



Introduction to Agile Project Management

Main Focus

- The course explores **Project Management using the Agile Methodology**.
 - Agile aims to **create value early** and ensure a **measurable return on investment (ROI)** as soon as possible.
-

How Agile Works

- Projects are delivered in **small, iterative increments**.
- Each increment adds value to the product.
- **Changes are welcomed**, even in requirements — promoting flexibility and improvement.

 *Agile mindset:* “Deliver early, learn fast, and adapt quickly.”

Agile vs Predictive (Waterfall) Approach

Aspect	Agile	Predictive / Waterfall
--------	-------	------------------------

Approach	Iterative and flexible	Linear and sequential
----------	------------------------	-----------------------

Change Handling	Welcomes changes anytime	Difficult to change after planning
-----------------	--------------------------	------------------------------------

Delivery	Continuous, in small parts	Delivered only at project end
----------	----------------------------	-------------------------------

Goal	Early value creation	Final value after full completion
------	----------------------	-----------------------------------

Feedback	Ongoing	At the end only
----------	---------	-----------------

Recommended Background

If you've taken Dr. Bozek's earlier courses:

1. **Foundations and Initiation in Project Management**
2. **Project Management II – Planning and Execution**

→ You already know the **predictive (waterfall)** basics.

If you **haven't**:

- Do the **optional reading assignment** provided.
- It covers **predictive methods**, helping you understand **why Agile is different and better** in certain scenarios.

Key Takeaways

- Agile = Early ROI + Continuous improvement + Adaptability
 - Understanding predictive methods builds a solid base for learning Agile.
 - The optional readings are important for beginners new to project management.
-

Agile Project Management — Overview Notes

Why Agile?

- Technology is transforming every industry — faster delivery and adaptability are crucial.
 - Traditional (predictive/waterfall) models are slow and resist change.
 - Agile allows quick response, transparency, and customer satisfaction through early value creation.
-

What is Agile Project Management?

- A flexible, iterative approach that aims for early, measurable ROI.
 - Delivers products in small increments (sprints).
 - Encourages continuous customer involvement and frequent feedback.
 - Focuses on value creation early and often instead of waiting until project completion.
-

Agile vs Predictive (Waterfall)

Aspect	Agile	Predictive / Waterfall
Approach	Iterative, adaptive	Sequential, rigid
Scope	Uncertain, evolving	Well-defined, fixed
Customer Role	Continuous involvement	Minimal after requirements phase
Change	Welcomed anytime	Avoided (scope creep feared)
Delivery	Frequent, small releases	One big release at end

 Agile = Exploratory and flexible
Predictive = Planned and fixed

Why Agile Works Today

- Designed for **uncertainty**, **speed**, and **innovation**.
 - Perfect for **technology-driven** markets where needs change rapidly.
 - Enables **faster delivery** and **quicker market response**.
-

History of Agile

- **2001:** 17 software engineers met to improve software delivery.
- Created the **Agile Manifesto**, focusing on faster development and feedback loops.

Four Core Values of the Agile Manifesto

1. **Individuals & interactions** over processes & tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

 *Agile values flexibility, teamwork, and customer-driven progress.*

Agile as a Process

1. Start with a **vision** (problem/opportunity).
 2. Break vision into **small increments** called **sprints** (1–4 weeks).
 3. Work repeatedly through **multiple sprints** until project completion.
 4. After each sprint → **review, adapt, and improve**.
-

Key Agile Components

1. Product Backlog

- Prioritized “to-do” list of features, fixes, and functions.
- Continuously updated based on feedback.

2. User Story

- Small, simple description of a feature from the **end-user's perspective**.
- Focuses on how it **delivers value** to the customer.

3. Sprint (or Scrum)

- A short cycle (~2 weeks) where a small part of the project is developed.
- Each sprint includes:
 - **Sprint Planning** – What to do this sprint

- **Daily Scrum** – Short daily progress meeting
 - **Sprint Review** – Present work to customer
 - **Sprint Retrospective** – Reflect and improve for next sprint
-

Customer Role in Agile

- The **customer is part of the team**.
 - Gives **continuous feedback** on backlog, user stories, and deliverables.
 - Ensures product evolves to match real needs.
-

Core Agile Principles

- Deliver **value early and often**
 - Embrace **change** rather than resist it
 - Ensure **open communication**
 - Prioritize **working results** over heavy documentation
 - Focus on **continuous improvement** through iterations
-

In Short

Predictive Model

Agile Model

Plan → Design → Build → Test → Deliver Vision → Sprint → Review → Repeat

Fixed plan and scope

Flexible and adaptive

Customer reviews at end

Customer feedback throughout

Slower to adapt

Rapid and iterative

Agile vs Waterfall (Predictive) vs Hybrid

1. Predictive / Waterfall Model

- **Sequential process** — each phase (plan → design → build → test → deliver) happens **one after another**.
- **Plan first, execute later.**
- **Changes are difficult** once the project starts.
- Best suited for projects with **clear requirements** and **low uncertainty** (e.g., construction, manufacturing).

- ◆ **Pros:** Clear structure, easy to manage progress.
 - ◆ **Cons:** Slow to adapt; customer feedback comes too late.
-

⚡ 2. Agile Model

- **Iterative and flexible** process — work is done in **small cycles (sprints)**.
 - Focuses on **early delivery, continuous feedback**, and **constant improvement**.
 - Welcomes **changes anytime**, even late in development.
 - Best for projects with **evolving requirements** (e.g., software, startups).
- ◆ **Pros:** Quick feedback, customer satisfaction, adaptable.
- ◆ **Cons:** Harder to predict cost/time, requires active customer involvement.
-

⌚ 3. Hybrid Model (Agile + Waterfall)

- **Combines strengths** of both Agile and Waterfall.
 - Commonly used in **projects involving both hardware and software**.
 - Example:
 - Hardware part → **Waterfall** (fixed design, testing)
 - Software part → **Agile** (iterative updates, flexibility)
- ◆ **Pros:** Balanced flexibility and structure.
- ◆ **Cons:** More complex coordination between teams.
-

🧠 Quick Comparison Table

Feature	Waterfall (Predictive)	Agile	Hybrid
Process Flow	Linear, step-by-step	Iterative, flexible	Mix of both
Change Handling	Hard to manage	Welcomed anytime	Controlled flexibility
Customer Involvement	At start & end	Continuous	Moderate
Delivery	Single final delivery	Frequent small releases	Both partial + final
Best For	Fixed-scope projects	Dynamic-scope projects	Mixed hardware/software

💡 Key Takeaway

- **Waterfall** = Structured & predictable

- **Agile** = Adaptive & customer-driven
 - **Hybrid** = Practical balance for mixed or complex projects
-

Agile is a Mindset

More Than Just a Method

- Agile isn't just a **set of tools or processes** — it's a **way of thinking and working**.
 - It focuses on "**being agile**", not just "**doing agile**."
-

What It Means to Have an Agile Mindset

- **Embrace change** instead of resisting it.
 - **Focus on delivering value** continuously.
 - **Collaborate openly** with team members and customers.
 - **Learn and adapt quickly** from feedback.
 - Believe that **improvement is ongoing**, not a one-time step.
-

Being Agile vs. Doing Agile

Doing Agile (Process)

Following steps & tools (Scrum, Kanban) Thinking flexibly & valuing collaboration

Checking boxes in the process

Focusing on outcomes & value

Rigidly following framework rules

Adapting to the situation

Managing tasks

Empowering people

Key Idea

Agile success comes not just from **using** the process — but from **thinking** and **acting** with agility in every decision.

1. Agile Project Management approaches aim to accomplish early return on investment for customers by: (Select all that apply)

1 / 1 point

involving the customers in the development cycle.

Good job. Customers are involved in the development.

iterative delivery of the products.

Correct. Agile aims to create value through incremental deliveries or releases.

well-defined processes that define project outcomes early.

2. Which project management methodology offers a documented and prescriptive approach to planning? (Select best answer)

1 / 1 point

Agile Project Management

Lean Project Management

Waterfall Project Management

Correct. Waterfall or predictive project management favors planning and follows a strict planning process.

Kanban Approach

3. You are managing a complex project and anticipate the customer's needs and requirements to change as the project progresses. Which approach would be most appropriate for your particular project?: (Select best answer)

Predictive Project Management

Waterfall Project Management

Agile Project Management

Excellent. The agile approach is appropriate for changing customer requirements.

4. The agile mindset is built around the concept of: (Select best answer)

1 / 1 point

Delighting the Customer

Co-creating with the customer

Embracing failure

All the above

Excellent. The agile mindset embraces all of these concepts.

Agile Manifesto

Origin

- **Year:** 2001
 - **Place:** Snowbird Ski Resort, Utah
 - **Who:** 17 software developers (called themselves **The Agile Alliance**)
 - **Why:** They wanted an **alternative to the rigid, document-heavy “waterfall” approach** — a faster, more flexible way to develop software.
-

Purpose

To create a **lightweight framework** that focuses on:

- **People over processes**
- **Results over paperwork**
- **Adaptability over strict planning**

This framework became known as the **Agile Manifesto**, consisting of **4 Core Values** and **12 Principles**.

◆ The 4 Core Values of the Agile Manifesto

Agile Values	Meaning
1. Individuals and interactions over processes and tools	People and teamwork matter more than strict rules or software tools.
2. Working software over comprehensive documentation	Deliver something that works — not just detailed plans or reports.
3. Customer collaboration over contract negotiation	Work <i>with</i> the customer continuously, not just by following a signed contract.
4. Responding to change over following a plan	Be flexible and adapt when things change, instead of sticking rigidly to a plan.

 *Agile doesn't reject the right side (plans, tools, documents) — it simply values the left side more.*

The Founding Developers (The Agile Alliance)

Some key members who created the Agile Manifesto:

- **Kent Beck** (Extreme Programming)

- **Mike Beedle**
- **Alistair Cockburn**
- **Martin Fowler**
- **Ron Jeffries**
- **Ken Schwaber**
- **Jeff Sutherland**
(and 10 others)

These pioneers shaped the foundations of **modern Agile frameworks** like **Scrum** and **Extreme Programming (XP)**.

Key Takeaway

The Agile Manifesto marked a **shift from rigid planning to adaptive collaboration** — focusing on **people, working results, and flexibility** as the heart of project success.

Four Foundational Values of Agile Project Management

Agile is built upon **four key values** from the **Agile Manifesto**, which focus on flexibility, collaboration, and value creation.

1 Individuals and Interactions > Processes and Tools

- People are the most important part of any project.
- Tools and processes are helpful, but they cannot replace **communication, teamwork, and creativity**.
- When teams focus on people, they respond to change better and deliver what the customer actually needs.
- **Agile empowers teams** to make decisions and develop their skills to thrive.

