

Object Oriented System Analysis and Design (OOSAD) **INSY3063**



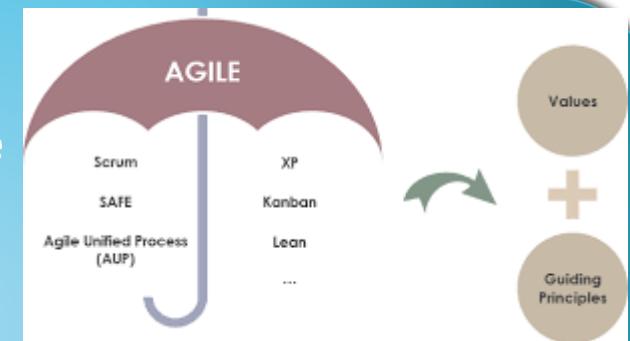
CHAPTER III

Chapter Three: Agile Development



CHAPTER THREE: AGILE DEVELOPMENT

- Agile Methods in General
- Large Scale SCRUM (LeSS) Structure
- LeSS Product
- LeSS Sprint
- More or LeSS
- Fundamental Information Gathering Skills
 - Interviewing
 - Observation
 - Brainstorming
- Agile Documentation





Objectives

- Understand the Agile Methods and methodology in general
- Differentiate Less rules, principles, practices and Less versus single-team Scrum.
- Identify fundamental information gathering techniques
- Explain and understand what Agile documentation is.

Agile Methods in General

- ❖ **Object-oriented development** is not directly concerned with the **organization** of the project but with **the product itself, from conceptual to physical**.

It has three requirements:

- ❶ **an object-oriented technology,**
- ❷ **an object-oriented analysis and design, and**
- ❸ **a project plan that is friendly to an object-oriented approach.**

- ❖ Object-oriented development is highly iterative. An expanded object-oriented development also includes the following concepts: **component-based development, architectural design, learning from patterns, and a model-driven approach.**

Agile Methods in General

- ❖ **Definition:-**
- ❖ Out of the object-oriented approach emerged the latest view of systems development, collectively called the **agile methodologies**.
- ❖ **The agile methodology** is one of the easiest and uncomplicated routes to transform an idea and varied needs into feasible software solutions.
- ❖ **The agile method** is an **iterative and incremental** tactic for software design that utilizes **constant planning, understanding, upgrading, team partnership, development, and delivery**. The agile process is fragmented into separate models that teams work on, there by **encouraging flexibility to changes**.
- ❖ As the development process is **iterative, errors are resolved in the intermediate stage of the project**. This process enables the final deliverable product to match the customer's wants better(systems is **user-friendly**).

Agile Methods in General

- ❖ In February 2001, many of the proponents of these alternative approaches met in Utah and reached a consensus on many of the **underlying principles** their various approaches contained. This consensus turned into a document they called **“The Agile Manifesto.”**
- ❖ **Agile software development** is a **lightweight approach** that was proposed to overcome other development methods' limitations and **To reduce the overhead and the cost** while providing flexibility to adopt the changes in requirements at any stage.

Agile Methods in General

- ❖ Agile methodologies form a distinct from other methodologies of software development but diverse category.
- ❖ Their common goal is being “adaptive” rather than “predictive.”
- ❖ Generally, Agile methodologies consider “upfront” documentation, “upfront” modeling, and “upfront” design as a waste of time.
- ❖ This is done by managing the tasks and their coordination through a certain set of values and principles.
- ❖ Agile is a wide umbrella of software development beliefs.

Agile Methods in General

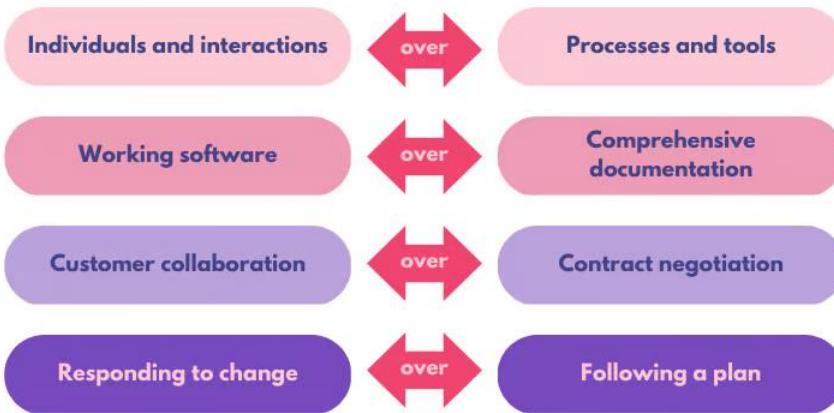
- ❖ Agile is a **conceptual framework** for software engineering that begins with a starting planning phase and follows the road toward the deployment phase with **iterative and incremental interactions throughout the life-cycle of the project.**
- ❖ The **initial goal for the agile methods** is **to reduce the overhead** in the software development process with the ability to adopt the changes **without risking the process or without excessive rework.**

Agile Methods in General

- ❖ Agile methods are processes that support the agile philosophy, i.e. agile values and principles.
- ❖ The agile methodologies share three key principles:
 1. A focus on adaptive rather than predictive methodologies
 2. A focus on people rather than roles, and
 3. A self-adaptive process.
- ❖ All agile development methodologies are based on the agile manifesto and a set of twelve principles.

Agile Methods in General

4 Agile Manifesto values



The emphasis of the agile manifesto is to focus the developers on

- the working conditions of the developers,**
- the working soft ware,**
- the customers, and addressing changing requirements** instead of focusing on detailed systems development processes, tools, all- inclusive documentation, legal contracts, and detailed plans.

Agile Methods in General

- ❖ Each Agile method consists of a different combination of practices, which is **a description of how the day-to-day work** is done by the software developer.
- ❖ Each method differs from the other by choosing its **appropriate set of terminology and practices**.
- ❖ **Agile methodologies** are typically based only on the **twelve principles** of agile software. These principles include the following:
 1. Software is delivered early and continuously through the development process, satisfying the customer.
 2. Being open to changes in requirements even late in the project.

Agile Methods in General

3. Delivering completed work at regular intervals, preferably short ones.
4. Customers and developers work together to solve the business problem.
5. Assembling a motivated team, providing them with the right environment and support, and trusting them.
6. Communicating face-to-face regularly. Face-to-face communication within the development team is the most efficient and effective method of gathering requirements.
7. Using completed work to measure progress. The primary measure of progress is working, executing software.
8. Creating processes that promote sustainable efforts and a constant pace of work. Both customers and developers should work at a pace that is sustainable. That is, the level of work could be maintained indefinitely without any worker burnout.

Agile Methods in General

9. Agility is heightened through attention to both technical excellence and good design.
10. Encouraging simplicity, the avoidance of unnecessary work, is essential.
11. Recognizing that the best work emerges from self-organized teams that deliver the best architectures and designs.
12. Reflecting regularly on how the development team can be more effective and fine-tuning and adjusting the approach or their development process

Agile Methods in General

- **What is the definition of agility?**
- ❖ **Agility** is the ability to rapidly change body direction, accelerate, or decelerate. It is influenced by balance, strength, coordination, and skill level. Agility can be improved by first developing an adequate base of strength and conditioning that is appropriate for the difficulty level of the athlete.
- ❖ **Agility** refers to the ability to adapt quickly to changing circumstances and make decisions efficiently.
- ❖ There are several **types of agility**:
- 1. **Operational Agility:** - is the ability of an organization to quickly respond to changes in the **market, customer demands, or other external factors**. It involves having **efficient processes, flexible systems, and a responsive workforce**. Operational agility is important for **companies that operate in rapidly changing industries**, such as technology and fashion.

Agile Methods in General

- **Types of agility, including:**
- 2. **Strategic Agility:** - is the ability of an organization **to anticipate and respond to changes in the competitive landscape, industry trends, or other long-term factors.** It involves having a **clear vision, a flexible strategy, and the ability to make decisions quickly and decisively.** **Strategic agility** is important for companies that **need to stay ahead of the curve and stay relevant in their industry.**
- 3. **Leadership Agility:**- is the ability of leaders to **adapt to changing circumstances and lead their teams effectively.** It involves being able **to communicate effectively, make decisions quickly, and build strong relationships with team members.** **Leadership agility** is important for companies that need to be able to **respond quickly to changing market conditions.**

Agile Methods in General

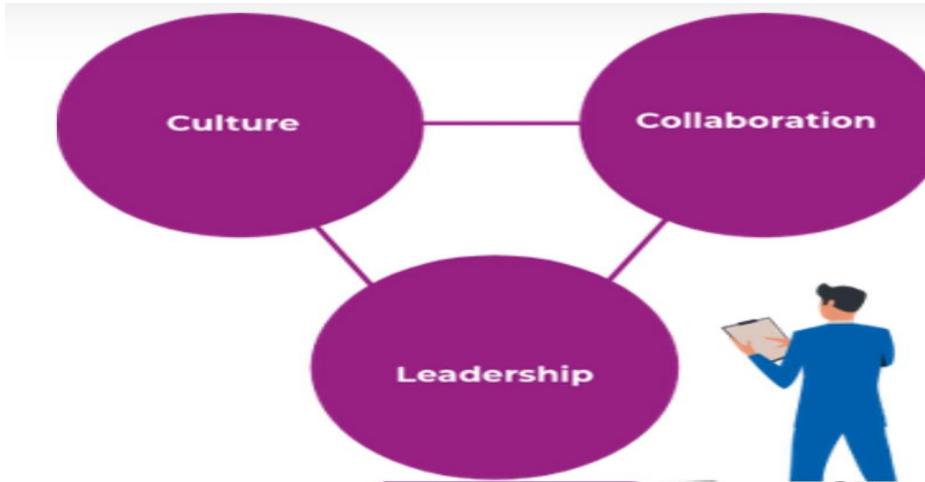
➤ Types of agility, including:

4. **Team Agility**:- is the ability of teams **to work together effectively and adapt quickly to changing circumstances**. It involves having **a shared vision, clear communication, and the ability to collaborate and problem-solve together**. **Team agility** is important for companies that need to be able to respond quickly to customer demands and changing market conditions.
5. **Learning Agility**:- is an individual's ability **to learn quickly and adapt to new situations**. It involves being open to **new experiences, seeking feedback, and continuously developing new skills and knowledge**. **Learning agility** is important for companies that need to **stay up-to-date with the latest technology and industry trends**.
6. **Emotional Agility**:- is an individual's ability **to regulate their emotions and adapt to changing circumstances**. It involves being **able to manage stress, communicate effectively, and build strong relationships with others**. **Emotional agility** is important for companies that want to create a positive and supportive work environment.

Agile Methods in General

➤ Types of agility, including:

7. **Business Agility:-** is the ability of an organization to **quickly adapt to changes in the market, customer demands, or other external factors**. It involves having the flexibility and agility **to respond to new opportunities and challenges**. **Business agility** is important for companies that need to be able to respond quickly to changing market conditions and customer demands.
- ❖ It has **three key elements of the Framework for Business Agility (FBA)** – **Culture**, **Leadership**, along with the crucial role of **Collaboration** throughout.
- ❖ **The Business Agility Toolkit** consists of tools designed to support **individuals, teams**, and **organizations** on their agile journey.



Agile Methods in General

- **Types of agility, including**
- 8. **Enterprise Agility:-** is the ability of an organization **to quickly adapt to changes in the market, customer demands, or other external factors across the entire organization.** It involves having a culture of agility and collaboration across all departments and levels of the organization.
 - **Enterprise agility** is important for companies that need to be able to respond quickly and efficiently to changes in the market and customer demands, while also maintaining a high level of customer satisfaction and employee engagement.

Agile Methods in General

- **Evolution of agile methodologies**
- ❖ The agility of each element will influence the way the organization operates and evolves in response to the **ever-changing needs of its customers and stakeholders.**
- ❖ The evolution of Agile methodologies traces back to the 1990s, primarily as a response to traditional software development methodologies that were perceived as slow, inflexible, and unable to adapt to rapidly changing requirements. Here's a brief overview of the key stages in the evolution of Agile methodologies:
- ❖ **Early Influences (1980s-1990s):** Agile methodologies were influenced by various concepts and practices that emerged in the 1980s and early 1990s. These include iterative and incremental development, as advocated by methodologies like Rapid Application Development (RAD) and the Spiral model. Other influences include Lean manufacturing principles, which emphasize eliminating waste and optimizing value delivery.

Agile Methods in General

- **Evolution of agile methodologies**
- ❖ **Manifesto for Agile Software Development (2001):** The pivotal moment in the formalization of Agile methodologies was the drafting of the Manifesto for Agile Software Development in 2001 by a group of software developers, who outlined the core values and principles of Agile.
- ❖ The manifesto emphasized individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.
- ❖ **Scrum (mid-1990s):** Scrum is one of the earliest and most popular Agile frameworks. It was initially introduced by Ken Schwaber and Jeff Sutherland in the mid-1990s. Scrum emphasizes iterative development, with small, cross-functional teams working in short cycles called sprints to deliver potentially shippable increments of software.

Agile Methods in General

- **Evolution of agile methodologies**
- ❖ Scrum also incorporates **roles (such as Scrum Master and Product Owner), artifacts (such as the Product Backlog and Sprint Backlog), and ceremonies (such as Sprint Planning and Daily Stand-ups)** to facilitate collaboration and transparency.
- ❖ **Extreme Programming (XP) (late 1990s):** Extreme Programming, developed by Kent Beck in the late 1990s, is another influential Agile methodology.
- ❖ XP emphasizes practices such as test-driven development (TDD), pair programming, continuous integration, and frequent releases.
- ❖ XP places a strong emphasis on delivering high-quality software and adapting to changing requirements through continuous feedback and communication.

