# AGILE

## Methodology


BY

VISHWANATH ARABATI

CORPORATE TRAINER


DELIVERY INCREMENTALY….. INSTEAD OF ALL AT ONCE

| Topic |
| --- |
| History of Agile Methodologies |
| Agile and Lean Software Development |
| Basics and Fundamentals |
| Extreme Programming, Scrum |
| Agile and Scrum Principles, Agile Manifesto |
| Twelve Practices of XP |
| Agile Estimation &amp |
| Planning |
| Agile Requirements |
| User Stories, Backlog Management |
| Agile Architecture |
| Tracking Agile Projects. |
| Lean Software Development |
| Agile Risk management |
| Agile Project Tools |
| Agile Project Tools |
| Continuous Integration (CI). |
| Agile Testing |
| Scaling Agile for Large Projects |
| Scaling Agile for Large Projects. |

AGILE

| Agile |
| --- |
| It originated from the Agile Manifesto in 2001. |
| The focus is on customer collaboration and iterative delivery. |
| Scrum, Extreme Programming (XP), Iterative Development. |
| Flexible and adaptive to changes as they arise. |
| It involves frequent reassessment and adaptation in short cycles. |
| It involves continuous collaboration and feedback from customers. |

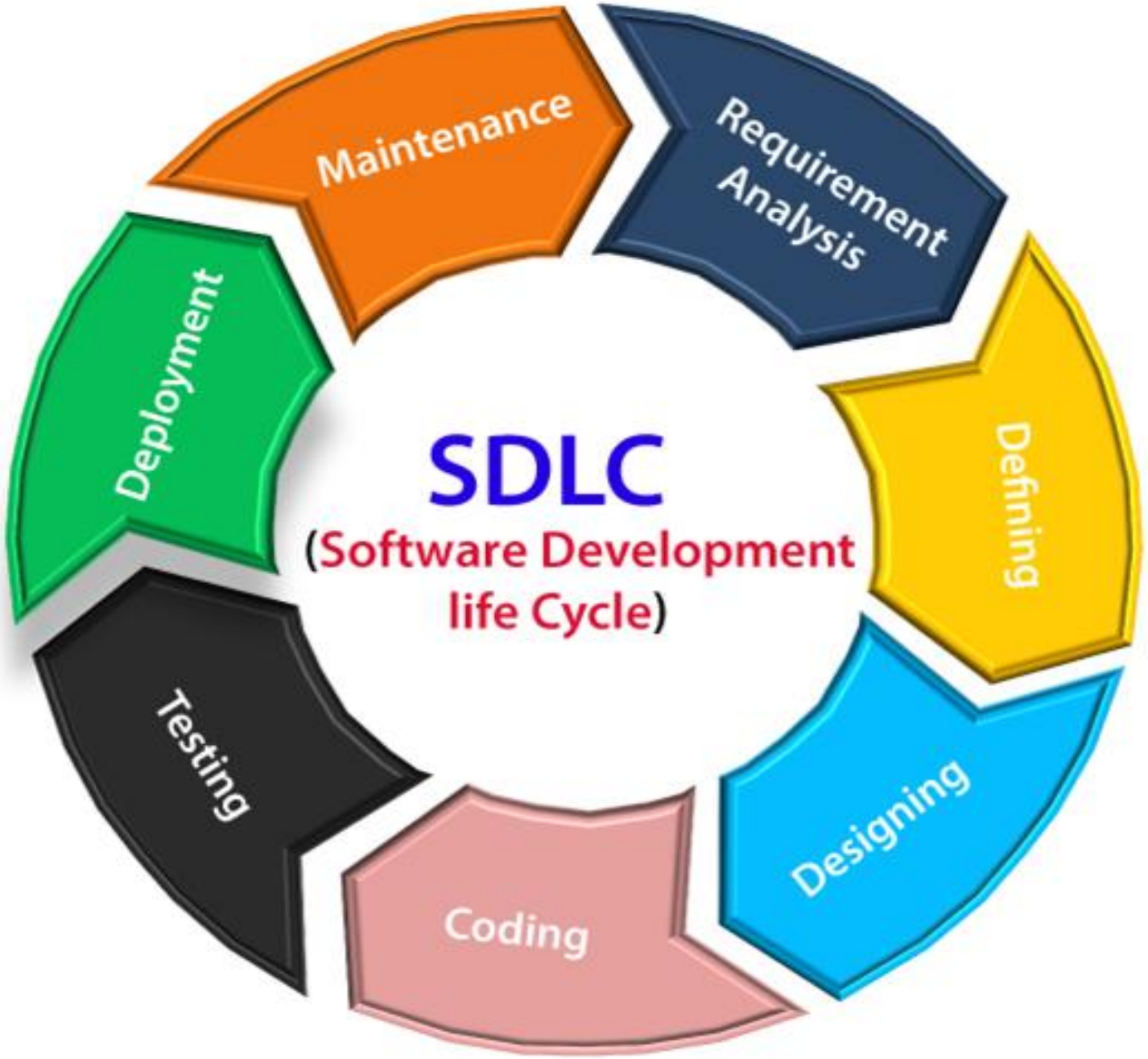
VCUBE Software Solutions Pvt. Ltd.

# SDLC

- Without using an exact life cycle model, the development of a software product would not be in a systematic and disciplined manner.
- developing a software product, there must be a clear understanding among team representative about when and what to do.
- describes entry and exit criteria for each phase.
- A phase can begin only if its stage-entry criteria have been fulfilled.
- without a software life cycle model, the entry and exit criteria for a stage cannot be recognized.
- Software Development Life Cycle (SDLC)
- pictorial and diagrammatic representation of the software life cycle.
- represents all the methods required to make a software product transit through its life cycle stages.
- maps the various activities performed on a software product from its inception to retirement.

VCUBE
Software Solutions Pvt. Ltd.

# SDLC



VCUBE Software Solutions Pvt. Ltd.

# SDLC

- **Stage1: Planning and requirement analysis**
  - senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry
- **Stage2: Defining Requirements**
  - document the software requirements and get them accepted from the project stakeholders.
  - "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.
- **Stage3: Designing the Software**
  - requirements, analysis, and design of the software project

VCUBE Software Solutions Pvt. Ltd.

# SDLC

- **Stage4: Developing the project**
  - development begins, and the programming is built.
- **Stage5: Testing**
  - During this stage, unit testing, integration testing, system testing, USER (UAT) acceptance testing are done.
- **Stage6: Deployment**
  - Once the software is certified, and no bugs or errors are stated, then it is deployed.
- **Stage7: Maintenance**
  - Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.

VCUBE Software Solutions Pvt. Ltd.

# SDLC

- Flexibility
  - More flexible than other methodologies (sucha as Waterfall which has a fixed timeline)
  - Agile's schedule adapts as the project progresses.
- Collaboration
  - Emphasizes collaboration between self-organizing, cross-functional teams.
- Reflection
  - After each sprint, teams reflect on what could be improved
  - adjust their strategy for the next sprint

# SDLC

- (SDLC) is a spiritual model used in project management that defines the stages
- include in an information system development project, from an initial feasibility study to the maintenance of the completed application.