 *Key idea:* Great people create great results — not rigid processes.

2 Working Software > Comprehensive Documentation

- Traditional (Waterfall) methods emphasize heavy documentation — detailed plans, reports, and specifications before building anything.
- Agile focuses on **delivering working outputs early**, rather than spending too much time documenting.
- Documentation still exists in Agile, but it's **lightweight and only as needed**.

- The goal is to **create value for the customer quickly** with real, working deliverables.

💡 *Key idea:* A functional product is more valuable than pages of documentation.

3 Customer Collaboration > Contract Negotiation

- In traditional models, the client is involved mainly at the start (requirements) and end (delivery).
- Agile changes this — the **customer is part of the team** and collaborates throughout the process.
- Regular feedback from customers ensures the product stays aligned with their needs.
- Continuous engagement **reduces misunderstandings and rework**.

💡 *Key idea:* Work *with* customers, not just *for* them.

4 Responding to Change > Following a Plan

- Waterfall models resist change because it affects cost, schedule, and scope.
- Agile **welcomes and adapts to change** — it's seen as an opportunity to improve the product.
- Work is prioritized in short cycles (sprints), so adjustments can be made quickly based on customer feedback.
- Change leads to **better alignment with real-world needs and faster value delivery**.

💡 *Key idea:* Change is not a disruption — it's a pathway to improvement.

❖ Summary

Waterfall (Predictive)	Agile
Focus on plans and processes	Focus on people and collaboration
Heavy documentation	Working product emphasized
Customer involved at start and end	Customer involved continuously
Change is a risk	Change is an opportunity

⌚ The 12 Principles of Agile

The **Agile Manifesto** defines 12 principles that guide how teams should think and work in an **agile mindset** — focusing on flexibility, collaboration, and continuous improvement.

1 Satisfy the Customer through Early & Continuous Delivery

- The top priority is to **deliver value early and often**.
- Shorter release cycles mean faster feedback and higher customer satisfaction.
- Teams aim to release an **MVP (Minimum Viable Product)** early so customers can test and provide input.

 *Key Idea:* Deliver fast, get feedback, and improve continuously.

2 Welcome Changing Requirements — Even Late in Development

- Agile **embraces change**, as customer needs evolve over time.
- Flexibility helps turn change into a **competitive advantage**.
- Regular iterations allow teams to adapt quickly.

 *Key Idea:* Change is not a problem — it's an opportunity to make things better.

3 Deliver Working Software Frequently

- Deliver small, usable outputs (called **user stories**) every few weeks.
- Frequent releases = faster progress and quicker feedback.
- This principle applies to all projects — replace “software” with “deliverable” if not in IT.

 *Key Idea:* Work in **small, quick sprints** to deliver value faster.

4 Business People & Developers Must Work Together Daily

- Constant **collaboration** keeps everyone aligned.
- Daily “**stand-up meetings**” ensure progress and resolve issues quickly.
- Builds strong communication and teamwork.

 *Key Idea:* Daily collaboration = fewer misunderstandings and faster results.

5 Build Projects Around Motivated Individuals

- Choose the right people and **trust them** to do their job.
- Give teams **autonomy, tools, and support** — then step back.

- Motivated teams are **self-organizing** and deliver the best results.

 *Key Idea:* Empower the team; don't micromanage.

6 Face-to-Face Communication is Most Effective

- Direct conversation is faster and clearer than emails or texts.
- Daily stand-ups or planning meetings help maintain connection.
- In remote teams, video calls can replace physical meetings.

 *Key Idea:* Communication drives collaboration — talk directly.

7 Working Software (Deliverables) is the Primary Measure of Progress

- Success = delivering **functional, usable outputs**, not just documents.
- Find failures early, fix them fast, and deliver value quickly.

 *Key Idea:* The product's usefulness shows real progress.

8 Promote Sustainable Development

- Work at a **steady, realistic pace** — avoid burnout.
- Sponsors, developers, and users should maintain balance and consistency.

 *Key Idea:* Sustainability ensures long-term productivity and quality.

9 Continuous Attention to Technical Excellence & Good Design

- Focus on quality from the start — reduces rework and errors.
- Skill development and strong design enhance agility.

 *Key Idea:* Doing it right the first time saves time later.

10 Simplicity — Maximize the Work Not Done

- Focus only on **what adds value** to the product.
- Avoid over-complicating designs or unnecessary features.
- Regularly **re-prioritize tasks** to stay efficient.

 *Key Idea:* Simplicity = focus + speed.

11 Self-Organizing Teams Create the Best Results

- Teams decide **how** to do the work themselves.

- Self-organization fosters **creativity, responsibility, and ownership**.
 *Key Idea:* Trust the team — they know the best way to achieve goals.
-

1 2 Regularly Reflect & Adjust for Continuous Improvement

- After each iteration (sprint), teams **review what went well and what didn't**.
 - They then **adapt** to improve the next cycle.
 *Key Idea:* Learning and improving never stop.
-

In Summary

Agile Principles encourage:

- **Customer satisfaction** through early delivery
 - **Embracing change** rather than fearing it
 - **Strong teamwork and communication**
 - **Continuous improvement and simplicity**
 - **Empowered, motivated, and self-organizing teams**
-

From Agile Mindset to Agile in Practice

1. Scrum Framework

- **Most popular Agile framework**
- Works in **Sprints** – short, time-boxed iterations for incremental progress.
- Can be used in **any industry**, not just software.

Scrum Roles:

1. **Development Team:** Builds the product.
 2. **Product Owner:** Prioritizes tasks based on business value.
 3. **Scrum Master:** Facilitates the process, provides *servant leadership*, and removes roadblocks.
-

2. Kanban Framework

- Uses a **visual board** (Kanban board) to track workflow.
- Work items are represented by **Post-it notes** or cards.

Common Columns:

- **Backlog:** Tasks to be done
- **Work in Progress (WIP):** Tasks currently being done
- **Done:** Completed tasks

As work progresses, cards move from *Backlog* → *WIP* → *Done*.

 **Focus:** Continuous flow of work and visual transparency.

3. Lean Framework

- Focuses on **eliminating waste** — both in **materials** and **time**.
- Goal: Deliver value efficiently and avoid unnecessary steps.

 **Focus:** Maximize value, minimize waste.

4. XP (Extreme Programming)

- Designed for **software development**.
- Emphasizes:
 - **Simplicity**
 - **Communication**
 - **Feedback**
 - **Respect**

Key Technique:

 **Pair Programming** — Two developers work together on the same computer, improving code quality and reducing errors.

Combining Frameworks

- Teams often **mix frameworks** for flexibility.
 - Example: Using **Kanban boards** within **Scrum**.
 - The goal is to **choose and adapt** methods that best fit the project and organization.
-

⌚ Agile Frameworks Overview

Agile has several frameworks that put the **Agile mindset** into practice.

The most common ones are **Scrum**, **Kanban**, **Lean**, and **Extreme Programming (XP)**.

✳️ 1. Scrum – *Iterative & Role-Based Framework*

Definition:

Scrum is a structured Agile methodology that organizes work in short, time-boxed iterations called **sprints** (usually 1–4 weeks).

💡 Scrum Roles

Role	Responsibility
Product Owner	Defines product features, prioritizes requirements (creates Product Backlog)
Scrum Master	Acts as a servant leader ; removes obstacles and ensures team follows Scrum practices
Development Team	Builds the product; cross-functional members (developers, testers, designers, etc.)

📁 Scrum Artifacts

Artifact	Description
Product Backlog	A prioritized list of features or tasks maintained by the Product Owner
User Stories	Short descriptions of functionality from the end user's perspective
Sprint Backlog	A selection of the highest-priority items from the Product Backlog for the next sprint
Burndown Chart	Graphical representation showing remaining work vs. time during a sprint

⌚ Scrum Ceremonies

Ceremony	Purpose
Sprint Planning	Define sprint goals and select user stories for the sprint
Daily Standup (Daily Scrum)	15-minute meeting to discuss progress, plans, and blockers

Ceremony	Purpose
Sprint Review	Team showcases completed work to stakeholders
Sprint Retrospective	Team discusses what went well and what can be improved for the next sprint

 **Goal:** Continuous improvement and fast, incremental delivery.

2. Kanban – *Visual Workflow Management*

Definition:

A **visual method** for managing work through a **Kanban board** with columns such as:

- Backlog → In Progress → Done

Features:

- Uses **cards or sticky notes** for tasks
- Encourages **continuous delivery**
- Limits **Work in Progress (WIP)** to prevent overload
- Provides **transparency** into the team's progress

 **Goal:** Improve flow and efficiency.

3. Lean – *Eliminate Waste, Maximize Value*

Definition:

Lean focuses on eliminating all forms of waste — whether in **materials, time, or processes** — to deliver value more efficiently.

Principles:

1. Define value from the customer's perspective
2. Map the value stream
3. Eliminate waste (non-value-adding activities)
4. Create flow
5. Strive for continuous improvement

 **Goal:** Faster, more efficient processes with minimal waste.

4. Extreme Programming (XP) – *Agile for Software Development*

Definition:

XP is an Agile framework focused on **engineering practices** to produce high-quality software rapidly and sustainably.

Five Core Values:

1. **Communication**
2. **Simplicity**
3. **Feedback**
4. **Courage**
5. **Respect**

Practices:

- **Pair Programming:** Two programmers work together on the same code.
- **Test-Driven Development (TDD):** Writing tests before writing code.
- **Continuous Integration:** Frequently merging and testing code.

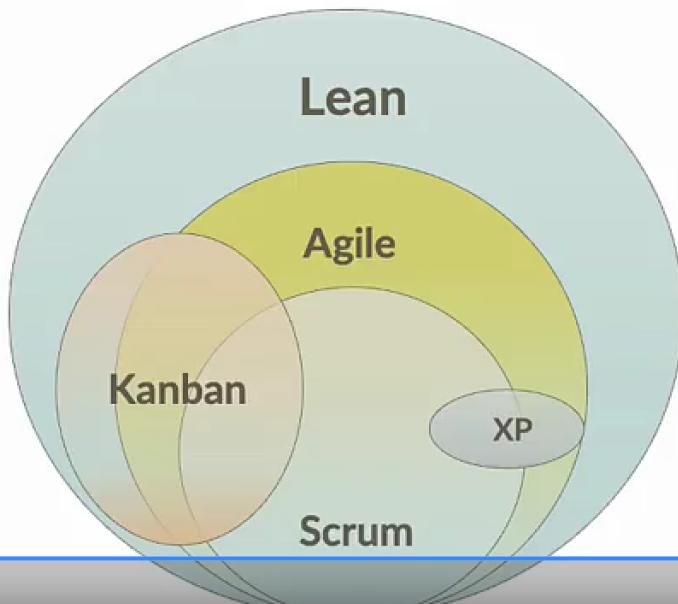
 **Goal:** High-quality software with fewer defects and strong collaboration.