Agile Methods in General

- **Evolution of agile methodologies**
- ❖ **Expansion and Hybridization (2000s-present):** Since the early 2000s, Agile methodologies have continued to evolve and expand.
- ❖ Various frameworks and methodologies have emerged, each with its own unique approach to Agile software development.
- ❖ These include **Kanban**, which emphasizes visualization and flow; **Lean software development**, which focuses on **maximizing customer value while minimizing waste; and**
- ❖ **Development operations**, which emphasizes **collaboration between development and operations teams to enable continuous delivery and feedback.**

Agile Methods in General

- **Evolution of agile methodologies**
- To conclude the evolution of Agile methodologies has been characterized by a shift towards lightweight, adaptive approaches to software development, with a focus on delivering value to customers quickly and responding to change effectively.
- Agile has become the dominant paradigm in software development, with its principles and practices being applied not only in software development but also in various other domains and industries.

Agile Methods in General

➤ Types of Agile methodology

1. Kanban
 2. Scrum method,
 3. Extreme Programming (XP) method,
 4. Test-Driven Development (TDD) method,
 5. Feature Driven Development (FDD) method,
 6. Dynamic System Development Model (DSDM) method and
 7. Crystal methods etc.
 8. Lean Software Development
 9. Scaled Agile Framework
- ❖ Each method has its own principles, life cycle, roles, advantages and disadvantages etc.
 - ❖ All of these agile software development methods **build the software in iterations and incremental processes.**

Agile Methods in General

➤ Types of Agile methodology

1. Kanban

- ❖ Originating from the Japanese language, the translation of the word ‘Kanban’ is “**visual board or signboard**” and is connected to the concept of “**just in time**”!
- ❖ Initially, the Kanban concept was introduced as a lean manufacturing system and slowly drove its way to agile software development teams.
- ❖ This method uses visual methods for developing and managing projects.
- ❖ Projects through Kanban are overseen with the help of the Kanban Board, which is divided into columns to depict the process flow of the software development.
- ❖ This helps in increasing the visibility of teams as the teams can see the progress through every stage of development and prepare for the upcoming tasks **to deliver the product “just in time”!**
- ❖ This method requires thorough interaction and transparency to enable the team members to be equipped with the right stage of development at any time and have a cohesive flow of work at all times.

Agile Methods in General

2. Types of Agile methodology

2. Scrum method

- ❖ One of **the most popular agile methodology** examples is the agile scrum development methodology, which is depicted by various cycles of development.
- ❖ Similar to Kanban, Scrum breaks down the development phases into stages or cycles called '**sprints**'.
- ❖ The development time for each sprint is maximized and dedicated, thereby managing only one sprint at a time.
- ❖ Scrum and agile methodologies focus on **continuous deliverables**, and thus this method lets designers adjust priorities to ensure that any incomplete or overdue sprints get more attention.
- ❖ Scrum Team has exclusive project roles such as **a scrum master** and **a product owner** with constant communications on the daily scrum where the activities are harmonized to devise the best way to implement the **sprint**.

Agile Methods in General

2. Types of Agile methodology

3. Extreme Programming (XP)

- ❖ **Extreme Programming (XP)** is a methodology **founded on four core values: communication, simplicity, feedback, and courage.** It focuses on constant development and customer satisfaction.
- ❖ Similar to scrum, this method also uses **sprints** or **short development cycles**. This is developed by a team to create a productive and highly efficient environment.
- ❖ The extreme Programming technique is **very supportive in a situation of constant and varying demands from the customers.**
- ❖ It motivates the developers to accept changes in the customer's demands, even if they pop up in an advanced phase of the development process.
- ❖ In Extreme Programming, **the project is tested from the initial stages by collecting feedback that progresses the output of the system.** This also presents a spot check to implement easily any customer requirements.

Agile Methods in General

2. Types of Agile methodology

4. Crystal

- ❖ Introduced by Mr Alistair Cockburn, one of the monumental persons in formulating the Agile manifesto for software development, **Crystal** is a group of smaller agile development methodologies comprising **Crystal Yellow, Crystal Clear, Crystal Red, Crystal Orange, and more.**
- ❖ Each has its peculiar and exclusive framework that is characterized by **factors such as system criticality, team size, and project priorities.**
- ❖ Depending on the nature of the project or system criticality such as Comfort (C), Essential Money (E), Discretionary Money (D), and Life (L), the kind of crystal agile methodology is chosen.
- ❖ Similar to other methodologies of Agile, Crystal also **addresses prompt delivery of software, regularity, less administration with high involvement of users, and customer satisfaction.**
- ❖ **The Crystal** family advocates that each system or project is inimitable and necessitates the solicitation of diverse practices, processes, and policies to achieve the best results, earning the name of the most lightweight methods of agile methodology.

Agile Methods in General

2. Types of Agile methodology

5. Dynamic Systems Development Method (DSDM)

- ❖ To address the need for a standard industry charter for the swift delivery of software, the **Dynamic Systems Development Method (DSDM)** was developed.
- ❖ **DSDM** gives a comprehensive structure that is defined and modified to create a plan, execute, manage, and scale the procedure of software development.
- ❖ Based on a business-driven approach and eight principles, the DSDM believes that modifications to the project are always expected, and quality with timely delivery must never be negotiated.

Agile Methods in General

2. Types of Agile methodology

6. Feature-Driven Development (FDD)

- ❖ Several industry-recognized best practices are incorporated into this iterative, **customer-centric, and incremental agile method.**
- ❖ Its primary goal is to consistently produce working software in a timely fashion.
- ❖ Lifecycle stages include developing an overarching model of the project; **creating feature lists; planning by feature; designing by feature;** and finally **building by feature.**
- ❖ Using this **five-step process**, large project teams will be able to move their products forward at a steady pace.

Agile Methods in General

2. Types of Agile methodology

7. Lean Software Development

- ❖ This agile methodology is based on seven principles:
- 1. **Deleting what doesn't matter-** Anything that doesn't add value is removed from the project
- 2. **Quality development-** The discipline and control of the number of residuals created are essential to quality development
- 3. **Knowledge creation-** The team is driven to document the entire infrastructure to preserve this value in the future
- 4. **Defer commitments-** This point encourages the team to focus less on planning and anticipating ideas without first having a prior and complete understanding of the business requirements
- 5. **Delivery promptly-** Providing value to the customer as quickly as possible
- 6. **Respecting the team-** two essential points are communication and conflict management
- 7. **Optimize the whole-** To create a flow of true value, the development sequence must be perfected enough to remove errors from the code
- ❖ Using this lean methodology, **development time and resources are optimized.** This method is easily scalable and adaptable to projects of any size.

Agile Methods in General

2. Types of Agile methodology

8. Lean Scaled Agile Framework

- ❖ A set of workflow and organizational patterns for implementing agile practices at an enterprise scale is known as the **Scaled Agile Framework** .
- ❖ **Adopting this framework** allows you to take advantage of a framework that is relatively light while still maintaining *the centralized decision-making required at the enterprise level for software development efficiency*.
- ❖ To put it another way, it takes *the agile philosophy and applies it to software executives who are tasked with addressing more strategic issues*.



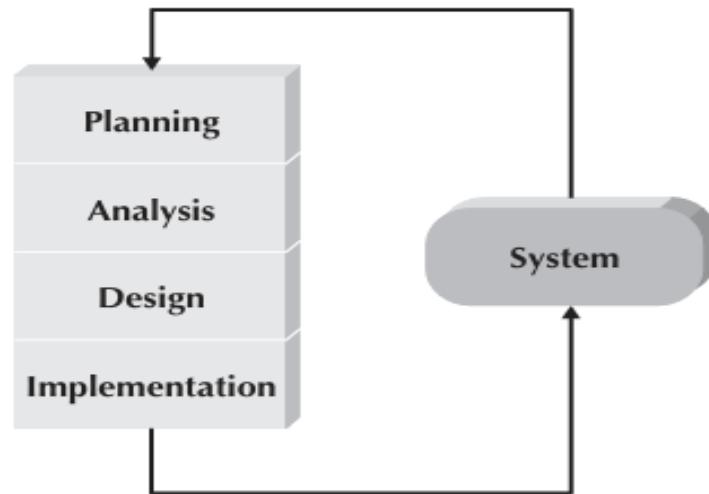
Agile Methods in General

2. Types of Agile methodology =

➤ Conclusion

- ❖ A dynamic approach is required in choosing the right agile methodology among the different types of agile methodology.
- ❖ The advantages and disadvantages of agile methodology must always be considered to choosing the framework for one's business to entice talent and convey remarkable digital experiences in this aggressively competitive market.

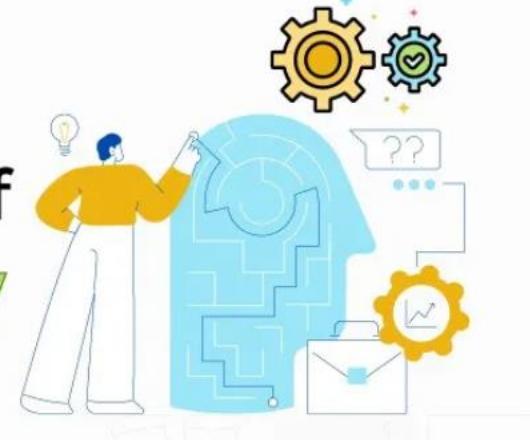
➤ Typical Agile Development Methodology



Agile Methods in General

- **Simplicity in Agility**
- ❖ At its core, **simplicity** in Agile isn't about doing **less work**; it's about **doing exemplary work**. It advocates for **eliminating unnecessary complexities** and **encourages teams to focus on what truly matters**. So **prioritization** will be the key.

The Power Of
Simplicity



Agile Methods in General

- **Simplicity in Agility**
- ❖ **Simplicity** is a fundamental principle in Agile methodologies, emphasizing the value of doing the simplest thing that could possibly work. Here's **how simplicity is integrated into Agile practices:**
 1. **Minimalism in Requirements:** Agile methodologies advocate for minimalistic requirements documentation. Instead of exhaustive documentation upfront, Agile teams focus on capturing just enough information to get started and iterating based on feedback. This approach ensures that efforts are not wasted on unnecessary features or details that may not be valuable to the end-users.
 2. **Iterative Development:** Agile promotes iterative development, where software is built incrementally in small, manageable chunks called iterations or sprints. Each iteration focuses on delivering a working, usable subset of functionality. By breaking down work into smaller pieces, teams can prioritize effectively and deliver value more frequently.
 3. **Continuous Refactoring:** Agile teams emphasize continuous improvement and refactoring of code to keep it simple and maintainable. Refactoring involves restructuring code without changing its external behavior, with the goal of improving readability, maintainability, and efficiency. By keeping the codebase clean and simple, teams can reduce technical debt and adapt to changing requirements more easily.

Agile Methods in General

➤ Simplicity in Agility

4. **Test-Driven Development (TDD):** TDD is a practice in Agile development where tests are written before the code. This encourages simplicity by focusing on the desired behavior of the software before writing any implementation code. TDD helps prevent over-engineering by guiding developers to write only the code necessary to pass the tests, resulting in simpler, more focused implementations.
 5. **Continuous Feedback:** Agile methodologies emphasize continuous feedback from stakeholders, customers, and team members throughout the development process. This feedback loop helps identify opportunities to simplify the product, remove unnecessary features, and prioritize effectively based on changing requirements and market conditions.
 6. **Empowered Teams:** Agile principles empower self-organizing teams to make decisions and adapt their processes based on their unique context. This allows teams to simplify their workflows, eliminate bottlenecks, and optimize for efficiency. By trusting teams to find the simplest solutions to complex problems, Agile fosters creativity and innovation.
- Overall, simplicity in Agile **is about prioritizing value, minimizing waste, and continuously seeking ways to do things more efficiently**. By embracing simplicity, Agile teams can deliver **high-quality software that meets the needs of their customers while maintaining flexibility and adaptability in a rapidly changing environment**.

Agile Methods in General

- **Tools for achieving Agility :-**
- ❖ **Agility** can be achieved through various tools and techniques that can help **individuals, teams, and organizations** to adapt quickly to changing circumstances.
- ❖ **Some of the tools required for agility are:**

1. Agile Methodologies
2. Design Thinking
3. Development Operations
4. Lean Startup
5. Cloud Computing
6. Continuous Improvement
7. Cynefin Framework
8. SAFe (Scaled Agile Framework)

Agile Methods in General

- **Cost of Implementing Agility**
- Implementing agility can have both costs and advantages for an organization. Here are some potential costs and advantages:
- **Costs:**
 1. **Training and Development Costs:** As mentioned earlier, adopting agility requires a significant shift in mindset, culture, and behavior, which may require training and development opportunities for employees to learn Agile principles and practices.
 2. **Process Improvement Costs:** Implementing Agile processes and practices may require changes to the existing processes, such as adopting new tools, revising workflows, and modifying communication channels, which may come at a cost.
 3. **Cultural Change Costs:** Adopting agility requires a cultural shift towards collaboration, transparency, and continuous improvement, which may require changes to the organization's culture, leadership, and management practices.

