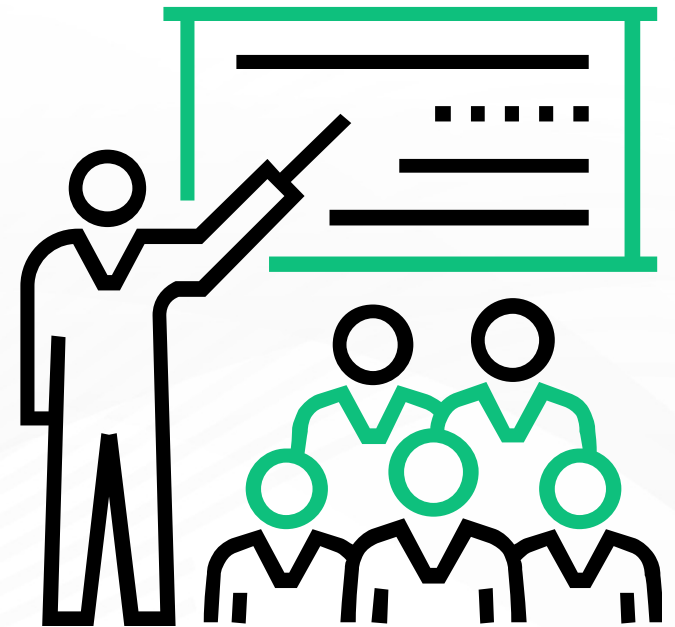


Module 1: Agile Practices

Module Objectives

At the end of this module, you will be able to:

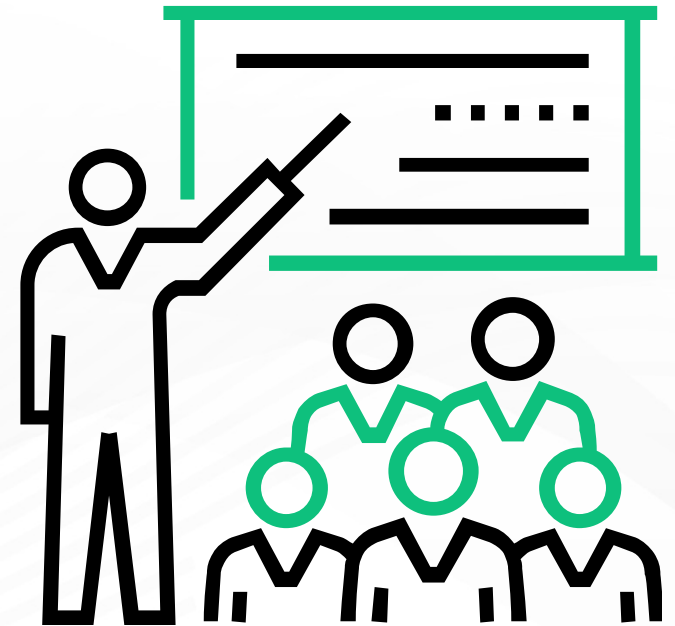
- Learn Agile methodology
- Define software, history of software engineering, and software development methodologies
- Identify the traditional software development models
- Discuss the waterfall model and classical waterfall model
- Understand about traditional IT organizations
- Learn about developers vs IT operations conflict
- Explain birth of Agile, and four values of the Agile manifesto
- Describe about Scrum, Scrum theory, Scrum values, Scrum roles, Scrum master, Scrum sprints, benefits of Scrum, etc.
- Describe planning and estimation, Agile planning, and its levels
- Define conditions of satisfaction and velocity



Module 1: Agile Practices

Module Objectives (Contd.)

- List the various estimating techniques
- Discuss about soft skills in Agile

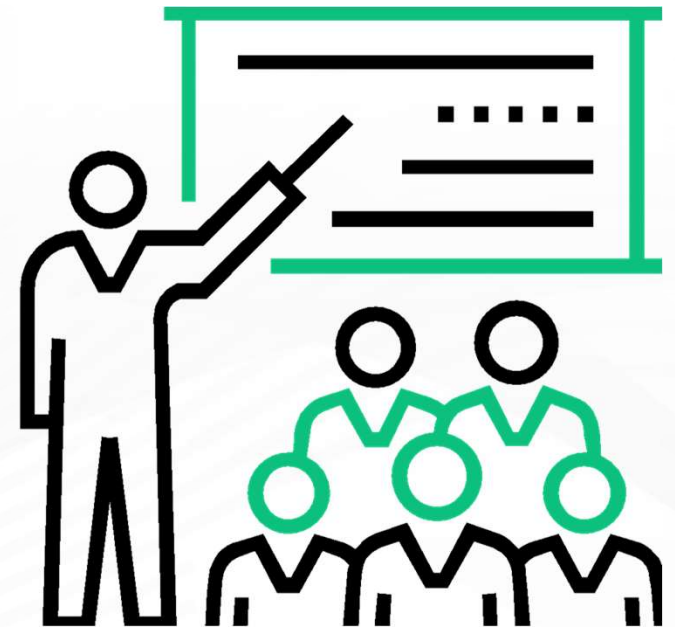


Module 1: Agile Practices

Module Topics

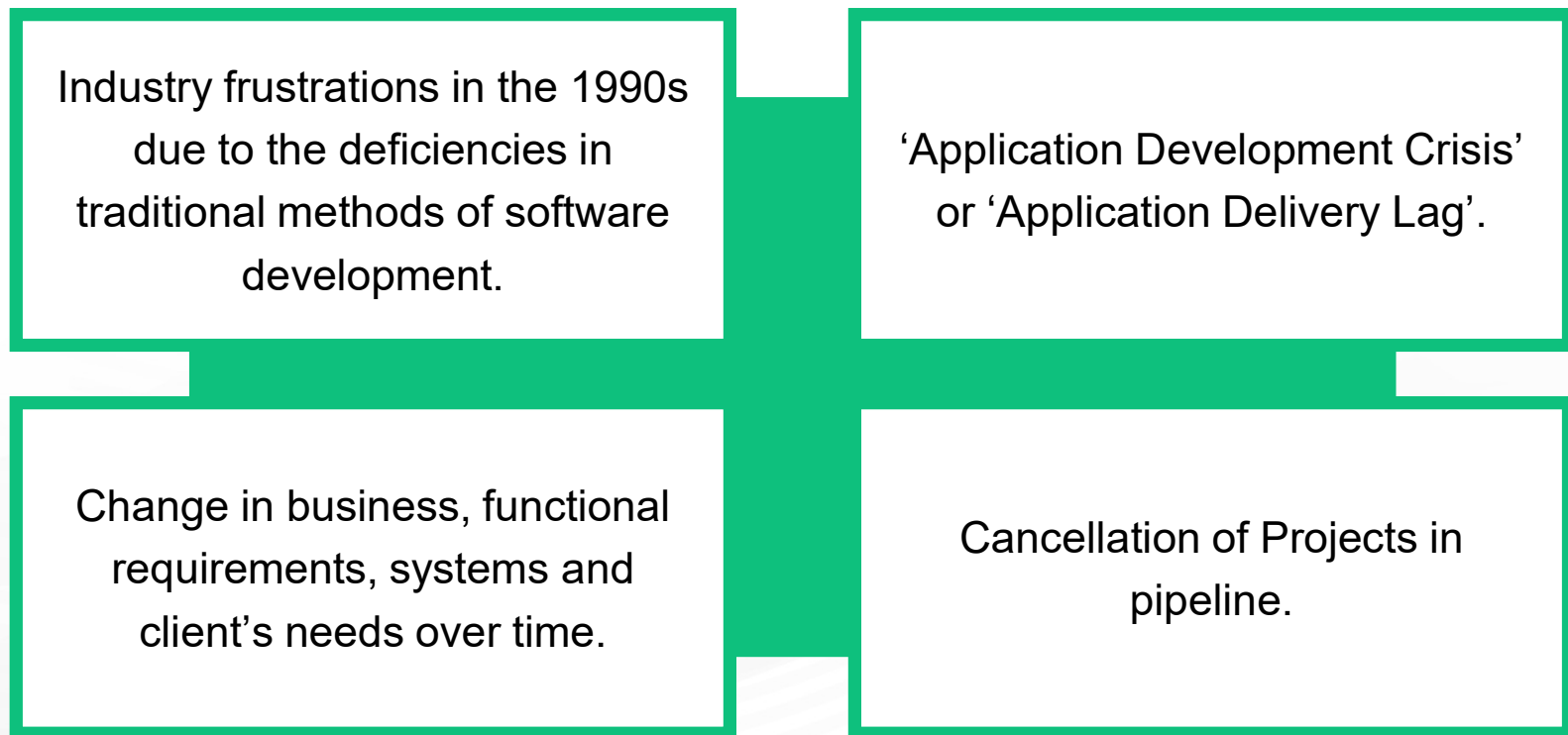
Let us take a quick look at the topics that we will cover in this module:

- Agile methodology
- Software, History of Software Engineering and Software Development Methodologies
- Traditional Software Development Models
- Waterfall Model, Classical Waterfall Model
- Traditional IT Organizations
- Developers vs IT Operations Conflict
- Birth of Agile, Four Values of the Agile Manifesto
- Scrum, Scrum Theory, Scrum Values, Scrum Roles, Scrum Master Scrum Sprints, Benefits of Scrum
- Planning and Estimation, Agile Planning, Levels of Agile Planning
- Conditions of Satisfaction, Velocity
- Estimating Techniques
- Soft Skills in Agile
- Kanban Model



1.1 Agile Methodology - Evolution

The following diagram illustrates the evolution of Agile methodology.

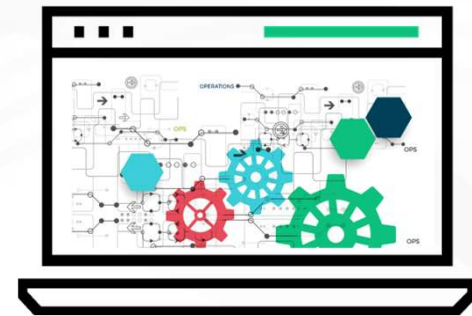


Module 1: Agile Practices

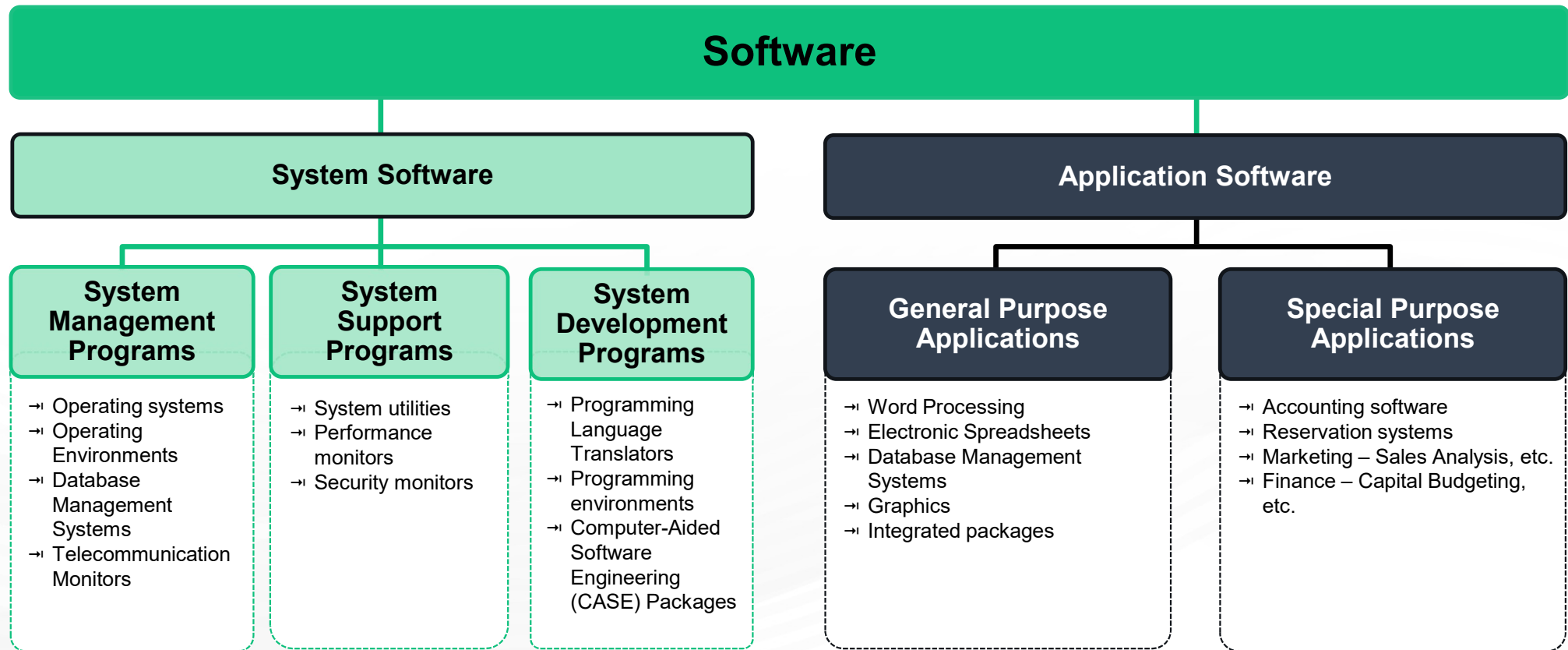
1.2 Software

Following are the key details of the software:

- A software is an organized information in the form of operating systems, programs, utilities and applications that enable a computer to work.
- Software comprises a set of machine-readable instructions that directs a computer's processor to perform specific operations.
- It includes computer programs, libraries and their associated documentation.
- Software programs are stored as files on a storage device such as hard disk, DVD or memory sticks. When needed, they are loaded into the computer's memory (RAM).
- In a nutshell, a software is everything that governs the functioning of a computer, an interface between computers and humans who use them.



1.2.1 Different Types of Software



What did You Grasp?



1. Which of the following is not a system software?

- A) Operating software
- B) Performance monitor
- C) Database management systems
- D) Word processing software

Module 1: Agile Practices

1.3 Definition of Digital Transformation



```
<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li><a href="multi-col-menu.html">Multiple Column Men
<li class="has-children"> <a href="#" class="current"
<ul>
<li><a href="tall-button-header.html">Tall Bu
<li><a href="image-logo.html">Image Logo</a></li>
<li><a href="tall-logo.html">Tall Logo</a></li>
```

Roger S. Pressman

President of R.S. Pressman & Associates

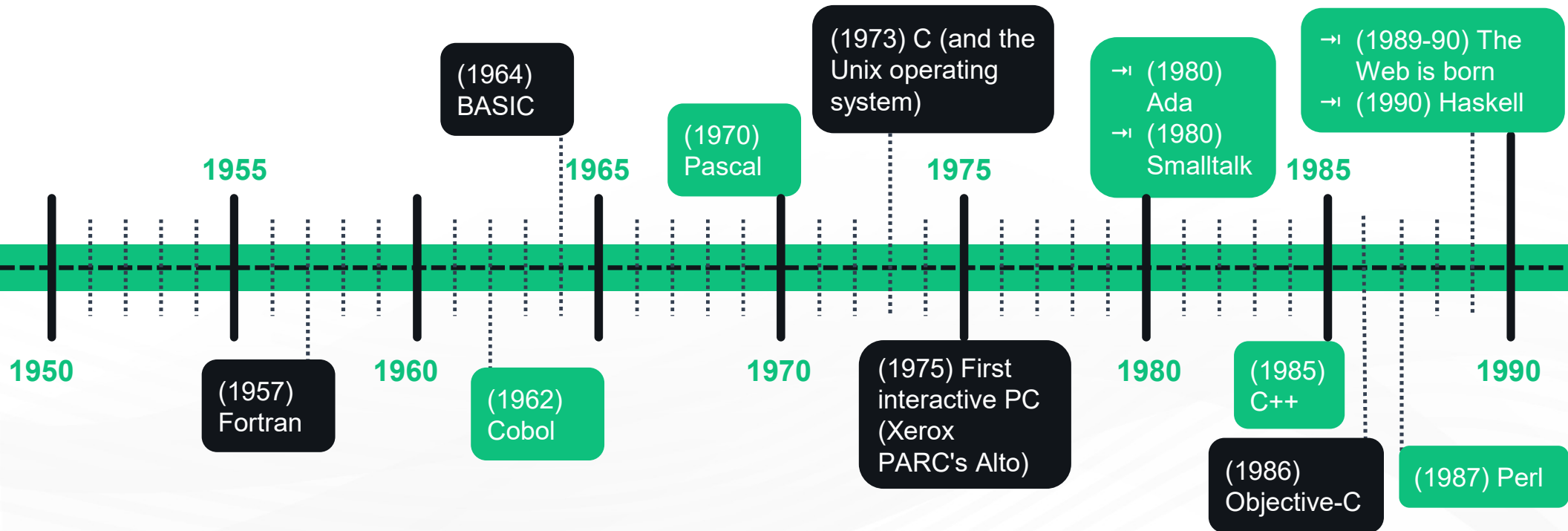
“Software engineering is the technology that encompasses a process, a set of methods and an array of tools that allow professionals to build high quality computer software.”