 **Summary Table**

Framework Focus	Key Tools / Concepts	Best For	
Scrum	Time-boxed sprints, roles, and ceremonies	Product backlog, Sprint backlog, Burndown chart	Projects with evolving requirements
Kanban	Continuous flow and visualization	Kanban board, WIP limits	Ongoing tasks or maintenance work
Lean	Efficiency and waste reduction	Value stream, Continuous improvement	Process optimization
XP	High-quality software via best engineering practices	Pair programming, TDD, CI	Software development teams

PMI Agile Practice Guide

"One way to think about the relationship between lean, agile, and the Kanban method is to consider agile and the Kanban method as descendants of lean thinking...focusing on delivering value, respect for people, minimizing waste, being transparent, adapting to change, and continuously improving".



 Lockheed Martin
Engineering Management Program

🔗 Relationship Between Agile Frameworks

🌱 Core Idea:

All Agile frameworks (**Scrum**, **Kanban**, **XP**) are built on the foundation of **Lean thinking** — the philosophy of eliminating waste, focusing on value, and improving continuously.

🔗 Framework Connections

Framework	Derived From / Related To	Key Relationship
Lean	Origin philosophy	The base concept for all Agile methods — focus on efficiency and value.
Agile	Evolved from Lean	Applies Lean principles to project management and teamwork.
Kanban	Derived directly from Lean manufacturing	Provides visual workflow management; often used within Scrum teams.
Scrum	Built on Agile and Lean ideas	Uses time-boxed sprints for iterative delivery; can integrate Kanban boards.
XP (Extreme Programming)	Agile + Lean practices for software development	Often used inside Scrum sprints to ensure high code quality.

💡 Key Takeaways

- All Agile frameworks share **Lean roots** — focusing on **customer value** and **continuous improvement**.
 - **Kanban** evolved directly from **Lean manufacturing systems**.
 - **XP** works well within **Scrum** for technical excellence in software projects.
 - **Lean principles** (reduce waste, improve flow, respect people) appear **across all frameworks**.
 - It's important to **understand all frameworks to combine best practices** that fit your team or organization.
-

Simple Summary:

Think of **Lean** as the “parent,” and **Agile** as its “child.”

Then **Scrum**, **Kanban**, and **XP** are the “grandchildren” — each applying Lean-Agile ideas in their own unique way.

1. The agile manifesto believes that individuals and interactions are _____ processes and tools. (Select best answer)

1 / 1 p

- Less important than
 As important as
 More important than

Correct. Individuals and interactions are have greater value than processes and tools.

2. Which of the following is true about Agile with respect to change? (select best answer)

1 / 1 point

- Agile does not address change.
 Agile methods do not respond well to change.
 Agile methods invite change.

Great job. Agile seeks change.

3. There are ____ principles of Agile. (Select best answer) 1 / 1 point

4
 12

Great job. There are 4 Values and 12 Principles of Agile.

16

4. Which Agile method uses visual boards to maximize efficiency? (Select best answer) 1 / 1 point

Kanban

Correct. Kanban uses boards to maximize efficiency and communication.

XP
 Waterfall

5. One of the practices of Kanban is to limit work-in-progress. Why is it important to limit WIP? (Select best answer) 1 / 1 point

Limiting WIP allows for more effective customer feedback loops.
 Teams need to be able to visualize the workflow.
 The team needs to be able to switch focus and multi-task effectively.
 Limiting work-in-progress ensures a more manageable number of active items so the team can focus.

Correct. Limiting WIP allows the team to focus on priorities.

6. Pair-programming, refactoring, continuous integration are all components of which Agile framework? (Select best answer) 1 / 1 point

Kanban
 Waterfall
 XP

Correct. Pair-programming, refactoring, and continuous integration are all associated with Extreme programming.

🌀 What Are Sprints?

✳️ Definition

Sprints are short, fixed periods of time — usually **1 to 4 weeks** — during which an Agile team works to complete a **specific, small portion of the overall project**.

They are also known as **time boxes**, meaning the duration is fixed and not extended even if the work isn't finished.

⌚ Purpose

- Break large projects into **manageable chunks**
 - Deliver **working, usable product increments** frequently
 - Allow **continuous feedback** and **improvement**
 - Keep the team **focused** and **motivated** through short-term goals
-

⚙️ Key Activities Within a Sprint

Phase	Description
1. Sprint Planning	The team and Product Owner decide what work will be done during the sprint and define the sprint goal.
2. Daily Scrum (Stand-up)	A short daily meeting (about 15 minutes) where team members share what they did yesterday, what they'll do today, and discuss any obstacles.
3. Sprint Review	Held at the end of the sprint. The team presents what they've built to stakeholders for feedback.
4. Sprint Retrospective	The team reflects on the sprint — what went well, what could improve, and how to make the next sprint more effective.

💡 Key Takeaway

A **Sprint** is like a mini-project within the project — a short, focused cycle that ends with a tangible, working result and a chance to improve before the next cycle.

Who Are the Players in a Sprint?

Every **Sprint** involves three key roles that make up the **Scrum Team**.

Each has specific **responsibilities** but works closely together to ensure success.

1. Product Owner (PO)

Main Focus: *Maximize product value.*

Responsibilities:

- Defines the **vision** and **goal** of the product.
- Manages the **Product Backlog** (list of all desired work).
- **Prioritizes** features based on business value and customer needs.
- Acts as the **bridge between stakeholders and the team** — ensuring everyone understands what's needed.

 *Think of the Product Owner as the “voice of the customer.”*

2. Scrum Master (SM)

Main Focus: *Enable and support the team.*

Responsibilities:

- Serves as a **servant leader** — removes obstacles that slow the team.
- Ensures the Scrum process is followed properly.
- Facilitates Scrum ceremonies (planning, standups, review, retrospective).
- Coaches and mentors the team in Agile principles.

 *The Scrum Master doesn't manage people — they help the team manage itself.*

3. The Development Team

Main Focus: *Build the product.*

Responsibilities:

- Delivers potentially **shippable product increments** each sprint.
- Self-organizes — decides how to complete the work.
- Collaborates continuously with the Product Owner and Scrum Master.
- Maintains **quality** and **transparency** through open communication.

 The team is cross-functional — it includes everyone needed to deliver the work (developers, testers, designers, etc.).

Summary Table

Role	Focus	Key Responsibility
Product Owner	Customer & Business Value	Defines and prioritizes what to build
Scrum Master	Process & Team Support	Guides the team and removes obstacles
Team	Product Delivery	Builds and delivers working product

The Scrum Master – The Servant Leader

Overview

- The **Scrum Master** is one of the **three key Scrum roles** (along with the Product Owner and Development Team).
 - Their main role is to **serve and support the team** so they can focus on building the product effectively.
 - Often called a “**servant leader**” — they lead by **helping**, not by commanding.
-

“Carrying the Water for the Team”

- This phrase means the Scrum Master **supports the team completely** — handling all distractions, blockers, and administrative tasks.
 - Allows the team to **focus fully on development and sprint goals**.
-

Core Responsibilities of the Scrum Master

1.  **Fosters Communication**
 - Builds a **collaborative environment**.
 - Acts as a **liaison** between the team and external stakeholders.
 - Ensures smooth information flow during standups and meetings.
2.  **Protects the Team**
 - Guards the team’s **focus and time**.
 - Keeps outside interruptions and distractions away during the sprint.
3.  **Maintains Tools and Reports**

- Keeps tools like **Jira** or **Kanban boards** updated.
- Creates **charts, reports**, and **burndown visualizations** showing progress.

4. **Facilitates Scrum Ceremonies**

- Leads key meetings:
 - **Sprint Planning**
 - **Daily Standups**
 - **Sprint Reviews & Retrospectives**

5. **Acts as an Agile Coach**

- Promotes **Agile principles** across the organization.
- Helps drive the transition to an **Agile mindset** company-wide.

6. **Supports the Team**

- Resolves **internal conflicts**.
- Handles logistical issues or even small morale-boosting tasks (like “coffee runs”).
- Removes **roadblocks** and ensures the team has everything it needs to succeed.

Key Roles of a Scrum Master

Role	Description
 Guard	Protects the team’s focus and removes outside distractions.
 Handler	Builds trust, resolves conflicts, and keeps teamwork healthy.
 Coach	Guides continuous improvement and professional growth.
 Communicator	Ensures clear communication inside and outside the team.
 Servant Leader	Leads by helping, enabling, and empowering others.

In Short

The **Scrum Master** is not a manager — they are the **heart of the Scrum Team**, ensuring smooth workflow, removing obstacles, promoting collaboration, and guiding the team toward success through **servant leadership**.

1. What is a Sprint? (Select all that apply)

1 / 1 point

- A process that offers fast and frequent deliverables.

Good job. Sprints allow for fast and frequent deliverables that give the team flexibility.

- A short, time-boxed period when the team works to complete a specific amount of work.

Correct. Sprints are short interactions of work.

- Agile

2. There are three primary roles in scrum. These are the Scrum Master, Scrum Team, and the _____? (Select best answer).

1 / 1 point

- Servant Leader
 Stakeholders
 Project Manager
 Product Owner

Correct. The product owner is part of the Scrum.

3. When a team finishes a sprint and discusses areas for improvement for the next sprint they are conducting a _____? (Select best answer)

1 / 1 point

- Sprint Review
 Sprint Retrospective

Correct. Sprint retrospectives allow the team to identify areas for improvement during the next iteration.

- Sprint Backlog
 Daily Scrum

4. Which best describes the product owner?: (Select all that apply)

1 / 1 point

- The product owner represents the voice of the customer.

Correct. The product owner acts as the voice of the customer

- The product owner prioritizes the product backlog.

Feedback. Good job. It is the responsibility of the product owner to prioritize the product backlog.

- The product owner protects the team from distractions.
 The product owner sets the product goals.

Correct. Product owners carry out product discovery and strategy.

- The product owner is the servant leader.

5. Who decides how to accomplish the work in the sprint? (Select best answer)

1 / 1 point

- The Developers

Correct. The developers decide how the work will be done.

- The Scrum Master
 The Tester
 The Product Owner

Scrum Process Overview

This video gives a **deeper dive into the Scrum process**, one of the most popular **Agile frameworks**.