# MODELS

- WATERFALL MODEL
- RAD MODEL
- V-MODEL
- INCREMENTAL MODEL
- SPIRAL MODEL
- INTERATIVE MODEL
- BIGBAND MODEL
- AGILE MODEL

# WATERFALL MODEL

- waterfall is a universally accepted SDLC model.
- waterfall model is a continuous software development model
- Phases and steps of requirements analysis, design, implementation, testing (validation), integration, and maintenance.
- **1. Requirements analysis and specification phase:**
- large document called **Software Requirement Specification (SRS)** document is created which contained a detailed description of what the system will do in the common language.
- **2. Design Phase:**
- overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

VCUBE Software Solutions Pvt. Ltd.

# WATERFALL MODEL

- **3. Implementation and unit testing:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD.
- **4. Integration and System Testing:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.
- **5. Operation and maintenance phase:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

# When to use SDLC Waterfall Model?

- Some Circumstances where the use of the Waterfall model is most suited are:

- When the requirements are constant and not changed regularly.
- A project is short
- The situation is calm
- Where the tools and technology used is consistent and is not changing
- When resources are well prepared and are available to use.

VCUBE Software Solutions Pvt. Ltd.

# Advantages of Waterfall model

- This model is simple to implement also the number of resources that are required for it is minimal.
- The requirements are simple and explicitly declared; they remain unchanged during the entire project development.
- The start and end points for each phase is fixed, which makes it easy to cover progress.
- The release date for the complete product, as well as its final cost, can be determined before development.
- It gives easy to control and clarity for the customer due to a strict reporting system.

# Disadvantages of Waterfall model

- In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.
- This model cannot accept the changes in requirements during development.
- It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.
- Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

- Definition

- Agile methodology is a project management framework.
- This breaks down the projects into phases, or sprints.
- This emphasizes continuous improvement and collaboration.
- It's an iterative process which involves
  - planning,
  - execution,
  - and evaluation.
-

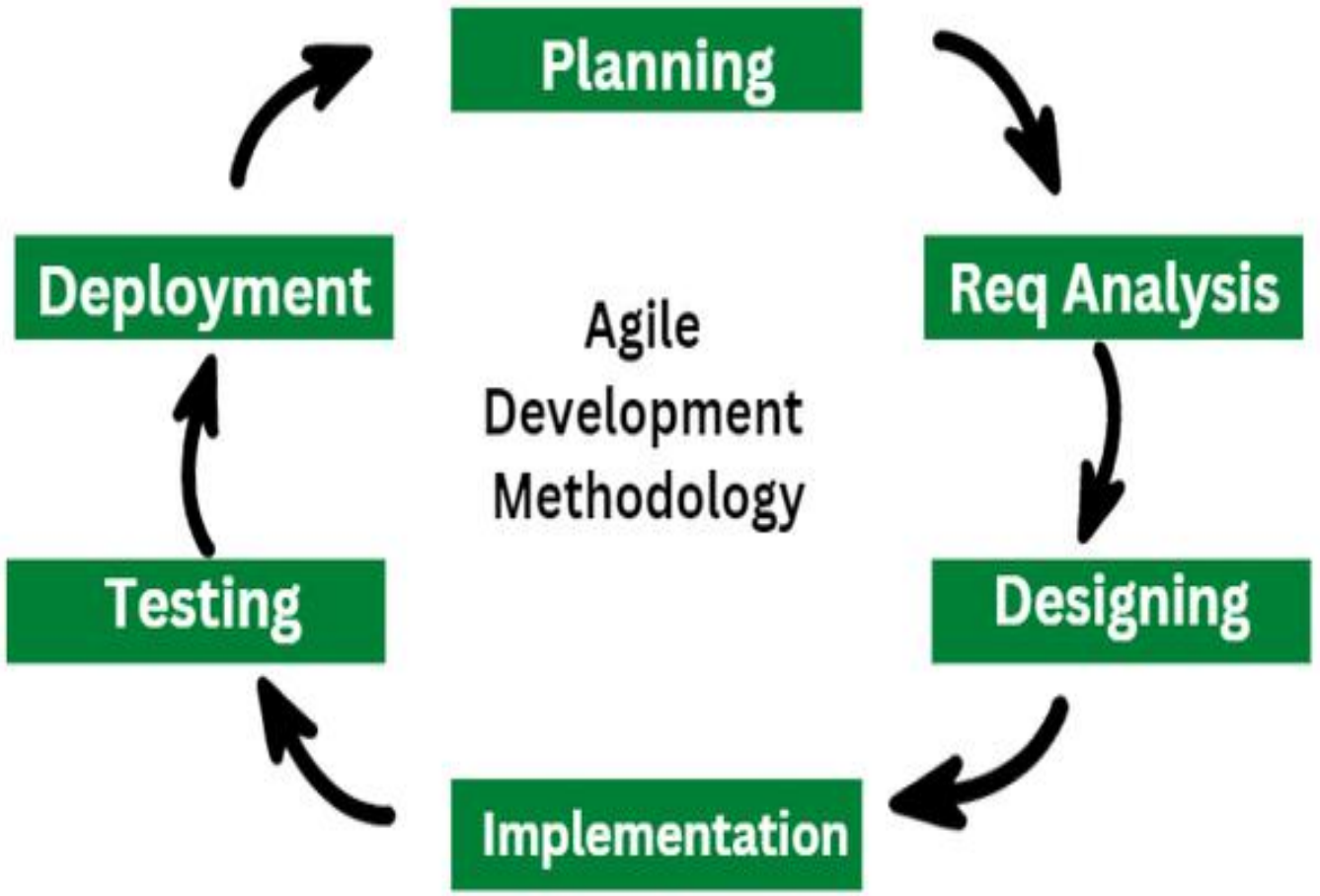
VCUBE Software Solutions Pvt. Ltd.

# AGILE

- Individuals and interactions:
- Working software
- Customer collaboration
- Responding to change

VCUBE Software Solutions Pvt. Ltd.

# AGILE Life Cycle



# Requirement Gathering

- In this stage, the project team identifies and documents the needs and expectations of various stakeholders, including clients, users, and subject matter experts.
- It involves defining the project's scope, objectives, and requirements.
- Establishing a budget and schedule.
- Creating a project plan and allocating resources.

VCUBE Software Solutions Pvt. Ltd.

## 2. Design

- Developing a high-level system architecture.
- Creating detailed specifications, which include data structures, algorithms, and interfaces.
- Planning for the software's user interface.

VCUBE Software Solutions Pvt. Ltd.

# 3. Development (Coding)

- Writing the actual code for the software. Conducting unit testing to verify the functionality of individual components.

VCUBE Software Solutions Pvt. Ltd.

# 4. Testing

- This phase involves several types of testing:

1. **Integration Testing:** Ensuring that different components work together.
2. **System Testing:** Testing the entire system as a whole.
3. **User Acceptance Testing:** Confirming that the software meets user requirements.
4. **Performance Testing:** Assessing the system's speed, scalability, and stability.