Source adopted from “Software Engineering: A Practitioner’s Approach” written by Roger S. Pressman



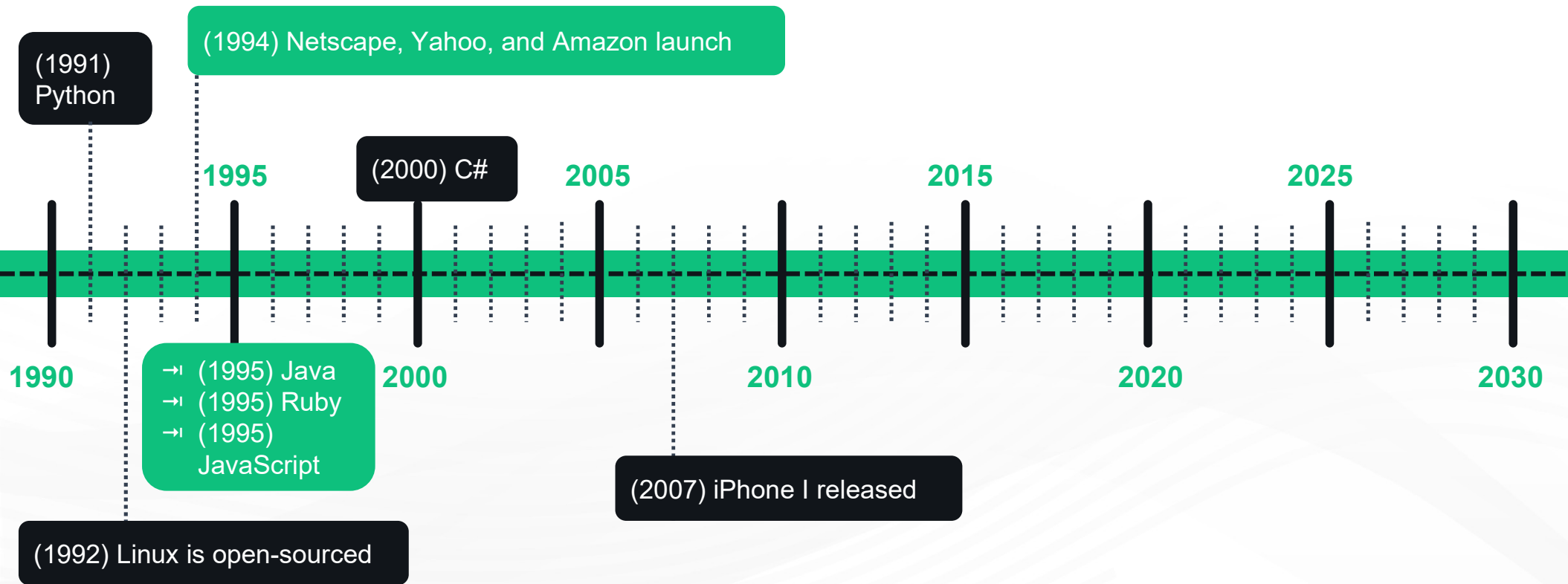
Copyright © 2019, Xebia Group. All rights reserved. This course B.TECH CSE with specialization in DevOps is licensed to UPES.

1.4 History of Software Engineering

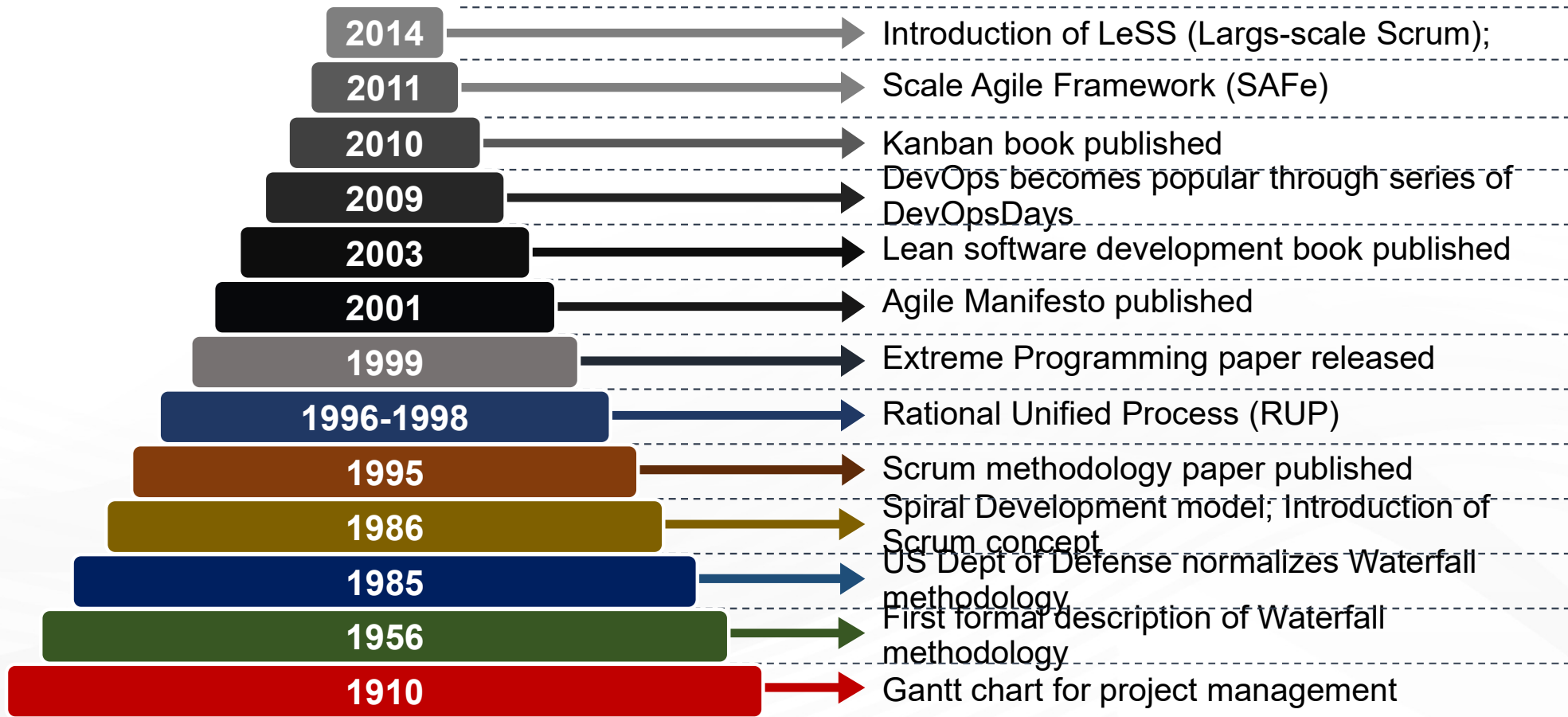


Module 1: Agile Practices

1.4 History of Software Engineering (Contd.)

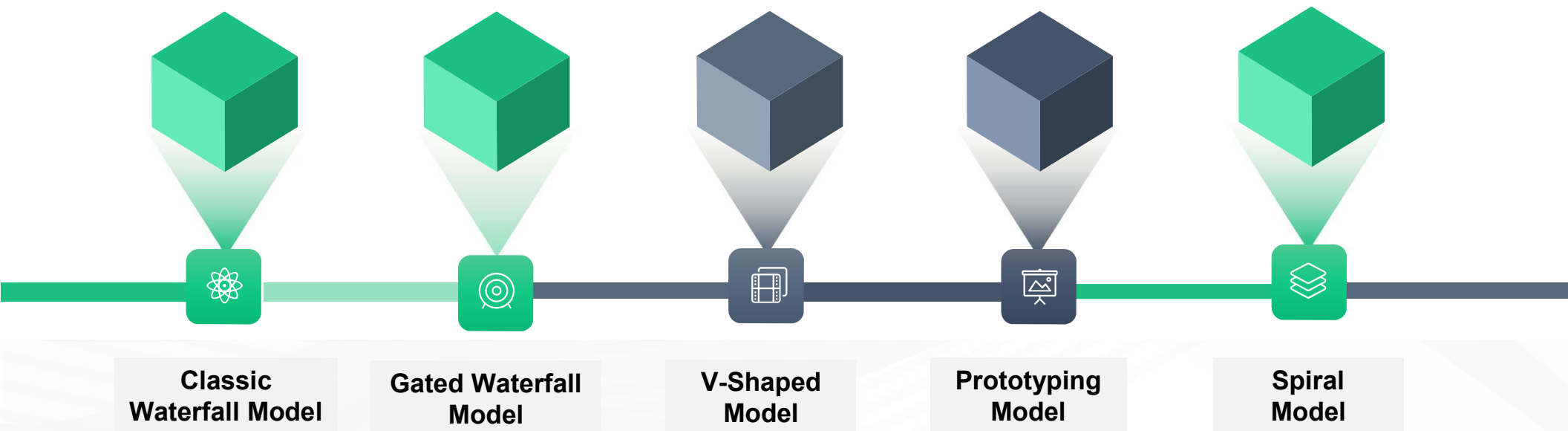


1.5 History of Software Development Methodologies

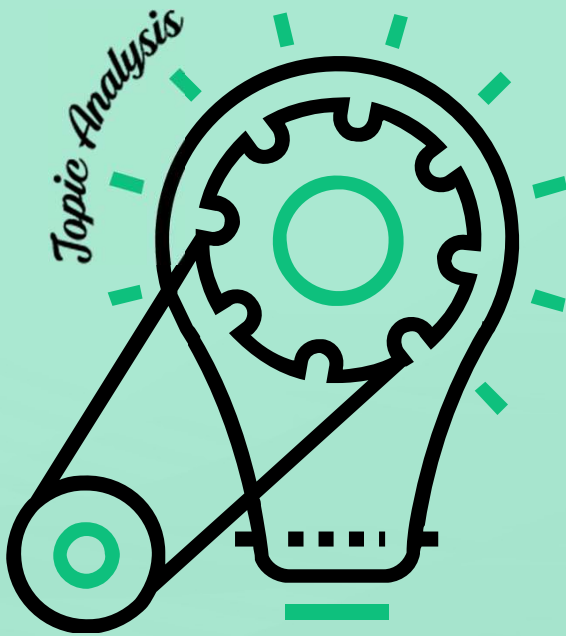


1.6 Traditional Software Development Models

Many life cycle models have been proposed so far, with their own advantages and disadvantages. A few important traditional software development models are as follows:

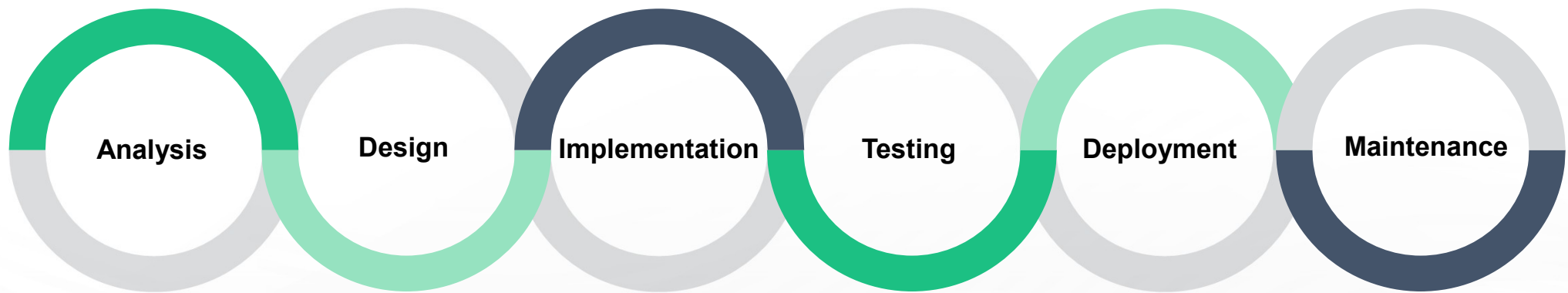


What did You Grasp?



1. In which of the following prototyping methods, initially created prototypes will not become part of the final version of the software?
 - A) Extreme prototyping
 - B) Rapid throwaway prototyping
 - C) Evolutionary prototyping
 - D) Incremental prototyping
2. Which of the following statements about Spiral model is true?
 - A) Define artifacts sequentially
 - B) There are four essential spiral tasks
 - C) Risk determines level of effort
 - D) Risk determines degree of details

1.7 Waterfall Model

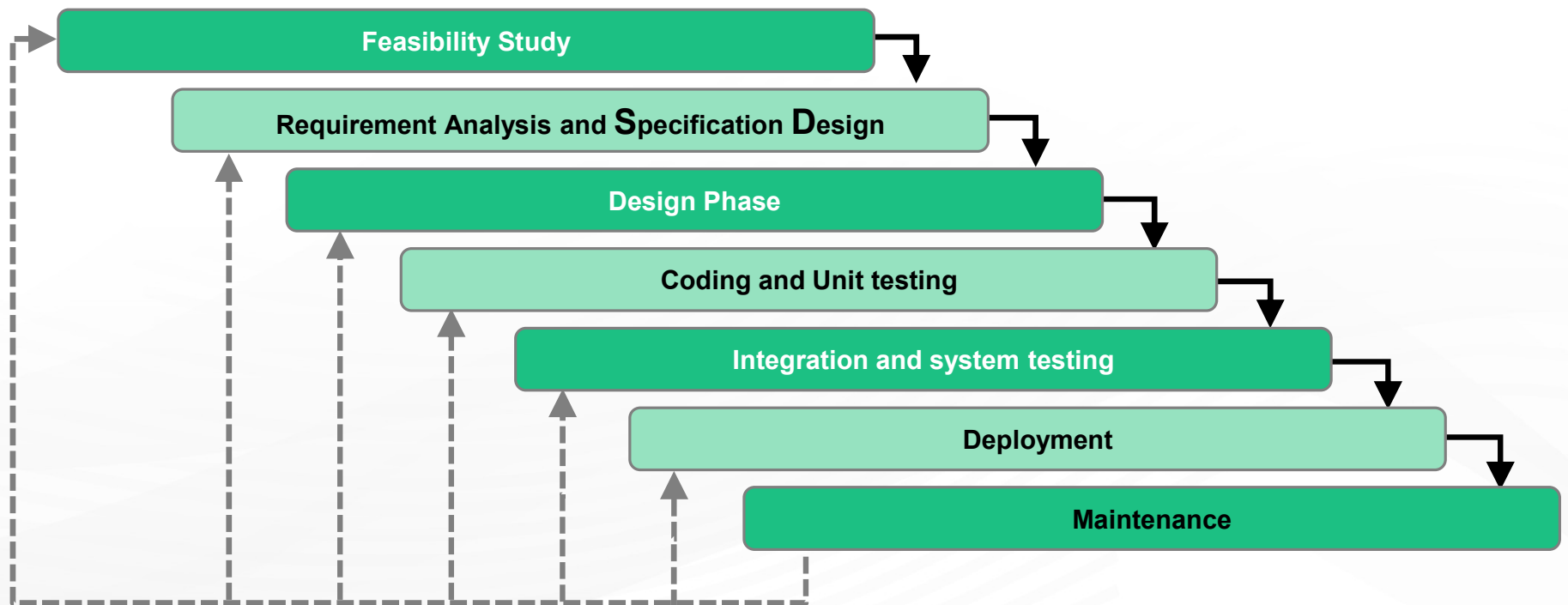


Source adopted from "International Journal of Engineering & Technology (IJET)"

Copyright © 2019, Xebia Group. All rights reserved. This course B.TECH CSE with specialization in DevOps is licensed to UPES.

1.8 Classical Waterfall Model

In the waterfall approach, the outcome of one phase acts as the input for the next phase sequentially. The following picture illustrates the flow of steps in the waterfall model.



1.8.1 First Stage – Feasibility Study

The following activities are covered during the feasibility study phase:

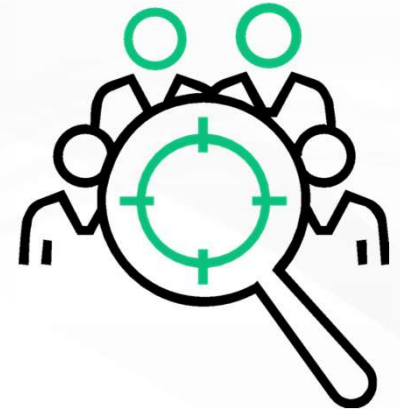
Financial Feasibility

Technical Feasibility

Client Visit

Study Of Input-output Data

Case Study



1.8.2 Second Stage – Requirements Analysis and Specification Design

The two major activities carried out during the requirements analysis and specification design phase.

Requirements gathering and analysis

- All relevant information regarding the product is collected from the customer and users through interviews and discussions.
- All the ambiguities and contradictions are identified and resolved.

Requirements specification

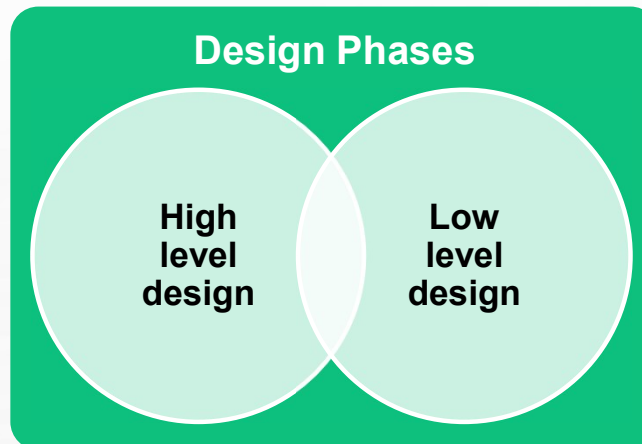
- User requirements are captured and organized into a Software Requirements Specification (SRS) document.
- The main components of SRS are functional requirements, non-functional requirements and goals for implementation.
- Very little information on top-level analysis and design is also included in this document.

