Chapter-4

Agile Teams, Size and Schedule

Agile Teams

An Agile team's main idea is to have a group of people with a common goal who are flexible in the way they work and more adaptable to changing customer requirements. One thing that distinguishes them from traditional teams is that they are self-directed and self-organized individuals who practice shared leadership.

Another common characteristic of Agile teams is that they are cross-functional teams. It places a focus on generalizing specialists who can contribute to several domains instead of just their own. The idea is to cross-skill people on a single team with the purpose of eliminating hand-offs and dependencies.

The Disciplined Agile Delivery (DAD)* is a people-first, learning-oriented agile approach in delivering software solutions. It aims to have teams that deliver consumable solutions, not just working software. Similar to Scrum, DAD provides a more cohesive approach, taking into consideration some aspects of software development that are deliberately omitted in Scrum. These include architecture, testing, documentation, stakeholders, etc.

According to the DAD toolkit, there are primary and secondary roles for agile solution delivery. every DAD team has five primary roles regardless of the team size and the scope of tasks it handles. Secondary (supporting) roles such as specialist, technical expert or independent tester, are there to address scaling issues, and they are oftentimes temporary.

TEAM LEAD

The team lead ensures that team members are focused on solution delivery and that iteration goals are achieved.

TEAM MEMBER

A DAD team member is not necessarily a developer writing code. It can be any person on the team doing any kind of work on the project, such as estimations, analysis, architecture, planning, testing.

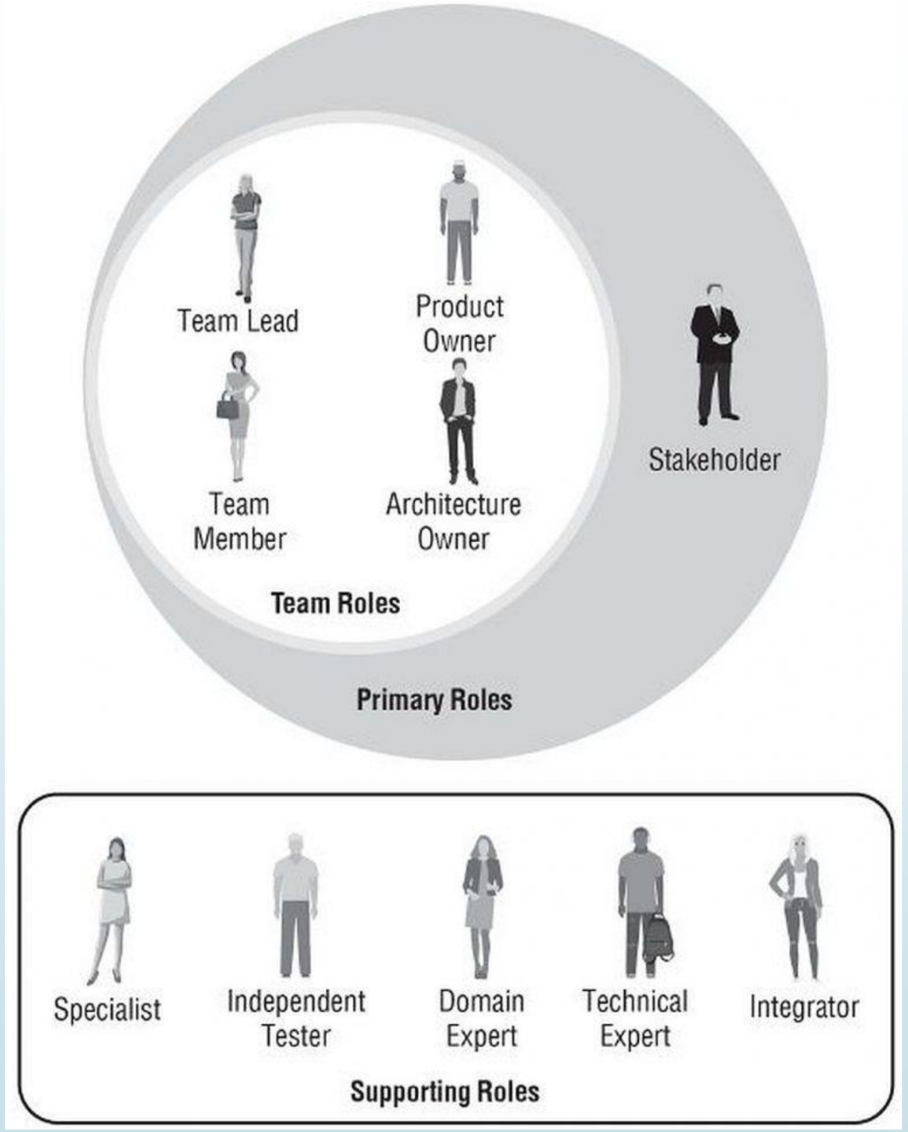
PRODUCT OWNER
Product owner is the link between the team and the stakeholders, both internal and external ones. "The voice of the customer"

ARCHITECTURE OWNER

This role is responsible for the architecture decisions of the team and facilitates the creation and development of the solution design, using a method like design thinking. Architecture owner ensures that software solutions address both functional and quality requirements.