VCUBE Software Solutions Pvt. Ltd.

# 5. Deployment

1. Deploying the software to a production environment.
2. Put the software into the real world where people can use it.
3. Make sure it works smoothly in the real world.
4. Providing training and support for end-users.


# 6. Review (Maintenance)

1. Addressing and resolving any issues that may arise after deployment.
2. Releasing updates and patches to enhance the software and address problems.

- History of Agile
- • In 1957, people started figuring out new ways to build computer programs. They wanted to make the process better over time, so they came up with iterative and incremental methods.
- • In the 1970s, people started using adaptive software development and evolutionary project management. This means they were adjusting and evolving how they built software.
- • In the 1990s, there was a big change. Some people didn't like the strict and super-planned ways of doing things in software development. They called these old ways "waterfall." So, in response, lighter and more flexible methods showed up.
- These included:

1. Rapid Application Development (RAD) in 1991.
2. Unified Process (UP), Dynamic Systems Development Method (DSDM) in 1994.
3. Scrum in 1995.
4. Crystal Clear and Extreme Programming (XP) in 1996.
5. Feature-Driven Development (FDD) in 1997.

- Even though these came before the official "Agile Manifesto", we now call them agile software development methods.

- In 2005, Alistair Cockburn and Jim Highsmith added more ideas about managing projects, creating the PM Declaration of Interdependence.
- Then, in 2009, a group, including Robert C. Martin, added principles about software development. They called it the Software Craftsmanship Manifesto, focusing on being professional and skilled.
- In 2011, the Agile Alliance, a group of agile enthusiasts, made the Guide to Agile Practices (later called Agile Glossary). This was like a shared document where agile people from around
- the world put down their ideas, terms, and guidelines. It’s a bit like a dictionary for how to do agile things.

- What is LSD?

- Lean Software Development (LSD) is an approach derived from lean manufacturing principles aimed at optimizing efficiency and minimizing waste in the software development process.
-
- Prevent Defects: It integrates quality assurance throughout the development process to prevent defects.
- Eliminate Waste: It focuses on activities that add value to the customer and eliminates those activities that do not add value.
- Fast Delivery: Reduces cycle time to deliver software quickly and respond to feedback and changing requirements rapidly.
- Delay Decisions: Delay decisions until they can be made based on facts.

Seven Principles of LSD

There are 7 established lean principles that come with a set of tactics, practices, and processes that build more efficient software products:

1. Eliminating the Waste

To identify and eliminate wastes e.g. unnecessary code, delay in processes, inefficient communication, issues with quality, data duplication, more tasks in the log than completed, etc. regular meetings are held by Project Managers. This allows team members to point out faults and suggest changes in the next turn.

2. Fast Delivery

Previously long-time planning used to be the key to success in business, but with time, it has been found that engineers spend too much time on building complex systems with unwanted features. So they came up with an MVP strategy which resulted in building products quickly that included a little functionality and launching the product to market and seeing the reaction. Such an approach allows them to enhance the product based on customer feedback.

3. Amplify Learning

Learning is improved through ample code reviewing and meetings that are cross-team applicable. It is also ensured that particular knowledge isn't accumulated by one engineer who's writing a particular piece of code so paired programming is used.

4. Builds Quality

LSD is all about preventing waste and keeping an eye on not sacrificing quality. Developers often apply test- driven programming to examine the code before it is written. Quality can also be gained by getting constant feedback from team members and project managers.

5. Respect Teamwork

LSD focuses on empowering team members, rather than controlling them. Setting up a collaborative atmosphere, keeping perfect balance when there are short deadlines and immense workload. This method becomes very important when new members join a well-established team.

6. Delay the Commitment

In traditional project management, it often happens when you make your application and it turns out to be completely unfit for the market. LSD method recognizes this threat and makes room for improvement by

postponing irreversible decisions until all experiment is done. This methodology always constructs software as flexible, so new knowledge is available and engineers can make improvements.

7. Optimizing the Whole System

Lean’s principle allows managers to break an issue into small constituent parts to optimize the team’s workflow, create unity among members, and inspire a sense of shared responsibility which results in enhancing the team’s performance.

LSD Process

Here is the overview of the lean software development process:

1. Identify Value: Understand the customer values and focus on delivering features that meet these needs.

2. Map the Value Stream: This involves mapping out the entire software development process to identify and eliminate wasteful activities that do not add value.

3. Create Flow: Ensure a smooth and continuous flow of work by minimizing delays and interruptions.

4. Establish Pull: Develop features based on customer demand rather than pushing features through the process.

5. Seek Perfection: Regularly review and refine the development process. Always encourage the team members to identify the areas of improvement and implement changes iteratively.

6. Build Quality In: Use practices such as test-driven development (TDD) and continuous integration to integrate quality assurance throughout the development process.

7. Empower Teams: Empower development teams by providing them with the necessary tools, resources, and autonomy to make decisions.

| LSD vs Agile | | |
|---|---|---|
| **Aspect** | **Lean Software Development (LSD)** | **Agile** |
| **Origin** | It originated from lean manufacturing, especially the Toyota Production System. | It originated from the Agile Manifesto in 2001. |
| **Focus** | The focus is on waste elimination and value optimization. | The focus is on customer collaboration and iterative delivery. |
| **Process and Practices** | Kanban, Value Stream Mapping, Continuous Improvement (Kaizen). | Scrum, Extreme Programming (XP), Iterative Development. |
| **Decision Making** | Delays decisions until necessary and are based on facts. | Flexible and adaptive to changes as they arise. |
| **Iteration and Feedback** | It involves continuous improvement through regular feedback. | It involves frequent reassessment and adaptation in short cycles. |
| **Customer Involvement** | It involves understanding and delivering customer value continuously. | It involves continuous collaboration and feedback from customers. |

Benefits of LSD

Here are some key benefits of LSD that help organizations to improve their software development processes and outcomes:

1. Increased Efficiency: LSD reduces delays and inefficiencies by identifying and eliminating non-value-adding activities.

2. Higher Quality: It integrates quality assurance throughout the development process, thus preventing defects and ensuring quality products.

3. Faster Delivery: Shorter development cycles allow for quicker release of features and updates, thus meeting customer demands more rapidly.

4. Adaptability: Delaying decisions until they are necessary and are based on facts, allowing teams to adapt to changes and new information.

5. Enhanced Collaboration: Engages customers throughout the development process, ensuring that their needs and feedback are continuously addressed.

Limitations of LSD

1. Cultural Resistance: Implementing LSD requires a significant cultural shift and if there is resistance to change from team members and management then it can hinder its adoption and effectiveness.

2. Learning Curve: There is a steep learning curve associated with understanding and applying lean principles and practices effectively.

3. Requires Strong Leadership: Successful implementation of LSD requires strong and committed leadership to guide the transition.

4. Difficulty in Measuring Waste: In LSD, determining waste is subjective and challenging. It requires a deep understanding of processes and value streams.