1.8.3 Third Stage – Design Phase

The following activities are covered during the design phase:

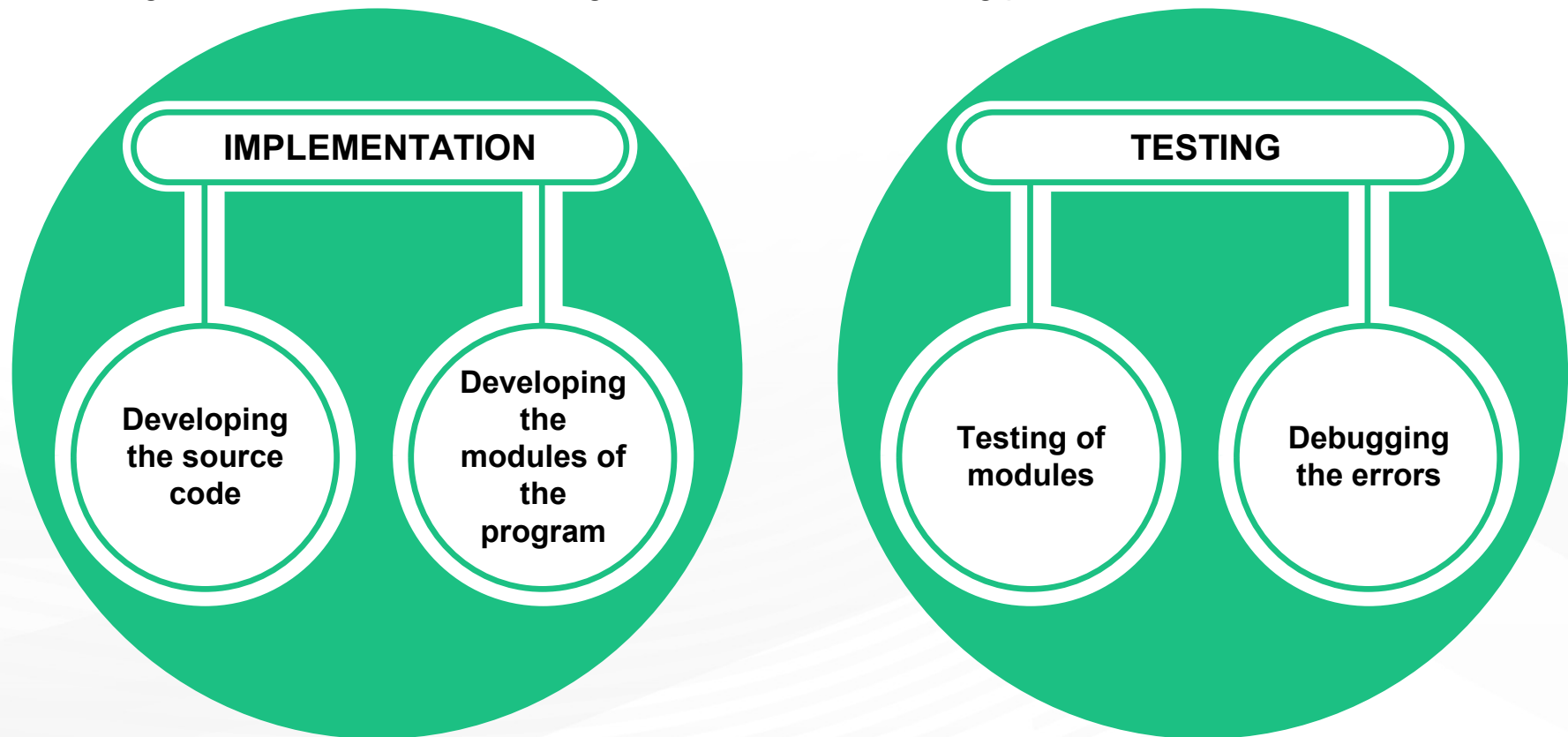
- Creation of system design as per the requirements gathered during the previous phase, i.e., based on SRS document
- Functionality of hardware and software are separated out and the software modules are designed
- Definition of overall system architecture
- Documentation of design

There are two levels in the design phase:



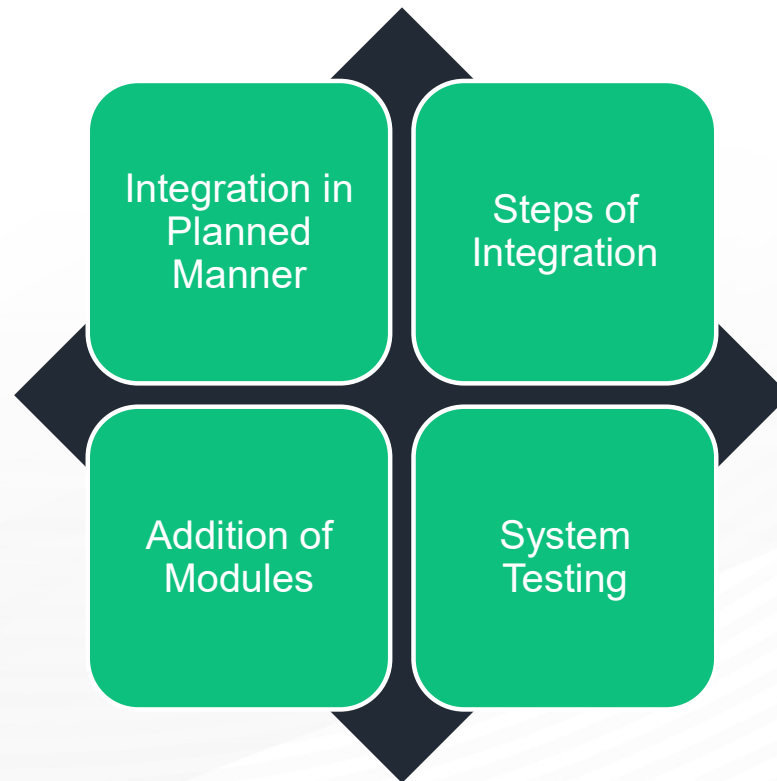
1.8.4 Fourth Stage – Coding and Unit Testing

The following activities are covered during the code and unit testing phase:



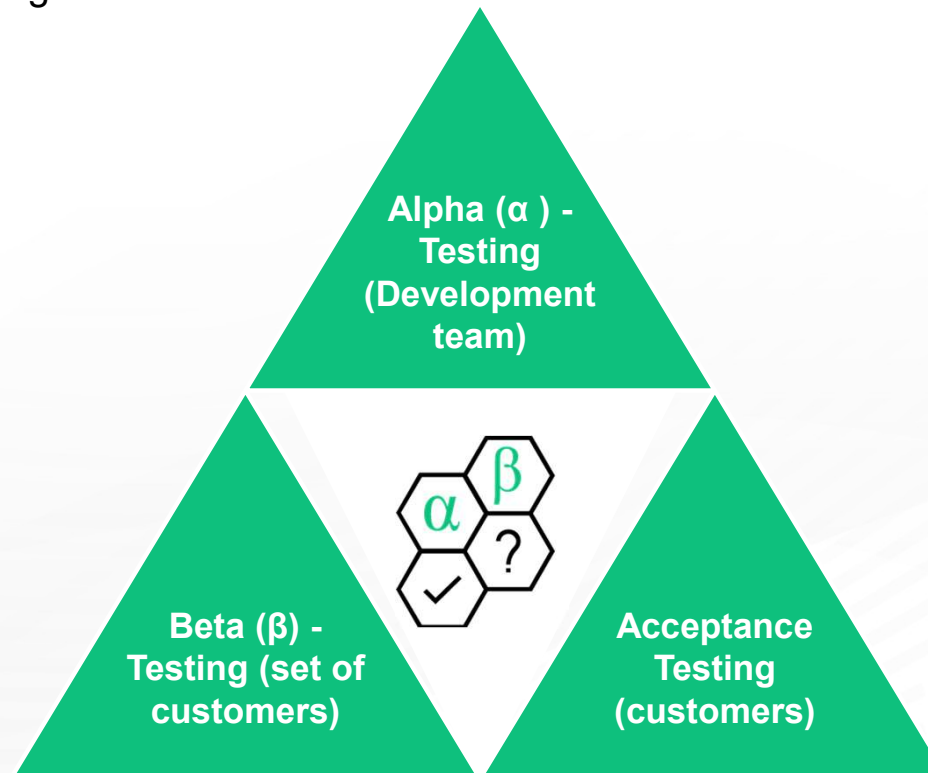
1.8.5 Fifth Stage – Integration and System Testing

The following activities are covered during the integration and system testing phase:



1.8.5 Fifth Stage – Integration and System Testing (Contd.)

This is a level of software testing where a complete and integrated software is tested. The various tests may include the following:



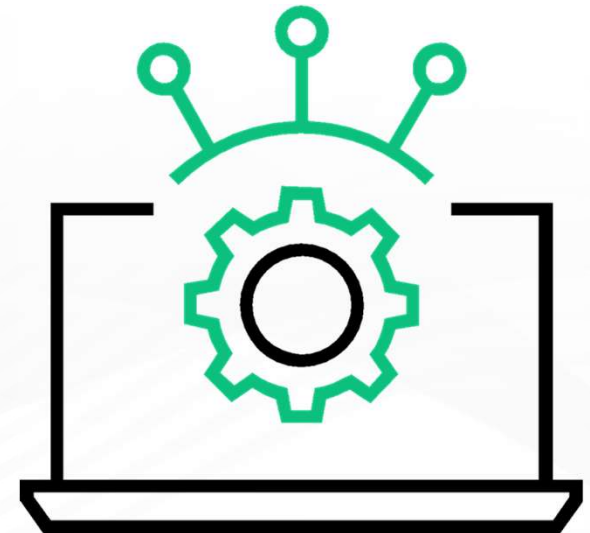
1.8.6 Sixth Stage – System Deployment and Maintenance

The following activities are covered during the system deployment and maintenance phase.

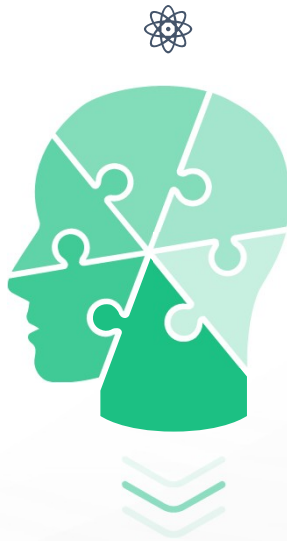
- Product Deployment
- Development and Maintenance

Maintenance

- Corrective Maintenance
- Perfective Maintenance
- Adaptive Maintenance

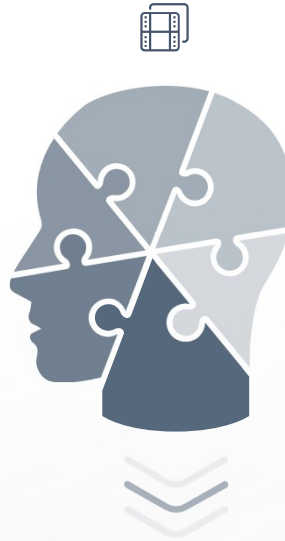


1.9 Advantages of Waterfall Model



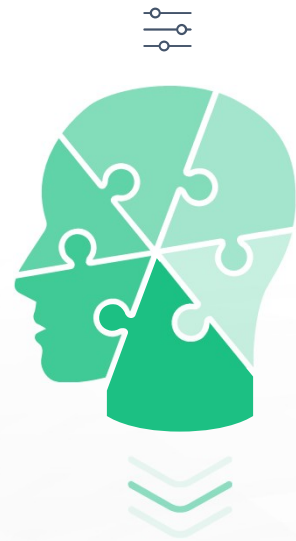
Discipline

Every phase has a defined start and end point. The progress can be distinctly identified by both, the client and the software developer, through the use of milestones.



Time and Cost Effective

The emphasis on requirements and design, before writing the codes/programs ensure the minimal wastage of time, effort and cost and decreases the risk of slipping of schedule.



Quality Improvement

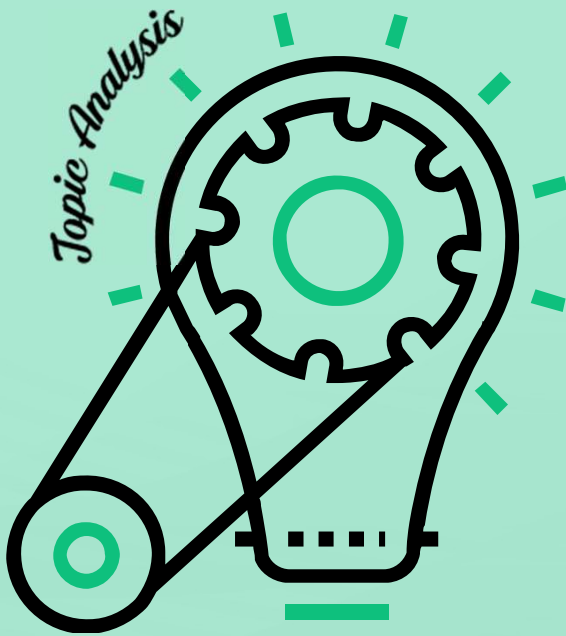
The flaws can be easily caught and taken care of at the design stage, much earlier than the testing stage.

1.10 Shortcomings of the Waterfall Model

- ✓ **Possibilities of errors**
- ✓ **Not suitable for all projects**
- ✓ **Implicit assumptions**
- ✓ **High risk and uncertainty**
- ✓ **Non-adaptive design constraints**
- ✓ **Ignores mid-process user/client feedback**
- ✓ **Delayed testing period**
- ✓ **Makes changes difficult**

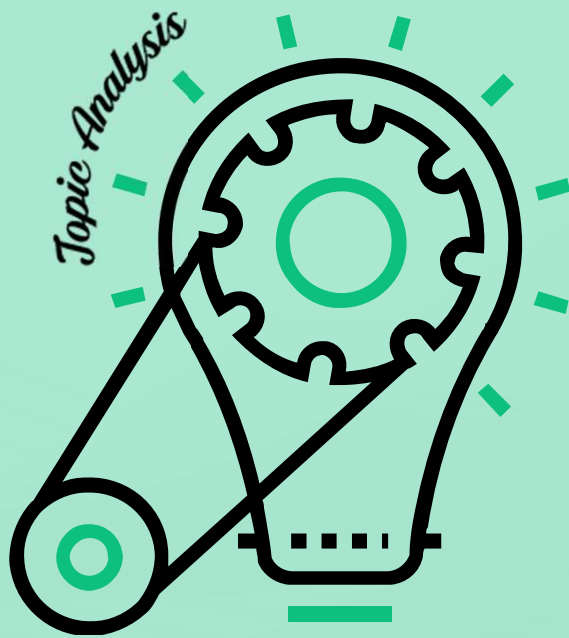


What did You Grasp?



1. Which model is also called as the classic life cycle or the Waterfall model?
A) Iterative Development
B) Linear Sequential Development
C) RAD Model
D) Incremental Development
2. What is the simplest model of software development paradigm?
A) V-model
B) Spiral model
C) Big Bang model
D) Waterfall model

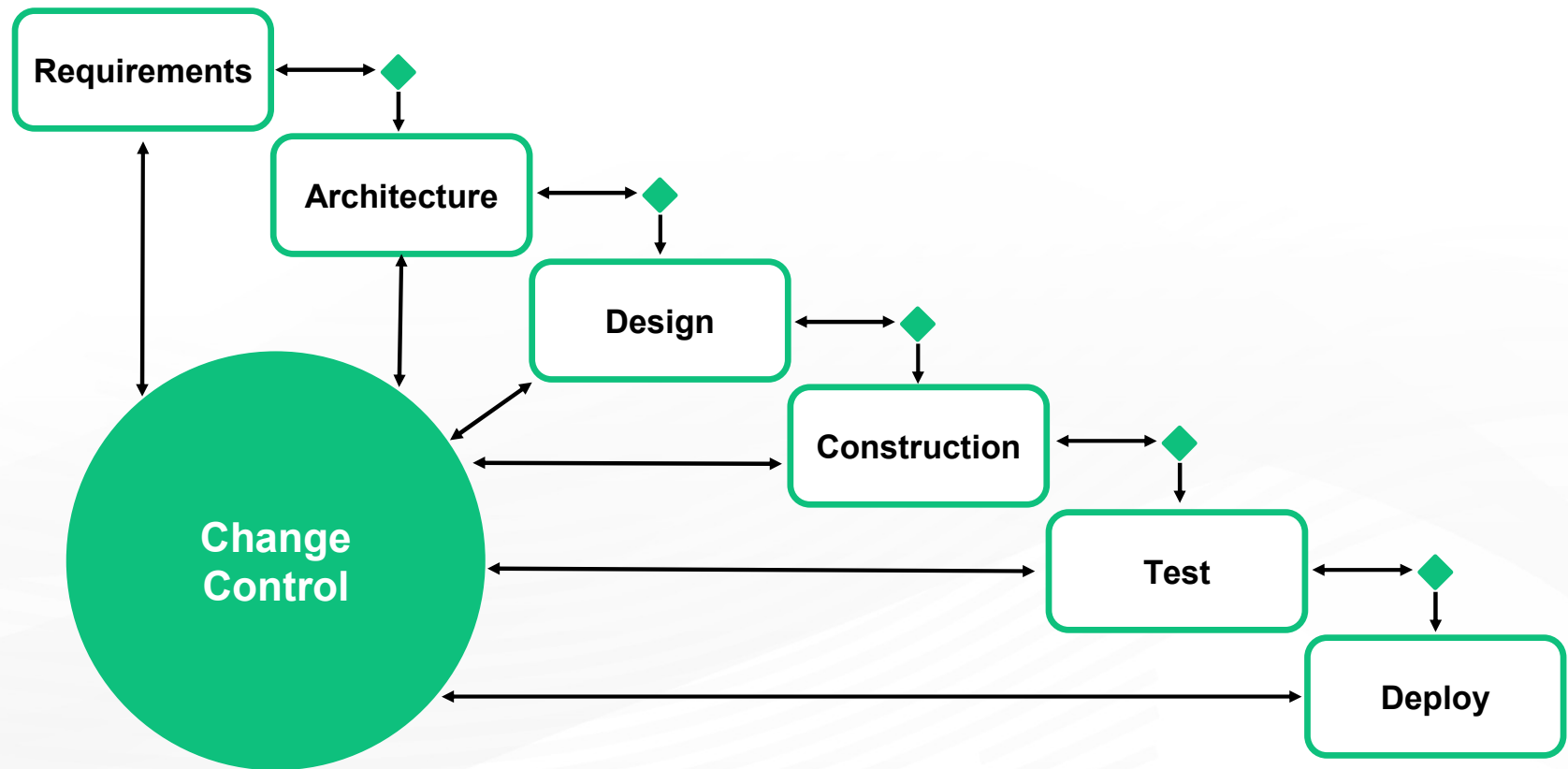
What did You Grasp?



3. In which of the following phases of maintenance, the software is ported to work in a new environment?
 - A) Corrective maintenance
 - B) Perfective maintenance
 - C) Adaptive maintenance
 - D) None of the above
4. Which of the following testing is carried out by a primary group of customers?
 - A) Alpha-testing
 - B) Beta-testing
 - C) Gamma-testing
 - D) Delta-testing

1.11 Gated Waterfall Model

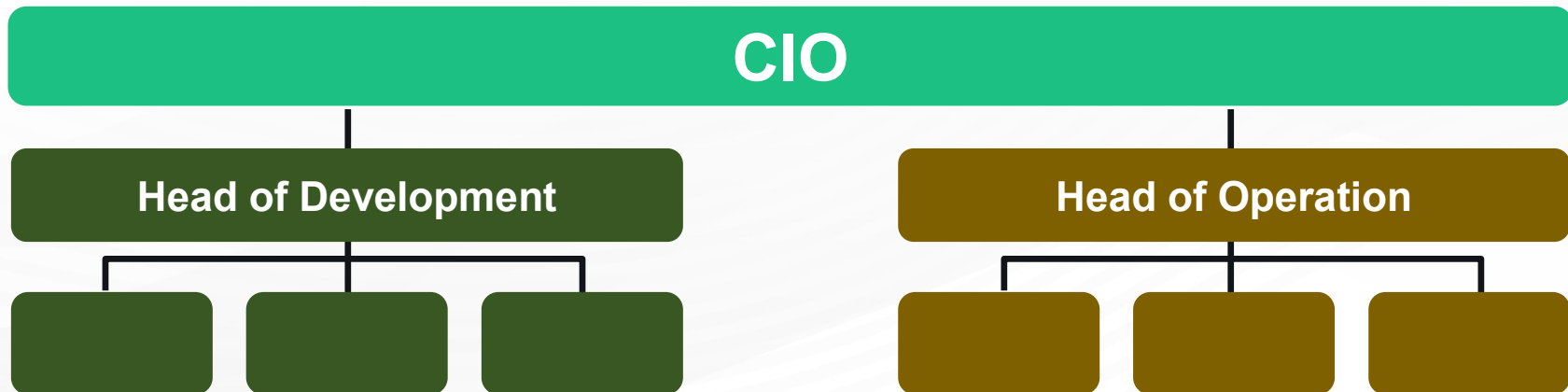
The following picture depicts the phases in the gated waterfall model.



