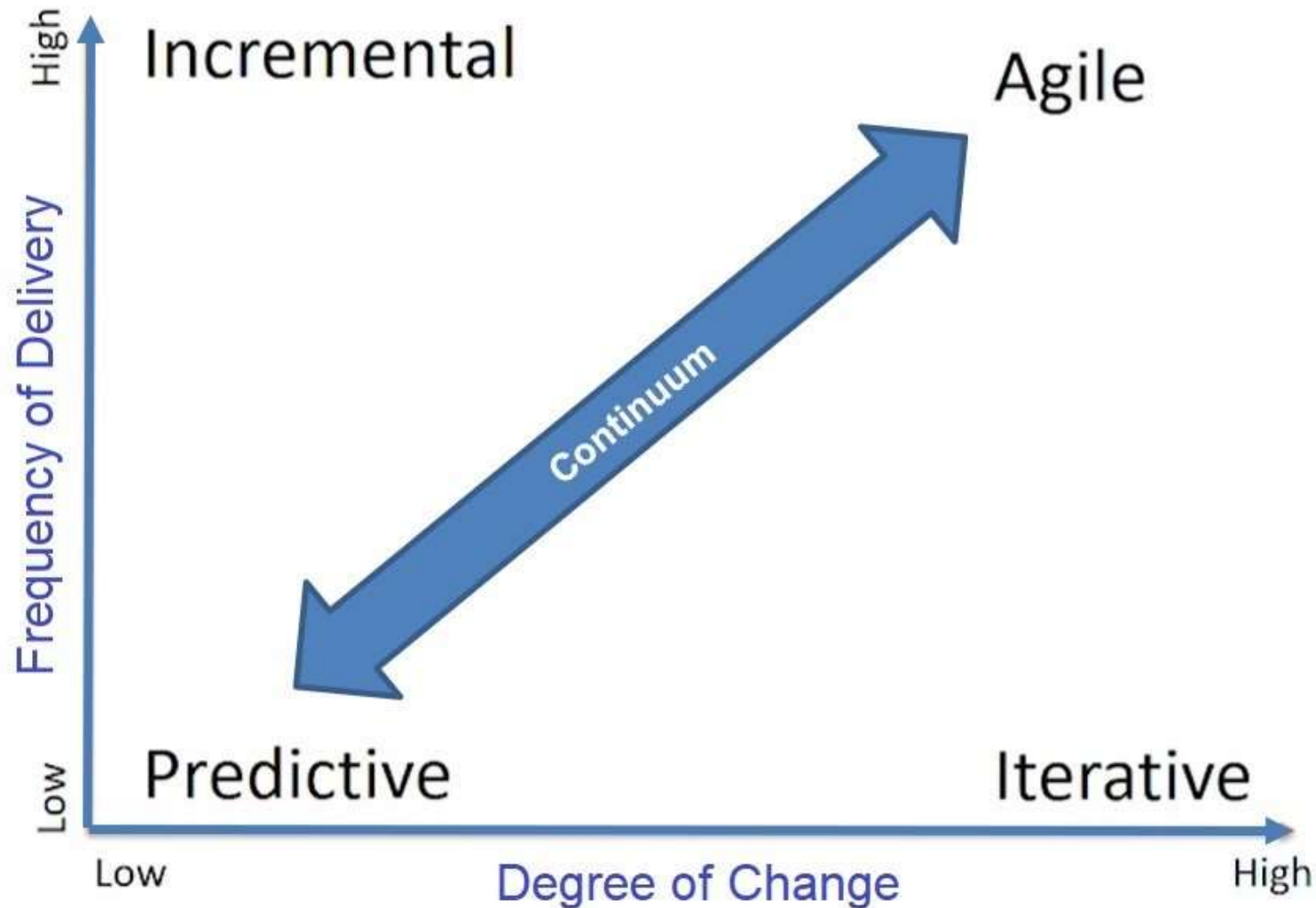
- ❖ Early delivery
- ❖ Allowing stakeholders to see tangible progress
- ❖ Ability to accommodate changing requirements in later increments
- ❖ Potential for more focused testing and feedback on specific functionalities