5. Resource Intensive: Implementing LSD requires an initial investment in training, tools, and process redesign. This can be significant.

VCUBE Software Solutions Pvt. Ltd.

# AGILE

- Customer satisfaction by early and continuous delivery of software
- Changing requirements
- Working software is delivered frequently
- Close, daily cooperation between business people and developers
- Applications built around motivated individuals, who should be trusted.
- F2F conversation
- Sustainable development, able to maintain a constant pace
- Light weighted methodology
- Small to medium team size

VCUBE Software Solutions Pvt. Ltd.

# AGILE

- SERIALIZED PROCESS: Iterative approach with tasks broken into small increments.
- Planning far in advance: plan for what we know and we have left sufficient allowance in our plans for what we don’t know.
- Lack of Visibility: Teams don’t realize they are behind schedule.
- Project timeline: Allows the development effort to get feedback from the customer throughout.
- Static requirements: scope is never closed; con  - requirement priorities by the business.

VCUBE Software Solutions Pvt. Ltd.

# AGILE

VCUBE Software Solutions Pvt. Ltd.

# AGILE

| Agile benefits | Agile challenges |
|---|---|
| Ability to manage changing priorities | Organisations can resist change in adoption |
| Increased project visibility | Teams may use inconsistent practises |
| Improved business/IT alignment | Needs support of leadership and management |
| Delivery speed/time to market | Organisational culture can be at odds with agile values |
| Project risk reduction & predictability | |

VCUBE Software Solutions Pvt. Ltd.

# Advantage(Pros) of Agile Method:

1. Frequent Delivery
2. Face-to-Face Communication with clients.
3. Efficient design and fulfils the business requirement.
4. Anytime changes are acceptable.
5. It reduces total development time.

# Disadvantages(Cons) of Agile Model:

1. Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.

2. Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become a difficulty.

*Four Core Values*

### Individuals and interactions over processes and tools

Agile emphasizes the importance of people and their interactions as the primary drivers of project success. Effective communication, collaboration, and teamwork are vital in Agile environments, fostering a sense of ownership and responsibility among team members.

### Working software over comprehensive documentation

While documentation remains essential, Agile prioritizes the delivery of working software that meets customer needs. Frequent and incremental releases allow stakeholders to see tangible progress and provide valuable feedback throughout the development process.

### Customer collaboration over contract negotiation

Agile encourages close collaboration with customers and end-users. This customer-centric approach ensures that the software being developed aligns with their evolving needs, increasing the likelihood of delivering a product that satisfies their requirements.

### Responding to change over following a plan

Agile acknowledges that change is inevitable in software development. Rather than rigidly adhering to a fixed plan, Agile teams embrace change and view it as an opportunity for improvement.

Frequent iterations enable teams to adapt to new information and feedback, fostering a more responsive development process. Agile software development thrives on change and adaptability, making flexibility the heartbeat of its success.

Twelve Agile Development Principles:

the 12 principles from the Agile Manifesto :

1. Prioritize satisfying the customer through early and continuous delivery of valuable software.

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

3. Deliver working software frequently, with a preference for shorter timescales.

4. Collaborate closely between business people and developers throughout the project.

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

6. Use face-to-face communication whenever possible for effective information sharing.

7. Measure progress primarily through working software.

8. Maintain a sustainable pace of work for the development team. Continuous work is sustainable work.

9. Focus on technical excellence and good design to enhance agility.

10. Keep things simple and maximize the amount of work not done (avoid unnecessary tasks).

11. Allow self-organizing teams to make decisions on how to accomplish their work.

12. Reflect at regular intervals on team effectiveness and adjust behavior accordingly.

VCUBE Software Solutions Pvt. Ltd.

# AGILE METHODOLOGY

- Agile means swift or versatile.
- development approach based on iterative development
- break tasks into smaller iterations.
- parts do not directly involve long term planning.
- Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.
- Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks.

VCUBE Software Solutions Pvt. Ltd.

# AGILE METHODOLOGY

- division of the entire project into smaller parts
- to minimize the project risk and to reduce the overall project delivery time requirements.
- development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

VCUBE Software Solutions Pvt. Ltd.

# Agile Methodology Steps

- These are the deliverables found in Agile project management:
  1. **Product vision statement:** A concise summary that vividly articulates the overarching goals and purpose of the product. This serves as the North Star, providing a guiding vision for the entire project team.
  2. **Product roadmap:** Offering a high-level overview, the product roadmap delineates the key requirements essential for realizing the product vision. It acts as a strategic blueprint, aligning the team's efforts with the broader objectives.
  3. **Product backlog:** A comprehensive, prioritized list outlining all the necessary elements for the project's success. The product backlog serves as a dynamic repository that evolves over time, capturing the evolving needs and priorities of the project.
  4. **Release plan:** A structured timetable detailing the planned releases of the working product. This component provides a strategic timeline, ensuring that deliverables align with project goals and external milestones.
  5. **Sprint backlog:** Focused on the current sprint, the sprint backlog encompasses user stories, goals, and tasks. It serves as a detailed action plan, guiding the team through the specific objectives of the ongoing iteration.
  6. **Increment:** The tangible outcome of each sprint, the increment represents the functional aspects of the product presented to stakeholders. This not only facilitates continuous feedback but also allows for potential customer delivery, enhancing transparency and collaboration.

# Phases of Agile Model:

- Following are the phases in the Agile model are as follows:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback

- **1. Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

- **2. Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

- **3. Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

- **4. Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
- **5. Deployment:** In this phase, the team issues a product for the user's work environment.
- **6. Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

VCUBE Software Solutions Pvt. Ltd.

# Agile Manifesto Values:

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

- This iterative approach offers several advantages:

- **Early Value Delivery:** Customers can start using and benefiting from the software early in the development process, gaining tangible value with each iteration.
- **Continuous Feedback:** Frequent releases allow stakeholders to provide feedback, guiding the development direction and ensuring that the final product aligns with their expectations.
- **Risk Mitigation:** By breaking the project into smaller chunks, Agile reduces the risk associated with large-scale development, making it easier to adjust and adapt to changes.
- **Increased Transparency:** Teams and stakeholders have a clear view of progress, making it easier to identify and address potential issues or delays.

VCUBE Software Solutions Pvt. Ltd.

# Agile Manifesto Principles:

1. Customer satisfaction through early and continuous software delivery
2. Accommodate changing requirements throughout the development process
3. Frequent delivery of working software
4. Collaboration between the business stakeholders and developers throughout the project
5. Support, trust, and motivate the people involved
6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Simplicity
11. Self-organizing teams encourage great architectures, requirements, and designs
12. Regular reflections on how to become more effective

VCUBE Software Solutions Pvt. Ltd.

# Agile Testing Methods:

- Scrum
- Crystal
- Dynamic Software Development Method(DSDM)
- Feature Driven Development(FDD)
- Lean Software Development
- eXtreme Programming(XP)

VCUBE Software Solutions Pvt. Ltd.