STAKEHOLDER

The stakeholder is someone impacted by the outcome of the solution. Oftentimes, people think of stakeholders as being exclusively CEOs, gold owners who fund the project.

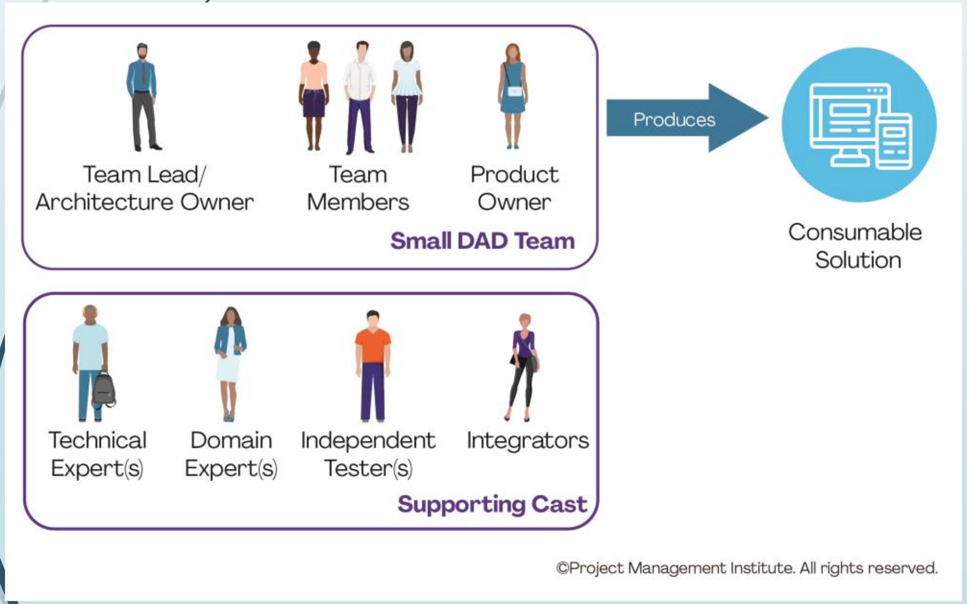


Agile Team Size

There is no exact definition of what it means for a team to be small, medium, and large.