by Sebastia Baghulyan

Continuum of project Lifecycles

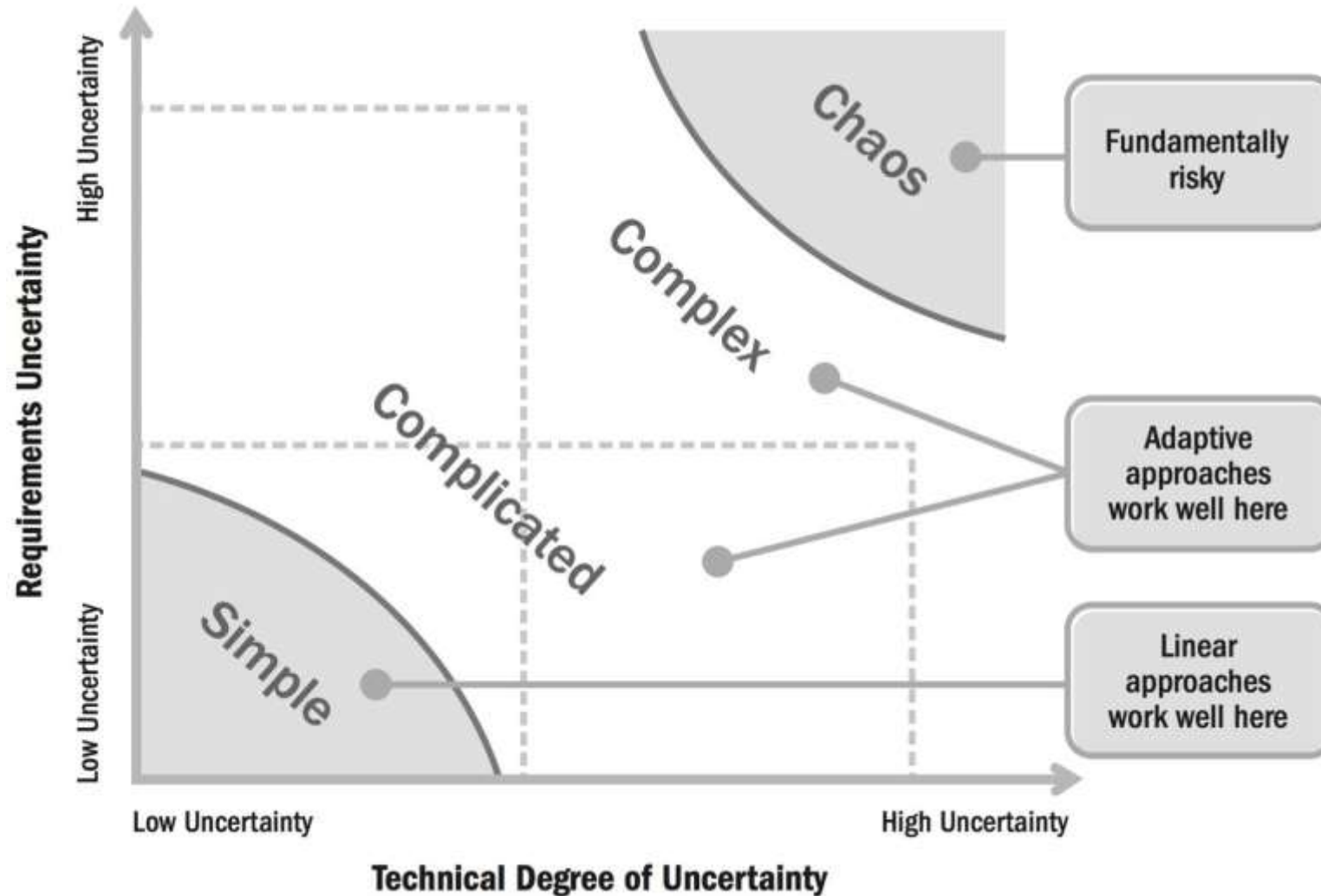
{code}



by Sebastia Baghulyan

Uncertainty and Complexity Model

{code}



by Sebastia Baghulyan

Cross-Functional teams

{code}

Cross-functional teams refer to groups of individuals with different functional expertise or skills working together to achieve a common goal or project.

Functional



Cross - Functional



by Sebastia Isaghulyan

Key features and benefits of cross-functional teams

- ❖ **Diverse Expertise:** Cross-functional teams include members with a variety of skills, knowledge, and backgrounds. This diversity can lead to more creative problem-solving and innovative solutions.
- ❖ **Faster Decision-Making:** With representatives from different functional areas present, decisions can be made more quickly as there is a range of expertise available to assess and address various aspects of the project.
- ❖ **Improved Communication:** Cross-functional teams encourage better communication and understanding between different parts of the organization. This can lead to increased transparency and a shared understanding of goals and challenges.
- ❖ **Increased Flexibility:** These teams are often more adaptable to changes in the project or external factors since they have a broader range of skills and perspectives to draw upon.
- ❖ **Holistic View:** Cross-functional teams can take a holistic approach to problem-solving, considering various aspects of a project or issue. This can lead to more comprehensive and effective solutions.
- ❖ **Enhanced Accountability:** Team members are collectively responsible for the success of the project, which can foster a sense of accountability and collaboration.
- ❖ **Employee Development:** Working in cross-functional teams provides opportunities for individuals to learn from others with different skills and experiences, promoting professional growth and development.
- ❖ **Customer-Centric Approach:** Cross-functional teams can focus on delivering value to the customer since they bring together individuals from different areas, including those directly connected to customer needs and feedback.