1. Sprints

- **Sprints** are short, fixed time periods (called **time boxes**) where the team completes a part of the overall project.
- Each sprint includes:
 - **Sprint Planning**
 - **Daily Scrum**
 - **Sprint Review**
 - **Sprint Retrospective**

2. Agile Frameworks

- There are multiple Agile frameworks like **Scrum** and **Extreme Programming (XP)**.
- Scrum is widely used and borrows ideas from other frameworks such as **Kanban**.
- Many Scrum teams use **Kanban boards** to visualize and manage:
 - Product backlogs
 - Work in progress

3. Scrum Roles

Scrum defines three main roles:

- **Scrum Master** – Facilitates the Scrum process and removes obstacles.
- **Product Owner** – Defines the product vision and prioritizes features.
- **Scrum Team (Developers)** – Build and deliver the product increment.

4. Scrum Artifacts

Artifacts are the key **tools and documents** used in Scrum to track progress and define work:

- **Product Backlog** – A list of all desired features and requirements.
 - **User Stories** – Short, user-focused descriptions of product functionality.
 - **Burndown Charts** – Visuals that track progress of work completion over time.
-

5. Scrum Ceremonies

Ceremonies are **structured meetings or discussions** that keep the team aligned:

- **Sprint Planning** – Define what will be done in the sprint.
 - **Daily Standup (Daily Scrum)** – Short meeting to discuss progress and blockers.
 - **Sprint Review** – Showcase completed work and gather feedback.
 - **Sprint Retrospective** – Reflect on the process and find areas for improvement.
-

6. Tools & Metrics

The video also mentions **Scrum tools** and **metrics** used to measure and visualize progress:

- **Velocity measures** – Track how much work is completed per sprint.
 - **Burndown/Burnup charts** – Show progress towards goals over time.
-

In Short

Scrum organizes work into short, manageable cycles (sprints) with **clear roles, tools, and ceremonies** that help teams **collaborate effectively, adapt to changes, and deliver value continuously**.

The Sprint (or Scrum) Charter

The **Sprint Charter** is a foundational tool in **Scrum** that defines how the team will work together. It's essentially a **team agreement** or **working contract** that helps build trust, commitment, and transparency — all essential for a high-performing Agile team.

1. Purpose of the Sprint Charter

- It serves as a **mutual agreement** among all members of the Scrum team — the **Product Owner, Scrum Master, and Developers**.
- It outlines **team norms, roles, rules, and processes** for collaboration.
- It promotes **alignment, clarity, and accountability** before starting actual sprint work.

2. Why the Entire Team Participates

- Every team member helps **create and agree** on the charter.
 - This encourages **shared ownership** and ensures everyone understands the team's values, working style, and responsibilities.
 - It's revisited whenever **team members change or project scope evolves**.
-

3. Key Areas Covered in a Sprint Charter

(a) WHO – The Team

- Defines **who the team members are**, their **roles**, and **skills**.
- Lists **strengths and weaknesses** of individuals and the team.
- Identifies **skill gaps** and **training needs**.
- Helps anticipate and mitigate future **blockers**.

(b) WHY – The Purpose

- Defines **why the team exists** and **what goals** they're working toward.
- Clarifies **the value** being created for the customer.
- Describes the **mission** — for example, developing a new app or integrating a system.

(c) HOW – Team Dynamics

- Outlines **how the team will collaborate and make decisions**.
- Defines **conflict resolution methods** and **decision-making processes**.
- Sets **rules of engagement** — how the team will communicate, share feedback, and respect boundaries.

(d) WHAT – The Process

- Defines the **Scrum process details**, including:
 - Sprint length
 - Meeting schedules (stand-ups, planning, reviews, retrospectives)
 - Ownership of **external communications**
 - How to manage the **product backlog** and **iteration timing**
 - Includes **rules for breaking ties**, handling **issues**, and maintaining **team discipline**.
-

4. Components of a Typical Sprint Charter

A well-designed Sprint Charter might include:

Section	Description
Team Roster	Names, roles, and responsibilities
Purpose/Mission	Why the team exists and what it aims to achieve
Team Values	Principles that guide behavior (e.g., transparency, respect, collaboration)
Rules & Decisions	Decision-making methods, tie-breaking, handling rule violations
Sprint Process	Sprint length, stand-up timing, backlog management
Communication Norms	How issues are raised, meeting etiquette
Risks & Boundaries	Anticipated challenges, constraints, and team agreements

5. Benefits of the Sprint Charter

- Builds **trust and cohesion** among team members.
 - Clarifies **roles, expectations, and working norms** early.
 - Helps prevent **misunderstandings** and **conflicts** later.
 - Provides a **reference point** for accountability throughout the project.
-

In Summary

The **Sprint Charter** is more than just a document — it's a **team-building process** that unites everyone under a shared understanding of *who they are, why they're here, how they'll work, and what process they'll follow*.

It lays the foundation for a **collaborative, transparent, and high-performing Scrum team**.

Sprint Planning in Agile (Scrum)

Sprint Planning is the **first step** in every Scrum sprint. It sets the direction and scope for what the team will accomplish in the upcoming **time-boxed iteration** (usually 1–4 weeks). Although Agile focuses on flexibility, **planning is still essential** to ensure everyone understands what needs to be done and how success will be measured.

1. What Is Sprint Planning?

Sprint Planning is a **collaborative meeting** where the entire Scrum team — **Product Owner, Scrum Master, and Developers** — meet to decide:

- **What work** will be done during the sprint.

- How the team will complete it.

It turns the **Product Backlog** (a list of all desired features or tasks) into a **Sprint Backlog** (the specific items chosen for that sprint).

2. Who Attends the Sprint Planning Meeting

Role	Responsibility
Product Owner (PO)	Presents and explains the prioritized product backlog items, clarifies goals and acceptance criteria.
Scrum Master (SM)	Facilitates the meeting, ensures it stays focused, and removes obstacles.
Developers (Team)	Select which tasks they can realistically complete and break them into smaller tasks.

3. Key Inputs for Sprint Planning

1. **Product Backlog** – A list of features or tasks prioritized by the Product Owner.
 2. **Team Capacity** – The available time and skills of team members during the sprint.
 3. **Past Performance** – Data from previous sprints (like velocity) to estimate workload.
 4. **Definition of Done (DoD)** – A checklist that ensures all work meets quality standards.
-

4. Main Components of Sprint Planning

Sprint planning usually answers **two key questions**:

1 What can be delivered in this sprint?

- The **Product Owner** explains the goal for the sprint (the **Sprint Goal**).
- The **team** discusses which **Product Backlog Items (PBIs)** can be completed within the sprint time frame.
- The team commits to a set of items — this becomes the **Sprint Backlog**.

2 How will the work get done?

- The **team** breaks each backlog item into smaller, actionable **tasks**.
 - Tasks are estimated (often in hours or story points).
 - Dependencies, risks, and required resources are discussed.
-

5. The Outcome of Sprint Planning

By the end of the meeting, the team should have:

- A **Sprint Goal** — the main purpose or outcome of the sprint.
 - A **Sprint Backlog** — the list of items the team commits to completing.
 - A **clear plan** for how the work will be completed.
-

6. Duration

- For a **one-month sprint**, sprint planning typically lasts **up to 8 hours**.
 - For **shorter sprints** (like two weeks), it's usually **2–4 hours**.
-

7. Tips for a Successful Sprint Planning Meeting (from Lucidchart)

- **Keep the meeting focused and time-boxed.**
 - **Prepare ahead** – Product Owner should ensure backlog items are ready and prioritized.
 - **Encourage collaboration** – Every team member should contribute.
 - **Define a clear and achievable Sprint Goal.**
 - **Use visual tools** (like Kanban boards or Lucidchart diagrams) to map progress.
 - **End with team commitment** – Everyone agrees on what they will deliver.
-

Example:

Imagine a sprint for an **e-commerce app**:

- **Sprint Goal:** Enable users to add products to the wishlist.
- **Selected Backlog Items:**
 - Design wishlist UI
 - Create backend API for wishlist
 - Integrate “Add to Wishlist” button
 - Test and deploy the feature

At the end of sprint planning, these items move from the **Product Backlog → Sprint Backlog**, ready for development.

In Summary

Sprint Planning is where the **team aligns around a shared goal**, commits to realistic work, and sets a clear path for the next iteration.

It transforms high-level ideas from the **Product Backlog** into **actionable tasks** that move the project closer to completion — one sprint at a time.

Product Backlog & Backlog Refinement (Grooming)

1. What Is a Product Backlog?

The **Product Backlog** is a **dynamic list** of everything that needs to be built or delivered in a product. It includes **features, bug fixes, technical tasks, and improvements** — basically, **all work items** the team might ever need to do.

It is **not a static document**; instead, it evolves as the product and customer needs change.

2. Who Owns the Product Backlog?

The **Product Owner (PO)** is **responsible for creating, managing, and prioritizing** the product backlog.

- They ensure the backlog reflects the **product vision and goals**.
 - They **prioritize** items based on business value, urgency, and customer needs.
 - The **development team** and **stakeholders** can suggest items, but only the **PO decides the final order**.
-

3. Structure of the Product Backlog

Each item in the backlog is called a **Product Backlog Item (PBI)** — these can be:

- **User Stories** (new features written from a user's perspective)
- **Bugs**
- **Technical tasks**
- **Research spikes** (time-boxed investigations)
- **Improvements**

Each item often includes:

- **Title/Description** – What needs to be done.
 - **Priority** – Importance or business value.
 - **Estimate** – Effort or complexity (in story points).
 - **Acceptance Criteria** – Conditions for completion.
-

4. Purpose of the Product Backlog

- Acts as the **single source of truth** for all future work.
 - Keeps the **team focused** on business priorities.
 - Helps the **Scrum team plan each sprint**.
 - Allows for **transparency** — everyone can see what's next.
-

5. What Is Backlog Refinement (a.k.a. Grooming)?

Backlog Refinement is an **ongoing process** of reviewing, updating, and improving the product backlog.

It ensures that items are:

- Up to date
- Clearly defined
- Properly estimated
- Prioritized according to current goals

In simple terms:

 **Refinement keeps the backlog clean, relevant, and ready for sprint planning.**

6. Who Participates in Backlog Refinement?

- **Product Owner** → Leads the session and updates priorities.
 - **Scrum Master** → Facilitates the discussion and ensures it stays efficient.
 - **Development Team** → Provides technical input, effort estimates, and identifies dependencies.
-

7. When & How Often?

- Typically happens **once per sprint** (around 5–10% of sprint time).
 - Can also occur **continuously** if the project evolves rapidly.
-

8. Activities in Backlog Refinement

1. **Review existing backlog items** – Are they still needed?
2. **Add new items** – Based on feedback or new requirements.
3. **Remove outdated items** – Delete anything no longer relevant.
4. **Update priorities** – Reorder items based on business needs.
5. **Break down large items (Epics)** into smaller user stories.