Agile Methods in General

➤ Advantages of Agility

- 1. Faster Delivery:** Agility enables organizations to deliver products and services faster by focusing on iterative and incremental development, continuous integration, and delivery.
- 2. Higher Quality:** Agility emphasizes continuous improvement, collaboration, and feedback, which can lead to higher quality products or services.
- 3. Increased Customer Satisfaction:** By delivering high-quality products or services faster and more efficiently, organizations can increase customer satisfaction and loyalty.
- 4. Increased Innovation:** Agility fosters a culture of experimentation, learning, and continuous improvement, which can lead to increased innovation and creativity.

Agile Methods in General

➤ Advantages of Agility

5. **Reduced Costs:** By delivering products or services faster and more efficiently, organizations can reduce costs associated with delays, rework, and wasted effort.
 6. **Competitive Advantage:** By adopting agility, organizations can become more responsive to changing market conditions, customer needs, and competitive pressures, *which* can give them a competitive advantage.
- Overall, **agility is a versatile and powerful approach** that can help organizations become more **responsive, innovative, and customer-centric**. *It requires a shift in mindset, culture, and behavior, and is not just about implementing processes and tools.*

Agile Methods in General

➤ Critics of Agility

❖ Agile methodologies do have critics.

1. One of the major criticisms **deals with today's business environment**, where much of the actual information systems development is off shored, outsourced, and/or subcontracted. Given agile development methodologies requiring co-location of the development team, this seems to be a very unrealistic assumption.
2. A second major criticism is that **if agile development is not carefully managed**, and by definition it is not, the development process can devolve into a prototyping approach that essentially becomes a "**programmers gone wild**" environment where programmers attempt to hack together solutions.
3. A third major criticism, based on the **lack of actual documentation created during the development of the software**, raises issues regarding the auditability of the systems being created. **Without sufficient documentation, neither the system nor the systems-development process can be assured.**
4. A fourth major criticism is based on **whether agile approaches can deliver large mission-critical systems.**

SCRUM

➤ **What is Scrum?**

- ❖ **Scrum** is the most popular Agile framework because of its flexibility, and it can adapt to any size of team or project.
- ❖ One of the prime benefits to Scrum is that the work is done by the development team concurrently rather than chronologically.
- ❖ **Scrum** is a **popular framework** that **facilitates collaboration between teams** while they are working on **complex projects and products**.

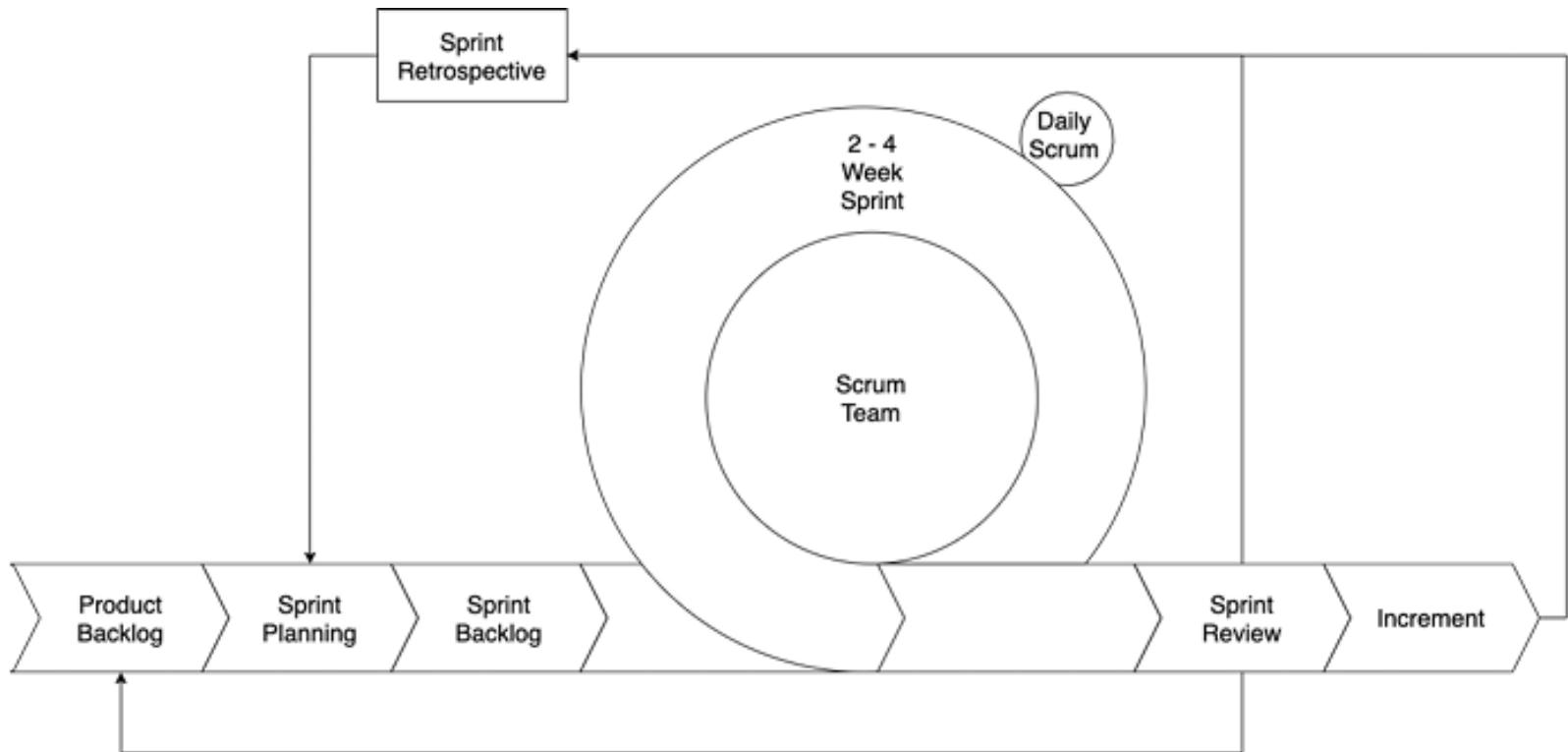
Scrum Roles

➤ What is Scrum?

- ❖ Scrum helps teams learn from their experiences, organize themselves while handling issues, and reflect on their successes and failures, so that they can improve.
- ❖ Scrum groups tasks into columns based on progress.
- ❖ Scrum usually breaks down a project into sprints, planning, and managing, one sprint at a time.
- ❖ Scrum also has unique project roles, which include a scrum master and Product Owner.

SCRUM

➤ What is Scrum?



Scrum Methodology

Scrum Roles

SCRUM

➤ **What is Scrum?**

□ **Practices that distinguish Scrum from other approaches**

❖ **Practices** that distinguish Scrum from other approaches to complex work include the following:

- ❖ The team organizes work in short cycles.
- ❖ Management doesn't interrupt the team during a work cycle
- ❖ The team reports to the client, not to the manager.
- ❖ The team estimates how much time and effort that work will take
- ❖ The team decides how much work it can do in an iteration
- ❖ The team decides how to do the iteration's work

SCRUM

➤ **What is Scrum?**

❑ **Practices that distinguish Scrum from other approaches**

- ❖ The team measures its own performance
- ❖ The team defines work goals before each cycle starts.
- ❖ The team defines work and desired outcomes through a progressively refined description of outcomes (called the Product Backlog).
- ❖ The team works to systematically and continuously improve and to remove impediments

Scrum Roles

SCRUM

➤ Scrum Practices Prescribed by the Scrum grouping

3 Roles

- Product Owner
- Scrum Master
- The Team
 - Skillset: cross-functional team to complete all their work with minimal dependencies
 - Size: 5-9 team members

4 Artifacts

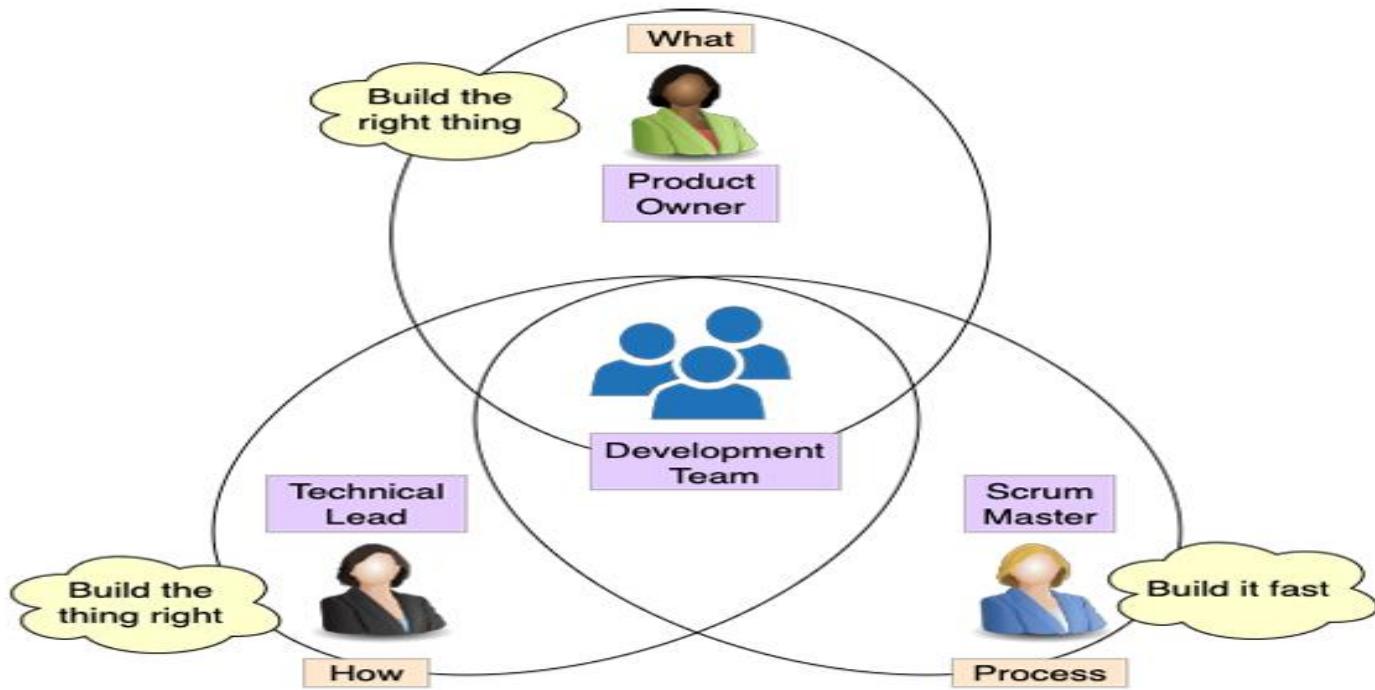
- Product Backlog
- Sprint Backlog
- Product Increment
- Definition of Done

5 Activities

- Product Backlog Refinement(PBR)
- Sprint Planning
- Daily Scrum
- Sprint Reviews (Demos)
- Sprint Retrospectives

SCRUM

➤ Scrum roles?



Scrum Roles

SCRUM

➤ Scrum Roles?

1. **The Product Owner (PO)** plays the liaison between customers and the development team. PO is the champion for their product. PO ensures that the expectation for the completed product has been communicated and agreed upon accordingly.
2. **The Development Team (DT)** works together to develop and test incremental releases of the final product. DT members are the champions for sustainable development practices. Scrum teams usually are five or seven members, tight-knit and co-located.
3. **Scrum Master (SM)** is the champion of Scrum within the team. It is the primary responsibility of SM **to ensure the best scrum practices in the project**. In addition, SM schedules the **needed resources for sprint planning, stand-up, sprint review, and the sprint retrospective**.

Scrum Roles

SCRUM

➤ How Does Scrum Work?

❖ It has the following Steps (Activities)

❑ Step 1: Product backlog

- ❖ The product backlog is used to draw up a list of tasks that a team must complete to successfully achieve the stakeholders' goals.

❑ Step 2: Sprint planning

- ❖ Selected tasks from the product backlog are chosen for teams to focus and work on, then ultimately be delivered during the sprint.

❑ Step 3: Sprint backlog

- ❖ Tasks discussed in the previous phase are added into the sprint backlog.

SCRUM

➤ How Does Scrum Work?

□ Step 4: Scrum team

- ❖ A scrum team is usually a team of **five to nine members** that work on the tasks mentioned in the **sprint backlog**.

□ Step 4.1: Daily scrum

- ❖ The team has **daily scrum meetings, 15 minute long sessions** during which the team members synchronize their activities with each other, **report on the bottlenecks** they are facing, and plan on what they aim to achieve in **the next 24 hours**.

SCRUM

➤ How Does Scrum Work?

□ Step 5: Sprint Review

- ❖ After the sprint is completed, it's time for a sprint review.
- ❖ The product owner, scrum master, stakeholders, and the scrum team attend the meeting.
- ❖ During this stage, the team discusses what they accomplished in the previous sprint.
- ❖ The session also opens up opportunities to ask questions, make observations, and provide feedback and suggestions.

SCRUM

❑ Step 5.1: Sprint review –

- ❖ **Product backlog**, the **product owner presents the backlog's top to the stakeholders.**
- ❖ This lets the former receive feedback for upcoming sprints and other things related to the backlog.

❑ Step 5.2: Sprint retrospective

- ❖ The sprint retrospective meeting follows the sprint review.
- ❖ Here, the **team identifies potential mistakes and issues, as well as ways to handle them.** Data from this stage is incorporated while planning the next sprint.

❑ Step 6: Increment

- ❖ The stakeholders receive a workable and usable output.