# SCRUM

- Scrum
- SCRUM is an agile development process focused primarily on ways to manage tasks in team-based development conditions.
- There are three roles in it, and their responsibilities are:
- **Scrum Master:** The scrum can set up the master team, arrange the meeting and remove obstacles for the process
- **Product owner:** The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.
- **Scrum Team:** The team manages its work and organizes the work to complete the sprint or cycle.

- **Silent features of Scrum**
- Scrum is a light-weighted framework
- Scrum emphasizes self-organization
- Scrum is simple to understand
- Scrum framework helps the team to work together
- Lifecycle of Scrum

VCUBE Software Solutions Pvt. Ltd.

# AGILE

## CUSTOMER

- More responsive to requests
- High-value features
- Delivered more quickly with short cycles

## DEVELOPMENT TEAMS

- Enjoy development work
- Work is valued and used
- Reduced non-productive work

## SCRUMMASTER

- Planning/task-level tracking in daily meetings
- Tremendous awareness of project state/status
- Catching and addressing issues quickly

## VENDOR

- Focused development on high-value features
- Increased efficiency
- Reduce wastage and decreased overhead

## PRODUCT OWNER

- Development work aligns with customer needs
- Frequent opportunities to re-prioritize work
- Maximum delivery of value

## PMOS AND C-LEVEL EXECUTIVES

- High visibility of daily project development
- Adjust strategies based on hard information
- Plan effectively with less speculation

VCUBE Software Solutions Pvt. Ltd.

# Scrum Team Roles

- **Product owner**: The product expert and voice of stakeholders in Agile Scrum, the Product Owner defines the product vision, prioritizes features, and ensures development aligns with business goals. This role demands a deep understanding of market demands and a strategic vision.
- **Development team**: Comprising skilled professionals like developers, programmers, and designers, the Development Team drives product delivery. Emphasizing self-organization, collective ownership, and continuous improvement, they collaborate closely to transform requirements into tangible outcomes.
- **Scrum master**: The organized servant-leader integral to applying Scrum principles seamlessly. Beyond facilitating events and removing impediments, the Scrum Master nurtures a culture of continuous improvement, guides the team in embracing Agile values, and champions efficiency, fostering an environment for the Scrum framework to thrive.

VCUBE Software Solutions Pvt. Ltd.

# Scrum Events

- **Sprint**: In Scrum, it's a brief period for the development team to complete specific tasks, milestones, or deliverables—essentially dividing the project schedule into manageable time blocks not exceeding one month.
- **Sprint planning**: At the start of every Sprint, the entire Scrum team gathers to plan the upcoming sprint.
- **Daily Scrum**: A 15-minute daily meeting during the Sprint to discuss the previous day's achievements and expectations for the next one.
- **Sprint review**: An informal end-of-sprint meeting where the Scrum team presents their Increment to stakeholders and discusses feedback.
- **Sprint retrospective**: A meeting where the Scrum team reflects on the previous Sprint and establishes improvements for the next one.

VCUBE Software Solutions Pvt. Ltd.

# Scrum Artifacts

- **Product backlog**: Managed by the Product Owner, it lists all requirements for a viable product in order of priority. Includes features, functions, requirements, enhancements, and fixes authorizing changes in future releases.
- **Sprint backlog**: A list of tasks and requirements for the next Sprint, sometimes visualized using a Scrum task board in a ‘To Do, Doing, and Done’ format.

VCUBE Software Solutions Pvt. Ltd.

## The ScrumMasters responsibilities include

- Teach the Product Owner how to maximize return on investment (ROI), and meet his/her objectives through Scrum.
- Improve the lives of the development Team by facilitating creativity and empowerment.
- Improve the productivity of the development Team in any way possible.
- Improve the engineering practices and tools so that each increment of functionality is potentially shippable.
- Keep information about the Team’s progress up to date and visible to all parties.

## Scrum: A Comprehensive Approach

Scrum is one of the most widely adopted Agile frameworks in the software development industry. It provides a structured and comprehensive approach to managing projects, enabling teams to deliver high-quality software efficiently.

In this section, we will explore Scrum in detail, understanding its framework, key roles, artifacts, ceremonies, and the benefits it brings to software development teams.

### Overview of Scrum Framework

The Scrum framework is built on the foundation of Agile principles and is designed to maximize productivity, foster collaboration, and deliver value to customers.

It consists of three essential elements:

#### *Scrum Roles*

**Product Owner:** The Product Owner is the voice of the customer and stakeholders. They are responsible for defining and prioritizing the product backlog, ensuring that the development team is working on the most valuable features.
The Product Owner collaborates with stakeholders to gather requirements and provide feedback on delivered increments.

**Scrum Master:** The Scrum Master acts as a facilitator and servant-leader for the development team. Their primary role is to ensure that the Scrum framework is understood and followed correctly.
They remove any impediments that hinder the team's progress, promote a collaborative team environment, and facilitate the various Scrum ceremonies.

**Development Team:** The Development Team consists of professionals who do the actual work of delivering a potentially shippable product increment in each sprint. They are self-organizing, cross-functional, and collaborate closely to complete the tasks from the sprint backlog.

*Scrum Artifacts*

**Product Backlog:** The Product Backlog is a prioritized list of all the work items required to complete the project. These items can include features, enhancements, bug fixes, and technical tasks.

The Product Owner continuously refines and updates the backlog based on feedback and changing requirements.

**Sprint Backlog:** Before each sprint, the Development Team pulls a set of work items from the Product Backlog and creates the Sprint Backlog.

The Sprint Backlog contains the tasks the team commits to completing during the sprint. It provides transparency and a clear plan for the upcoming iteration.

**Increment:** The Increment represents the sum of all completed Product Backlog items at the end of each sprint. It is a potentially shippable piece of software that should be in a usable state and adhere to the team's definition of "*done*."

VCUBE Software Solutions Pvt. Ltd.

# Agile methodologies

- Scrum is one of the most widely used.
- This is prescriptive framework.
- Scrum excels at managing iterative and incremental projects.
- Using the Scrum Agile methodology, a Product Owner sets a list of priorities, the Product Backlog, to be completed by a cross-functional team.
- The team works to deliver “potentially shippable increments” of software in 2-4-week sprints, at the end of which the Product Backlog is reevaluated and prioritized.
- Agile teams like Scrum because it’s easy to follow and scale.
- It enables management teams to identify problems early on and fosters strong, active collaboration between teams and colleagues.