1.12 Traditional IT Organizations

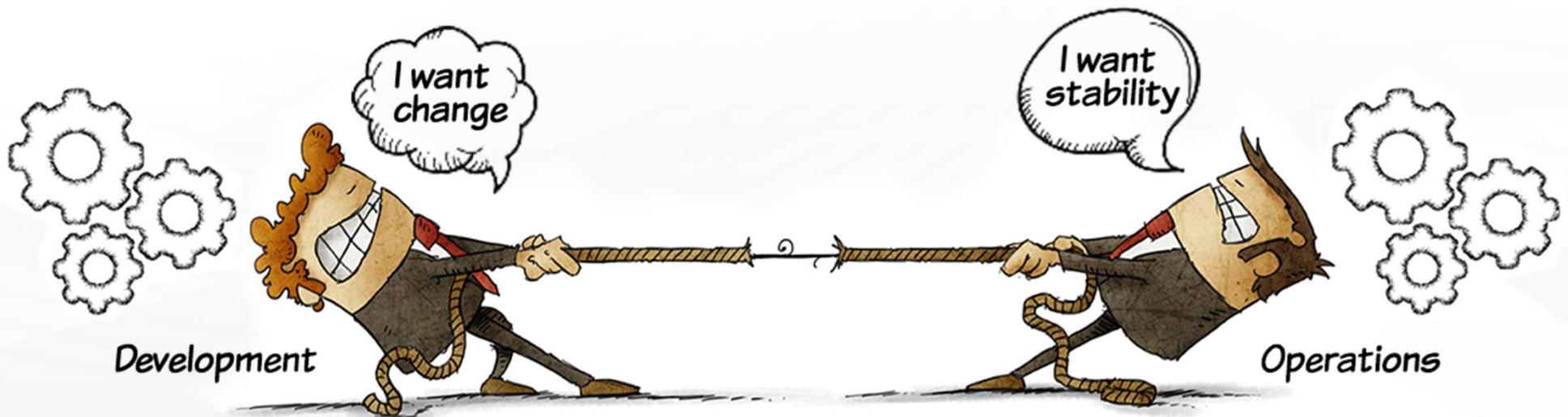
Cultural hindrance between development and operations teams in traditional SDLC.

- Development and operations are two sides of an equation holding their own roles and responsibilities.
- The development team works independently on code. The code then sent to the testing team for validation.
- Operations team comes in toward the end of the process, handover of the release.
- There is no collaboration between these teams.



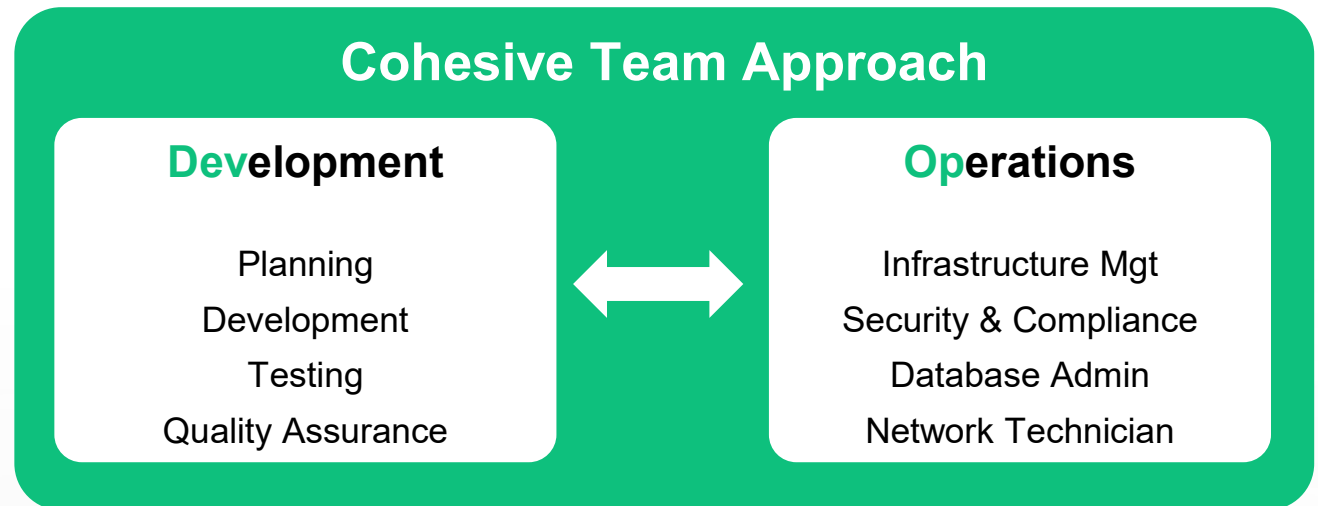
1.13 Developers vs IT Operations Conflict

1. Meaning
2. Development Changes
3. Confusions /Lack Of Communication
4. Operations Stability

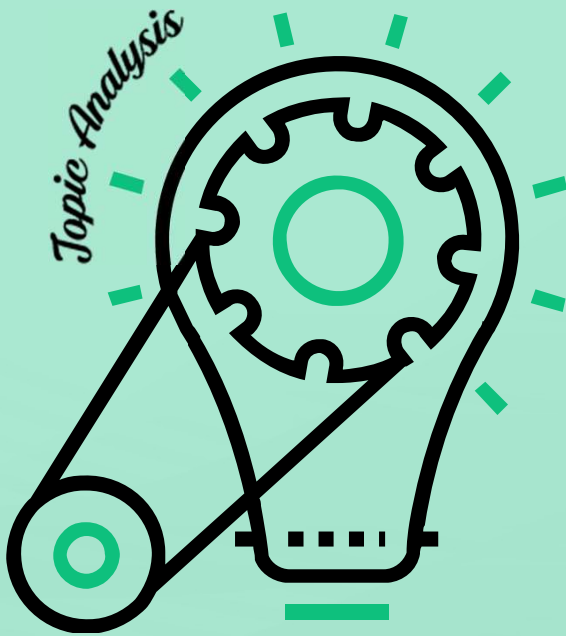


1.14 Problems with the Traditional Development and the Operations

- Organizational silos
- Different mindsets
- Different implementations
- Different tools
- Lack of interest in learning other tools
- Different environments
- Loss of work
- Blame game
- Build rollback
- Disintegrated process
- No feedback loop



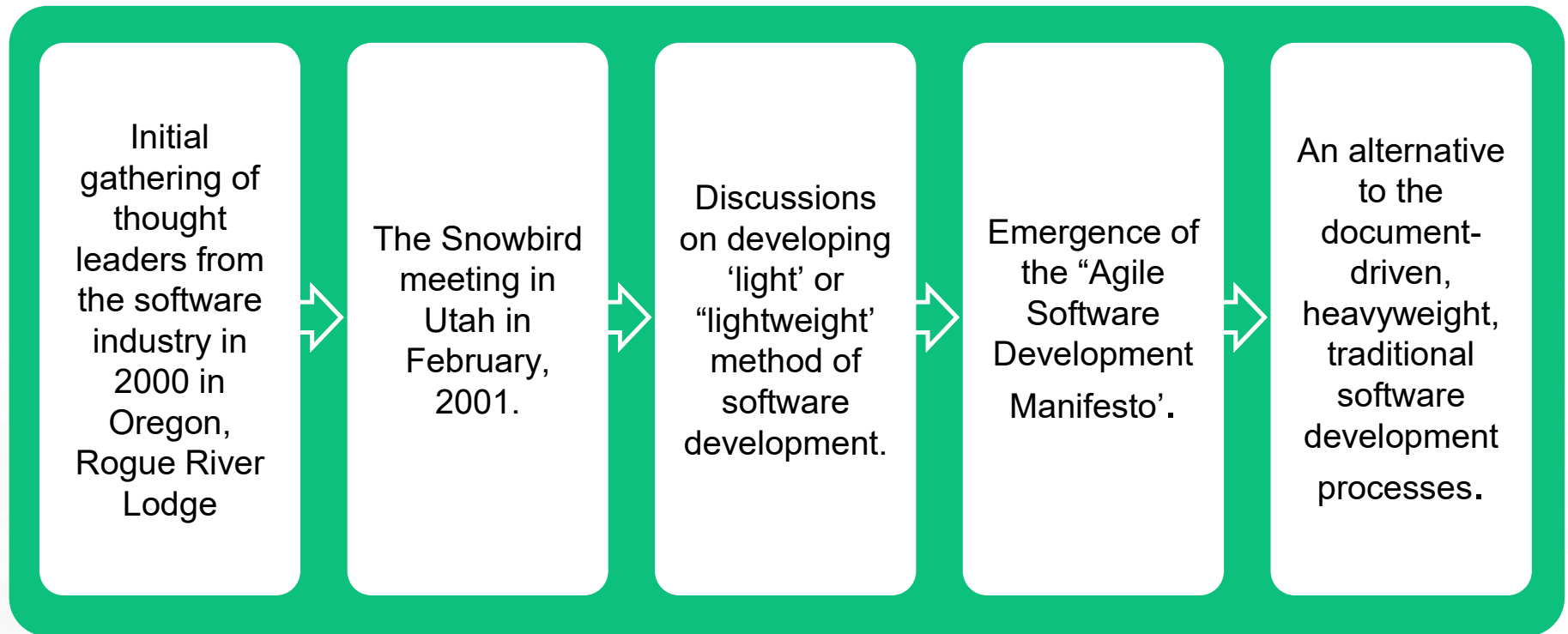
What did You Grasp?



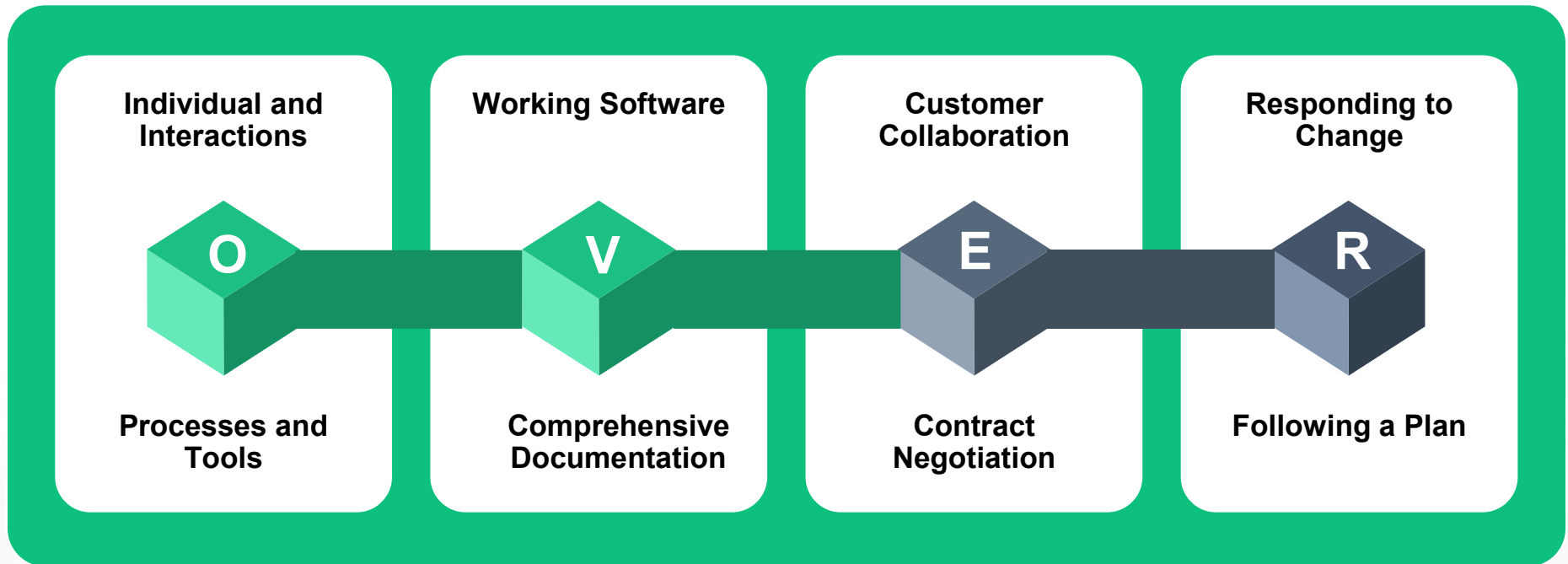
1. There is a negative effect on deployment and delivery dates in case of conflict between the developers and the IT operations.
A) True
B) False
2. Which of the following is NOT a consequence of wall of confusion?
A) Lack of communication
B) Errors and bugs due to the use of different tools
C) Application delivery in a fast pace
D) No methodical handover

1.15 Birth of Agile

The following is the process illustrates the birth of Agile:



1.16 Four Values of the Agile Manifesto



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

1.16.1 Agile Manifesto

Agile Manifesto Value 1: Individuals and interactions over processes and tools

INDIVIDUALS AND INTERACTIONS

- Communication—the key to the success of a project
- People build Products
- Have the focus on people and the source of energy
- Self-organization of cross-functional teams—to identify scope, negotiate, accept, define, collaborate, share, and solve problems
- Individuals need to be motivated
- Interactions need to be fostered among team members, customers and other stakeholders

1.16.1 Agile Manifesto (Contd.)

Agile Manifesto Value 2: Working software over comprehensive documentation

WORKING SOFTWARE

- Create and deliver value
- Customer satisfaction is important
- Deliver frequently and consistently; offer business value to the customer
- The primary goal of software development is to create software, not lengthy documents
- Write documentation that adds value
- Deliver what the customer wants; documentation is always supplementary
- Customer-focus is the primary focus

1.16.1 Agile Manifesto (Contd.)

Agile Manifesto Value 3: Customer collaboration over contract negotiation

CUSTOMER COLLABORATION

- Flexibility and co-operation in terms of customer's needs
- Work with the customer
- Make sure the intent of the contract is satisfied
- Understand customer's product vision by close collaboration
- Let the contracting models be flexible
- Maintain relationships

1.16.1 Agile Manifesto (Contd.)

Agile Manifesto Value 4: Responding to change over following a plan

RESPONDING TO CHANGE

- Change is the reality, the worst enemy of any plan
- Changes in customer's business needs - direct impact on developer's plans
- Adaptive planning for accepting change
- Change to be reflected in the product
- Show improvement

1.16.2 Twelve Principles of the Agile Manifesto

- 1 Our highest priority is to satisfy the customer through an early and continuous delivery of valuable software.
- 2 Welcome changing requirements, even late in the development. Agile processes harness change for the customer's competitive advantage.
- 3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4 Business people and developers must work together daily 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 The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.
- 7 Working software is the primary measure of progress.
- 8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 9 Continuous attention to technical excellence and good design enhances agility.
- 10 Simplicity—the art of maximizing the amount of work not done—is essential.
- 11 The best architectures, requirements, and designs emerge from self-organizing teams.
- 12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

1.17 Scrum: An Introduction

Following are the key details of Scrum:

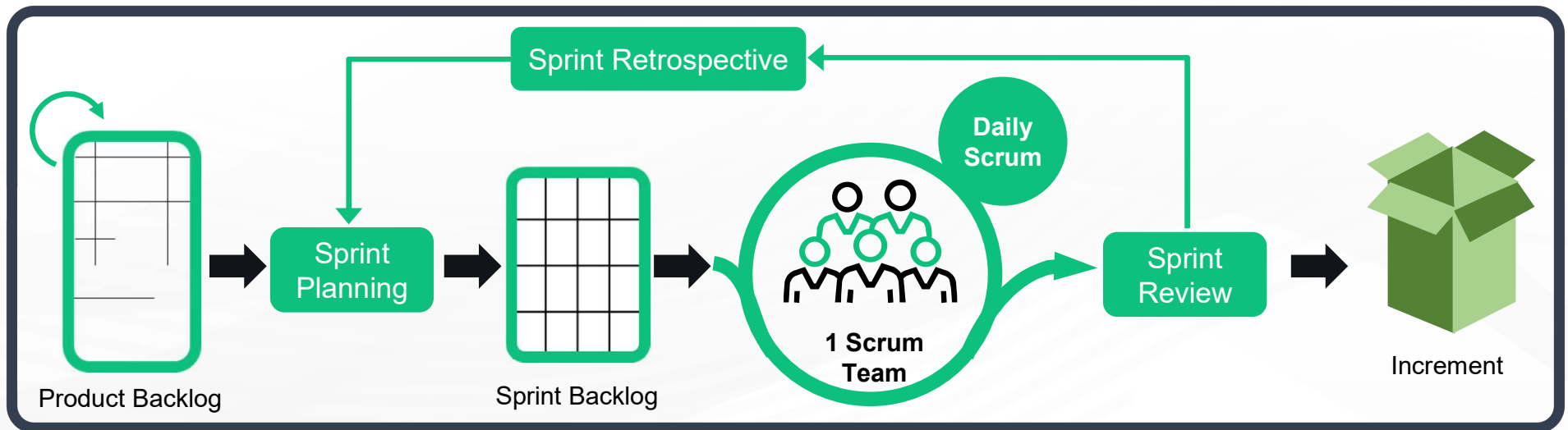
- Scrum is one of the most popular and widely used frameworks to implement Agile in industry projects.
- Scrum has gained so much popularity that it is used synonymously with Agile, but in reality they are different.
- Scrum is a framework used for incremental product development by deploying one or more cross-functional, self-organizing teams.
- Using scrum, product development is carried out in small, fixed-length iterations, called sprints.
- Sprints are not longer than 30 days, short sprints are preferred. Every sprint is intended to build a potentially shippable product increment.