SCRUM

➤ Professional Scrum

- ❖ **Professional Scrum** is a framework for implementing Agile principles and practices developed by Scrum.org, founded by Ken Schwaber, one of the creators of Scrum.
- ❖ **Professional Scrum** provides a set of guidelines and tools for organizations to apply Scrum effectively in various contexts, emphasizing professionalism, empiricism, and continuous improvement.
- ❖ **Professional Scrum** plays a significant role in fostering business agility by **promoting key Agile values and principles, including:**
 1. **Customer-Centricity:** Professional Scrum emphasizes the importance of understanding and prioritizing customer needs, delivering value early and often, and incorporating customer feedback to drive product development.

SCRUM

➤ Professional Scrum

2. **Empirical Process Control:** Professional Scrum is based on empiricism, which involves making decisions based on observation, experimentation, and evidence. By regularly inspecting and adapting their processes, teams can respond rapidly to changes and optimize their performance over time.
3. **Iterative and Incremental Delivery:** Professional Scrum promotes iterative and incremental delivery, allowing teams to deliver valuable product increments frequently. This approach enables organizations to respond quickly to market feedback, mitigate risks, and adapt to changing requirements.
4. **Cross-Functional Teams:** Professional Scrum encourages the formation of self-organizing, cross-functional teams that have all the skills and capabilities necessary to deliver valuable product increments independently. This enables teams to respond quickly to changing priorities and requirements without relying on external dependencies.

➤ Professional Scrum

5. **Transparency and Collaboration:** Professional Scrum emphasizes transparency, collaboration, and open communication within and across teams. By fostering a culture of trust and collaboration, organizations can leverage the collective intelligence of their teams to address challenges and seize opportunities effectively.
 6. **Continuous Improvement:** Professional Scrum promotes a culture of continuous improvement, where teams regularly reflect on their practices, identify areas for enhancement, and experiment with new approaches. This iterative approach to improvement enables organizations to adapt to changing circumstances and continuously optimize their performance.
- **By embracing Professional Scrum and its Agile principles, organizations can enhance their business agility, enabling them to innovate, respond to market changes, and deliver value to customers more effectively and efficiently**

Large Scale SCRUM (LeSS)

➤ Common mistakes

- ❖ Achieving business agility with Professional Scrum, or any Agile framework, requires more than just adopting a set of practices.
- ❖ It involves a **cultural shift, mindset change, and continuous improvement**.
- ❖ **Here are some common mistakes** hindering teams and organizations from fully enabling business agility with Professional Scrum:
 1. **Treating Agile as a Process, Not a Mindset:** One common mistake is viewing Agile frameworks like Professional Scrum solely as a set of processes or practices to be implemented rather than embracing the underlying values and principles.
 - ❖ Business agility requires a cultural shift towards **collaboration, transparency, and adaptability**.

Large Scale SCRUM (LeSS)

➤ Common mistakes

2. **Lack of Leadership Support and Engagement:** Business agility relies on strong leadership support and active engagement. **Leaders** must champion **Agile values, empower teams, and remove organizational impediments to enable agility**. Without leadership buy-in, teams may struggle to implement Agile practices effectively.
3. **Incomplete or Inconsistent Implementation:** Implementing Agile practices in isolation or inconsistently across teams can hinder business agility. It's essential **to have a clear and consistent approach to Agile adoption, including standardized practices, shared norms, and continuous improvement mechanisms**.

Large Scale SCRUM (LeSS)

➤ Common mistakes

4. **Resisting Change and Embracing Status Quo:** Resistance to change is a significant barrier to business agility. Some organizations may resist Agile transformation due to **fear of the unknown, reluctance to disrupt existing processes, or ingrained cultural norms.** Overcoming resistance requires effective change management, communication, and stakeholder engagement.
5. **Micromanagement and Lack of Trust:** Micromanagement undermines **autonomy, creativity, and trust—the cornerstones of Agile teams.** Organizations that continue to micromanage teams, dictate solutions, or enforce command-and-control structures will struggle to achieve business agility. Trusting teams to self-organize and make decisions fosters innovation and agility.

Large Scale SCRUM (LeSS)

➤ Common mistakes

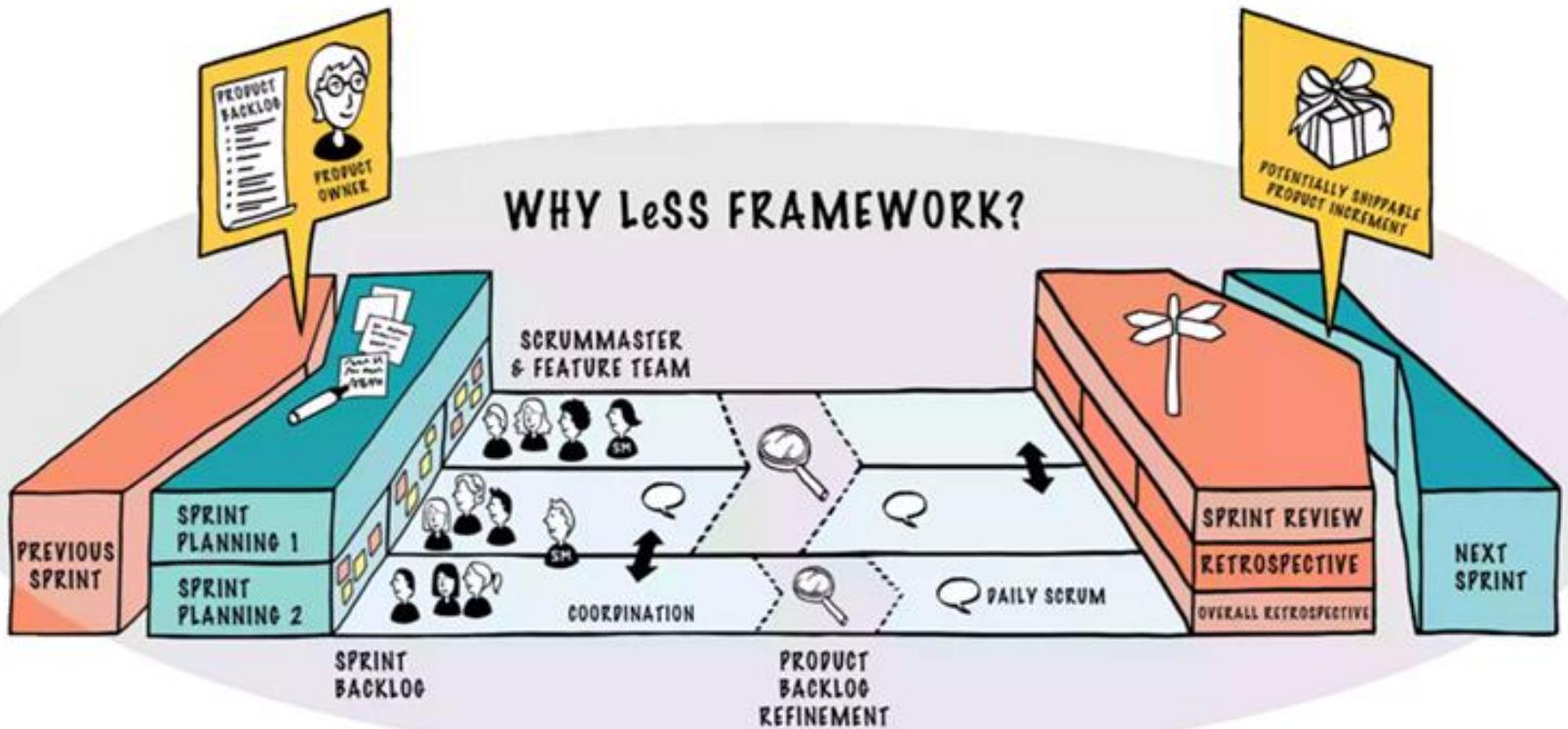
6. **Ignoring Technical Excellence:** Business agility relies on a foundation of technical excellence. Neglecting technical practices such as test automation, continuous integration, and refactoring can lead to technical debt, reduced productivity, and compromised quality. Organizations must prioritize technical excellence to sustain Agile delivery.
7. **Focusing Solely on Delivering Features:** Agile is not just about delivering features faster but also about delivering value to customers. Organizations that focus solely on output metrics, such as velocity or story points, without considering outcomes or customer feedback may miss the mark on business agility. Instead, they should prioritize delivering valuable, high-quality solutions that meet customer needs.

Large Scale SCRUM (LeSS)

- **Common mistakes**
- 8. **Lack of Continuous Learning and Improvement:** Business agility requires a culture of continuous learning and improvement. Organizations that fail to embrace a growth mindset, solicit feedback, or reflect on their practices will struggle to adapt and evolve. Continuous learning and improvement are essential for optimizing processes, fostering innovation, and sustaining business agility over time.
- By addressing these **common mistakes** and **fostering a culture of collaboration, empowerment, and continuous improvement, organizations can fully enable business agility with Professional Scrum or any Agile framework.**

Large Scale SCRUM (LeSS)

- Large-Scale Scrum is Scrum(LeSS)



Less Framework

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- ***Two Agile Scaling Frameworks***
 - ❖ LeSS provides two different large-scale Scrum frameworks. Most of the scaling elements of LeSS are focused on directing the **attention of all of the teams onto the whole product instead of “my part.”** Global and “end-to-end” focus are perhaps the dominant problems to solve in scaling.
 - ❖ The two frameworks – which are basically single-team Scrum scaled up – are:
 - 1. LeSS: Up to eight teams (of eight people each).**
 - 2. LeSS Huge: Up to a few thousand people on one product.**

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- ❖ **Less is an Agile framework designed to scale Scrum principles and practices for larger organizations working on complex products.**
- ❖ LeSS was developed by Craig Larman and Bas Vodde based on their extensive experience in scaling Agile in various organizations.
- ❖ **LeSS is about applying the principles, elements, and purpose of Scrum in a large-scale context.**
- ❖ **LeSS is more than a set of principles and experiments.**
- ❖ It also provides a framework with rules.

Large Scale SCRUM (LeSS)

- Large-Scale Scrum is Scrum(LeSS)
- ❖ The **LeSS Rules** define **what is LeSS (and what isn't)** and they **provide a concrete framework for applying LeSS**.
- ❖ Within the LeSS Framework, **product groups can apply the experiments and discover what works best for them at a certain moment**

Large Scale SCRUM (LeSS)

- Large-Scale Scrum is Scrum(LeSS)
- ❖ Properties of Large-Scale Scrum:
 - 1. Simplicity: LeSS emphasizes simplicity in scaling Agile. It strives to keep the framework as lightweight as possible, focusing on the core principles of Scrum and minimizing additional roles, artifacts, and ceremonies.
 - 2. Scrum at Scale: LeSS extends the Scrum framework to address the challenges of scaling Agile beyond single teams. It maintains the core elements of Scrum, including the *roles of Product Owner, Scrum Master, and Development Team, as well as the ceremonies of Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective*.

Large Scale SCRUM (LeSS)

- Large-Scale Scrum is Scrum(LeSS)
 - ❖ Properties of Large-Scale Scrum:.
3. **One Product Backlog:** In LeSS, **multiple Scrum teams working on the same product collaborate using a single Product Backlog.** This promotes **alignment, transparency, and shared understanding** of priorities across the organization. The Product Owner is responsible for managing and prioritizing the Product Backlog in collaboration with stakeholders.
4. **Feature Teams:** LeSS encourages the use of feature teams, where cross-functional teams are responsible for delivering end-to-end features or components of the product. ***Feature teams are self-organizing and autonomous, with all the skills necessary to deliver valuable increments of work.***

Large Scale SCRUM (LeSS)

- Large-Scale Scrum is Scrum(LeSS)
 - ❖ Properties of Large-Scale Scrum:.
- 5. Integration and Coordination:** LeSS emphasizes integration and coordination mechanisms **to ensure alignment and coherence across multiple teams**. This may include frequent synchronization events, such as Sprint Reviews and Scrum of Scrums, as well as architectural and technical practices to facilitate collaboration and integration.
- 6. Lean Thinking:** LeSS incorporates **principles of Lean thinking, emphasizing customer value, waste reduction, and continuous improvement**. Teams are encouraged to adopt Lean practices such as ***minimizing work in progress, optimizing flow, and maximizing the delivery of value to customers.***

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- ❖ **Properties of Large-Scale Scrum:**
- 7. **Organizational Design:** LeSS encourages organizations to adopt a flatter, more decentralized organizational structure to support agility. This may involve reducing layers of hierarchy, empowering teams to make decisions, and fostering a culture of collaboration and trust.
- 8. **Continuous Improvement:** Like Scrum, LeSS promotes a culture of continuous improvement, where teams regularly reflect on their practices, identify opportunities for enhancement, and experiment with new approaches. This iterative approach to improvement helps organizations adapt to changing circumstances and optimize their performance over time.

Large Scale SCRUM (LeSS)

- Large-Scale Scrum is Scrum(LeSS)
- ❖ Properties of Large-Scale Scrum:..
- In general, **Large-Scale Scrum (LeSS)** provides a framework for scaling Agile principles and practices in large and complex organizations.
- By focusing on *simplicity, collaboration, and continuous improvement, LeSS enables organizations to achieve greater agility, responsiveness, and value delivery at scale.*
- Large-Scale Scrum (LeSS) is built upon the same principles as traditional Scrum, but **it extends them to address the complexities of scaling Agile practices across multiple teams working on a single product.** Here are the main principles of Large-Scale Scrum (LeSS):

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- **Main principles of Large-Scale Scrum (LeSS):**
 1. **Empirical Process Control:** LeSS, like traditional Scrum, is based on the principle of empirical process control, which emphasizes transparency, inspection, and adaptation. Teams regularly inspect their progress, adapt their plans based on feedback, and strive for continuous improvement.
 2. **Customer Focus:** LeSS maintains a strong focus on delivering value to customers. All decisions and activities within LeSS are driven by the goal of meeting customer needs and delivering high-quality products that provide value.
 3. **Self-Organizing Teams:** Teams in LeSS are self-organizing and cross-functional, with all the skills necessary to deliver valuable increments of work. Self-organizing teams have the autonomy to make decisions, plan their work, and collaborate effectively to achieve their goals.