-
6. **Estimate effort** – Assign story points to each item.
-

9. Benefits of Backlog Refinement

- Keeps the **product roadmap aligned** with goals.
 - Helps the team **prepare for sprint planning** (no surprises).
 - Reduces confusion and miscommunication.
 - Ensures work is **actionable and well understood**.
-

Example

For an **online shopping app**:

- Product backlog items might include:
 1. Add “Wishlist” feature
 2. Improve payment gateway speed
 3. Fix login bug
 4. Add product filters
 - During refinement:
 - “Add product filters” may move higher if users request it.
 - “Wishlist” might be broken into smaller stories:
 - Create backend API
 - Add UI button
 - Test feature
-

In Summary

Term	Meaning	Owner	Goal
Product Backlog	Master list of all tasks/features for the product	Product Owner	To organize and prioritize work
Backlog Refinement	Regular update and clean-up of the backlog	Product Owner (with team)	To ensure items are clear, relevant, and ready for the next sprint

User Stories in Scrum

1. What is a User Story?

A **User Story** is a short, simple description of a **feature or functionality** written from the **end user's perspective**.

It defines **what the user wants**, **why they want it**, and **how it provides value**.

- It focuses on **value delivery**, not technical details.
-

2. Purpose of a User Story

- To **capture customer needs** clearly and simply.
 - To **help teams understand** how each piece of work benefits the user.
 - To **guide development** toward delivering customer value.
 - To make **communication easier** between product owners, developers, and stakeholders.
-

3. Structure / Format

A common format is:

As a (who), I want (what), so that (why).

Example:

As a shopper, I want to save my favorite products so that I can buy them later.

This format ensures that the user's goal and value are always clear.

4. Source of User Stories

- User stories are **derived from the Product Backlog**.
 - During **Sprint Planning**, the Scrum team selects a few stories from the backlog to work on during the sprint.
-

5. Definition of Done (DoD) / Acceptance Criteria

Each user story must include **Acceptance Criteria**, also called the **Definition of Done** — a clear list of conditions that define when the story is considered complete.

 Acceptance Criteria help ensure:

- Everyone has the same understanding of “done.”
- The feature meets user expectations.

- Testing and validation are easier.

Example:

For the story “As a shopper, I want to add items to my cart”:

Acceptance Criteria:

- User can add items to the cart.
 - The total updates automatically.
 - The cart retains items until checkout.
-

6. Key Qualities of a Good User Story (INVEST Model)

Letter Stands For Meaning

I	Independent	Each story should be self-contained.
N	Negotiable	It's open for discussion and refinement.
V	Valuable	Delivers clear user or business value.
E	Estimable	The team can estimate its effort.
S	Small	Should be completed within one sprint.
T	Testable	It has clear acceptance criteria.

7. Benefits of Using User Stories

- Keeps development **user-focused**.
 - Promotes **collaboration** between team members.
 - Makes planning **more flexible** and **easier to estimate**.
 - Helps ensure **each sprint delivers tangible value**.
-

8. Summary Table

Concept	Description
User Story	Short description of a feature from the user's perspective
Format	“As a [who], I want [what], so that [why]”
Derived From	Product Backlog
Used In	Sprint Planning

Concept	Description
Acceptance Criteria	Defines when the story is complete and successful
Goal	Deliver value to the user quickly and clearly

Daily Stand-Up Meeting (Scrum)

Purpose:

A short daily meeting to ensure **team communication, collaboration, and transparency**. It helps the team stay aligned and resolve blockers early.

Duration and Frequency

- **Held every day** during the sprint.
 - **Fixed time each day** for consistency.
 - **Duration:** Maximum **15 minutes** — short enough that everyone can stand.
-

Agenda — The Three Questions

Each team member answers:

1. **What did I work on yesterday?**
 2. **What will I work on today?**
 3. **What issues or blockers are preventing my progress?**
-

Benefits

- Keeps everyone **aware** of each other's progress.
 - Allows team members to **offer help** if someone faces blockers.
 - **Prevents surprises** or delays later in the sprint.
 - Encourages **daily communication** and **team accountability**.
-

Best Practices

1. **Hold at the same time daily** – builds routine and predictability.
 2. **Decide a sharing order** – e.g., go around in a circle or same sequence each day.
 3. **Keep it under 15 minutes** – respect time and maintain focus.
-

What to Avoid

- Don't turn it into a long status meeting.
 - Avoid problem-solving discussions during the stand-up — handle those **after** the meeting if needed.
-

Sprint Review

The **Sprint Review** is a key Scrum event held at the end of each sprint. It provides an opportunity for **stakeholders** to **inspect** the outcome of the sprint and **offer feedback** for future improvements.

During this meeting, the **Scrum Team** presents the work completed — typically in the form of **working product increments** — and discusses what was accomplished versus what was planned.

The **Product Owner** reviews the product backlog and explains which items have been “Done” and which remain incomplete. Stakeholders then collaborate with the team to suggest changes, improvements, or new features based on the sprint outcome. This helps ensure the product continues to evolve in alignment with customer needs and business goals.

Overall, the Sprint Review promotes **transparency, collaboration, and continuous improvement**, setting the stage for more effective **sprint planning** in the next iteration.

Sprint Retrospective

The **Sprint Retrospective** is a dedicated meeting held at the **end of each sprint** where the **Scrum Team** reflects on its process, teamwork, and overall performance. The primary goal is **continuous improvement** — identifying what went well, what didn’t, and what actions can be taken to enhance future sprints.

During this session, team members discuss challenges, communication gaps, workflow issues, or any obstacles faced during the sprint. Together, they define **actionable steps** to improve efficiency, collaboration, and quality in upcoming sprints.

Unlike the **Sprint Review**, which focuses on the **product and deliverables**, the **Sprint Retrospective** focuses on the **process and people** — ensuring that the team evolves and grows stronger after every iteration.

Key Takeaway:

- **Sprint Review** → What we built.
 - **Sprint Retrospective** → How we built it and how to do it better next time.
-

Top 5 Common Retrospective Mistakes

1. Skipping or Rushing the Retrospective

Teams often skip this meeting under time pressure, losing valuable learning opportunities.

2. Blaming Instead of Problem-Solving

The meeting should be about collaboration and improvement, not finger-pointing or assigning fault.

3. No Action Items or Follow-Up

Without actionable steps or accountability, the same issues repeat in future sprints.

4. Lack of Honest Communication

If members fear speaking up or feedback is ignored, the retrospective loses its value.

5. Not Changing the Format or Approach

Repeating the same meeting style every sprint can make it monotonous. Mixing formats keeps it engaging and productive.

1. What questions are answered in the Daily Standup meeting? (select all that apply)

1 / 1 point

What did I work on yesterday?

Correct. You discussed what you accomplished the day before to keep communication flowing and transparent.

What will I work on tomorrow?

What issues are blocking my progress?

Correct. This is discussed so the team can help each other.

What will I work on today?

Correct. You discuss what you plan to work on that day.

2. A comprehensive list of the details or features that need to be developed is also known as a: (select best answer):

1 / 1 point

Product Backlog

Correct: A product backlog is a list of features that the team needs to develop.

Sprint Planning

User story

Sprint Charter

3. “As an online customer, I want to recall my previous payment method so that I can easily checkout with one click.” is an example of: (Select best answer)

Product Backlog

Try again. The product backlog is a list of product requirements.

A User Story

MVP

Scrum Charter Statement

ANSWER: A USER STORY

4. In order for a user story to move into a sprint, the _____ must be defined. (Select best answer)

1 / 1 point

- product backlog
- Kanban
- acceptance criteria

Correct. Acceptance criteria is important to define so you know the level of effort necessary to complete the functionality.

5. What is the purpose of the Sprint Review? (Select best answer)

1 / 1 point

- Obtain feedback from stakeholders on products from the sprint.

Correct. The Sprint Review allows stakeholders to offer feedback for future iterations.

- Demo to Product Owner
- To identify improvements in the team process.

6. How long should the Daily Standup meeting last? (select best answer)

1 / 1 point

- As long as necessary
- 15 minutes

Correct. The daily standup should be brief.

- 1 hour

Simplicity Principle (Agile Manifesto)

Definition:

Simplicity is “*the art of maximizing the amount of work not done.*” It means **focusing only on work that adds direct value** and avoiding unnecessary effort or complexity.

Key Ideas:

- **Not about avoiding work** — it’s about avoiding *wasteful* work.
- Rooted in **Lean principles**, which focus on **eliminating anything unnecessary**.
- Encourages **simple, clear, and minimal** design or code that achieves the goal without over-engineering.
- Supports faster delivery of the **MVP (Minimum Viable Product)** — allowing customers to reach the market quickly.

• Benefits:

- Reduces errors and rework.
- Speeds up development.
- Makes systems easier to understand and maintain.

Key Takeaway:

“More” does **not** equal “better.”

The **simplest effective solution** is always the best one.

Kanban Board (in Scrum & Agile)

Definition:

A **Kanban Board** is a visual tool used to **track workflow and task progress** during a project. It helps teams see what needs to be done, what's in progress, and what's completed — at a glance.

Key Components:

- **To Do** – Tasks that are planned but not started yet.
 - **Work In Progress (WIP)** – Tasks currently being worked on.
 - **Done** – Tasks that have been completed.
-

Purpose & Benefits:

- Provides **real-time visibility** of project progress.
 - Helps **identify bottlenecks** (where work gets stuck).
 - Encourages **continuous flow** of work.
 - Improves **team collaboration and accountability**.
 - Promotes **transparency** and **efficiency** in task management.
-

Example Tools:

- **Physical boards** with sticky notes.
 - **Digital tools** like **Trello, Jira, Asana, or Kanban Tool**.
-

Key Takeaway:

The Kanban Board turns complex workflows into a **simple visual system**, helping teams manage and complete work more efficiently.

👉 Agile Team Velocity

Definition:

Agile **velocity** measures how much work a team can complete during a sprint (iteration). It helps predict **how long the project will take** to finish based on past performance.