Module 1: Agile Practices

1.17.1 Why Scrum?

- Scrum is an alternative to the traditional waterfall method.
- Where traditional approaches depends on complete understanding of the requirements before starting off, scrum includes all the development activities in each of the sprints.
- Scrum focuses on developing high-value features first with the frequent incorporation of feedback.

The following figure shows the overall scrum framework.

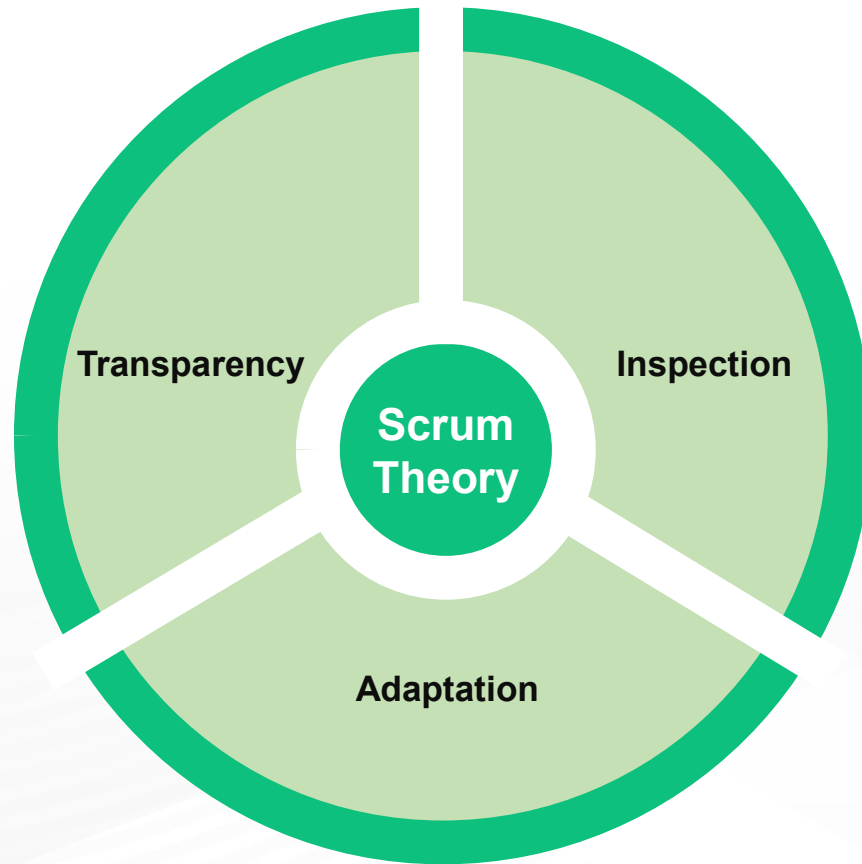


Source: Scrum.org.

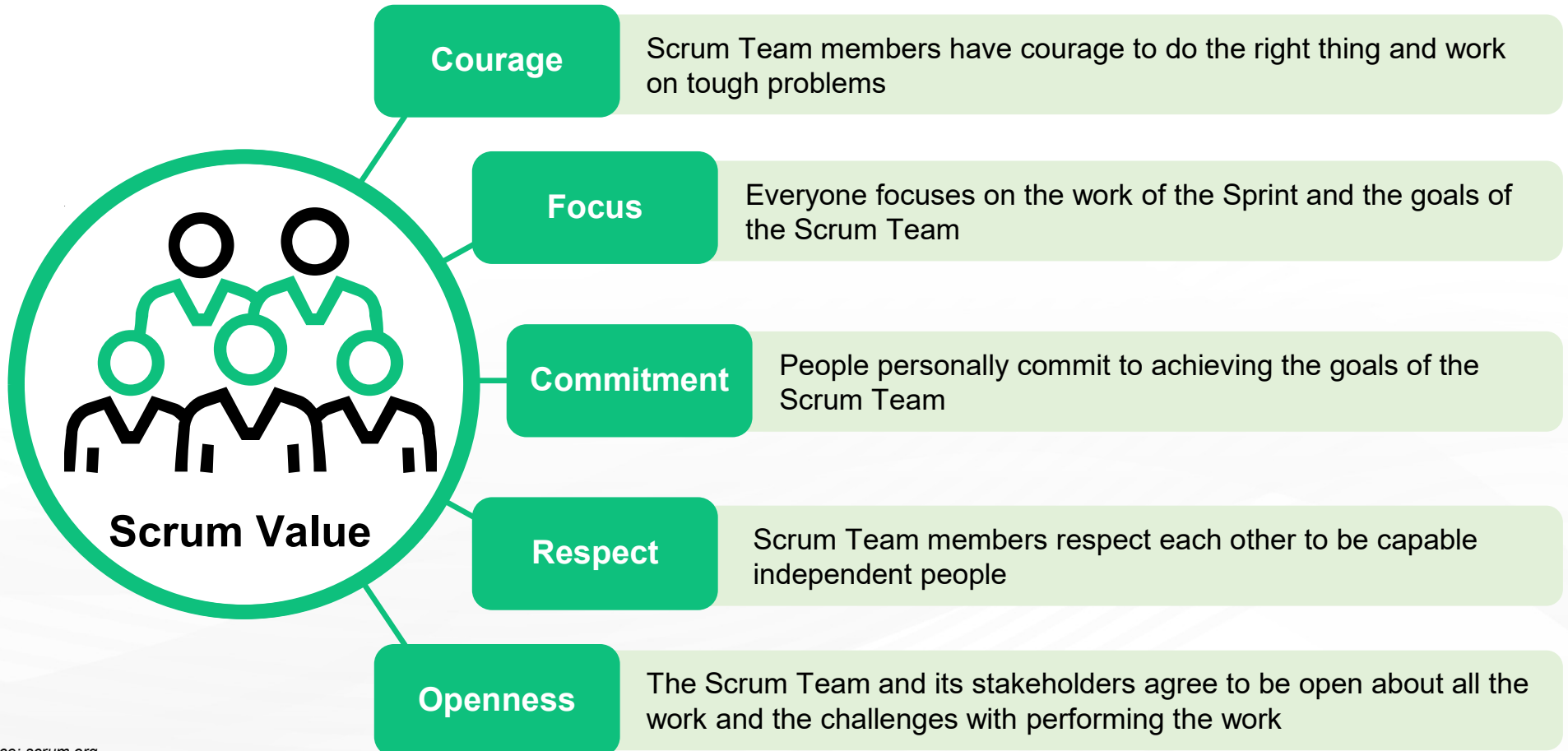
Copyright © 2019, Xebia Group. All rights reserved. This course B.TECH CSE with specialization in DevOps is licensed to UPES.

1.17.2 Scrum Theory

Scrum is based on empirical process control theory, called empiricism. This ensures that knowledge originates from experience and decisions are taken based on known things. The picture illustrates the three pillars of scrum theory.



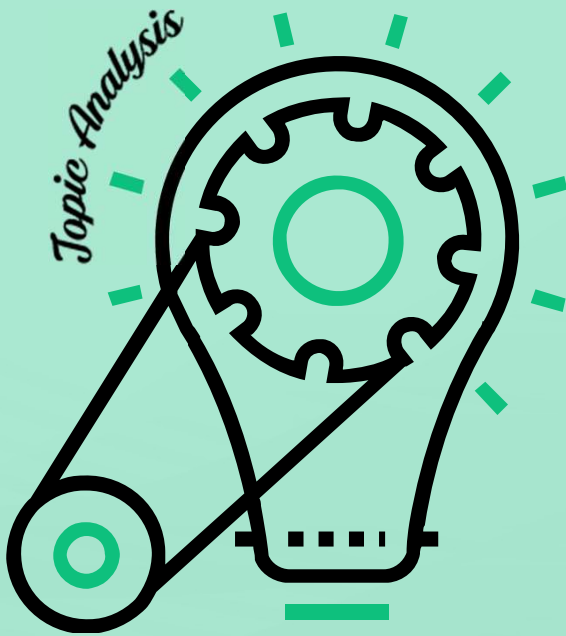
1.17.3 Scrum Values



Source: [scrum.org](https://www.scrum.org)

Copyright © 2019, Xebia Group. All rights reserved. This course B.TECH CSE with specialization in DevOps is licensed to UPES.

What did You Grasp?



1. What are the iterations in Scrum called?
A) Release
B) Sprint
C) Release cycles
D) Build
2. Which of the following aspects of the Scrum theory ensures that every stakeholder understands the aspects in the same way?
A) Transparency
B) Inspection
C) Adaptation
D) Commitment

1.18 Scrum Roles

The following picture illustrates the three scrum roles in comparison with the traditional approach.

Product Owner

The Holder of Product Value

Determines what needs to be done and sets the priorities to deliver the highest value

Traditional approach: Controls the work

Scrum Master

The Servant Leader

Protecting the Scrum process and preventing distractions

Traditional approach: No equivalent

Development Team

The Self-Organizing Group

Takes on and determines how to deliver chunks of work in frequent increments

Traditional approach: Gets told what to do by the project manager

1.18.1 Scrum Master

The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide.

It is the responsibility of the Scrum Master to support the product owner, development team and the organization.

The following are the qualities of a Scrum Master, who has a leadership role:

- Works with organizations in implementing Scrum
- Makes sure that Scrum is understood and implemented
- Makes the environment workable, that also ensures team self-organization
- Guards the team from external interference and distractions and ensures harmony
- Helps to encourage improved engineering practices
- Doesn't manage the team
- Helps overcome hurdles

1.18.2 Product Owner

Product Owner holds the complete responsibility of the product and is called Product Value Maximizer. Product owners understand the business and market requirements and plan the work to be done by the development team.

The responsibilities and qualities of a product owner include:

- Building and managing the product backlog
- Works closely with the organization and the Team to make sure everyone understands the work items in the product backlog
- Product Owner takes the sole responsibility of maximizing the return on investment (ROI) of the development effort
- Defines and constantly re-prioritizes the product backlog, makes adjustments to long-term expectations such as release plans
- Decides the set of requirements that the development team has to work on
- Remains accountable for the work done by the development team on product backlog
- Decides when to ship the product, by also taking into account frequent delivery

1.18.3 Scrum Development Team

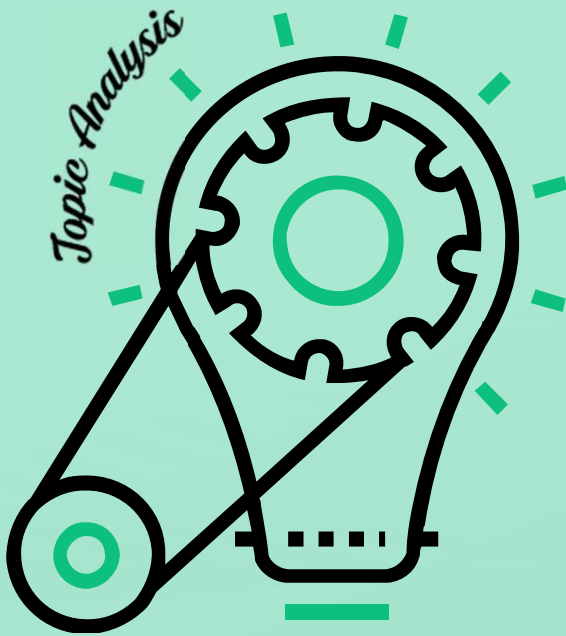
A scrum development team is a cross-functional, self-organizing team with varied skill sets.

Characteristics of a development team are as follows:

- Collaborative, cross-functional, i.e., people of varied roles, with varied skills, who carry out different functionalities
- The Development Team has the capabilities to self-organize/self-manage, with roles assigned internally
- Along with the Product Owner, the team plans sprints, one at a time
- Development Team holds the autonomous right to plan the increment
- Initial few sprints require the team to sit together and work
- Scrum is against moving people across or splitting them between teams
- Ideal number per team is 6 +/- 3 members

Module 1: Agile Practices

What did You Grasp?

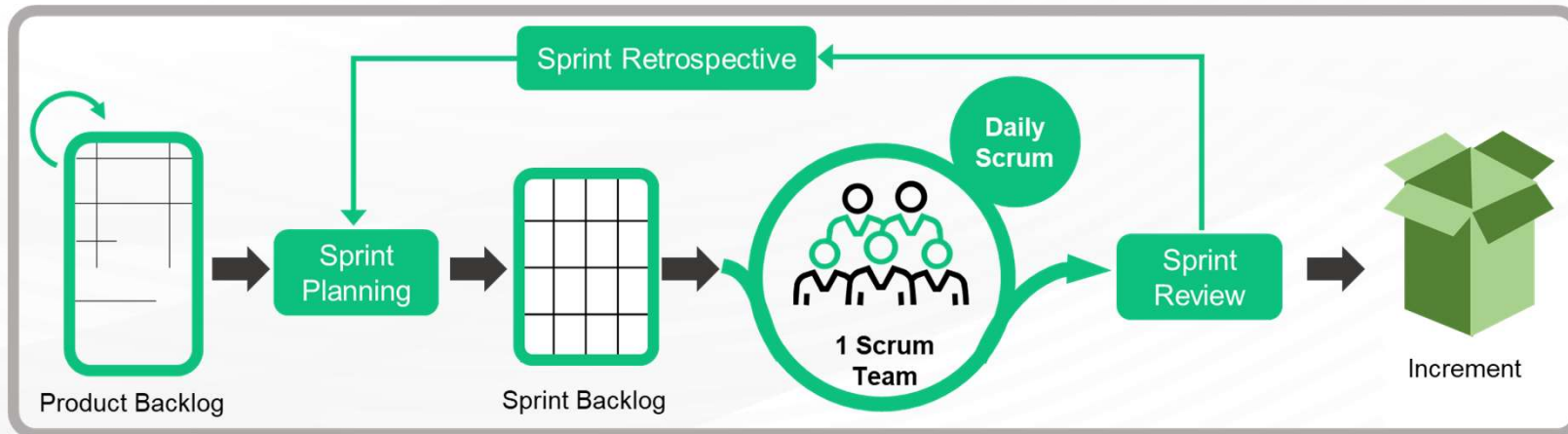


1. Who is responsible for protecting the team from internal and external distractions?
A) Product Owner
B) Scrum Master
C) The team itself
D) None of the above
2. Who builds the product backlog?
A) Product Owner
B) Scrum Master
C) The Development Team
D) Individual team member

1.19 Scrum Sprints

Following are the key details of Scrum sprints:

- A sprint is a short, time-boxed period, during which a scrum team works and completes a certain amount of work.
- With scrum, any product is built in a series of sprints, one iteration of the product in each sprint.
- Sprints are used to break down big, complex projects into bite-sized pieces.
- Sprint is the heart of Scrum, during which a useable and potentially releasable product Increment is created.
- Sprints can be typically be of one week to one month in length and happen one right after the other to keep projects moving. Shorter Sprints are generally preferred.



1.20 Scrum Ceremonies or Events

Events are described in Scrum for creating regularity and to reduce the need for meetings that are not described in Scrum. The picture illustrates the four major Scrum events:



1.20.1 Sprint Planning

Following are the key details of Sprint planning:

- At the beginning of each Sprint, the Product Owner and team hold a Sprint Planning Meeting to negotiate which Product Backlog Items (PBIs) will be converted to a working product during the Sprint.
- Scrum Master ensures that planning happens and the team understands the responsibilities.
- The Product Owner orders the product backlog based on business and technical requirements.
- The Development Team finalizes the amount of work that can be accomplished at the end of a Sprint.
- During the Sprint Planning Meeting, the team decides the ways of accomplishing the work.
- The time-box of Sprint Planning for a 30-day Sprint is eight hours, and is reduced proportionally for a shorter Sprint.

1.20.2 Daily Scrum

Following are the key details of daily Scrum:

- The Daily Scrum is a 15-minute time-boxed event for the Development Team, held every day of the Sprint.
- During the Daily Scrum, the development team plans the work for the next 24 hours. The team shares the work done on the previous day, what will be done that day and what are the roadblocks.
- The Daily Scrum is held at the same time and place to avoid the complexities and this also ensures that team collaboration and performance is optimized.
- The meeting is structured by the development team and it is the internal meeting of the development team.
- The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or re-plan, the rest of the Sprint's work.
- The Scrum Master ensures that the Development Team conducts the meeting, but the Development Team is responsible for conducting the Daily Scrum.

1.20.3 Sprint Review

Following are the key details of Sprint review:

- A Sprint Review is held at the end of each Sprint to inspect the Increment and adapt the Product Backlog if needed.
- The Product Owner reviews the PBIs selected during the Sprint Planning Meeting and explains the ones that are considered done.
- The Product Owner and stakeholders convert their feedback to new PBIs and the Product Owner does the ordering.
- The Sprint Review Meeting is helpful for external stakeholders, including the end users.
- Sprint review is time-boxed to 4 hours for a one-month Sprint and is proportionally reduced for shorter Sprints.

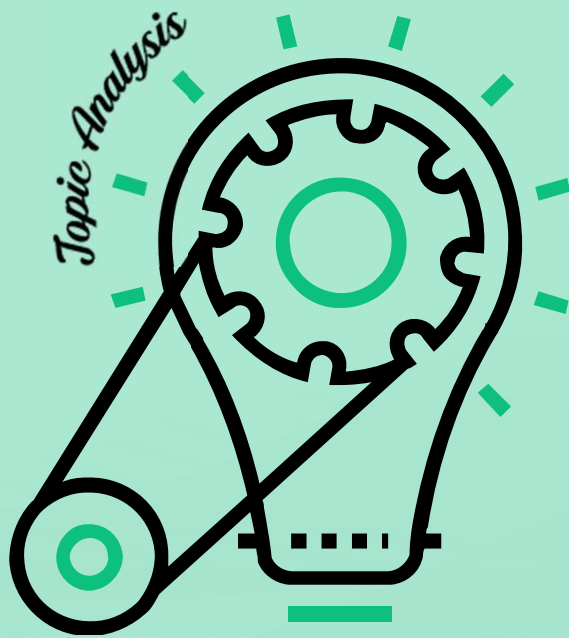
1.20.4 Sprint Retrospective

- During the Sprint retrospective, the team inspects itself and creates a plan for improvements that are to be implemented during the next Sprint.
- The Sprint retrospective occurs after the Sprint review and prior to the next Sprint planning.
- This meeting is time-boxed to three hours for a one-month Sprint and is proportionally reduced for shorter Sprints.