by Sebastia Isaghulyan

Self-Organized teams

Self-organized teams, also known as self-managed teams or self-directed teams, are groups of individuals within an organization that have the autonomy to manage their own work and make decisions without direct supervision. These teams are given the responsibility to plan, execute, and monitor their own tasks, with the goal of achieving specific objectives or project outcomes. The concept of self-organization is often associated with principles of empowerment, trust, and collaboration.



by Sebastia Isaghulyan

Key features and benefits of self-organized teams

- ❖ **Autonomy:** Self-organized teams have a high degree of autonomy in their decision-making processes. They are trusted to manage their work without constant oversight from a manager.
- ❖ **Collaboration:** Team members collaborate closely to share knowledge, skills, and responsibilities. This fosters a sense of shared ownership and accountability for the team's success.
- ❖ **Flexibility:** These teams are typically more flexible and adaptive to change. They can quickly respond to evolving project requirements or external factors without waiting for approval from higher levels of management.
- ❖ **Innovation:** Self-organized teams often encourage innovation and creativity. Team members, being close to the tasks at hand, can identify and implement creative solutions to challenges.
- ❖ **Increased Morale and Engagement:** Empowering individuals to take control of their work can lead to higher levels of job satisfaction, motivation, and overall engagement. Team members feel a sense of pride and ownership in their accomplishments.
- ❖ **Faster Decision-Making:** Decisions can be made more quickly because there is less hierarchy and bureaucracy involved. Team members are empowered to make decisions at the level closest to the task.
- ❖ **Continuous Learning:** Self-organized teams provide opportunities for continuous learning and skill development. Team members may take on different roles within the team, expanding their expertise.
- ❖ **Adaptability:** These teams are well-suited for dynamic and rapidly changing environments. They can adjust their strategies and plans in response to new information or shifts in priorities.
- ❖ **Reduced Micromanagement:** With a focus on self-management, there is less need for micromanagement from higher-level supervisors. This allows managers to focus on more strategic aspects of their role.
- ❖ **Customer Focus:** Self-organized teams often have a customer-centric approach, as they are closer to the actual work and can better understand and respond to customer needs.

by Sebastia Isaghulyan

What is the Agile Mindset?

An Agile mindset is a flexible way of thinking that enables people to react quickly and adapt to changing situations.



Time to market

`{code}`

- ❖ Time to market (also called TTM or time-to-market) is defined as the length of time from the conception of a product until it is released to the market.

by Sebastia Isaghulyan

The Crisis in 1990s

{code}

The companies had a big problem:
TTM is too long, and the process
takes years to validate customers'
needs and create the actual product.



by Sebastia Isaghulyan

Snowbird, Utah (2001)