✳️ Why Velocity Matters

- Shows the **team's productivity and capacity**.
 - Helps in **planning future sprints** more accurately.
 - Allows teams to **forecast project timelines**.
-

🌐 How Velocity is Calculated

1. Record how many **story points** were completed in each past sprint.
 2. Add them together and **divide by the number of sprints**.
 - Example:
 - 8 sprints → total 111 story points completed
 - $\text{Velocity} = 111 \div 8 = 14 \text{ story points per sprint}$
-

Ἐ Estimating User Story Effort

Before calculating velocity, teams estimate effort for each user story using:

1. **Story Points:**
 - A number that represents the **effort or complexity** of a task.
 - Example: 1 story point ≈ 5–8 hours of work.
 - 2 points = twice the effort of 1 point, and so on.
 2. **T-Shirt Sizing:**
 - Simple way to classify effort as **Small (S)**, **Medium (M)**, **Large (L)**, or **Extra-Large (XL)**.
 - Easier for quick comparisons.
-

📈 Using Velocity to Predict Completion

- Example:
 - **Velocity:** 14 story points/sprint
 - **Remaining backlog:** 80 story points

- $80 \div 14 \approx 6$ sprints
 - If each sprint = 2 weeks $\rightarrow 6 \times 2 = 12$ weeks to complete the project
-

Key Takeaways

- Velocity = **Average work done per sprint.**
 - It enables **accurate forecasting and planning.**
 - Focus is on **continuous improvement**, not competition.
 - Keep estimates **simple and consistent** across sprints.
-

Burn Down & Burn Up Charts

Both **Burn Down** and **Burn Up charts** are visual tools in **Agile project management** that track the team's progress throughout a sprint or project.

They help everyone **see how much work is done** and **how much remains** — making progress easy to monitor at a glance.

▼ Burn Down Chart

Definition:

A **Burn Down Chart** shows how much work **remains** to be done in the sprint or project over time.

Purpose:

- Tracks **progress toward completing all work items.**
- Helps identify whether the team is **ahead or behind schedule.**

How It Works:

- The **X-axis** = Time (e.g., days of the sprint).
- The **Y-axis** = Amount of work (in **story points, tasks, or hours**).
- The **ideal line** shows how work *should* burn down if everything goes perfectly.
- The **actual line** shows the team's *real* progress.

Example:

If the line goes **downward quickly**, the team is completing work faster.

If it flattens, progress has slowed.

 **Goal:** Reach **zero remaining work** by the end of the sprint.

▲ Burn Up Chart

Definition:

A **Burn Up Chart** shows how much work has been **completed** over time, rather than what's left.

Purpose:

- Displays **total scope** and **completed work** together.
- Helps visualize when **scope changes** (new work added or removed).

How It Works:

- The **X-axis** = Time.
- The **Y-axis** = Work (story points, tasks, etc.).
- Two lines are shown:
 - **Total Scope Line** (top)
 - **Completed Work Line** (bottom)
- As the project progresses, the completed line moves upward until it meets the total scope line.

 **Goal:** The two lines meet — meaning all planned work is done.

Burn Down vs. Burn Up

Feature	Burn Down Chart	Burn Up Chart
Focus	Work remaining	Work completed
Direction	Line goes down	Line goes up
Scope Changes	Hard to show	Clearly visible
Best for	Tracking sprint progress	Tracking project progress

Key Takeaways

- Both charts visualize **progress and team velocity**.
 - **Burn Down** = How much work is left.
 - **Burn Up** = How much work is done.
 - Great tools for **motivation, transparency**, and **early detection** of delays.
-

⌚ Prioritizing the Product Backlog and Selecting User Stories in Scrum

◆ Goal: Deliver Maximum Customer Value

One of the main goals in Agile and Scrum is to **create value early and often** for the customer. To do this effectively, teams must **prioritize the work** in the **product backlog** — ensuring that the **most valuable features** are developed first.

👤 Who Sets the Priorities?

The **Product Owner (PO)** is responsible for prioritizing the **Product Backlog**.

- The **Product Owner** acts as the **voice of the customer**.
 - They decide which **user stories** will be included in the upcoming **sprint** based on value, urgency, and feedback.
 - The **Scrum Team** (developers and Scrum Master) collaborates and provides input, but the **final decision** on backlog order lies with the Product Owner.
-

⌚ How Prioritization Works

At the end of every sprint:

1. The team and Product Owner review customer feedback.
 2. They re-evaluate priorities based on **new requirements or changes in business needs**.
 3. The **backlog is refined (groomed)** to reflect these updated priorities before the next sprint.
-

⚙️ Common Prioritization Techniques

1. 🟢 MoSCoW Method

A simple way to classify requirements based on importance:

Category	Meaning	Description
M – Must Have	Essential	Critical features without which the product cannot function.
S – Should Have	Important	High-value features that are not mandatory for launch.
C – Could Have	Nice to Have	Adds convenience or minor value, but not essential.
W – Won't Have (this time)	Not Needed	Features that will not be implemented in this release.

👉 Helps teams **focus on what truly matters** and avoid overengineering.

2. Kano Analysis

A **graphical method** to visualize customer satisfaction vs. feature fulfillment.

- **X-axis:** Implementation (None → Fully implemented)
- **Y-axis:** Customer Satisfaction (Dissatisfied → Delighted)

Position on Graph	Meaning
◆ Upper Right Corner	Must-have / Delight features → High customer satisfaction
◆ Lower Left Corner	Unnecessary or missing features → Low value or dissatisfaction
👉 Helps identify which features truly delight users versus those that don't add much value .	

3. Dot Voting

A **simple, democratic** prioritization method.

- Each team member or stakeholder gets a set number of **sticky dots**.
- They **vote** on features they believe are most important.
- The features with the **most dots** become the **highest priorities**.

👉 Encourages **team participation** and **shared ownership** in decision-making.

4. The \$100 (or 100-Point) Method

A **fun and quantitative** prioritization technique.

- Each participant gets **\$100** (or 100 points) to "spend" on features.
- They can allocate the money however they want.
- The **features with the most money** represent the **highest perceived value**.

👉 Highlights **which features are worth the most investment** from a value perspective.

💡 Key Takeaways

- **Product Owner** owns and prioritizes the **Product Backlog**.
- **Agile teams** focus on **maximizing customer value** through prioritization.
- Use tools like **MoSCoW**, **Kano**, **Dot Voting**, and **\$100 method** to decide what to build next.
- After each **iteration (sprint)**, priorities are **reviewed and adjusted** based on customer feedback and business goals.

1. What is usually written on cards that are used on Kanban Boards?: (select best answer)

1 / 1 point

- Results from sprint retrospectives
- User Stories

Correct. The cards used on Kanban Boards are usually User Stories.

- Product Backlog

2. Why is it important to limit work in progress? (select best answer)

1 / 1 point

- It allows the team to move through projects more efficiently.

Correct. It allows the project team to concentrate on the user story and increases efficiency.

- Kanban helps you visualize the work
- To reduce overall cost.

3. Which Scrum Tool shows how much work is remaining? (select best answer)

1 / 1 point

- T shirt sizing
- Story points
- Burn down chart

Correct. The Burn down chart shows the team how much project work is remaining.

4. My agile team has decided that one story point = 8 hours. We selected Feature X to be worked on during the next iteration. Feature X is assigned 3 story points. Approximately how much effort will Feature X require? (select best answer)

1 / 1 point

- Not enough information given.
- Approximately 16 hours
- Approximately 24 hours

Correct. 3 story points is approximately 3 times the value of one story point.

5. Who prioritizes the Product Backlog? (select best answer)

1 / 1 point

- Developers
- Product Owner

Correct. The Product Owner prioritizes the product backlog

- Stakeholders
- Scrum Team
- Scrum Master

⌚ What Does It Mean to Be an Agile Organization?

💡 “Doing Agile” vs. “Being Agile”

- **Doing Agile** means following the *processes, tools, and techniques* of Agile (like Scrum, Kanban, etc.).
- **Being Agile** means adopting the *mindset* — living by the *values and principles* of the **Agile Manifesto** across the entire organization, not just in projects.

👉 In short, an Agile organization doesn't just *do* Agile — it *is* Agile.

🌐 Definition of an Agile Organization

An Agile organization embraces the **values and principles** of the Agile Manifesto throughout the company — putting **people first** and **results ahead of process**.

This means the company:

- Focuses on **customer value** over rigid procedures.
 - Promotes **collaboration, adaptability, and learning**.
 - Encourages **continuous improvement** at all levels.
-

📘 PMI's Definition of Agile and Agility

From PMI's *2017 Pulse of the Profession* report (*Achieving Greater Agility*):

- **Agile:**

A mindset based on key values and principles that enable collaborative work and continuous value delivery through a people-first orientation.

- **Agility:**

The ability to quickly sense and adapt to external and internal changes to deliver relevant results productively and cost-effectively.

👉 So, being Agile means **embracing change and delivering value continuously**.

⚙️ Three Key Elements of an Agile Organization

1. Corporate Strategy

- Must be **adaptable** to rapidly changing markets.
- The company continuously **searches for opportunities** through change.
- A shared vision and strategy encourage **nimbleness** and **innovation** across teams.

✳️ *Example:* Instead of long-term fixed plans, strategy evolves based on market feedback.

2. Organizational Structure

- Agile organizations use **flat, distributed team structures**.
- Teams are **self-managed** and **cross-functional**, allowing faster decision-making.
- Open communication encourages **knowledge sharing** and **transparency**.

✳️ *Example:* Replacing silos with networks of teams that collaborate fluidly.

3. Leadership

- Agile organizations practice **distributed leadership**, following **servant leadership** principles.
- Leaders focus on **empowering teams, removing barriers**, and **supporting growth**.
- Leadership is **shared** — anyone can take initiative and influence outcomes.

✳️ *Example:* A servant leader asks, “*How can I help my team succeed?*” instead of commanding from above.

Agile in the Age of Remote Work

Even though Agile favors face-to-face collaboration, **remote and hybrid teams** have adapted successfully through technology:

- Tools like **Slack, Zoom, Jira, and Miro** enable real-time connection and collaboration.
- Virtual “face-to-face” communication continues to uphold Agile values.

Data from Digital.ai (2022):

- **94%** of surveyed companies practice Agile.
- **29%** began Agile adoption during 2020–2021 (remote era).
- Agile is expanding **beyond software**:
 - **Operations:** 29%
 - **Marketing:** 17%
 - **HR:** 16%
 - **Sales:** 11%

👉 Agile thinking is becoming an **organizational philosophy**, not just a software practice.

✳️ Remote Work and Agile Statistics

- **16%** were remote before 2020 and will stay remote.

- **25%** went remote during the pandemic and will remain remote.
- **56%** plan to stay **hybrid** (mix of remote and in-office).
- Only **3%** plan to return to full-time office work.

 Remote and hybrid work are fully compatible with Agile — and often enhance flexibility.