The purpose of the Sprint retrospective is to:

- Inspect as to how the last Sprint went in terms of people, relationships, processes and tools
- Identify the important items that went well and the areas for improvement
- Plan for implementing improvements in the Scrum team

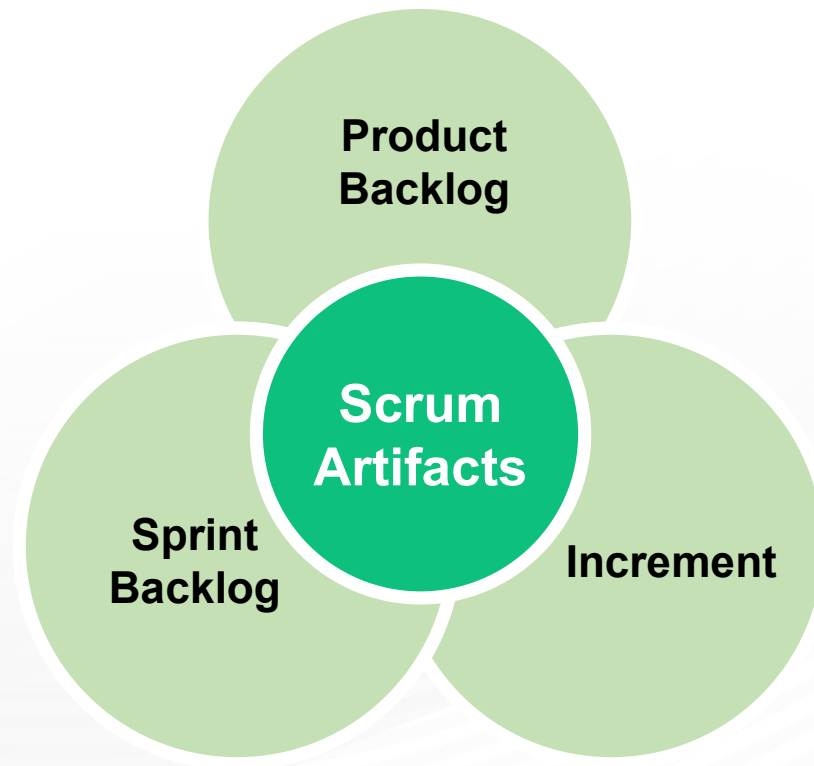
What did You Grasp?



1. What is the maximum time-box for a sprint?
A) 1 month
B) 1 week
C) 1 day
D) 15 days
2. Which of the following events is considered as an internal meeting for the development team?
A) Sprint Planning
B) Daily Scrum
C) Backlog Grooming
D) Sprint Review

1.21 Scrum Artifacts

Scrum defines three artifacts:



1.21.1 Product Backlog

The salient features of the product backlog artifact are as follows:

- Product backlog is a list of the desired functionalities in the order of priority.
- All stakeholders involved have access to the product backlog.
- Items can be added to the product backlog by the development team on discretion of product owner and Scrum Master.
- Product owner continuously reorders the PBIs.
- It undergoes constant refinement by the Scrum team.
- Items at the top of the product backlog should be smaller than the ones at the bottom.
- During product backlog refinement, more details, estimates and the order to the items in the product backlog are added.

1.21.2 Sprint Backlog

Sprint backlog is the set of PBIs selected for the Sprint, along with the plan to realize the Sprint goal.

- Every Sprint backlog consists of PBIs selected by the team and the product owner for the Sprint
- Sprint Backlog also consists of the plan for delivering the Product Increment and realizing the Sprint goal
- In general, Sprint Backlog is a forecast done by the Development Team on what will go into the next Increment and the effort needed to deliver the functionality into a 'Done' Increment
- The Development Team has access to the Sprint backlog
- Sprint backlog is the reference document during the daily Scrum meeting

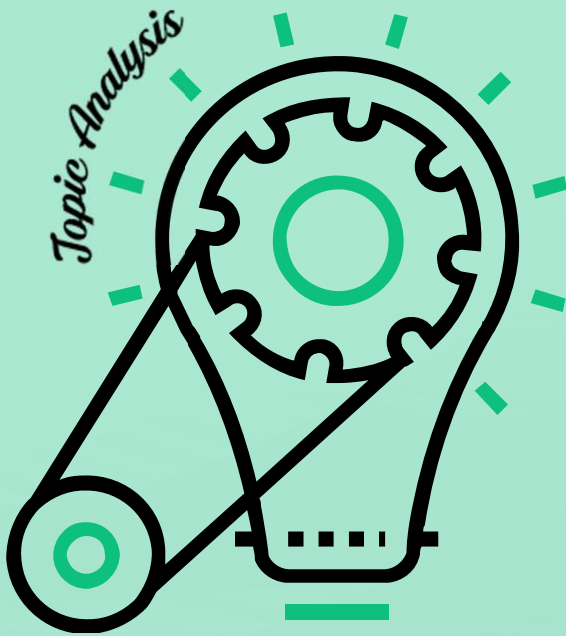
1.21.3 Increment

Increment refers to the sum of all the product backlog items completed during a Sprint and the value of the increments of all the previous Sprints.

The salient features of an increment are as follows:

- An increment consists of the product capabilities completed during the Sprints
- An increment should be a usable, releasable state by the end of each Sprint
- An increment is released at the discretion of the product owner
- During the Sprint review Meeting an increment is inspected

What did You Grasp?



1. Who is the authority to decide the items that go into the product backlog?
A) Scrum Master
B) Product Owner
C) Development Team
D) Customer
2. Which of these artifacts is considered as the reference document during the daily Scrum?
A) Product Backlog
B) Sprint Backlog
C) Increment
D) Sprint document

1.22 Benefits of Scrum

Scrum offers the following benefits over other traditional development methodologies. The key benefits of Scrum are as follows:

1

Scrum ensures quality of the product by means of different practices like defining and elaborating the requirements, incorporating feedback, regular and continuous improvement of the Scrum team and conducting Sprint retrospectives.

2

Scrum drastically reduces the time to market, since it employs an iterative development and each iteration produces a usable version of the product in a short time period.

3

Higher return on investment (ROI) is possible with Scrum because it reduces the time to market.

4

Scrum teams are committed to delivering products and services that satisfy the customers

5

Scrum promotes collaboration and team spirit among the stakeholders.

6

Scrum encourages teams to take ownership of the projects and products so they can produce great results.

7

Scrum uses more relevant metrics to measure the outcome of any project.

8

Scrum ensures visibility on progress and control over the project.

9

Most importantly Scrum mitigates the risk of complete project failure, as products are built over iterations. Any failure can be identified rectified before proceeding to the next iteration.

1.23 Planning and Estimation - An Introduction

- Estimation and planning are not just about determining an appropriate schedule or deadline for any given project.
- Planning for any project should be an iterative and ongoing process to determine what to build and when to deliver it.
- The factors to be considered while planning are the features to be built, available resources and the schedule.
- Planning is a very important step in implementing Agile or any other development method.

Planning is important because it helps in the following:

Reducing risk

Reducing uncertainty

Supporting better decision making

Establishing trust

Conveying information

1.23.1 Agile Planning

According to Mike Cohn, the author of *Agile Estimating and Planning*, “Agile planning balances the effort and investment in planning with the knowledge that we will revise the plan through the course of the project.”

- Plan changes only if there is a need that arises based on the learning acquired through the course of the project.
- Plan changes if the customer requirements change, or change in technology or change in priorities.

The most important aspects of Agile planning include:

It is focused more on the planning than the plan

It encourages change

It results in plans that are easily changed

It's spread throughout the project

1.24 Need for Agile Planning

Agile planning helps in overcoming the deficiencies associated with traditional planning, which include:

- Focusing more on activities than delivered features
- Ignoring the prioritization
- Not considering the existence of uncertainty
- Considering estimations as commitments

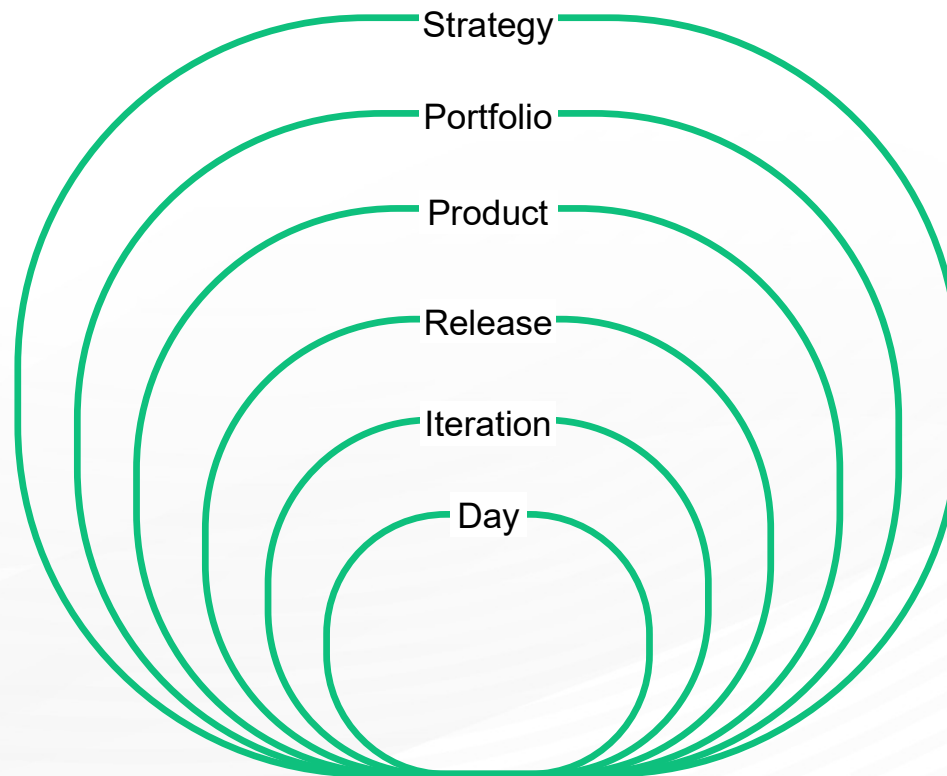
To keep up with the progress of Agile projects and the dynamic environments, the planning has to be Agile as well.

Agile planning factors in the user requirements and the plan is adapted to accommodate the changes, which makes it superior to traditional planning.

Module 1: Agile Practices

1.25 The Agile Planning Onion

The given image illustrates the relationship between the different planning horizons.

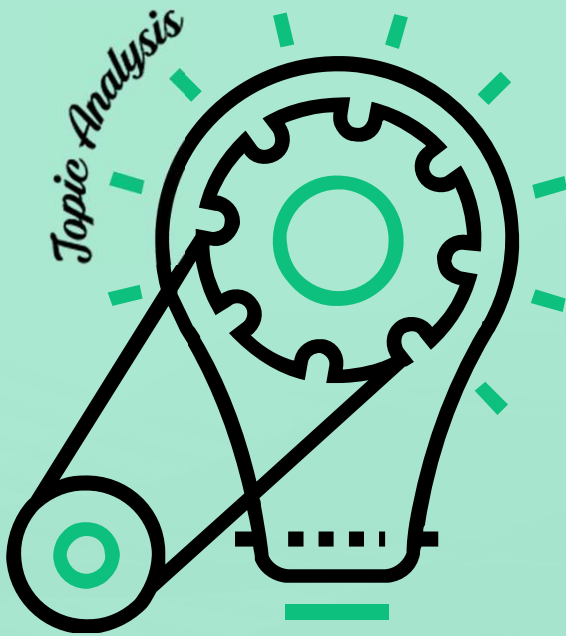


Credit: Mike Cohn

Copyright © 2019, Xebia Group. All rights reserved. This course B.TECH CSE with specialization in DevOps is licensed to UPES.

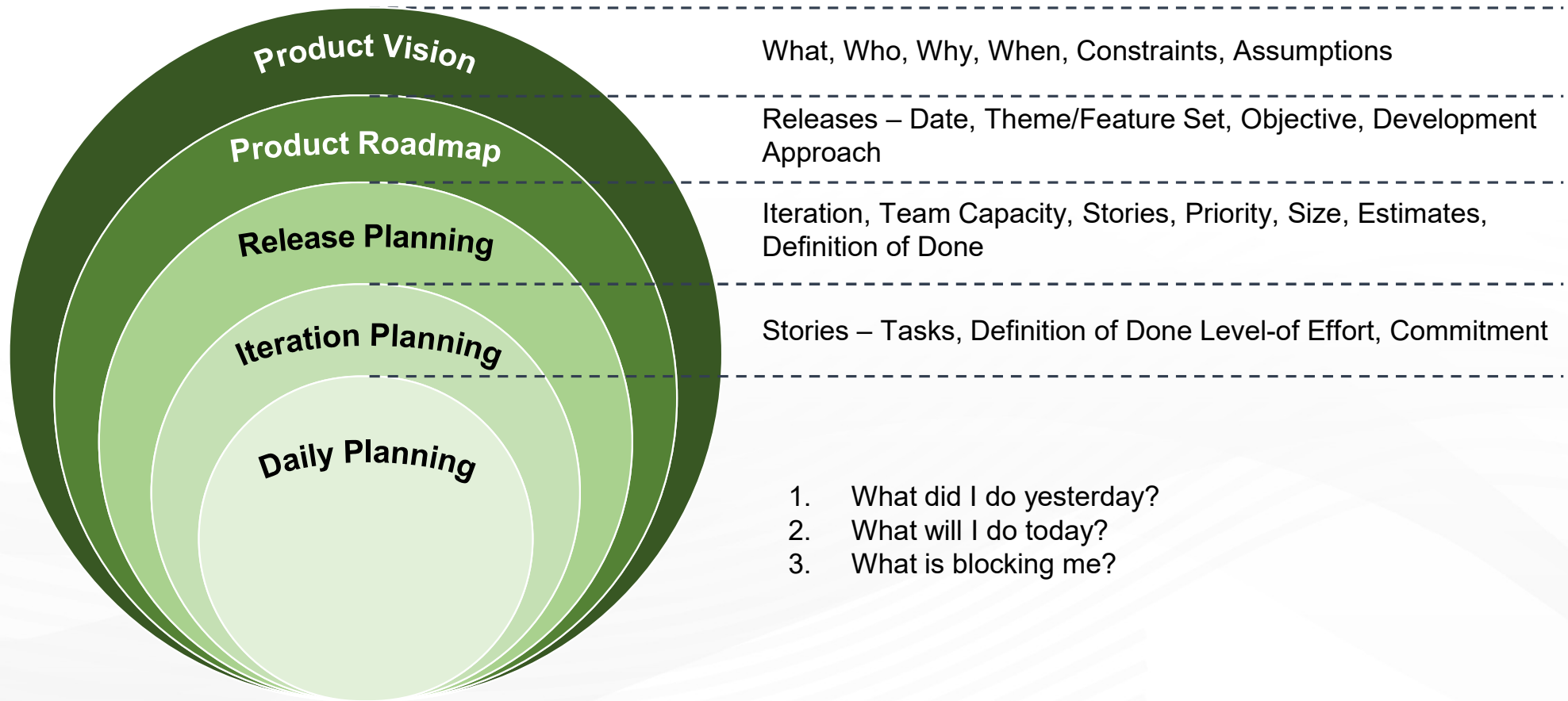
Module 1: Agile Practices

What did You Grasp?



1. State True or False.
In Agile planning, the plans cannot be changed easily.
A) True
B) False
2. Which among the following is the most visible horizon in the Agile planning onion?
A) Day
B) Iteration
C) Release
D) Product

1.26 Levels of Agile Planning



1.26.1 Product Vision

Following are the key details of product vision:

- The outermost level of the planning horizon and it concentrates on the future of the product.
- The macro image of the product, i.e., how it will look at the end of the project, is defined at this level.
- Product vision is set primarily by the product owner, with a little support from the project manager.
- Product vision creation step is to ensure that the strategies are aligned properly and the team spends the time and effort only on creating and delivering the valuable product.
- The vision statement also tells us how the product supports the organization's strategies. The picture shows a product vision template.

THE PRODUCT VISION BOARD

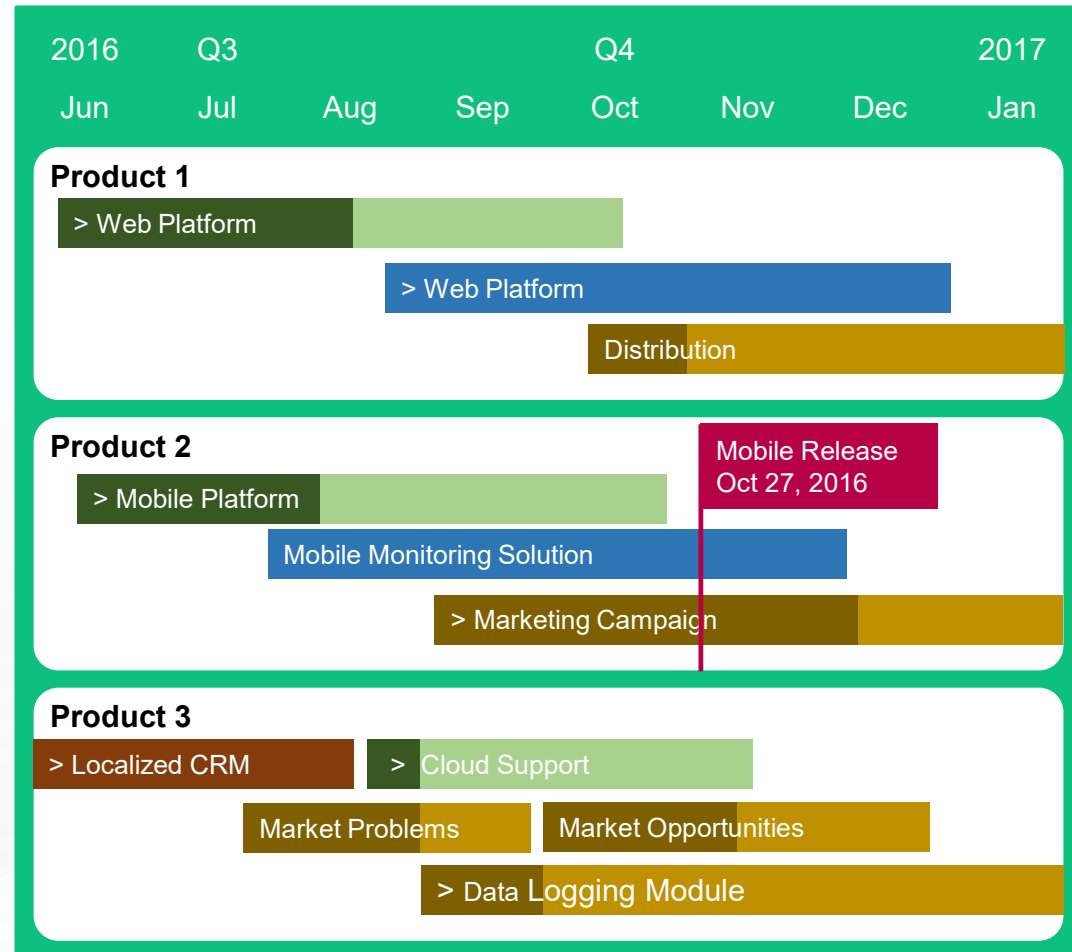


Module 1: Agile Practices

1.26.2 Product Roadmap

Following are the key details of product roadmap:

- Roadmap is the plan that describes the way that the product is likely to grow.
- Agile teams focus on a goal-oriented roadmap, as it is important for the team to know about the everyday work.
- Product roadmap also helps in estimating the budget for developing and delivering high value product as per the promised schedule.
- Agile environment is prone to changes because of the changing requirements from the customers, hence creating a roadmap is often a challenging task.
- Product roadmap should focus on goals, benefits, objectives, acquiring customers and increasing their engagement. The picture shows a sample product roadmap.