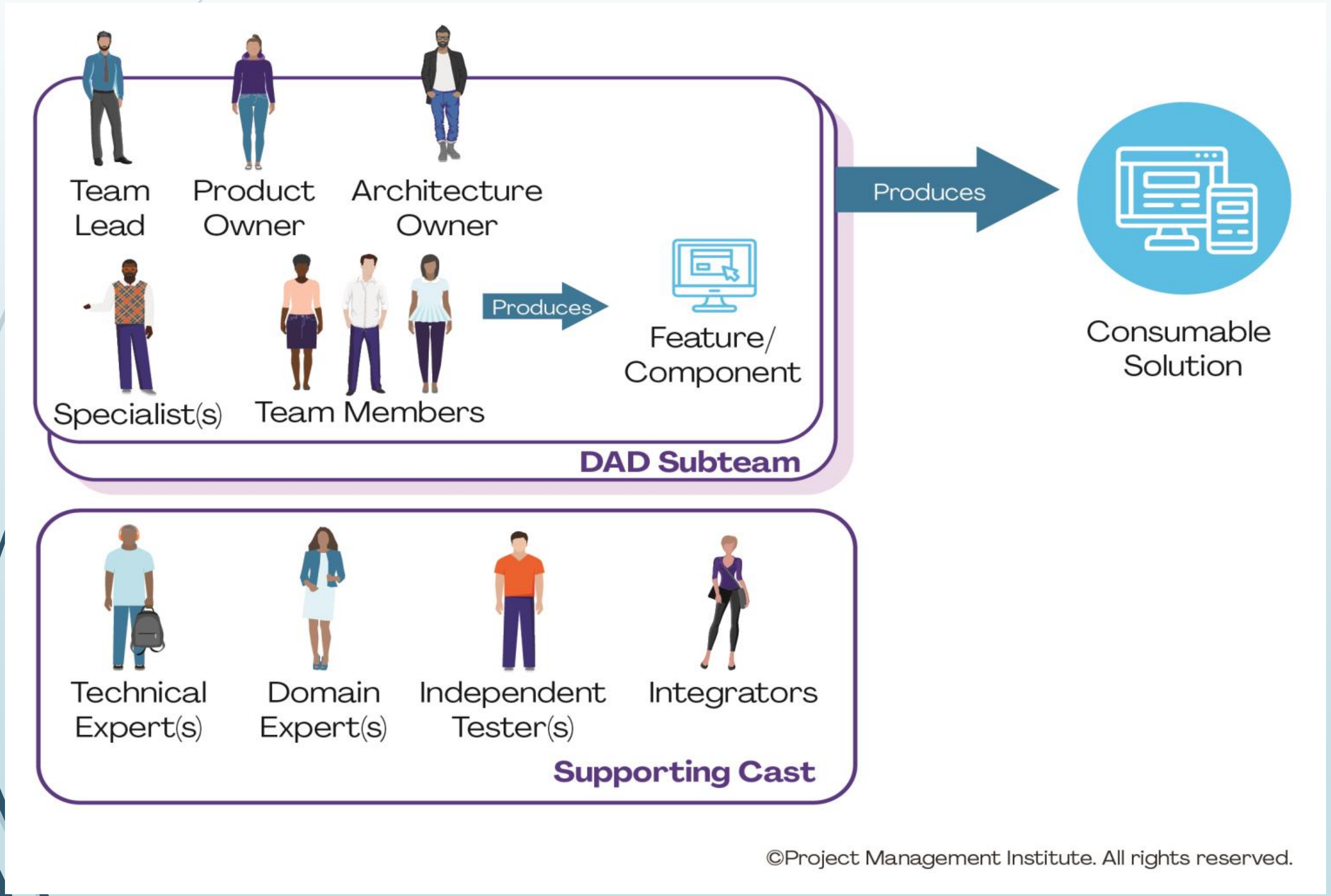
1. small teams are between two and fifteen people,
2. medium-sized teams twelve to fifty people,
3. large teams thirty-five people or above.

Small teams are between two and fifteen people, medium-sized teams twelve to fifty people, and large teams thirty-five people or above.