# When to use the Agile Model?

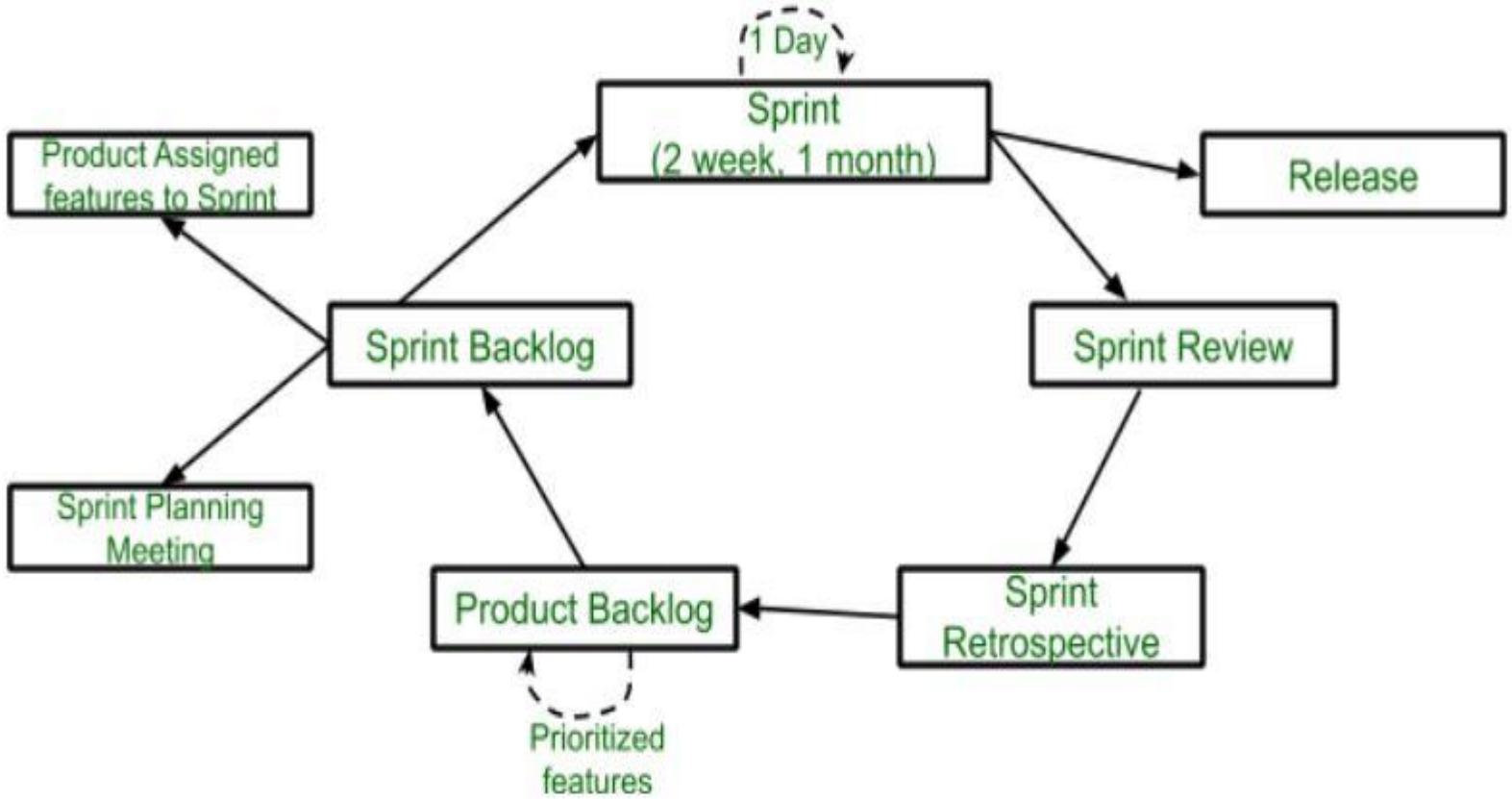
- When frequent changes are required.
- When a highly qualified and experienced team is available.
- When a customer is ready to have a meeting with a software team all the time.
- When project size is small.

| Criteria | Agile | Waterfall |
| --- | --- | --- |
| Adaptability | Extremely adaptable, allowing quick responses to changes and evolving technology. | More rigid structure, best suited for projects with a clear and unchanging vision. |
| Project Timeline | Flexible timeline dependent on project development. | Fixed timeline planned from the start. |
| Project Phases | Concurrent work on phases with tight deadlines, team-driven direction. | Linear progression through defined stages, driven by project manager. |
| Flexibility in Direction | Allows for changes even late in the process, suitable for evolving projects. | Less flexibility due to a predefined and unchanging vision. |
| Budget Flexibility | Budget subject to change as project direction evolves. | Less flexible budget planned from the start. |
| Ideal for | Software development where technology evolves rapidly. | Projects with a clear and specific vision that won't change. |
| Stakeholder Feedback | Continuous stakeholder feedback is incorporated throughout. | Deliverables for each stage are clearly defined before moving on. |

- **When to use the Agile Methodology?**
- If you want to know when to use the Agile methodology, then it is particularly well-suited for projects and organizations where the following conditions or needs are present:

1. **Unclear or Changing Requirements:** Agile is great for projects with requirements that aren't well-defined or might change.
2. **Complex Projects:** It's good for big, complex projects by breaking them into smaller pieces.
3. **Customer Focus:** Use Agile when making customers happy is a priority and you want to involve them throughout.
4. **Quick Time-to-Market:** If you need to get your product out fast, Agile can help.
5. **Small to Medium Teams:** Agile works well for teams of a few to a few dozen people.
6. **Improvement:** Agile fosters a culture of always getting better over time.

6. **Team Skills:** It’s best when you have a mix of skills in your team, like development, testing, design, and more.
7. **Collaboration:** Agile promotes working together and open communication.
8. **Regular Updates:** If you want to check progress often and make changes as needed.
9. **Transparency:** Agile emphasizes being open and clear with everyone involved in the project.
10. **Risk Control:** It helps manage risks by tackling issues as they come up.
11. **Innovation:** If you encourage trying new things and learning from experience, Agile supports that.
12. **Continuous**


1 Day
Sprint
(2 week, 1 month)
Release
Product Assigned
features to Sprint
Sprint Backlog
Sprint Review
Sprint Planning
Meeting
Product Backlog
Sprint
Retrospective
Prioritized
features

- **Sprint:** A Sprint is a time box of one month or less. A new Sprint starts immediately after the completion of the previous Sprint. **Release:** When the product is completed, it goes to the Release stage.
- **Sprint Review:** If the product still has some non-achievable features, it will be checked in this stage and then passed to the Sprint Retrospective stage.
- **Sprint Retrospective:** In this stage quality or status of the product is checked. **Product Backlog:** According to the prioritize features the product is organized.
- **Sprint Backlog:** Sprint Backlog is divided into two parts Product assigned features to sprint and Sprint planning meeting.