1.26.3 Release Planning

Following are the key details of release planning:

- Release is a set of increments that are released to the customer at frequent intervals.
- The release plan helps to estimate the amount of work that will be delivered by the team at the scheduled deadline.
- Release planning is a collaborative effort of the Scrum Master and the product owner, Agile team and the stakeholders.
- Release planning is focused on goals and benefits, by taking into account, the dependencies and uncertainties.

1.26.4 Iteration Planning

Following are the key details of iteration planning:

- Iteration planning is generally a subset of the release plan stories that will be done in the next iteration or a sprint.
- During iteration planning, the volume of the backlog items that a team can deliver during the next iteration, are determined.
- The team commits to deliver the features according to their velocity and the schedule.
- Iteration plan is about breaking down the features into multiple development tasks and estimating them.
- Use cases are also created to identify the stories that will be developed and to break them into specific tasks and acceptance criteria.

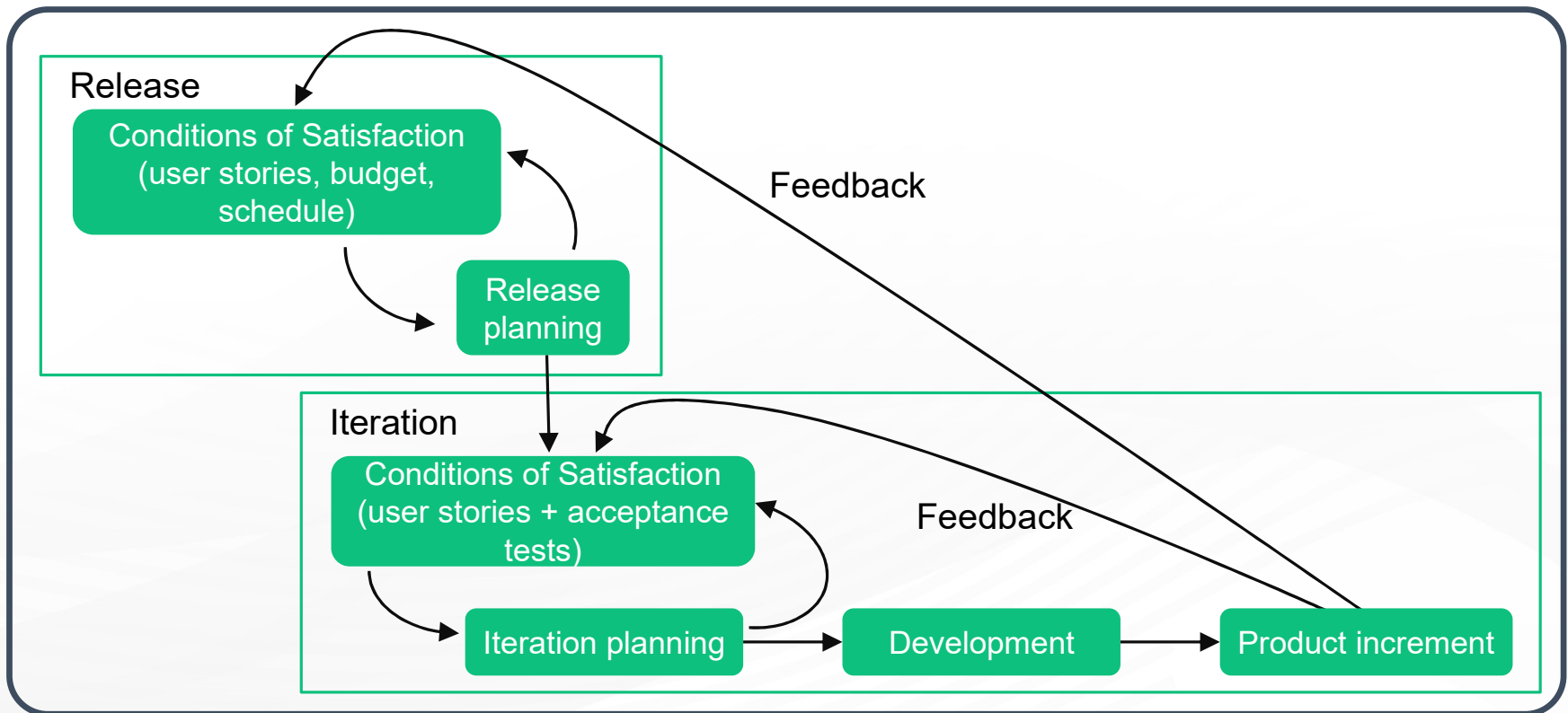
1.26.5 Daily Scrum/Standup

Following are the key details of daily scrum/standup:

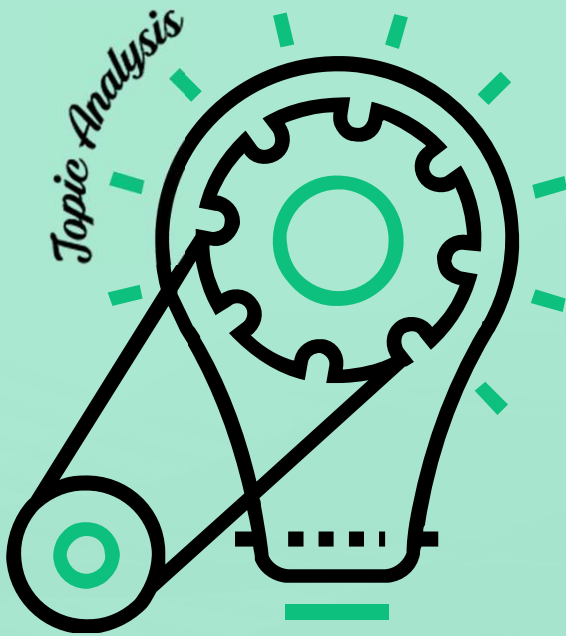
- Daily scrum is done to coordinate the work and synchronize the effort.
- Teams make, assess and revise their plans during this meeting.
- Daily scrum is focused on completing the top priority features. This meeting is focused on the individual's accomplishments, plan for the day and the issues faced by them and ways to resolve them.
- For the unresolved issues, 'Parking Lot' can be used. Anything out of the scope of the daily planning will be placed in the parking lot and dealt with later.

1.27 Conditions of Satisfaction

The picture shows how the conditions of satisfaction drive both release and iteration planning.



What did You Grasp?

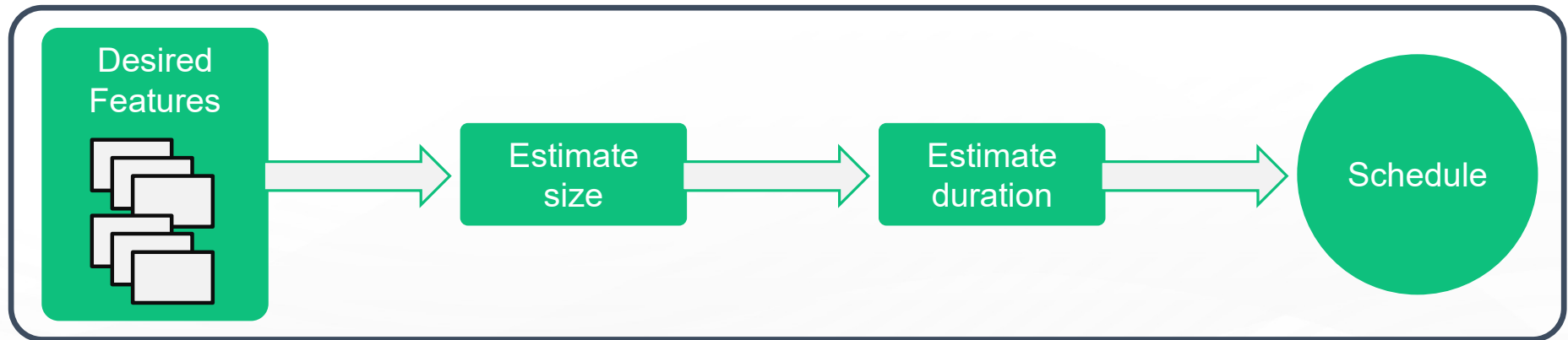


1. Who is responsible for devising the product roadmap?
A) Developer
B) Product Owner
C) Scrum Master
D) Customer
2. State True or False
Conditions of satisfaction are checked only at the iteration level.
A) True
B) False

1.28 Estimating the Size - Story Points

Estimate is defined as ‘the quantified evaluation of the effort necessary to carry out a given development task.’ Estimation is done in terms of size or duration.

→ Estimating the size helps in the estimation of duration, which is shown in the figure given below.



- Story points are a unit of measure for expressing the overall size of a user story, feature, or any other piece of work. A point value is assigned to each item while estimating with story points.
- Story points are relative and not absolute. A story point with a value two should be twice as much the story with with the value one.
- The number of story points associated with a story represents the overall size of the story.

1.29 Velocity

Following are the key details of velocity:

- Velocity is defined as a measure of a team's rate of progress, which is important to understand the story points.
- Velocity is by adding the number of story points assigned to each user story that the team completed during the iteration.
- For example, if a team completes three stories that are valued at three story points each, the velocity of the team would be nine.
- Based on the velocity of the team during the previous iteration, the estimate for the next iteration can be estimated.
- If the story point estimates of all the features are summed up, the total size estimate for the project can be done.

1.30 Estimating in Ideal Days

In Agile development, user stories or other tasks are estimated in ideal days.

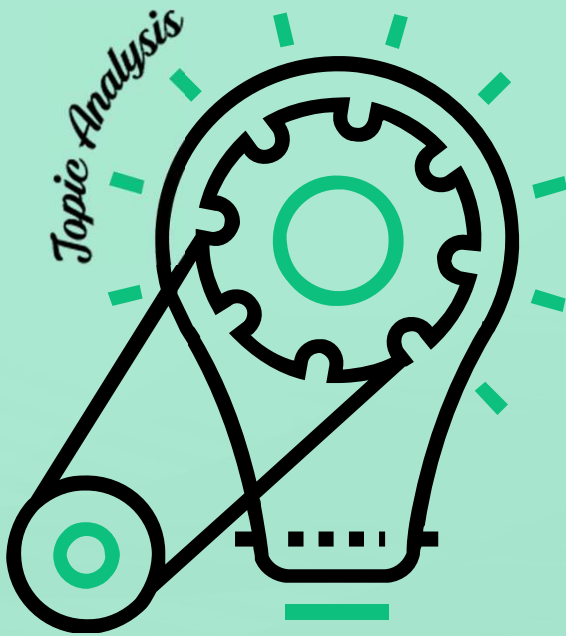
The assumptions behind ideal days estimation are as follows:

- The story being estimated is the only thing that the developer works on
- Everything the developer needs will be on hand when they start
- There will be no interruptions

While estimating the ideal days required to develop, test and accept a user story, it is not necessary to take into account, the overhead of the environment in which the team works.

If this organizational overhead is ignored, ideal days can be thought of as an estimate similar to story points.

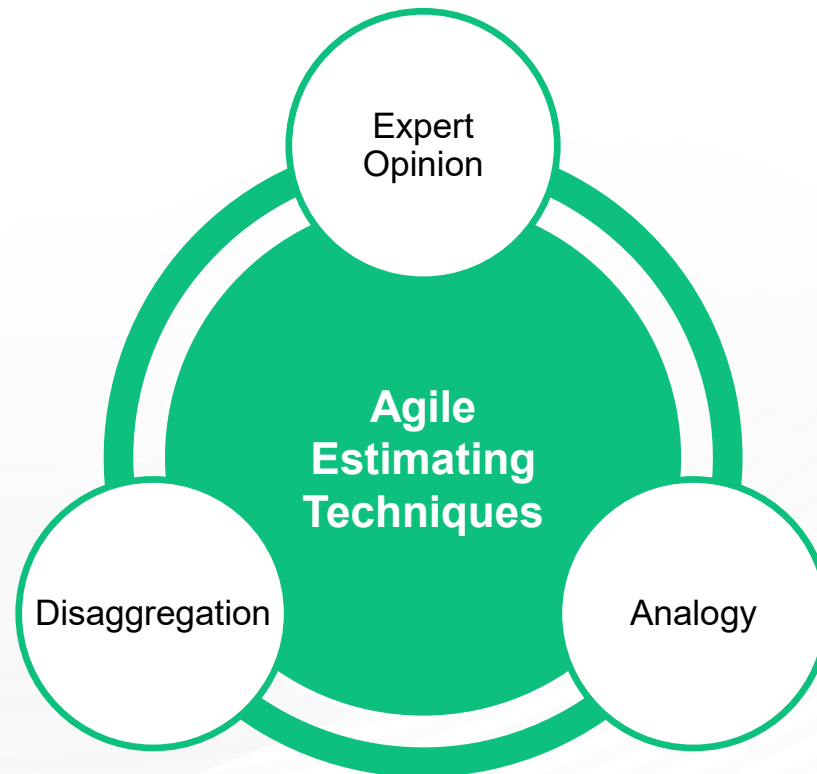
What did You Grasp?



1. What is the velocity of the team that completes five user stories that have 4 story points each?
A) 10
B) 15
C) 20
D) 30
2. State True or False.
While estimating in ideal days, it is assumed that the developer works only on the story being estimated.
A) True
B) False

1.31 Estimating Techniques

The three most common techniques for estimating are represented in the figure below:

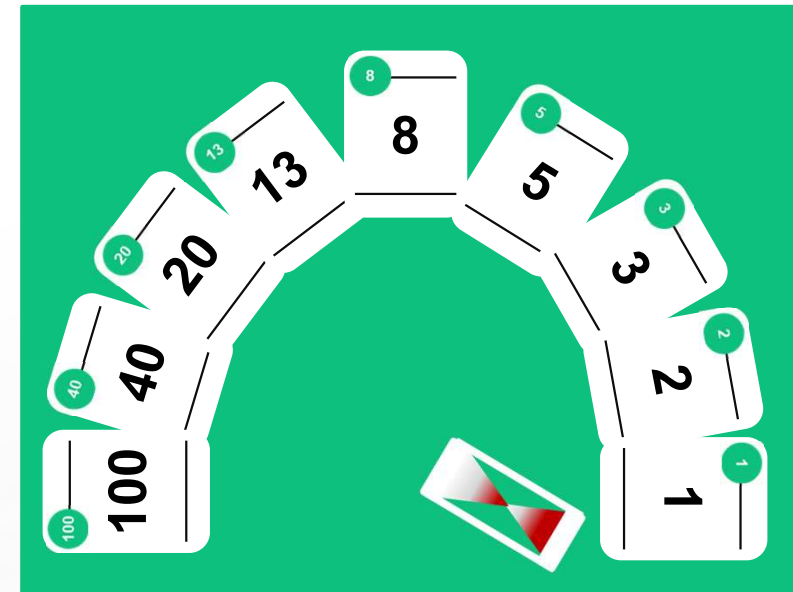


Module 1: Agile Practices

1.32 Planning Poker

Following are the key details of the Planning Poker game:

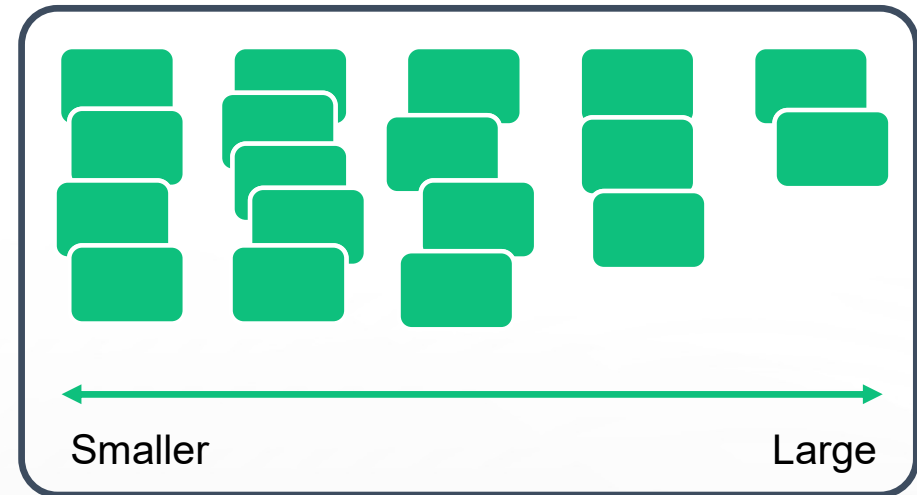
- Planning poker is one of the most popular gross level estimating technique, which uses Fibonacci sequence (adding previous two numbers to get the next number in the sequence) to assign point value to a story or a feature.
- Fibonacci sequence has been modified to get the following sequence for Agile estimation purposes: 0, 1, 2, 3, 5, 8, 13, 20, 40, 100.
- The numbers in the given sequence are represented in a set of playing cards as shown in the picture.
- Team members play the Planning Poker game and assign a point value to each item. The process is explained below in the notes.
- Fibonacci sequence is used as these numbers to represent relative size and this method also provide the correct level of detail for smaller and better understood projects.