Common Challenges in Becoming Agile

1. **Corporate policies** that don't support Agile principles.
 2. **Resistance to change** — many organizations fear large structural shifts.
 3. **Lack of Agile training** for teams and leaders.
 4. **Absence of strong leadership** in driving Agile transformation.
-

Key Takeaway

To truly **become Agile**, an organization must:

- Embed Agile **values and principles** in its **strategy, structure, and leadership**.
 - Encourage **adaptability, empowerment, and collaboration**.
 - Build a culture where **everyone — from leadership to teams — embodies the Agile mindset**.
-

◎ The Five Trademarks of Agile Organizations

1. North Star Embodied Purpose

- Agile organizations have a **clear vision or “North Star”** that guides all decisions and actions.
 - This purpose aligns employees, inspires innovation, and helps navigate change.
 - Everyone understands *why* the organization exists and *how* their work contributes to the overall mission.
 *Example:* Spotify's purpose to “unlock the potential of human creativity” drives all its product and cultural decisions.
-

2. Network of Empowered Teams

- Instead of traditional hierarchies, agile organizations operate as **networks of small, empowered, and cross-functional teams**.
- Teams are given ownership to make quick decisions and adapt to change.

- Leadership supports rather than controls — promoting trust, collaboration, and accountability.

👉 *Think of it as many mini-startups working toward one big goal.*

3. Rapid Decision and Learning Cycles

- Agile organizations encourage **continuous learning and experimentation**.
- They use **short cycles** (like sprints) to test ideas, get feedback, and improve quickly.
- Mistakes are treated as learning opportunities, not failures.

👉 *Focus = “Fail fast, learn faster.”*

4. Dynamic People Model that Ignites Passion

- Agile companies emphasize **people over processes** — they empower employees to grow and innovate.
 - Roles are flexible, and people are encouraged to use their strengths creatively.
 - Culture focuses on **trust, transparency, recognition, and personal development**.
-

5. Next-Generation Enabling Technology

- Technology acts as the **backbone** of agility — enabling collaboration, data-driven decisions, and automation.
 - Tools like cloud platforms, digital dashboards, and communication apps help teams stay connected and responsive.
 - Agile organizations continually adopt new digital tools to stay ahead of change.
-

💡 Summary:

Agile organizations = Purpose-driven + Team-empowered + Fast-learning + People-focused + Tech-enabled.

These five trademarks allow companies to **sense change quickly, respond effectively, and deliver continuous value** — even in uncertain environments.

⚡ Agile Strategic Vision

1. Traditional vs. Agile Strategy

- **Traditional strategic planning** → a **top-down** process.
 - Senior leaders analyze strengths, weaknesses, and market conditions.

- They create a **long-term mission and vision** assuming the market will stay stable.
- **Problem:** Today's world is **fast-changing and unpredictable**, so rigid plans often fail.

👉 Companies now need **Agile Strategic Planning** — strategies that **adapt quickly** to market shifts.

2. What Is Agile Strategic Vision?

- It's the **application of Agile principles** to the **strategic planning process**.
 - Instead of fixed multi-year plans, companies use **shorter planning cycles**, continuous feedback, and flexible goals.
 - The focus is on **responding to change**, not resisting it.
 - Strategy evolves through **experimentation, learning, and collaboration**.
-

3. Key Principles of Strategic Agility

(From the article “*6 Principles to Build Your Company’s Strategic Agility*”)

1. **Anticipate Change** – Constantly scan the environment for new trends and threats.
 2. **Adapt Quickly** – Be ready to pivot strategies when conditions change.
 3. **Empower Teams** – Give decision-making power to cross-functional teams closer to the action.
 4. **Focus on Value Creation** – Prioritize work that delivers the most value to customers and stakeholders.
 5. **Learn Continuously** – Encourage testing, feedback, and learning from both success and failure.
 6. **Align with a Clear Purpose** – Keep everyone guided by a strong mission (“North Star”) even as tactics shift.
-

4. Examples of Agile Strategic Thinking

- **Netflix** – Shifted from DVD rentals to streaming when the market evolved.
 - **Amazon** – Continuously experiments and adapts (e.g., AWS, Prime) based on user feedback.
 - **Spotify** – Uses squads and tribes to stay innovative while aligning to a common purpose.
-

⌚ Summary

Agile Strategic Vision = **Flexible, iterative, and purpose-driven planning** that helps companies **thrive in uncertainty** by adapting fast, empowering teams, and continuously learning.

Being Agile vs. Practicing Agile

1. Practicing Agile

- Means **following Agile frameworks like Scrum, Kanban, or XP.**
- Focuses on **processes, tools, and ceremonies** — e.g., daily standups, sprint reviews, retrospectives, etc.
- Teams **apply the Agile methodology** to improve development speed and efficiency.
- Often **limited to project or team level** (not the whole organization).

Example:

A software team uses Scrum to release updates every 2 weeks — but the company still makes slow, top-down decisions.

2. Being Agile

- Goes **beyond following the process** — it's a **mindset and culture**.
- The entire organization **embraces agility** in how it **thinks, decides, and adapts**.
- Promotes **collaboration, flexibility, empowerment, and continuous improvement**.
- Decisions are **decentralized**, and teams are trusted to act fast.

Example:

A company where every department — HR, Marketing, Operations — adapts quickly to change and values people over processes.

3. Key Difference

Aspect	Practicing Agile	Being Agile
Focus	Tools, processes, and frameworks	Mindset, culture, and adaptability
Scope	Teams or projects	Whole organization
Goal	Deliver working software efficiently	Create an adaptive, value-driven organization
Change Handling	Follows a process for change	Embraces change as a natural part of growth

Summary

- Practicing Agile** = Doing Agile (methods and tools).
 - Being Agile** = Thinking Agile (values and culture).
-

Embracing Technology — Agile Transformation vs. Digital Transformation

1. Agile Transformation

- Focuses on **changing how people work** — adopting an **agile mindset** and **organizational culture**.
- It's about **collaboration, adaptability, and customer-centric delivery**.
- Involves **teams, leadership, and structure** evolving to be more flexible and value-driven.
- Technology supports it — but the real goal is **organizational agility**.



A company trains cross-functional teams to deliver in short cycles, respond quickly to customer feedback, and self-organize.

2. Digital Transformation

- Focuses on **what tools and technologies** an organization uses.
- Involves adopting **next-generation technologies** such as:
 - Cloud computing 
 - Artificial intelligence 
 - Data analytics 
 - Automation 
 - IoT and digital platforms 
- The goal is to **improve efficiency, innovation, and customer experience**.



Migrating infrastructure to the cloud or automating manual workflows using AI.

3. The Connection Between the Two

- **Digital Transformation** gives the **tools** for speed and scalability.
- **Agile Transformation** gives the **mindset and process** to use those tools effectively.
- Together, they help companies **adapt faster, innovate continuously, and deliver customer value early and often**.

4. Why It Matters

- One of the **five trademarks of agile organizations** (as per McKinsey) is “**embracing next-generation enabling technologies**.”

- Agile organizations **leverage technology not just for automation**, but to **enhance collaboration, transparency, and responsiveness**.
 - It's not about just *having* the latest tech — it's about **using it in an agile way**.
-

Summary

- ✓ **Agile Transformation** = Changing *how* you work (culture & mindset).
 - ✓ **Digital Transformation** = Changing *what* you use (tools & technology).
 - 💡 Together → True organizational agility in a digital world.
-

1. All of the following are challenges companies face while implementing agile within their organizations. (Select all that apply) 1 / 1 point
 - Resistance to change

Correct. This is a challenge to implementing agile.
 - Lack of agile training

Correct. This is a challenge to implementing agile.
 - Corporate policy

Correct. This is a challenge to implementing agile.
 - Leadership buy-in

Correct. This is a challenge to implementing agile.
2. In the article, *The Five Trademarks of Agile Organizations*, the authors described the old paradigm of hierarchical and specialized companies as _____. (select best answer) 1 / 1 point
 - Living organisms
 - Optimus Prime
 - Machines

Correct. This is based on Morgan's descriptions of specialized organizations.

3. Which of the following statements below reflect an agile mindset? (Select all that apply)

1 / 1 point

- We live in an environment where everything is changing and evolving. The best way to minimize risk is to embrace uncertainty and be the quickest and most productive in trying new things.

Correct. This reflects an agile mindset.

- To deliver the right outcome, the most senior leaders should define where we are going, develop the plans to get there, and minimize risk.
- We succeed by capturing value from our competitors, customers, and suppliers for our stakeholders.
- Technology is integrated and core to every aspect of the organization to maximize value and meet business needs.

Correct. This is an agile mindset.

4. What is defined as "The ability to improve performance and thrive during market disruption"? (select best answer)

1 / 1 point

- Simplicity Principle
- Servant Leadership
- Strategic Agility

Correct. Strategic agility allows companies to quickly improve performance.

- Agile Project Management

5. What are the 3 A's of strategic agility? (select best answer)

1 / 1 point

- Aspire** to create a culture of agility, **Assure** employees autonomy through self-organizing teams, and **Attend** to the needs of the team through servant leadership.
- Adopt** technology to beat the competition, **Advance** your strategic vision by quick implementation plans, and **Agility** makes for quick decisions.
- Analyze** the competition through SWOT, **Adjust** your vision based on analysis, and **Approve** proposals quickly that meet the vision.
- Avoid** impacts by being nimble, **Absorb** impacts by being financially sound, and **Accelerate** on opportunities by moving faster than the competition.

Correct. Avoid, Absorb, and Accelerate are the 3 A's of strategic agility.

6. Which of the following describe Digital Transformation? (select all that apply)

1 / 1 point

- The integration of technology to fundamentally transform how you operate and deliver value.

Correct. Digital transformation is about transforming value delivery through technology

- Technology is a supporting capability that delivers specific services and tools to the organization as defined by priorities and budget.

- Integrating technology into all aspects of business to transform delivery of value to a customer.

Correct. Digital transformation integrates technology into all aspects of the business.

- Adopting technology to create or modify business processes to deliver a better customer experience

Correct. Digital transformation is about incorporating technology to create better experiences.

Agile Organizational Structure — Summary

1. Traditional Hierarchy vs. Agile Networks

- Traditional organizations follow a **pyramid structure** with layers of management (CEO → VPs → Managers → Team Members).
- This slows down decision-making, creates communication bottlenecks, and limits flexibility.
- **Agile organizations**, by contrast, operate as **networks of teams** — smaller, autonomous groups that collaborate closely and communicate openly.

2. Network of Teams

- Multiple small, **self-organizing teams** that collaborate across functions.
- Focus on **transparency, real-time communication, and collective problem-solving**.
- Teams **leverage each other's strengths** and share information freely.