- **Advantage of Scrum framework**
- Scrum framework is fast moving and money efficient.
- Scrum framework works by dividing the large product into small sub-products. It’s like a divide and conquer strategy
- In Scrum customer satisfaction is very important.
- Scrum is adaptive in nature because it have short sprint.
- As Scrum framework rely on constant feedback therefore the quality of product increases in less amount of time

| Parameters | Agile Methodology | Traditional Approach |
|---|---|---|
| **Definition** | Agile is like building a flexible and adaptable treehouse in stages. | Traditional approaches are like constructing a house with a detailed blueprint. |
| **Chronology of operations** | Testing and development processes are performed concurrently. | Testing is done once the development phase is completed. |
| **Organizational structure** | It follows iterative organizational structure. | It follows linear organizational structure. |
| **Communication** | Agile encourages face-to-face communication. | Traditional approach encourages formal communication. |
| **Number of phases** | It consists of only three phases. | It consists of five phases. |
| **Development cost** | Less using this methodology. | More using this methodology. |
| **User requirements** | Clearly defined user requirements before coding. | Requires interactive user inputs. |

## S*crum Ceremonies*

**Sprint Planning:** At the beginning of each sprint, the Product Owner and Development Team collaborate in the Sprint Planning meeting. They discuss and agree on the sprint goal, select the top items from the Product Backlog, and create the Sprint Backlog with associated tasks.
**Daily Standup (Daily Scrum):** The Daily Standup is a brief daily meeting where the Development Team synchronizes their work. Each team member shares what they worked on the previous day, what they plan to work on that day, and any impediments they are facing.

**Sprint Review:** At the end of each sprint, the team holds a Sprint Review meeting to demonstrate the completed Increment to stakeholders. Feedback is gathered, and the Product Backlog is updated based on the stakeholders' input.
**Sprint Retrospective:** Following the Sprint Review, the team conducts the Sprint Retrospective to reflect on the previous sprint. They identify what went well, what could be improved, and define actionable items to enhance their processes in the upcoming sprints.

## Benefits and Advantages of Scrum

Scrum offers a number of benefits that contribute to its popularity and success in Agile software development:

**Transparency:** The use of visible backlogs, frequent progress updates, and regular meetings ensures transparency among team members and stakeholders. This fosters a shared understanding of the project's status.

**Adaptability:** Scrum's iterative nature allows teams to adapt to changing requirements and priorities. This ensures that the delivered product remains aligned with the customer's needs.

**Continuous Improvement:** The Sprint Retrospective encourages continuous improvement by providing a platform for the team to reflect on their practices and identify opportunities for enhancement.

**Early Value Delivery:** The focus on delivering potentially shippable increments at the end of each sprint allows customers to see tangible progress early in the development process.

**Customer Collaboration:** The involvement of the Product Owner and regular Sprint Reviews promote active collaboration with customers, resulting in a product that better meets their expectations.

## Scrum Challenges and How to Overcome Them

While Scrum is highly effective, it is not without its challenges. Some common hurdles that teams may encounter include:

1. **Overcommitment:** Teams might take on too much work in a sprint, leading to incomplete tasks and a compromised Increment. Regularly evaluating capacity and being realistic about commitments can help avoid this pitfall.

2. **Lack of Empowerment:** If team members are not empowered to make decisions and are overly dependent on the Scrum Master, the efficiency and effectiveness of the team may suffer. Encouraging self-organization and trust within the team can mitigate this challenge.

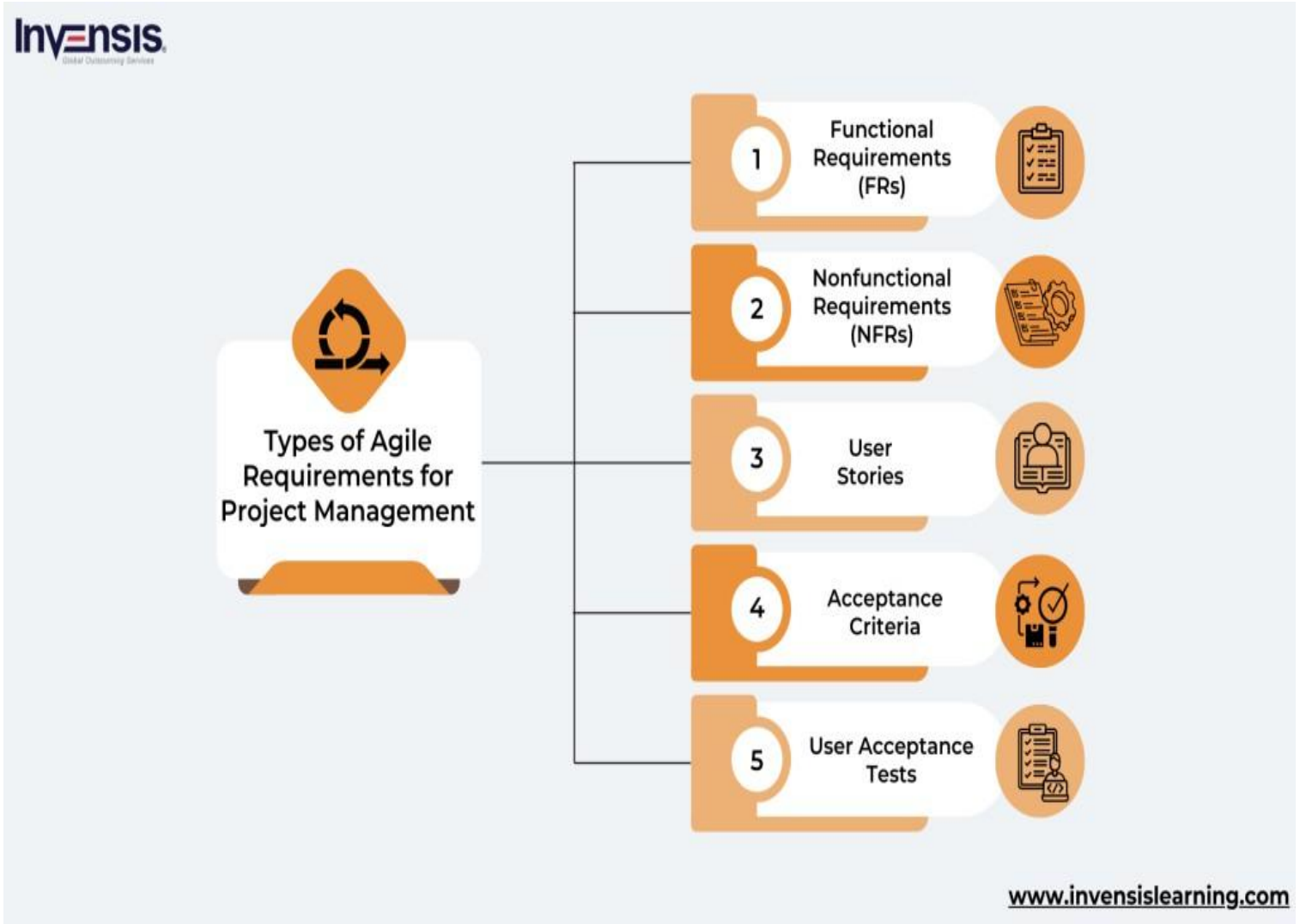
3. **Incomplete Definition of "Done":** Ambiguity about what constitutes a "done" user story can lead to misunderstandings and incomplete work. Clearly defining and agreeing upon the team's "definition of done" is crucial for consistent delivery.

4. **roduct Owner Availability:** Insufficient availability of the Product Owner can slow down decision- making and result in unclear requirements. Maintaining constant communication and involvement with

   - the team can help alleviate this issue.