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- **Main principles of Large-Scale Scrum (LeSS):**
- 4. **Single Product Backlog:** LeSS promotes the use of a single Product Backlog for the entire product, shared by all teams working on it. This promotes alignment, transparency, and collaboration across teams and ensures a unified view of priorities for the product.
- 5. **Whole Product Focus:** LeSS encourages a whole-product focus, where teams are responsible for delivering end-to-end features or components of the product. This helps to minimize dependencies and optimize flow, allowing teams to deliver value more quickly and efficiently.

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- **Main principles of Large-Scale Scrum (LeSS):**
 6. **Continuous Improvement:** Like traditional Scrum, LeSS promotes a culture of continuous improvement, where teams regularly reflect on their practices, identify areas for improvement, and experiment with new approaches. This iterative approach to improvement helps teams adapt to changing circumstances and optimize their performance over time.
 7. **Lean Thinking:** LeSS incorporates principles of Lean thinking, emphasizing **customer value, waste reduction, and optimization of flow**. Teams are encouraged to adopt Lean practices such as *minimizing work in progress, optimizing cycle times, and maximizing the delivery of value to customers*.
 8. **System Thinking:** LeSS encourages teams to take a holistic view of the system in which they operate, considering the interactions and dependencies between

Large Scale SCRUM (LeSS)

- **Large-Scale Scrum is Scrum(LeSS)**
- **Main principles of Large-Scale Scrum (LeSS):**
- 8. **System Thinking:** LeSS encourages teams to take a holistic view of the system in which they operate, considering the interactions and dependencies between teams, as well as the broader organizational context. **System thinking** helps teams identify and **address bottlenecks, optimize flow, and improve overall performance.**
- By adhering to these principles, organizations can **effectively scale Agile practices with LeSS, enabling them to achieve greater agility, responsiveness, and value delivery at scale.**

Large Scale SCRUM (LeSS) Structure

➤ Why Less

- ❖ Traditional sequential-lifecycle development doesn't work well.
- ❖ It doesn't work well for either small or large product development efforts
- ❖ Since 2001, Agile development and Scrum in particular have revolutionized software development, but when asked how to apply Agile development to large groups many people say **“don’t”** or **“just use a small team”** or **“do Scrum at the team level.”**

Large Scale SCRUM (LeSS) Structure

➤ Why Less

- ❖ None of those answers is particularly useful, and while it is true that it is best to avoid **adding people to your development effort.**
- ❖ It is also true that large scale product development isn't going away so we need to discover ways to do it well.

Large Scale SCRUM (LeSS) Structure

- **Why Less**
- ❖ **None felt right.** Scrum, on the other hand, **felt right for single-team development.**
- ❖ So, the question then became “**How can we scale Scrum without losing its strength?**”
- ❖ The concepts and principles behind Scrum, such as **Transparency, Empirical Process Control, Iterative development, and Self-managing teams** are critical.

Large Scale SCRUM (LeSS) Structure

➤ Why Less

- ❖ Scrum hits the sweet spot between **abstract principles** and **concrete practices**.
- ❖ Thus, in order to keep **Large-scale Scrum as Scrum**, we'll need to find a similar balance,
- ❖ so that we will be able to say: For large groups, LeSS hits the sweet spot between **defined concrete elements and empirical process control**.
- ❖ This leads to some decisions:
- ❖ **LeSS** needs to be simple when scaling, there is a tendency to add **roles, artifacts, processes, etc.**

Large Scale SCRUM (LeSS) Structure

➤ Why Less

- ❖ This should be avoided so that a process can empirically be created by the product group.
- ❖ Most other scaling frameworks fall into the trap of providing a defined process.
- ❖ In LeSS we want to avoid that trap.
- ❖ LeSS is **Scrum Scaled** Rather than having **Scrum as a building block for a scaled framework** .
- ❖ we need to look at Scrum and for each element ask “**Why is it there?**” followed by “**If we have more than one team, how can we achieve the same purpose on a larger scale?**”

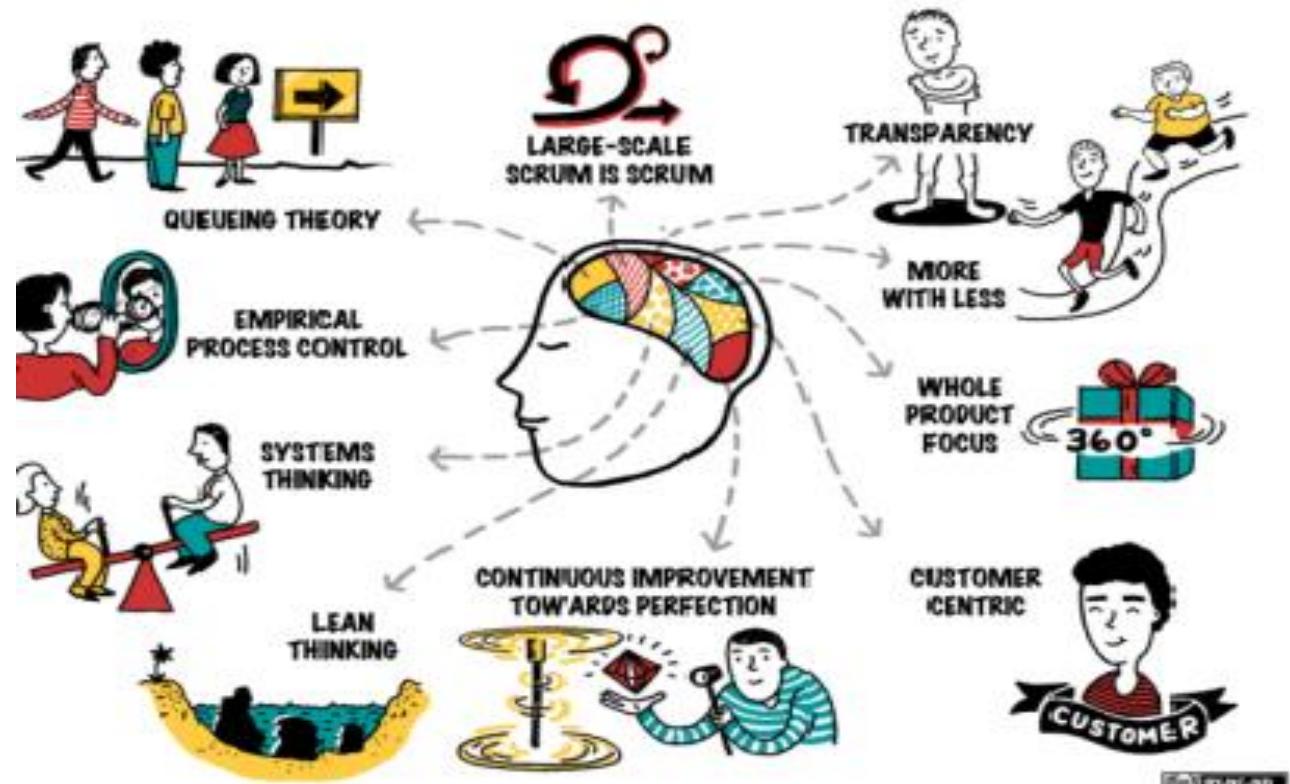
Large Scale SCRUM (LeSS) Structure

➤ Why Less

- ❖ **Scaled up instead of tailored down** - A common concept for process development is to define a **universal, overarching framework** and then contextually modify it down.
- ❖ This does not work well because people often assume everything is needed in their particular context.
- ❖ This assumption often leads to the creation of bloated processes.
- ❖ LeSS should be kept to the minimum. We acknowledge that scaling will require “more” but instead of “polluting” LeSS with optional elements, we have separated out **a different framework LeSS Huge**.

Large Scale SCRUM (LeSS) Structure

- LeSS Product
- LeSS Sprint
- More with LeSS





Large Scale SCRUM (LeSS) Structure

- **LeSS Framework Rules**
- **The LeSS framework applies to products with 2-“8” teams.**
 - ❖ **LeSS structure**
 - ❖ **LeSS Product**
 - ❖ **LeSS Sprint**
 - ❖ **More with LeSS**
- ❖ In Large-Scale Scrum (LeSS), the structure, product management, and sprint execution follow principles and practices adapted for larger-scale Agile implementations

- 
1. **Basic Structure:** LeSS maintains the basic structure of Scrum, including roles, events, artifacts, and rules. However, it extends these elements
 2. **Team Organization:** LeSS recommends organizing multiple teams into one Product Owner team, one Product Backlog, one Definition of Done, and a common Sprint length.
 3. **Area Product Owners (APOs):** In larger LeSS implementations, Area Product Owners may be introduced to work with the Product Owner in areas where one Product Owner cannot manage all the backlog items effectively. APOs help maintain alignment and clarity within their respective areas.

Large Scale SCRUM (LeSS) Structure

➤ LeSS structure

- ❖ **Basic Structure:** LeSS maintains the basic structure of Scrum, including **roles, events, artifacts, and rules**. However, it extends these elements to support coordination and collaboration across multiple teams.
- ❖ **Team Organization:** LeSS recommends organizing multiple teams into one **Product Owner team, one Product Backlog, one Definition of Done, and a common Sprint length**. Teams remain **autonomous and cross-functional**, but they **collaborate closely to deliver a single product increment**.
- ❖ **Area Product Owners (APOs):** In larger LeSS implementations, Area Product Owners may be introduced to work with the Product Owner in areas where one **Product Owner cannot manage all the backlog items effectively**. APOs help maintain alignment and clarity within their respective areas.

- 1. **Basic Structure:** LeSS maintains the basic structure of Scrum, including roles, events, artifacts, and rules. However, it extends these elements
- 2. **Team Organization:** LeSS recommends organizing multiple teams into one Product Owner team, one Product Backlog, one Definition of D
- 3. **Area Product Owners (APOs):** In larger LeSS implementations, Area Product Owners may be introduced to work with the Product Owner

Large Scale SCRUM (LeSS) Structure

➤ LeSS structure

- ❖ **Structure** the organization using real teams as the basic organizational building block.
- ❖ Each team is
 1. **self-managing,**
 2. **cross-functional,**
 3. **co-located, and**
 4. **long-lived.**
- ❖ The majority of the teams are customer-focused feature teams.
- ❖ Scrum Masters are responsible for a well-working LeSS adoption.

Large Scale SCRUM (LeSS) Structure

➤ LeSS structure

- ❖ Their focus is towards the **Teams, Product Owner, organization, and development practices.**
- ❖ **A Scrum Master** does not focus on just one team but on the overall organizational system.
- ❖ **A Scrum Master** is a dedicated full-time role.
- ❖ One Scrum Master can serve **1-3 teams**.
- ❖ **In LeSS, managers are optional**, but if managers do exist their role is likely to change. Their focus shifts from managing the day-to-day product work to improving the value-delivering capability of the product development system.

Large Scale SCRUM (LeSS) Structure

➤ LeSS structure

- ❖ **Managers'** role is to improve the product development system by practicing Go See, encouraging Stop & Fix, and “experiments over conformance”.
- ❖ For the product group, establish the complete LeSS structure “at the start”; this is vital for a LeSS adoption.
- ❖ For the **larger organization** beyond the product group, adopt LeSS evolutionarily using Go and See to create an organization where experimentation and improvement is the norm.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Product

- ❖ **Single Product Backlog:** LeSS promotes the use of **a single Product Backlog shared by all teams working on the product.** The Product Backlog contains all the features, enhancements, and technical items needed to deliver the product. It is owned and prioritized by the **Product Owner** in collaboration with **stakeholders**.
- ❖ **Definition of Done (DoD):** LeSS maintains a **single Definition of Done for the entire product, ensuring consistency and quality across all teams.** The Definition of Done specifies the criteria that must be met for a product increment to be considered complete and potentially shippable.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Product

- ❖ **Product Owner Collaboration:** Product Owners collaborate closely to ensure alignment, manage dependencies, and prioritize the Product Backlog effectively. They work together to define the overall product vision, goals, and roadmap.
- ❖ There is **one Product Owner and one Product Backlog** for the complete shippable product.
- ❖ The **Product Owner** shouldn't work alone on **Product Backlog** refinement; he is supported by the **multiple Teams working** directly with **customers/users** and **other stakeholders**.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Product

- ❖ All prioritization goes through the Product Owner, but clarification is as much as possible directly between the Teams and customer/users and other stakeholders.
- ❖ The definition of product should be as broad and **end-user/customer centric** as is practical.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Product

- ❖ Over time, the definition of product might expand. Broader definitions are preferred.
- ❖ One Definition of Done for the whole product common for all teams.
- ❖ Each team can have their own stronger Definition of Done by expanding the common one.
- ❖ The perfection goal is to improve the Definition of Done so that it results in a shippable product each Sprint (or **even more frequently**)
- **Common Sprint Length:** LeSS recommends using a common Sprint length across all teams working on the product. This promotes synchronization, predictability, and alignment within the organization. Sprints typically last 2-4 weeks, with shorter Sprints preferred to enable faster feedback and adaptation.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Sprint