3. Ideal Team Size

- Inspired by **Dunbar's Number** — humans can maintain effective communication with about 150 people.
- Recommended **team size**: no more than **15 members** for optimal communication.
- **Jeff Bezos' Two-Pizza Rule**: If a team needs more than two pizzas, it's too big.

4. Autonomous Teams

- Function like **mini-business units** with control over:
 - Hiring

- Marketing and sales decisions
 - Profit and loss management
 - Despite autonomy, teams must align with **overall organizational vision and values**.
-

5. Characteristics of Autonomous Teams

- **Self-driven:** Set and pursue their own goals.
 - **High trust:** Members collaborate closely and rely on each other's strengths.
 - **Strong communication:** Open sharing of ideas and information.
 - **Collective decision-making:** Every voice counts.
 - **Continuous improvement:** Focus on learning, upskilling, and feedback from other teams.
-

6. Leadership's Role in Agile Organizations

- Senior leaders define the **shared vision, values, and strategy**.
 - Teams align their missions with these goals but have **freedom in execution**.
 - Leadership focuses on **empowerment, not control** — enabling innovation and adaptability.
-

7. Outcome

- A **networked structure** replaces the slow-moving hierarchy.
 - Teams are **accountable, adaptable, and innovative**.
 - The organization becomes more **resilient to change and responsive to technology shifts**.
-

The Allegro Group Way to Agility (Case Study Summary)

Company Background

- **Allegro Group** — a leading **Polish e-commerce company**, similar to eBay or Amazon in Central Europe.
 - Before 2009, it faced **rapid growth challenges**:
 - Siloed departments (business vs. IT)
 - Slow product delivery cycles
 - Limited adaptability to user needs and market changes
-

Why the Shift to Agile

- Traditional (waterfall) project management wasn't flexible enough.
 - Leadership wanted **faster delivery, improved collaboration**, and **better responsiveness to customers**.
 - The company needed to align **business and IT** to achieve shared goals.
-

Transformation Journey

1. **Started Small:**
 - Began with **pilot Scrum teams** in 2009.
 - Focused on training employees in agile principles, not just frameworks.
 2. **Leadership Commitment:**
 - Senior management acted as **Agile champions**, emphasizing cultural change.
 - Shifted from command-and-control to **empowerment and trust**.
 3. **Structural Redesign:**
 - Reorganized teams around **products and customer value**, not functions.
 - Each team had cross-functional members (developers, testers, business analysts, UX).
 4. **Cultural Shift:**
 - Encouraged **transparency, collaboration, and experimentation**.
 - Adopted the mindset: "Fail fast, learn faster."
 5. **Continuous Improvement:**
 - Used regular **retrospectives** to identify bottlenecks and enhance processes.
 - Established **Agile Coaches** to support teams.
-

Outcomes

- Faster product releases.
 - Greater employee engagement and morale.
 - Improved communication between teams.
 - A strong **customer-first mindset**.
 - Agile principles became embedded in **company culture**, not just project management.
-

Allegro - Q1 2021 Earnings Breakdown (Chirag Modi Report)



Business Performance Highlights

- Allegro became **Poland's largest e-commerce platform**, with over **12 million active buyers**.
 - **Revenue growth:** ~60% YoY (year-over-year).
 - **Gross Merchandise Value (GMV):** surged by more than **54%** in Q1 2021.
 - **Profit margin and operational efficiency** improved significantly due to agility and digital optimization.
-



Agility in Practice

- Continuous product innovation (e.g., Allegro Smart!, logistics improvements, better mobile UX).
 - Agile teams helped **shorten release cycles** and **respond quickly to market shifts**, including pandemic-driven e-commerce surges.
 - Leveraged **data analytics and automation** for better decision-making.
 - Operated as a **network of empowered teams** — each focused on specific customer needs or platform features.
-



Key Takeaways

- **Agility = Competitive Advantage:** Allegro's transformation positioned it as a market leader in a highly competitive sector.
 - **Sustainable Growth:** The agile mindset enabled continuous evolution and rapid adaptation to market trends.
 - **Cultural Alignment:** Agile became part of "how Allegro works," not just a temporary methodology.
-



Conclusion

Allegro's journey from 2009 to 2021 shows that **being agile (mindset + culture)** is far more powerful than simply **doing Agile (following a framework)**.

Their success proves how embracing agility drives **innovation, speed, and resilience** in fast-changing digital markets.



Distributed Agile Teams — Summary



Context

- Traditionally, **Agile** emphasizes *face-to-face communication* as the most effective method for collaboration.

- However, with **remote work**, **global teams**, and **digital transformation**, organizations now rely on **distributed agile teams** — teams spread across locations, time zones, and even cultures.
-

Key Principles for Building Effective Distributed Agile Teams

1.  **Communication is Everything**
 - Use **synchronous** tools (Zoom, Teams, Slack) for daily stand-ups and retrospectives.
 - Use **asynchronous** tools (Jira, Confluence, Trello) for updates and documentation.
 - Maintain **transparency** — everyone should know what others are working on.
 2.  **Build Trust and Relationships**
 - Create opportunities for **virtual social interactions** and team bonding.
 - Encourage **psychological safety** — team members must feel comfortable sharing ideas or raising concerns.
 3.  **Clear Roles and Accountability**
 - Define each person's **responsibility and ownership** clearly.
 - Encourage **autonomous teams** that can make decisions without micromanagement.
 4.  **Manage Time Zones Effectively**
 - Use “**golden hours**” (overlapping working time) for real-time collaboration.
 - Record key meetings and share summaries for those who can't attend.
 5.  **Embrace Tools and Technology**
 - Collaboration: Slack, MS Teams
 - Project tracking: Jira, Trello, ClickUp
 - Version control: GitHub, GitLab
 - Continuous integration: Jenkins, GitHub Actions
 6.  **Continuous Improvement**
 - Conduct **regular retrospectives** to discuss challenges unique to remote work.
 - Gather feedback on **communication tools**, **time management**, and **team morale** to refine collaboration.
-

Benefits of Distributed Agile Teams

- Access to **global talent pool**
- **24-hour productivity cycle** (work continues as time zones shift)

- Increased **flexibility and resilience**
 - Cost savings on physical infrastructure
-

Challenges

- Miscommunication or delays due to time zone gaps
 - Reduced team bonding without physical interaction
 - Potential **loss of visibility** or accountability without proper tools
-

Conclusion

Distributed Agile Teams thrive when they combine:

- **Strong communication,**
- **Clear structure,**
- **Digital collaboration tools,** and
- **A culture of trust and transparency.**

This model keeps the **Agile spirit alive** even when teams are not co-located — ensuring adaptability, collaboration, and continuous delivery across borders. 

Servant Leadership — Summary

Core Idea

Servant leadership is a philosophy where a leader's primary goal is **to serve others first** — empowering, supporting, and growing their team members before focusing on personal authority or control.

In Agile, this is embodied by the **Scrum Master**, who leads by **facilitating, guiding, and enabling**, rather than commanding.

The “Jar” Metaphor – Ali Fett’s Story

Ali Fett uses a **jar** to illustrate servant leadership:

- The jar represents **your capacity to serve others**.
- Each act of kindness, listening, and support **fills the jar** of others.
- But leaders must also **refill their own jar** — through reflection, rest, and self-care — to continue serving effectively.

The key message:

“You cannot pour from an empty jar. To serve others well, you must also take care of yourself.”

Key Characteristics of a Servant Leader

1. **Empathy** – Understand and care about the emotions and perspectives of others.
 2. **Listening** – Prioritize understanding before responding.
 3. **Awareness** – Be mindful of the team’s environment, morale, and needs.
 4. **Healing** – Help others overcome challenges, mistakes, or burnout.
 5. **Stewardship** – Take responsibility for the success of the team and organization.
 6. **Commitment to Growth** – Focus on helping individuals learn and develop.
 7. **Building Community** – Foster trust, collaboration, and belonging.
-

Servant Leadership in Agile / Scrum

- The **Scrum Master** exemplifies servant leadership by:
 - Removing obstacles for the team
 - Facilitating collaboration and continuous improvement
 - Ensuring the team follows Agile principles
 - Protecting the team from unnecessary pressure or distractions

Rather than managing tasks, the Scrum Master **serves the team's success**.

Benefits of Servant Leadership

- Builds **trust and loyalty** within teams
 - Increases **engagement and ownership**
 - Promotes **open communication** and psychological safety
 - Encourages **continuous learning and innovation**
 - Results in **stronger, more motivated teams**
-

Key Takeaway

Servant leadership shifts the mindset from:

“How can my team serve me?”

to

“How can I best serve my team?”

By leading with empathy, humility, and service, leaders create environments where **people thrive and deliver their best work** — the very essence of Agile success.

1. Which of the following is an organizational structure that values many layers of management and clear lines of authority? (select best answer) 1 / 1 point

Traditional hierarchical organizational structure

Correct. The traditional structure resembles a pyramid with many layers and silos.

- Networked organizational structure
 Network of Teams

2. The Dunbar effect states that a person can effectively maintain a network of _____ people? (Select best answer) 1 / 1 point

- 250
 75
 150

Correct. The Dunbar effect states a person can comfortably maintain 150 personal relationships.

- 15

3. Which of the following is a characteristic of autonomous teams? (Select best answer). 1 / 1 point

- Discipline
 Promotability
 Leadership
 Individuality
 Trust

Correct. Trust is a characteristic of an autonomous team.

4. Why do you think Allegro was able to experience rapid growth? (Select best answer). 1 / 1 point

By adopting a more agile approach, Allegro was able to grow through innovation and acquisition.

Correct. By becoming more agile, they could react to market trends quickly.

- Focusing on return on investment
 By introducing their e-commerce to North America, Allegro was able to capture more of the marketplace.
 By incorporating a complex system of management tools, Allegro was able to better predict outcomes of rapid investments.

5. In the video *How to Build Distributed Teams - Agile Teams*, the presenter shared seven best practices for managing distributed agile teams. Which of the following options is NOT one of those practices? (Select best answer).

1 / 1 point

- Keep an eye on team health.
- Face-to-face conversation is the best method of communication.

Correct. While the principles of agile suggest face-to-face is the best method of communication, the video suggested methods on managing teams when in-person meetings aren't possible.

- Be present as a leader.
- Build and maintain a great product backlog.
- Create opportunities to socialize.

6. Focusing on the greatness of others is a trait of which leadership style? (Select best answer).

1 / 1 point

- Servant leader

Correct. Servant leaders build up their teams.

- ABC leader
- Micromanager
- Eager leader