- Key elements of Scrum project management include:

- **Sprints**: Short, time-boxed work cycles where the team focuses on completing a set of deliverables from the product backlog. These cycles typically last 1-4 weeks and keep the project focused and adaptable
- **Daily stand-up meetings**: Also known as daily scrums, these are brief meetings (usually 15-20 minutes) held each day during a sprint. The team uses this time to discuss progress, identify roadblocks, and ensure everyone is aligned.
- **Product backlog**: This is a prioritized list of features, requirements, and fixes for the entire project. It's a living document that evolves throughout the project as new information emerges
- **Sprint backlog**: A subset of the product backlog, it includes the specific list of items the development team will work on during a particular sprint. This list is created during sprint planning and reflects what the team believes they can accomplish in that timeframe
- **Sprint review meetings**: Held at the end of each sprint, the review meeting is an opportunity for the team to showcase what they've completed and gather feedback from stakeholders
- **Sprint retrospectives**: Another meeting held at the conclusion of a sprint, the retrospective is a chance for the team to reflect on what went well, what didn't, and how they can improve their process for the next sprint


Invensis
Types of Agile Requirements for Project Management
1
Functional Requirements (FRs)
2
Nonfunctional Requirements (NFRs)
3
User Stories
4
Acceptance Criteria
5
User Acceptance Tests
www.invensislearning.com

Why create user stories?

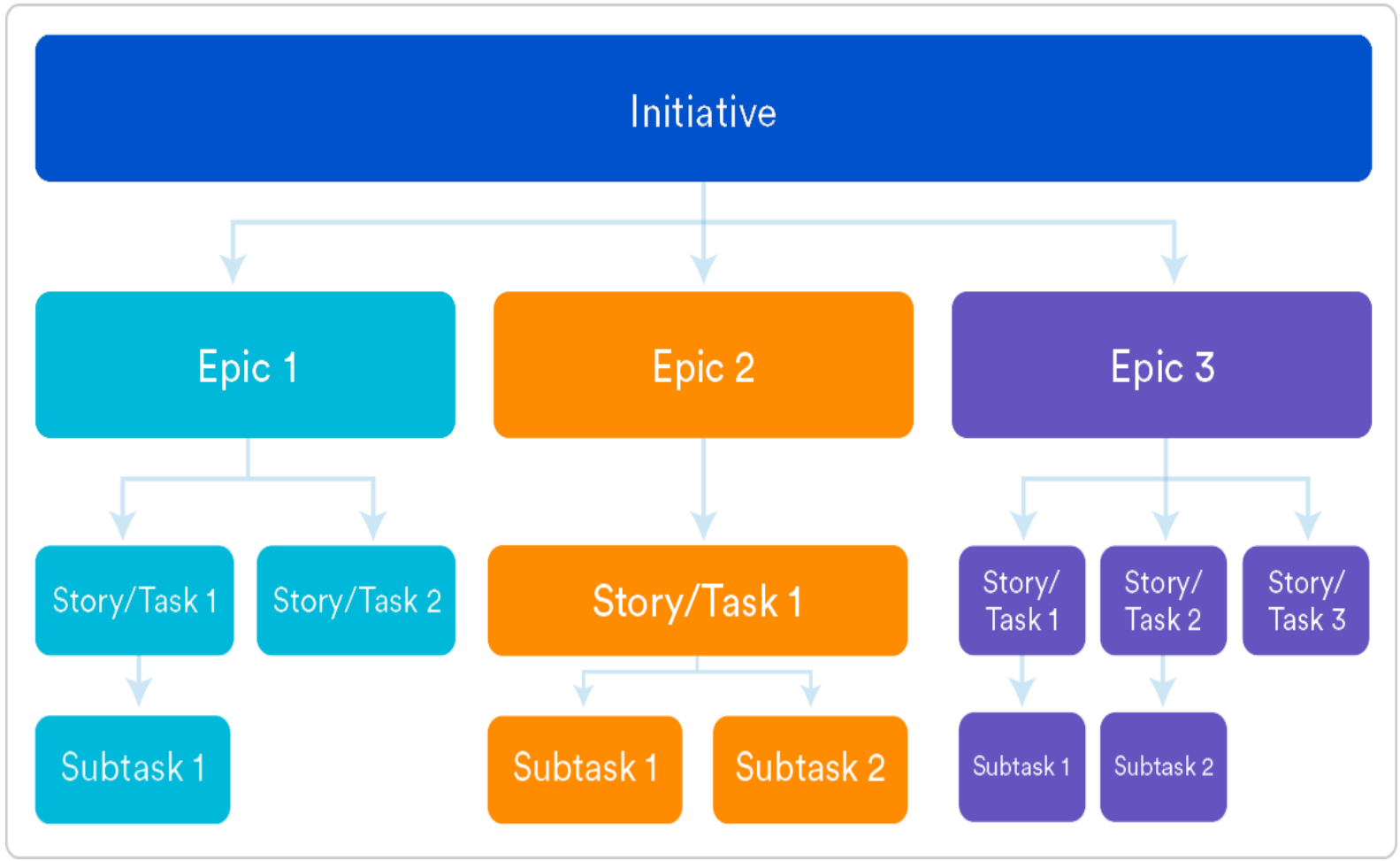
For development teams new to agile, user stories sometimes seem like an added step. Why not just break the big project (the epic) into a series of steps and get on with it? But stories give the team important context and associate tasks with the value those tasks bring.

User stories serve a number of key benefits:

**Stories keep the focus on the user.** A to-do list keeps the team focused on tasks that need to be checked off, but a collection of stories keeps the team focused on solving problems for real users.

**Stories enable collaboration.** With the end goal defined, the team can work together to decide how best to serve the user and meet that goal.
**Stories drive creative solutions.** Stories encourage the team to think critically and creatively about how to best solve for an end goal.

**Stories create momentum.** With each passing story, the development team enjoys a small challenge and a small win, driving momentum.


Initiative
Epic 1
Epic 2
Epic 3
Story/Task 1
Story/Task 2
Story/Task 1
Story/ Task 1
Story/ Task 2
Story/ Task 3
Subtask 1
Subtask 1
Subtask 2
Subtask 1
Subtask 2

## How to write user stories

Consider the following when writing user stories:

**Definition of "done"** — The story is generally "done" when the user can complete the outlined task, but make sure to define what that is.
**Outline subtasks or tasks** — Decide which specific steps need to be completed and who is responsible for each of them.

**User personas** — For whom? If there are multiple end users, consider making multiple stories.

**Ordered Steps** — Write a story for each step in a larger process.

**Listen to feedback** — Talk to your users and capture the problem or need in their words. No need to guess at stories when you can source them from your customers.
**Time** — Time is a touchy subject. Many development teams avoid discussions of time altogether, relying instead on their estimation frameworks. Since stories should be completable in one sprint, stories that might take weeks or months to complete should be broken up into smaller stories or should be considered their own epic.

# LETS START WORKING
# ON
# A PROJECT
# USING
# AGILE METHODOLOGY

## Agile is not for Listening or Teaching....
## In fact , To follow

## packup...