- ❖ **Common Sprint Length:** LeSS recommends using a **common Sprint length across all teams working on the product.** This promotes synchronization, predictability, and alignment within the organization. Sprints typically last 2-4 weeks, with shorter Sprints preferred to enable faster feedback and adaptation.
- ❖ **Sprint Planning:** During Sprint Planning, teams collectively plan the work for the Sprint based on the items from the Product Backlog. Teams collaborate **to identify dependencies, resolve ambiguities, and ensure a common understanding of the Sprint goals and deliverables.**
- ❖ **Sprint Review:** At the end of each Sprint, all teams participate in a joint Sprint Review to demonstrate the product increment to stakeholders and gather feedback. This promotes **transparency, collaboration, and shared learning across the organization.**

Large Scale SCRUM (LeSS) Structure

➤ LeSS Sprint

- ❖ **Sprint Retrospective:** Teams conduct joint Sprint Retrospectives to **reflect on their performance, identify improvement opportunities, and make adjustments to their processes.** Retrospectives focus on **enhancing collaboration, eliminating impediments, and optimizing performance across all teams.**
- ❖ There is **one product-level Sprint, not a different Sprint for each Team.**
- ❖ Each Team starts and ends the Sprint at the same time. Each Sprint results in an integrated whole product.
- ❖ Sprint Planning consists of two parts:
 - Sprint Planning One is **common for all teams** while
 - Sprint Planning Two is usually done **separately for each team.** Do multi-team Sprint Planning Two in a shared space for closely related items.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Sprint

- ❖ Sprint Planning One is attended by the **Product Owner and Teams or Team representatives.**
- ❖ They together tentatively select the items that each team will work on that Sprint.
- ❖ The Teams identify opportunities to work together and final questions are clarified.
- ❖ Each Team has their own Sprint Backlog.
- ❖ Sprint Planning Two is for Teams to decide how they will do the selected items.
- ❖ This usually **involves design and the creation of their Sprint Backlogs.**
- ❖ Each Team has their own Daily Scrum.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Sprint

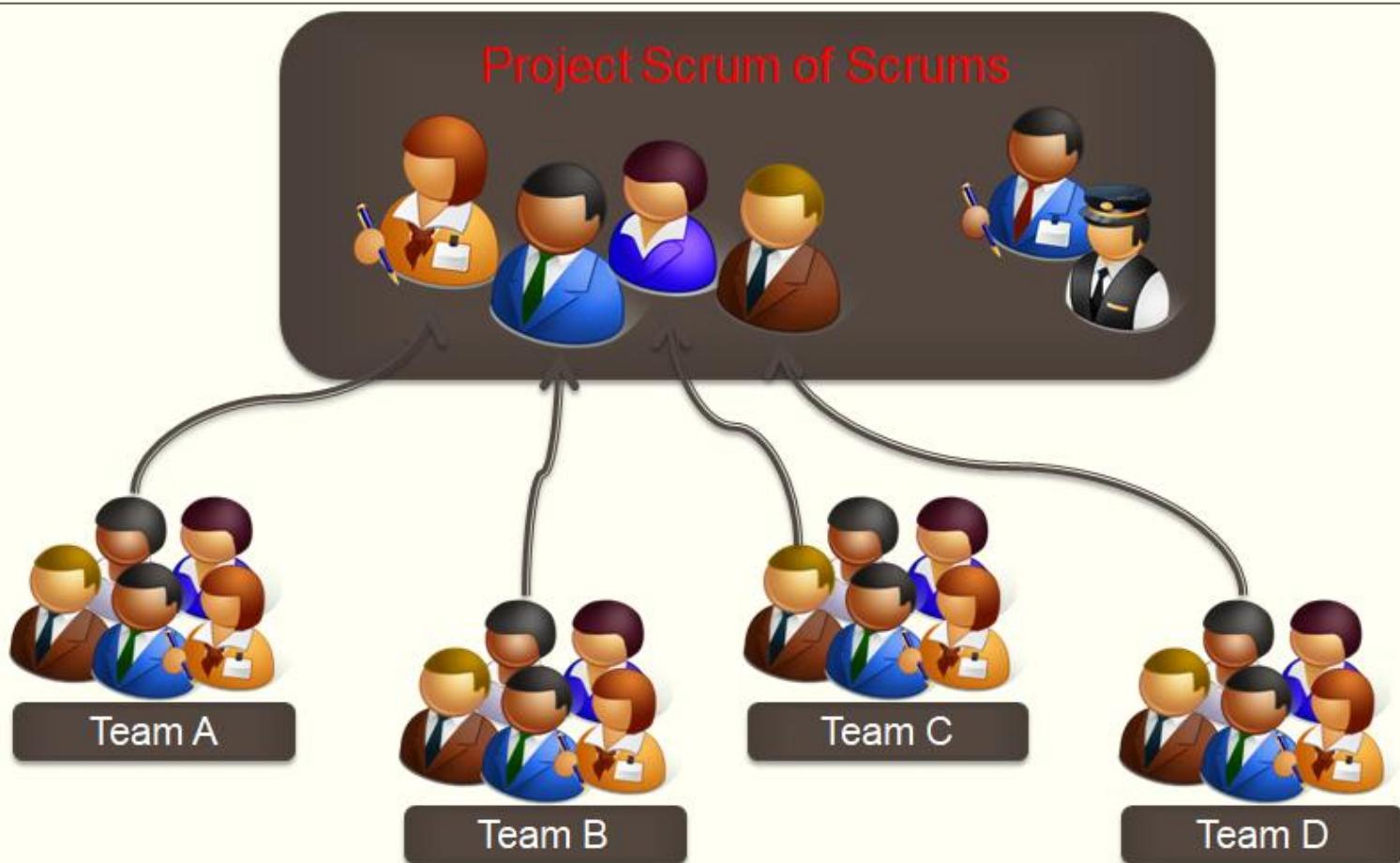
- ❖ Cross-team coordination is decided by the teams. Prefer **decentralized** and **informal coordination over centralized coordination.**
- ❖ Emphasize Just Talk and informal networks via communicate in code, cross-team meetings, component mentors, travelers, scouts, and open spaces.
- ❖ **Product Backlog Refinement (PBR)** is preferably done with multiple teams to increase shared learning and to exploit coordination opportunities.
- ❖ There is one product Sprint Review; it is common for all teams.
- ❖ Ensure that suitable stakeholders join to contribute the information needed for effective inspection and adaptation.

Large Scale SCRUM (LeSS) Structure

- **LeSS Sprint**
 - ❖ Each Team has their own Sprint Retrospective.
 - ❖ An Overall Retrospective is held after the Team Retrospectives to discuss **cross-team and system-wide issues, and create improvement experiments.**
 - ❖ This is attended by **Product Owner, Scrum Masters, Team representatives, and managers (if any).**
- **By adopting LeSS structure, managing the LeSS product effectively, and executing LeSS sprints collaboratively, organizations can scale Agile practices successfully and deliver value more efficiently in larger-scale environments.**
- **Scrum of scrum meetings**

Large Scale SCRUM (LeSS) Structure

- Scrum of scrum meetings



Large Scale SCRUM (LeSS) Structure

➤ Scrum of scrum meetings

- ❖ **Scrum of Scrums (SoS) meetings** are a coordination mechanism used in Agile and Scrum methodologies, especially in large-scale Agile implementations where multiple teams are working on the same product or project.
- ❖ **The purpose of Scrum of Scrums meetings** is to facilitate **communication, collaboration, and alignment between multiple Scrum teams**.
- ❖ **Composition:**
 - ❖ Scrum of Scrums meetings typically involve representatives from each Scrum team working on the same product or project.
 - ❖ Each team selects one or more members, often the Scrum Master or a designated team representative, to attend the Scrum of Scrums meeting.
 - ❖ The size and composition of the Scrum of Scrums group may vary depending on the number of teams and the complexity of the project.

Large Scale SCRUM (LeSS) Structure

➤ Scrum of scrum meetings

❖ Frequency:

- ❖ Scrum of Scrums meetings are typically held **at regular intervals**, usually **daily or several times per week**, depending on the cadence of the project and the need for coordination.
- ❖ The frequency of Scrum of Scrums meetings may be adjusted based **on the level of complexity, dependencies, and the pace of work**.

❖ Objectives:

- ❖ The primary objective of Scrum of Scrums meetings is to facilitate communication and coordination between teams.
- ❖ Scrum of Scrums meetings provide a forum for teams to share updates, discuss dependencies, identify impediments, and align on priorities.
- ❖ By sharing information and collaborating effectively, Scrum of Scrums meetings help ensure that all teams are aligned towards common goals and objectives.

Large Scale SCRUM (LeSS) Structure

➤ Scrum of scrum meetings

❖ Agenda:

- ❖ Scrum of Scrums meetings typically follow a structured agenda to ensure that key topics are addressed efficiently.
- ❖ Common agenda items may include updates on progress and blockers, discussions on cross-team dependencies, coordination of upcoming work, and resolution of impediments.
- ❖ The agenda may be flexible and may evolve over time based on the needs and priorities of the teams.

❖ Facilitation:

- ❖ Scrum of Scrums meetings are facilitated by one of the attendees, often rotating among team representatives.
- ❖ The facilitator is responsible for guiding the discussion, ensuring that all relevant topics are addressed, and keeping the meeting focused and on track.

Large Scale SCRUM (LeSS) Structure

➤ Scrum of scrum meetings

- ❖ Facilitation techniques such as timeboxing, visual aids, and active listening may be used to enhance the effectiveness of the meeting.

❖ Outputs:

- ❖ The outputs of Scrum of Scrums meetings may include action items, decisions, and agreements reached during the discussion.
- ❖ Teams may also identify dependencies, impediments, and risks that require further attention or resolution.
- ❖ The information shared and decisions made during Scrum of Scrums meetings help teams coordinate their efforts, address challenges, and deliver value more effectively.

- ❖ In general **Scrum of Scrums meetings** play a crucial role in large-scale Agile implementations by promoting **collaboration, alignment, and transparency between multiple Scrum teams**. By facilitating communication and coordination, Scrum of Scrums meetings help organizations **overcome challenges associated with scaling Agile practices and deliver value more efficiently**.

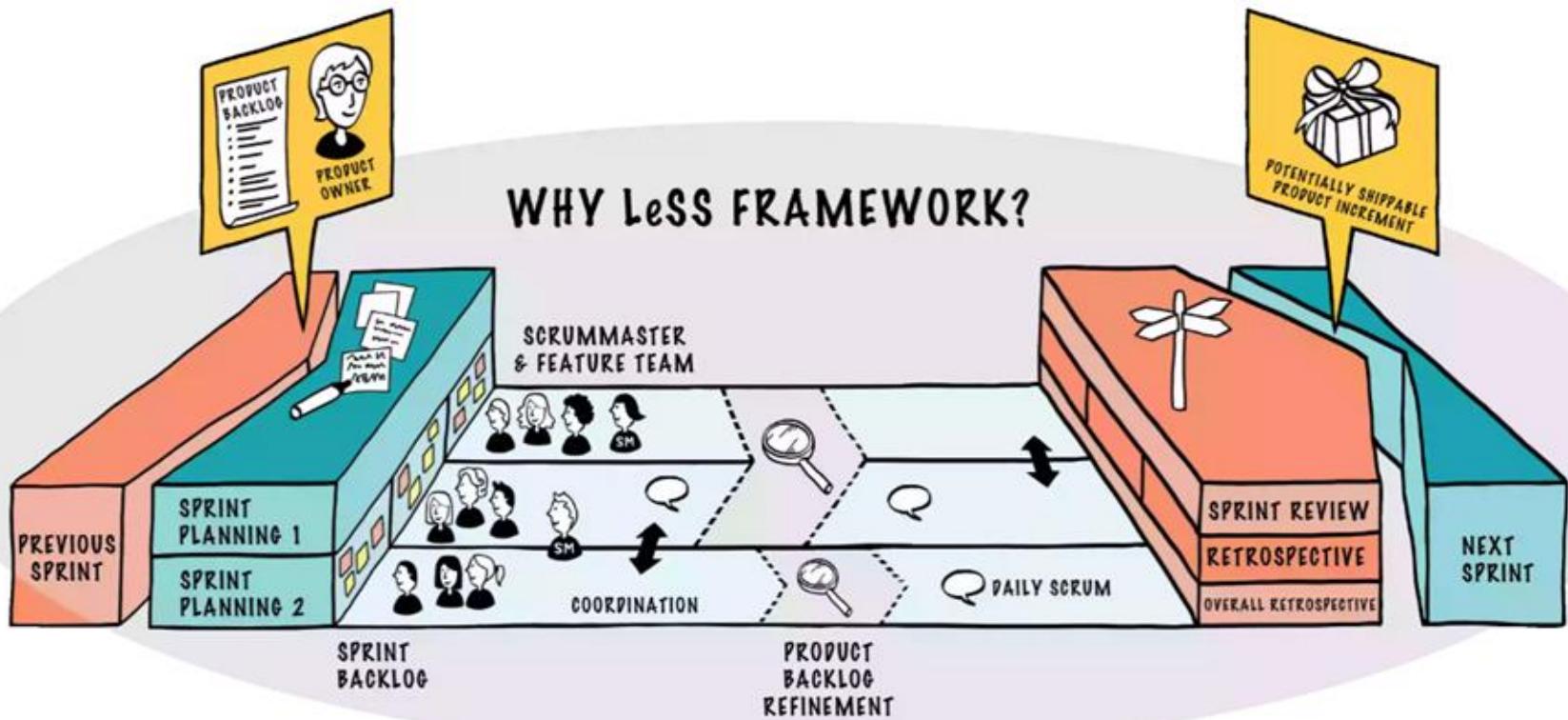
Large Scale SCRUM (LeSS) Structure

- **The Daily Scrum Meeting**
- ❖ **The daily scrum meeting** generally co-occurs to discuss the necessary task to reach the sprint goal.
- ❖ This informal meeting usually lasts **10-15 minutes**, and the scrum master (SM) must take the initiative to make it happen daily.
- ❖ It is **not an extensive discussion meeting**, and it is also called '**daily stand-up.**'
- ❖ The motive of the daily scrum meeting is **to keep each member of the scrum team in exact alignment.**
- ❖ A work plan is made for the **next 24 hours** without compromising the goal.