`{code}`



by Sebastia Isaghulyan

Goals

`{code}`

- ❖ Fast Delivery
- ❖ Rapid Feedback



by Sebastia Isaghulyan

2 keywords

`{code}`

Continuous
Delivery

Continuous
Improvement

by Sebastia Isaghulyan

Agile Manifesto

`{code}`



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

by Sebastia Isaghulyan

Agile Manifesto

`{code}`

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

by Sebastia Isaghulyan

Agile Manifesto

`{code}`

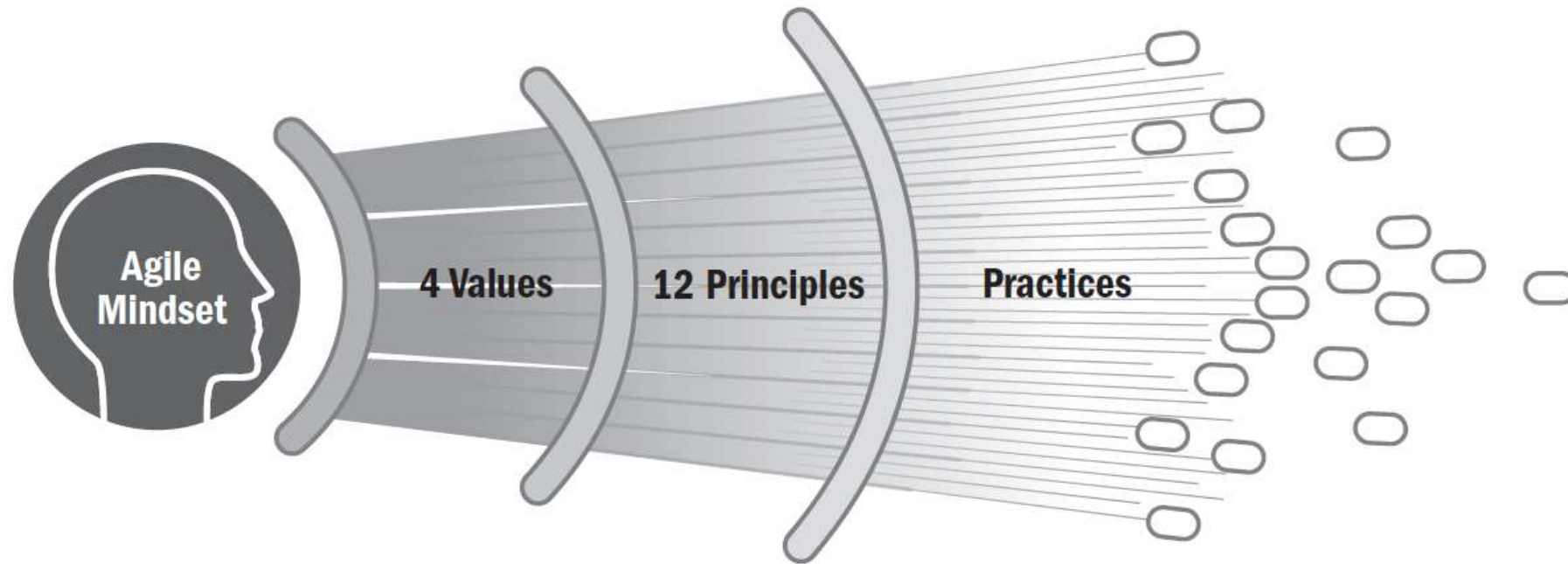
4
Values

12
Principles

by Sebastia Isaghulyan

Agile Manifesto

{code}



Agile is a mindset defined by values, guided by principles, and manifested through many different practices. Agile practitioners select practices based on their needs.

by Sebastia Isaghulyan

4 Values

`{code}`

- ❖ **Individuals and interactions** over processes and tools
- ❖ **Working software** over comprehensive documentation
- ❖ **Customer collaboration** over contract negotiation
- ❖ **Responding to change** over following a plan

by Sebastia Isaghulyan

Values

{code}



by Sebastia Isaghulyan

Values

`{code}`



by Sebastia Isaghulyan

Values

{code}



Customer Collaboration

over



Contract Negotiation

by Sebastia Isaghulyan

Values

{code}



Responding to Change

over



Following a Plan

by Sebastia Isaghulyan

4 Values

`{code}`

- ❖ **Individuals and interactions** over processes and tools
- ❖ **Working software** over comprehensive documentation
- ❖ **Customer collaboration** over contract negotiation
- ❖ **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

by Sebastia Isaghuilyan

Principle 1

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



Principle 2

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



Principle 3

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

`{code}`



by Sebastia Isaghulyan

Principle 4

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



Principle 5

Build projects around motivated individuals.

Give them the environment and support they need, and trust them to get the job done.

{code}



by Sebastia Isaghulyan

Principle 6

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



Principle 7

Working software is the primary measure of progress.

`{code}`



by Sebastia Isagulyan

Principle 8

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

`{code}`



by Sebastia Isaghuayan

Principle 9

Continuous attention to technical excellence and good design enhances agility.

`{code}`



by Sebastia Isaghulyan

Principle 10

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

`{code}`



by Sebastia Isagulyan

Principle 11

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

`{code}`



by Sebastia Isaghuayan

Principle 12

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

`{code}`



by Sebastia Isagulyan

{code}

Waterfall

VS

Agile

Specialists	People	Cross-functional
Linear	Process Flow	Cyclical
Fixed	Project Scope	Variable
Fixed	Project Cost	Variable
Upfront	Design	Just In Time
After Implementation	Verification	During Implementation
After Verification	Delivery	Continuous

by Sebastia Isaghulyan

Waterfall:Pros & Cons

{code}



PROS

- ✓ Thorough, clear documentation and planning
- ✓ Easy to implement and manage; fixed cost and predictability
- ✓ Strong, disciplined process
- ✓ Timely delivery; outcomes largely as expected
- ✓ Team changes are relatively easy to accommodate