1.33 Affinity Grouping

Following are the key details of affinity grouping:

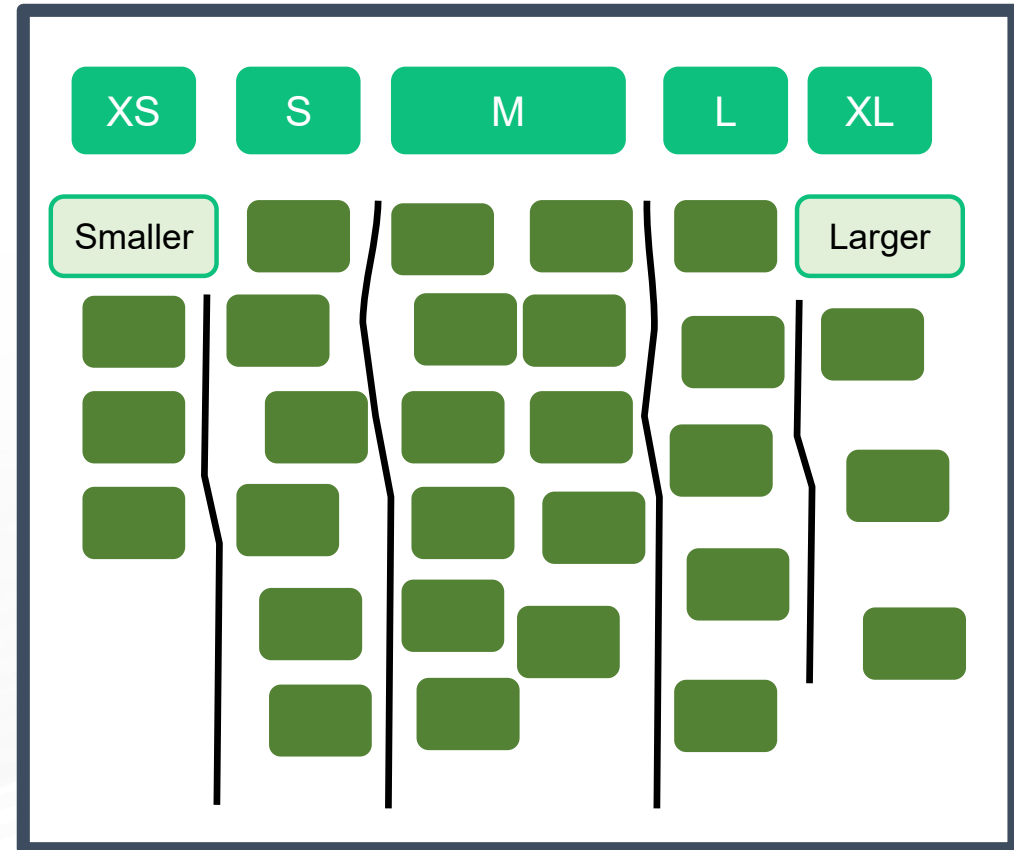
- Affinity grouping is a faster way to Agile estimating, especially when the number of items to estimate is very large.
- Stories or items that are like-sized are grouped together that results in affinity groups as shown in the picture.
- The first item is read out by the presenter to the team members and placed on the wall. The second item is then read out and compared with the size of the first item. If it is of smaller size, it goes to the left of the first one and if larger, goes to the right.
- All the items are thus read out, assessed and placed on the wall. The team then reviews the grouping and any inappropriate placement is modified.
- Once the affinity grouping is complete, point values are then assigned.



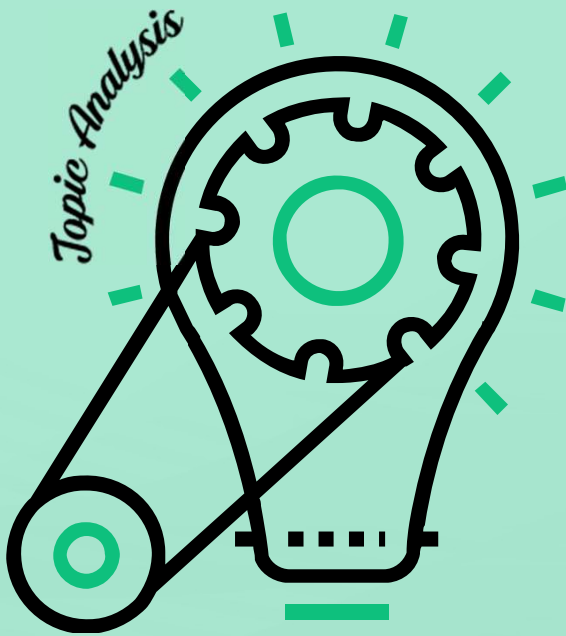
1.34 T-Shirt Sizes

The salient features of T-shirt sizes are as follows:

- T-Shirt sizes is a way of estimating relative sizes of features. Five different sizes Extra Small (XS), Small (S), Medium (M), Large (L), Extra Large (XL) are used for estimating the size of features.
- Using T-shirt sizes is an informal and fast way of assessing the size of backlog items.
- The picture shows how affinity grouping is done using T-shirt sizes.
- After mutual discussion and agreement of the team, a relative size is first decided, which will be medium (M) most often.
- Numbers are then assigned to the items, according to the relative size that is assigned to the medium size.



What did You Grasp?



1. What is the ideal number of points can be completed in a single iteration?
A) 0-13
B) 15-18
C) 20-25
D) 25-30
2. State True or False.
Affinity grouping is suitable only for projects with minimum number of items.
A) True
B) False

1.35 Agile Implementation in Industry Projects

Implementation of Agile across organizations can be done by several approaches. Some of these for Developers and Management include:

Developers

- Resistance
- Micromanagement
- Transition from traditional to Agile processes
- Distributed development
- Top talent is required
- Enthusiastic teams
- Testers

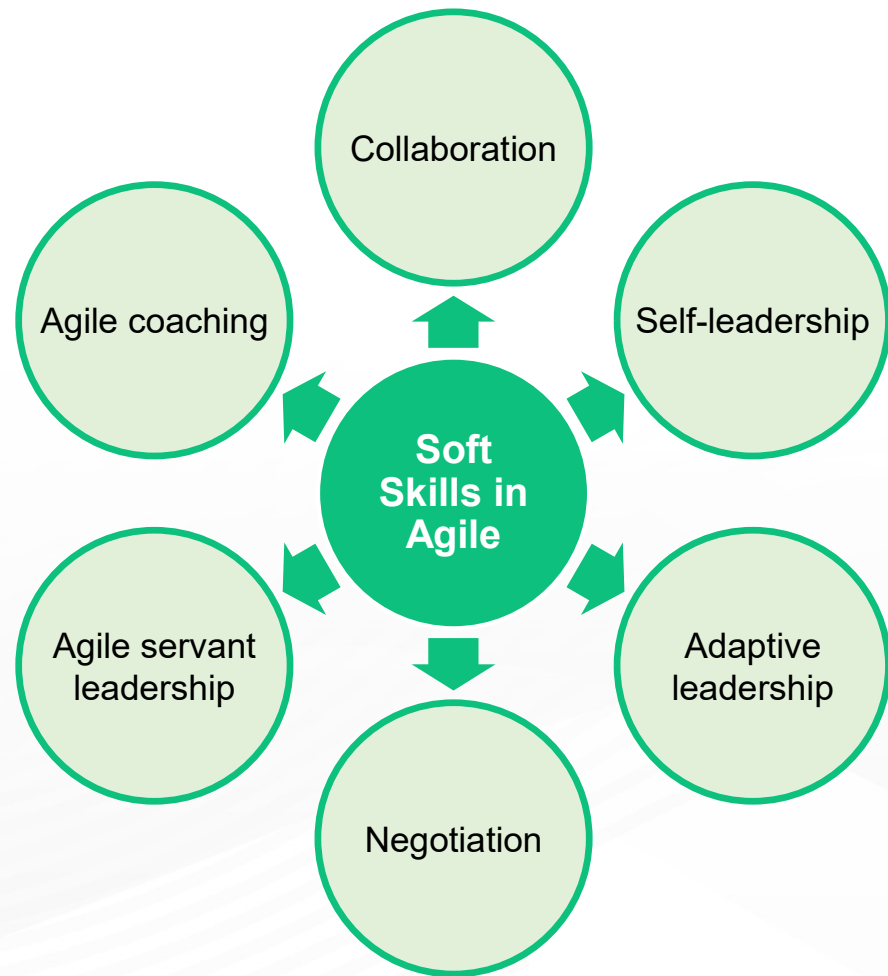
Upper management

- Customer Commitments
- Tracking Progress
- Impact on other groups
- Project completion

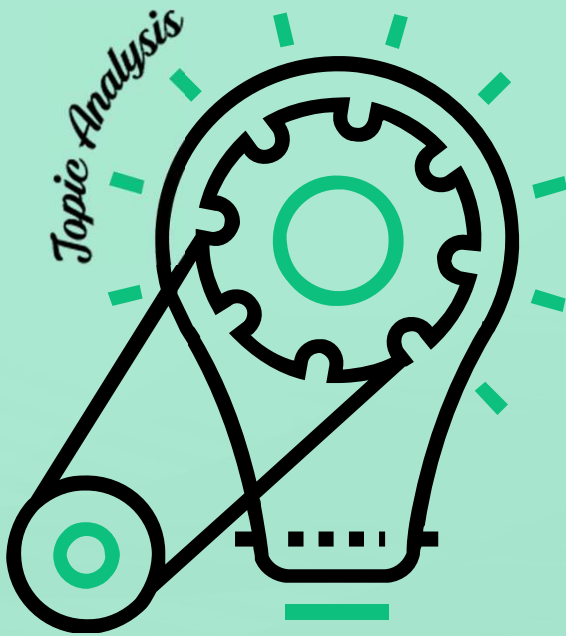
1.36 Soft Skills in Agile

Soft skills refer to the interpersonal skills and the ability to interact effectively and harmoniously with other people.

Agile teams and the project leaders need to have soft skills, a few of them are as follows:



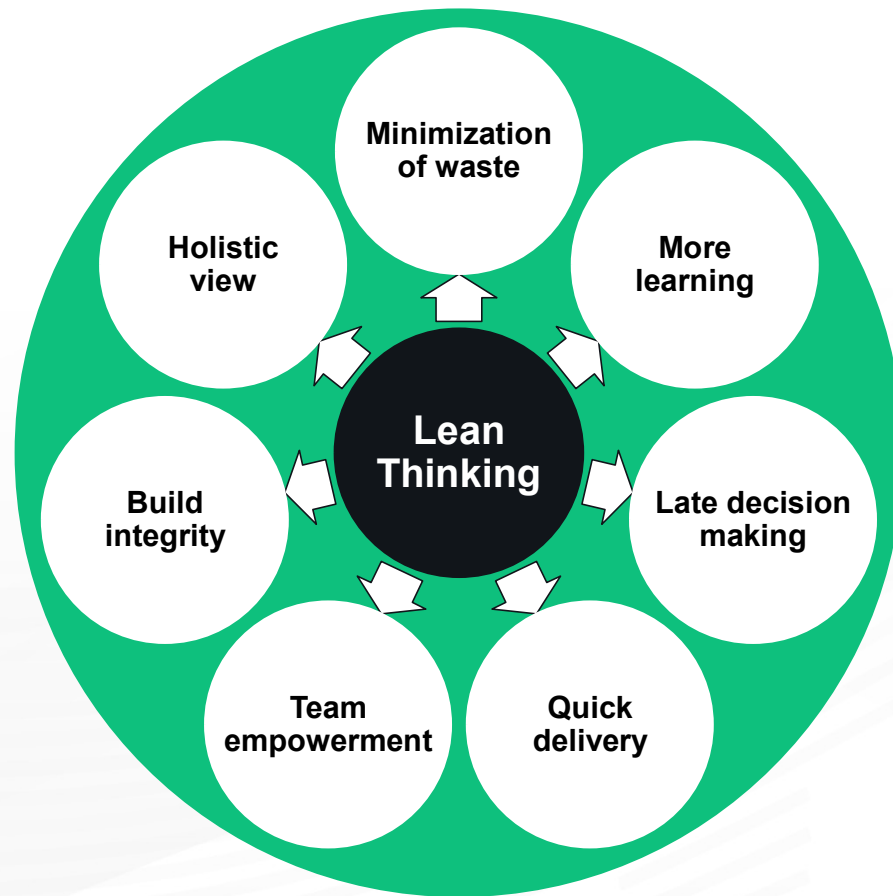
What did You Grasp?



1. Which of the following could not be a soft skill of an Agile leader?
A) Dictatorship
B) Self-leadership
C) Coaching
D) Negotiation

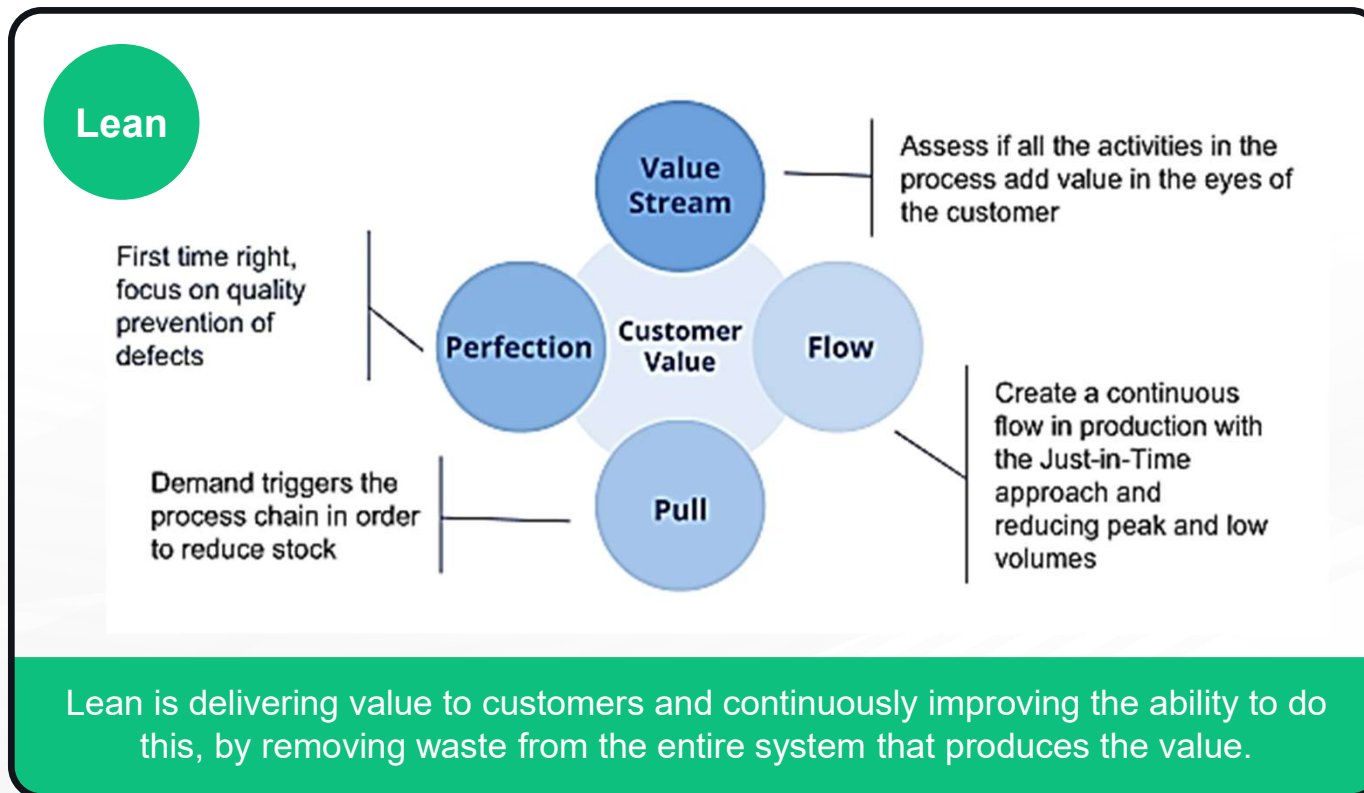
1.37 Lean Thinking

Lean methodology derives itself from Lean Principles in Manufacturing where more is achieved from less. Lean thinking is governed by core principles such as:

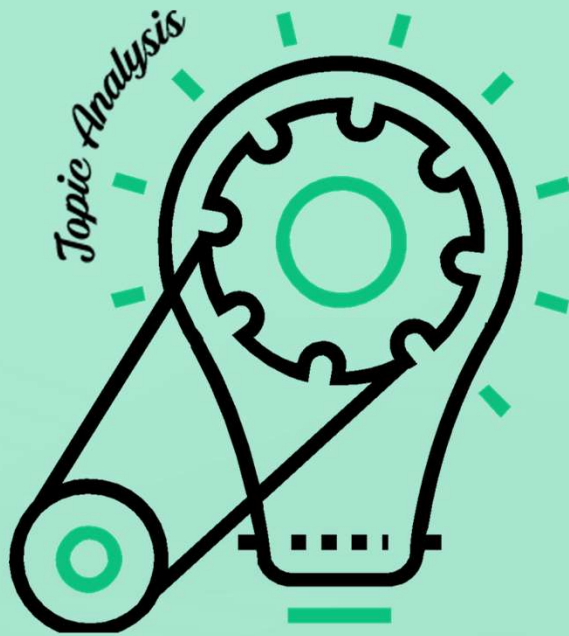


1.38 Lean Methodology: Customer is King

Lean places supreme importance on creating value for the customer.



What did You Grasp?



1. Which of the following options is not a LEAN principle?

- A) Elimination of fees
- B) Slow delivery of features
- C) Late decision making
- D) Holistic approach

Module 1: Agile Practices

In a nutshell, we learnt:



1. Agile methodology
2. Software, History of Software Engineering and Software Development Methodologies
3. Traditional Software Development Models
4. Waterfall Model, Classical Waterfall Model
5. Traditional IT Organizations
6. Developers vs IT Operations Conflict
7. Birth of Agile, Four Values of the Agile Manifesto
8. Scrum, Scrum Theory, Scrum Values, Scrum Roles, Scrum Master
Scrum Sprints, Benefits of Scrum
9. Planning and Estimation, Agile Planning, Levels of Agile Planning

Module 1: Agile Practices

In a nutshell, we learnt (Contd.):



- 10. Conditions of Satisfaction, Velocity
- 11. Estimating Techniques
- 12. Soft Skills in Agile