Large Scale SCRUM (LeSS) Structure

- **The Daily Scrum Meeting**
- ❖ The team members usually keep focusing on the following three questions for the sprint goal in this meeting –
 1. **What did I do yesterday?**
 2. **What do I plan to do today?**
 3. **Are there any obstacles that may avert meeting the goal?**

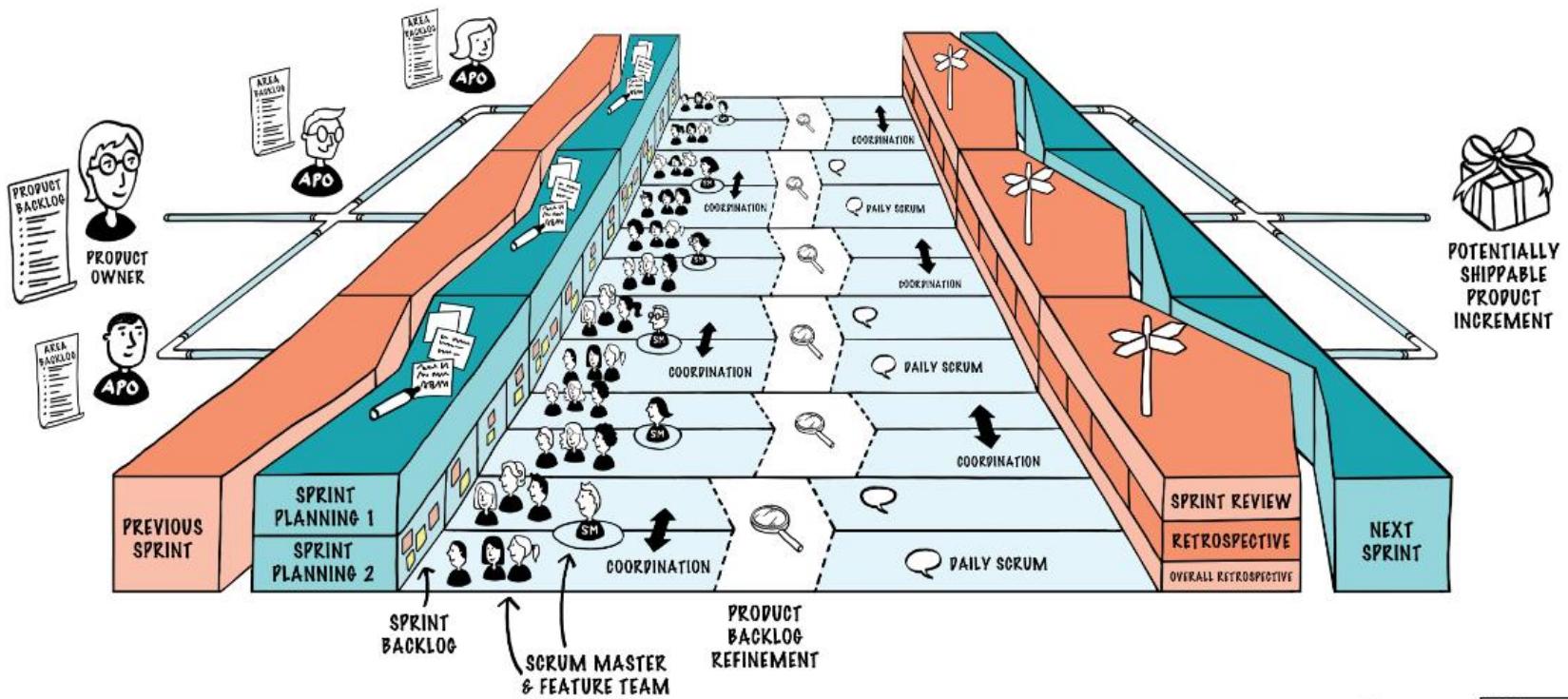
Large Scale SCRUM (LeSS) Structure



LeSS Framework

Large Scale SCRUM (LeSS) Structure

➤ LeSS Huge



➤ Less Huge Framework

Large Scale SCRUM (LeSS) Structure

➤ **LeSS Huge**

- ❖ LeSS Huge framework is an adaptation of the LeSS framework designed for organizations with hundreds or even thousands of people working on a single product.
- ❖ LeSS Huge maintains the core principles and practices of LeSS while providing additional guidance and structures to support larger-scale Agile implementations.
- ❖ Here's an overview of the rules in LeSS Huge framework:

1. One Product Backlog:

- ❖ LeSS Huge follows the principle of having one Product Backlog for the entire product, shared by all teams working on it.
- ❖ The Product Backlog contains all the features, enhancements, and technical items needed to deliver the product.
- ❖ Product Backlog items are prioritized based on business value and stakeholder input.

Large Scale SCRUM (LeSS) Structure

➤ **LeSS Huge Framework Rules**

2. Multiple Requirement Areas (MRAs):

- ❖ In LeSS Huge, the Product Backlog may be partitioned into Multiple Requirement Areas (MRAs) to manage large product backlogs more effectively.
- ❖ Each MRA is managed by an Area Product Owner (APO) responsible for prioritizing the backlog items within their area and coordinating with the overall Product Owner.

3. LeSS Huge Scrum Teams:

- ❖ Scrum Teams in LeSS Huge are organized into Feature Teams, each responsible for delivering end-to-end features or components of the product.
- ❖ Feature Teams are cross-functional and self-organizing, with all the skills necessary to deliver valuable increments of work.
- ❖ Feature Teams may be further organized into component teams or sub-teams to manage large-scale development efforts.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Huge Framework Rules

4. LeSS Huge Coordination:

- ❖ LeSS Huge emphasizes coordination mechanisms to ensure alignment and coherence across multiple teams.
- ❖ Coordination may involve regular synchronization events, such as the LeSS Huge Requirement Area Backlog Refinement and the LeSS Huge Product Backlog Refinement meetings.
- ❖ Coordination may also include architectural and technical practices to facilitate collaboration and integration between teams.

5. LeSS Huge Adoption:

- ❖ LeSS Huge encourages organizations to adopt the framework incrementally, starting with a small number of teams and expanding gradually.
- ❖ Organizations should focus on establishing a strong foundation of Agile practices, principles, and values before scaling further.
- ❖ LeSS Huge adoption should be guided by a continuous improvement mindset, with teams regularly reflecting on their practices and making adjustments based on feedback.

Large Scale SCRUM (LeSS) Structure

➤ LeSS Huge Framework Rules

6. Lean Thinking and Systems Thinking:

- ❖ LeSS Huge incorporates principles of Lean thinking and systems thinking to optimize flow, minimize waste, and enhance overall performance.
- ❖ Teams are encouraged to adopt Lean practices such as minimizing work in progress, optimizing cycle times, and maximizing the delivery of value to customers.

➤ In general **LeSS Huge framework** provides a structured approach to scaling Agile practices in large organizations, enabling them to achieve greater agility, responsiveness, and value delivery at scale. By following the rules and principles of LeSS Huge, organizations can effectively coordinate efforts across multiple teams and deliver high-quality products that meet customer needs.

Large Scale SCRUM (LeSS) Structure

- **LeSS Huge**
- **LeSS Huge Framework Rules**
- ❖ LeSS Huge applies to products with “8+” teams.
- ❖ Avoid applying LeSS Huge for **smaller product groups** as it will result in more overhead and local optimizations.
- ❖ All LeSS rules apply to LeSS Huge, unless otherwise stated. Each Requirement Area acts like the basic LeSS framework.

Large Scale SCRUM (LeSS) Structure

➤ **LeSS Huge**

- ❖ The Large-Scale Scrum (LeSS) Huge framework is built upon the same principles as the LeSS framework, but it provides additional guidance and structures to support larger-scale Agile implementations.

Large Scale SCRUM (LeSS) Structure

- **LeSS huge structure, LeSS huge product, and LeSS huge sprint**
- ❖ In the Large-Scale Scrum (LeSS) Huge framework, the structure, management of the product backlog, and sprint execution are adapted to support large-scale Agile implementations. Here's an overview of LeSS Huge structure, LeSS Huge product, and LeSS Huge sprint:
- **LeSS Huge Structure:**
 - ❖ Multiple Requirement Areas (MRAs): In LeSS Huge, the product is divided into Multiple Requirement Areas (MRAs) to manage large-scale development efforts effectively. Each MRA is a collection of related product features or components.
 - ❖ Area Product Owners (APOs): Each MRA is managed by an Area Product Owner (APO) who is responsible for prioritizing the backlog items within their area and coordinating with the overall Product Owner.
 - ❖ LeSS Huge Teams: Teams in LeSS Huge are organized into Feature Teams, each responsible for delivering end-to-end features or components of the product. Feature Teams are cross-functional and self-organizing, with all the skills necessary to deliver valuable increments of work.

Large Scale SCRUM (LeSS) Structure

- **LeSS huge structure, LeSS huge product, and LeSS huge sprint**
- **LeSS Huge product :**
 - ❖ **One Product Backlog:** LeSS Huge follows the principle of having one Product Backlog for the entire product, shared by all teams working on it. The Product Backlog contains all the features, enhancements, and technical items needed to deliver the product.
 - ❖ **Multiple Requirement Area Backlogs:** The Product Backlog may be partitioned into Multiple Requirement Area Backlogs to manage large product backlogs more effectively. Each MRA Backlog corresponds to an MRA and is managed by the respective Area Product Owner.

Large Scale SCRUM (LeSS) Structure

➤ LeSS huge structure, LeSS huge product, and LeSS huge sprint

□ LeSS Huge sprint :

- ❖ **Common Sprint Length:** LeSS Huge recommends using a common Sprint length across all teams working on the product. This promotes synchronization, predictability, and alignment within the organization.
- ❖ **Sprint Planning:** During Sprint Planning, teams collectively plan the work for the Sprint based on the items from the Product Backlog. Teams collaborate to identify dependencies, resolve ambiguities, and ensure a common understanding of the Sprint goals and deliverables.
- ❖ **Sprint Review:** At the end of each Sprint, all teams participate in a joint Sprint Review to demonstrate the product increment to stakeholders and gather feedback. This promotes transparency, collaboration, and shared learning across the organization.
- ❖ **Sprint Retrospective:** Teams conduct joint Sprint Retrospectives to reflect on their performance, identify improvement opportunities, and make adjustments to their processes. Retrospectives focus on enhancing collaboration, eliminating impediments, and optimizing performance across all teams.

Large Scale SCRUM (LeSS) Structure

- **LeSS huge structure, LeSS huge product, and LeSS huge sprint**
- **LeSS Huge sprint :**
 - ❖ By following the **LeSS Huge structure, managing the LeSS Huge product effectively, and executing LeSS Huge sprints collaboratively**, organizations can scale Agile practices successfully and deliver value more efficiently in large-scale environments.
 - ❖ Fundamental information gathering skills are crucial within Agile methodologies as they ensure teams have the necessary insights to make informed decisions and adapt their approaches effectively. Here's how key information gathering skills like interviewing, brainstorming, and observation are applied in Agile contexts:

Fundamental Information Gathering Skills

- ❖ Fundamental information gathering skills are crucial within Agile methodologies as they ensure teams have the necessary insights to make informed decisions and adapt their approaches effectively.
- ❖ **Data gathering** is a process of preparing for and then collecting data needed to facilitate project decision making.
- ❖ Data is collected to provide information regarding a specific topic, and the information obtained is kept on record and used to make decisions about important issues.
- ❖ Because of this, it is important to understand the quality of the data collected in terms of its **accuracy, reliability and relevance**.

Fundamental Information Gathering Skills

➤ Some Information Gathering Skills:

- ❖ Interviewing
- ❖ Observation
- ❖ Brainstorming

Fundamental Information Gathering Skills

❑ Interviewing:

- ❖ **Purpose:** Interviews help Agile teams gather firsthand information from stakeholders, users, or subject matter experts to understand their perspectives, needs, and requirements.
- ❖ **Techniques:** Agile teams conduct structured interviews where they ask open-ended questions to elicit detailed responses. They may use techniques such as user story mapping interviews or stakeholder interviews.
- ❖ **Benefits:** Interviews provide valuable insights into user needs, expectations, pain points, and priorities. They help teams gain empathy and develop a deeper understanding of the problem domain.

Fundamental Information Gathering Skills

Interviewing:

- ❖ **The interview** is the most flexible and direct tool for eliciting requirements. It also is more prone to misunderstanding and failure than any other technique.
- ❖ **Interview** with **stakeholders of the system—users, sponsors, and domain experts**— is an indispensable tool not only for its verbal responses but also because it allows one to observe and learn from nonverbal reactions such as: **attitudes, body language, and emotional overtones**.
- ❑ These nonverbal connotations put the requirements into a human context that must be taken into account to build a successful information system.
- ❖ The interview is also a **sensitive tool**: It is very flexible but exactly because of its flexibility it can lead us to blind alleys.