CONS

- ✓ Less flexible; process changes cannot be accommodated
- ✓ Does not accommodate simultaneous workflows
- ✓ Delivery only at the end
- ✓ Cannot be implemented without sufficient clarity
- ✓ Considerable time spent on documentation

by Sebastia Isaghulyan

Agile:Pros & Cons

{code}



PROS

- ✓ Product goals can be defined with stakeholders
- ✓ Strong collaboration
- ✓ Customer feedback is encouraged
- ✓ Adaptive; changes can be accommodated
- ✓ Rapid, continuously improving output



CONS

- ✓ Requires considerable expertise and discipline
- ✓ Planning may be weak
- ✓ Timelines should be clear to keep things on track
- ✓ Requires dedicated resources
- ✓ Final product may be entirely different from expectations

by Sebastia Isaghulyan

Myths About Agile



by Sebastia Isaghulyan

Myth 1

Agile Is Only For Software Development

`{code}`



by Sebastia Isaghuayan

Myth 2

Agile Isn't For Highly
Regulated Environments

`{code}`



by Sebastia Isagulyan

Myth 3

Agile=Scrum

`{code}`



by Sebastia Isaghulyan

Myth 4

Agile Can't Be Scaled

`{code}`



by Sebastia Isaghulyan

Myth 5

Agile Has No Governance

`{code}`



by Sebastia Isaghulyan

Myth 6

`{code}`

There Is No Documentation With Agile



by Sebastia Isaghulyan

Myth 7

`{code}`

Agile Doesn't Do Planning



by Sebastia Isaghulyan

Agile SDLC

`{code}`



by Sebastia Isaghulyan

{code}

Project Product

by Sebastia Isaghulyan

{code}

Discovery Delivery

by Sebastia Isaghulyan

DIFFERENCES BETWEEN

A PRODUCT MANAGER

AND

A PROJECT MANAGER



Product Managers own the "Vision."



Product managers synthesize and communicate "strategy."



A product manager is responsible for feature definition.



A vital focus area for a product manager is "Content."



A product manager spends a lot of time on "Feature Conceptualization."



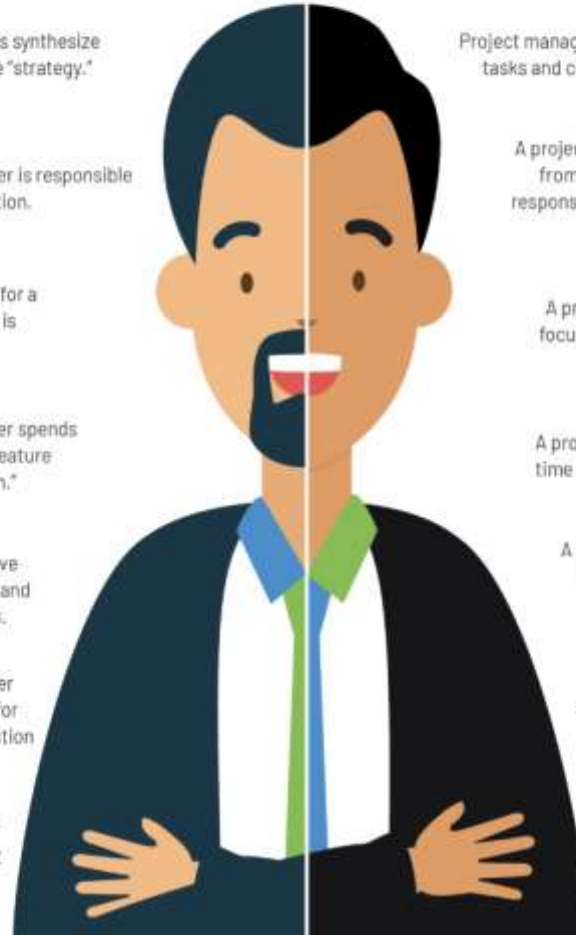
A product may have multiple projects and project managers.



A product manager tries to optimize for the form and function of the product.



Product manager creates a Product Roadmap.



Project managers own "Execution."



Project managers consolidate work tasks and communicate "status."



A project manager (with help from business analysts) is responsible for requirements definition.



A project manager's core focus is on "coordination."



A project manager spends time on "Feature Delivery."



A project manager may manage a project or projects that span multiple products.



A project manager strives for optimizing cost, speed, and quality.



Project Manager creates the "Schedule."

{code}

by Sebastia Isaghulyan

Do you have
questions?



by Sebastia Isaghulyan

Thank you



by Sebastia Isaghulyan