Fundamental Information Gathering Skills

- **Interviewing**
- ❖ Since an interview is limited by a timeframe it is also **a precious commodity**, the use of which must be planned with great care.
- ❖ **Interviewing** needs a great amount of “people skill” and experience, but there are things that we must know and actions that we must take to increase its chances of success.
- ❖ **Early interviews** in the requirements gathering process have a **broad scope**, while **later ones** are defined by **more focus on detail**.

Fundamental Information Gathering Skills

➤ Interview: The Building Blocks



To maximize the benefits of an interview and control its flow, the interviewer must construct the interview from building blocks with predefined scope and identifiable goals.

Types of the interview techniques

- ◆ Unstructured interview.
- ◆ Structured interview.
- ◆ Semi structured interview.

Fundamental Information Gathering Skills

- Assginment:
- ❖ Identify the difference of the 3 types of interview.
 1. Unstructured interview.
 2. Structured interview.
 3. Semi structured interview.

Fundamental Information Gathering Skills

➤ Observation

- ❖ **Purpose:** Observation involves directly observing users, stakeholders, or processes to gain insights into their behaviors, interactions, and pain points.
- ❖ **Techniques:** Agile teams may conduct user observation sessions, where they observe users in their natural environment as they interact with a product or service. They may also use techniques like shadowing or ethnographic research.
- ❖ **Benefits:** Observation provides firsthand, real-time insights into user behaviors, preferences, and challenges. It helps teams identify usability issues, uncover hidden needs, and validate assumptions, leading to more informed decision-making.

Fundamental Information Gathering Skills

➤ **Observation**

- ❖ **Observation** needs something that can be observed.
- ❖ **Observation** of individuals functioning within teams leads to the following list of characteristics of an effective team member.
- ❖ Such a person(observer):
 - Understands and is committed to group goals;
 - Is friendly, concerned, and interested in others;
 - Acknowledges and confronts conflict openly;
 - Listens to others with understanding;
 - Includes others in the decision-making process;

Fundamental Information Gathering Skills

➤ **Observation**

- Recognizes and respects individual differences;
- Contributes ideas and solutions;
- Values the ideas and contributions of others;
- Recognizes and rewards team efforts; and
- Encourages and appreciates comments about team performance

Fundamental Information Gathering Skills

➤ Brainstorming

- ❖ The term **brainstorming** has now become the accepted way of referring to group attempts to **solve specific problems or develop new ideas** by amassing spontaneous, unrestrained contributions by members.
- ❖ **The technique of brainstorming** utilizes the logical steps in problem solving by clearly separating the suggestion of solutions from the discussion of their value.
- ❖ **The basic rule in brainstorming** is that no one can comment on or in any way belittle the suggestion of another member.
- ❖

Fundamental Information Gathering Skills

➤ Brainstorming

- ❖ **Purpose:** Brainstorming sessions in Agile methodologies foster creativity and generate a wide range of ideas or solutions to address a particular challenge or opportunity.
- ❖ **Techniques:** Agile teams facilitate brainstorming sessions where team members openly share ideas, thoughts, and suggestions related to a specific topic. Techniques like **mind mapping, affinity diagramming, or silent brainstorming may be employed.**
- ❖ **Benefits:** Brainstorming **encourages collaboration, diversity of thought, and innovation within Agile teams.** It allows teams to explore different perspectives and generate potential solutions that can be further refined and evaluated.

Fundamental Information Gathering Skills

- **Brainstorming**
- ❖ Much of the value of **brainstorming** can be achieved by separating **step four**
- Brainstorming** can be used to generate possible solutions for simple problems, but it is unrealistic to expect it to accomplish most problem-solving or planning tasks.
- ❖ The technique has value as part of a larger effort that includes individual generation of information and ideas and subsequent compilation, evaluation, and selection.

Fundamental Information Gathering Skills

- **Brainstorming**
- ❖ **Brainstorming** can be used to generate components of **a plan, process, solution, or approach and to produce checklists.**
- ❖ The major benefits of brainstorm are as follows:
 - ❑ **It encourages “out-of-the-box” thinking;** that is, by challenging and inspiring each other, the participants are likely to go beyond conventional boundaries.
 - ❑ **Full solutions can emerge from the accumulation of ideas:** One participant may build upon another’s proposal or streamline it.
 - ❑ **The scope of the solutions will go beyond the narrow interests of individual groups.**

Fundamental Information Gathering Skills

➤ Brainstorming

- ❖ Among the **responsibilities** of the **conductor**, the following are the most important.
 - Establish a professional and objective tone for the meetings.
 - Establish and enforce the rules.
 - Introduce the meeting's goals and agenda.
 - Manage the meeting and keep the participants focused.
 - Facilitate the process of decision making and accord building, but avoid taking sides in the arguments.
 - Make certain that all stakeholders have an equal chance in voicing their opinions.
 - Control disruptive and unproductive behavior with subtlety but decisiveness

Fundamental Information Gathering Skills

- **Agile Documentation**
- ❖ Agile software development methods and their proponents suggest ‘just enough’ documentation on agile projects.
- ❖ Types of Software Documentation **Software documentation represents** all written documents and materials that accompany computer software.
- ❖ software documentation can be divided into two categories:
 1. **Process documentation**
 2. **Product documentation**

Fundamental Information Gathering Skills

➤ Agile Documentation

1. **Process documentation** records **the progress of the development cycle**.
 - ❖ This category fall materials that are created for the purpose of keeping track of **the project, passing and storing information** about the project as well as **plans, estimates and other information** that helps make the whole process visible.
 - ❖ Visible processes enable more effective management of projects.
 - ❖ Types of documents from this category include **project plans, test schedules, reports, standards, meeting notes or business correspondence**.

Fundamental Information Gathering Skills

- **Agile Documentation**
- 2. **Product documentation** focuses on the **developed software product**.
- ❖ These materials offer descriptions and information on the product and provide instructions on how to perform various tasks with the product.
- ❖ **Product documentation includes:**
 1. **User documentation**
 2. **System documentation**
- ❖ You must work closely with the customer for their documentation.
- ❖ When you do not, you are at risk of creating too much documentation, unnecessary documentation, or documentation that does not meet their actual needs.

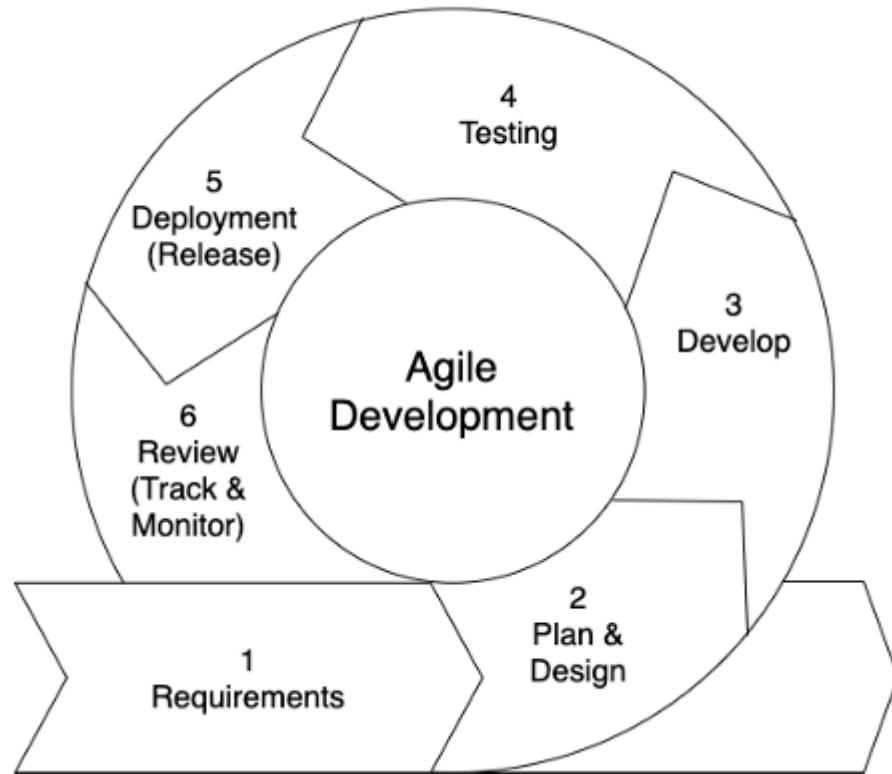
Fundamental Information Gathering Skills

- **Agile Documentation**
- ❖ **Agile documents** are sufficiently **accurate, consistent, and detailed**.
- ❖ **Agile documents** do not need to be perfect; they just need to be good enough.
- ❖ **Agile documents** are sufficiently indexed.
- ❖ **Documentation** is not effective if you cannot easily find the information contained in it;
therefore an index and a table of contents are important.
- ❖ In addition, the following points about documentation are critical to your success:
 - ❖ **The fundamental issue is effective communication, not documentation.**
 - ❖ **Documentation is as much a part of the system as the source code.**
 - ❖ **You should actually need the documentation, not just want it.**

Fundamental Information Gathering Skills

- **Agile Documentation**
- ❖ The investment in system documentation is a business decision, not a technical one
- ❖ It is your stakeholder's money, so they are the ones that should decide whether it will be invested in documentation, not you.
- ❖ You will often need to educate the stakeholder about the importance of certain documentation though.
- ❖ The reader, not the writer, determines whether documentation is sufficient.

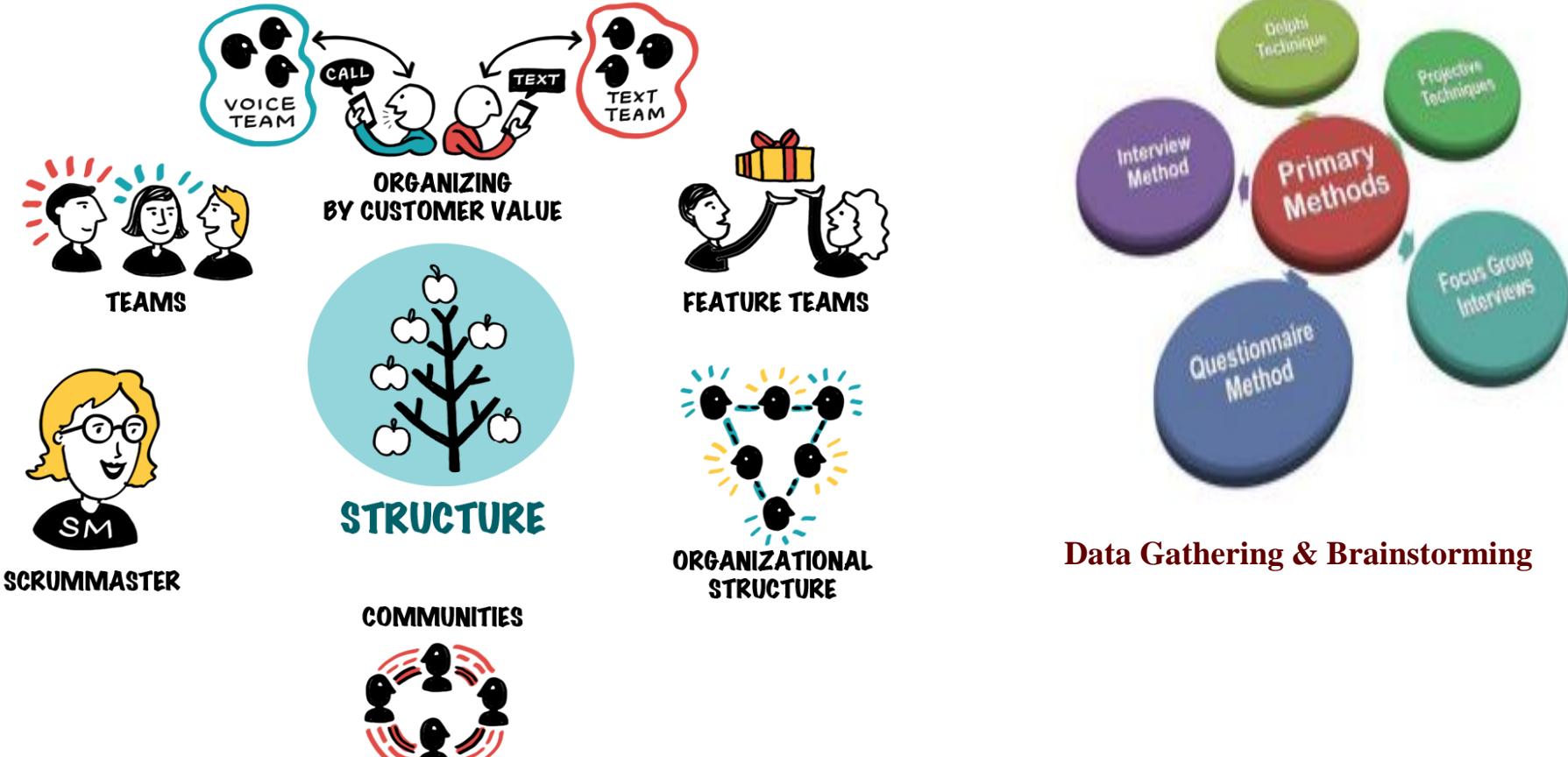
Agile Methods in General



Agile methodology

Chapter Three: Agile Development

The End of Unit Three



Scrum: less.works

<http://less.works>

Agile Methods in General