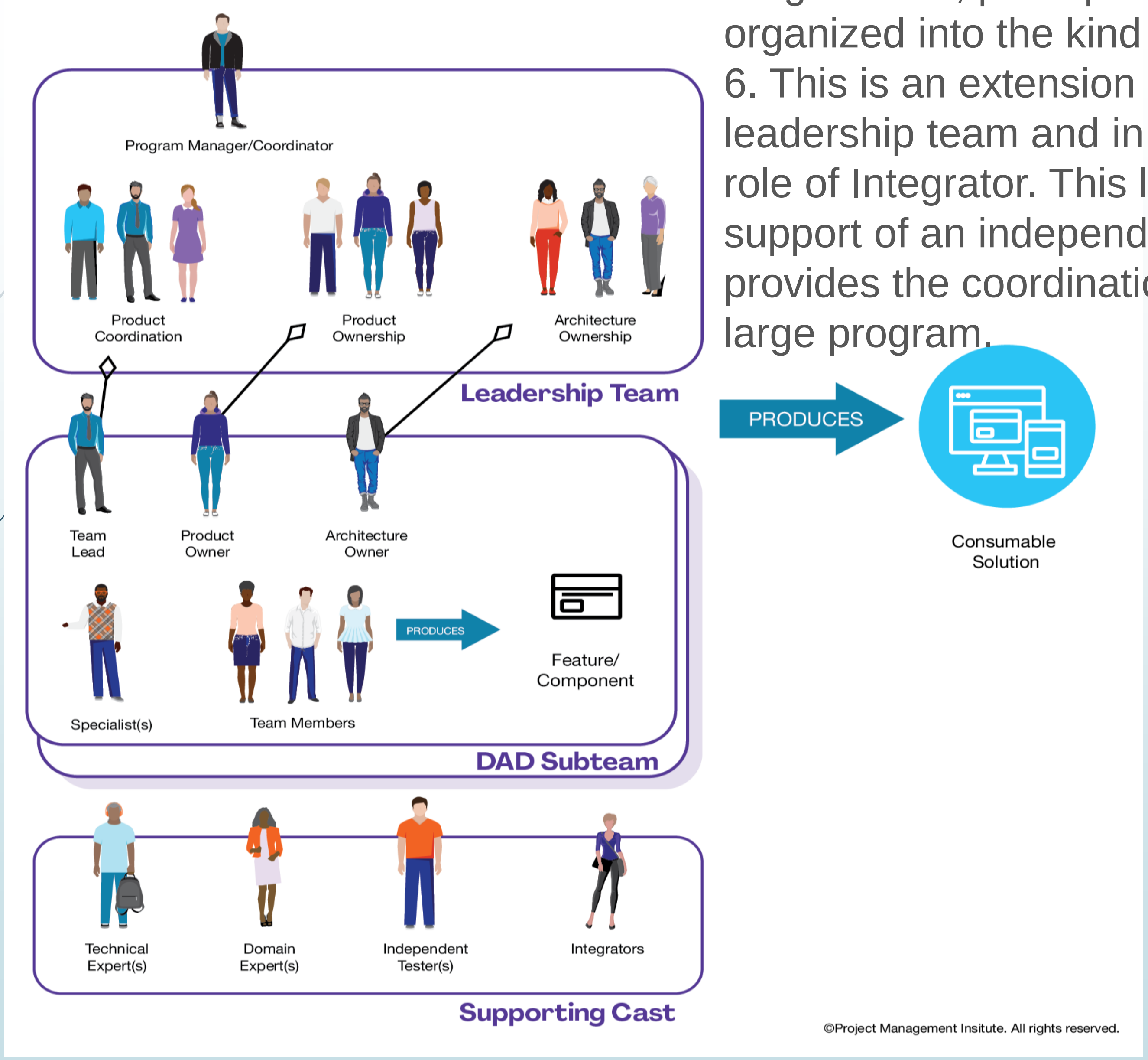


Presents a more common and realistic structure. No team is an island, not even an agile team. Fig shows that agile teams are often helped by what we call a “supporting cast”.

Each subteam typically coordinates their own efforts via a daily coordination meeting, often called a daily standup or a daily Scrum meeting. The subteams will coordinate with each other by sending one person to a second daily stand up called a "Scrum of Scrums" (SoS). The purpose of the SoS is for the representatives of each sub-team to coordinate how any issues that have arisen (perhaps one sub-team has a dependency on another, perhaps one sub-team needs help from another, and so on) within the overall team. We've found that SoS starts to fall apart after four of five sub-teams, so a more sophisticated large-team coordination strategy is needed.

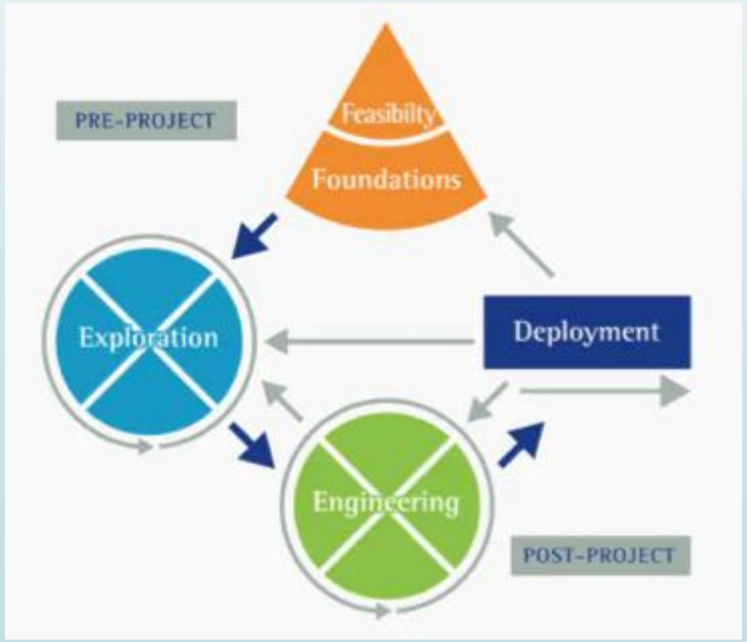


Large teams, perhaps 35 or more people, are often organized into the kind of structure as you see in Figure 6. This is an extension of Figure 5 to explicitly include a leadership team and in many cases someone(s) in the role of Integrator. This leadership team, with the potential support of an independent testing and integration team, provides the coordination structure required to support a large program.



Dynamic System Development Method

DSDM is an **agile project framework** based on an **iterative development methodology**. It was developed in the 90s after the rise of the **Rapid Application Development (RAD)** approach, which focused on adapting **traditional project management methodologies** such as the waterfall methodology to software and system development.



DSDM Framework

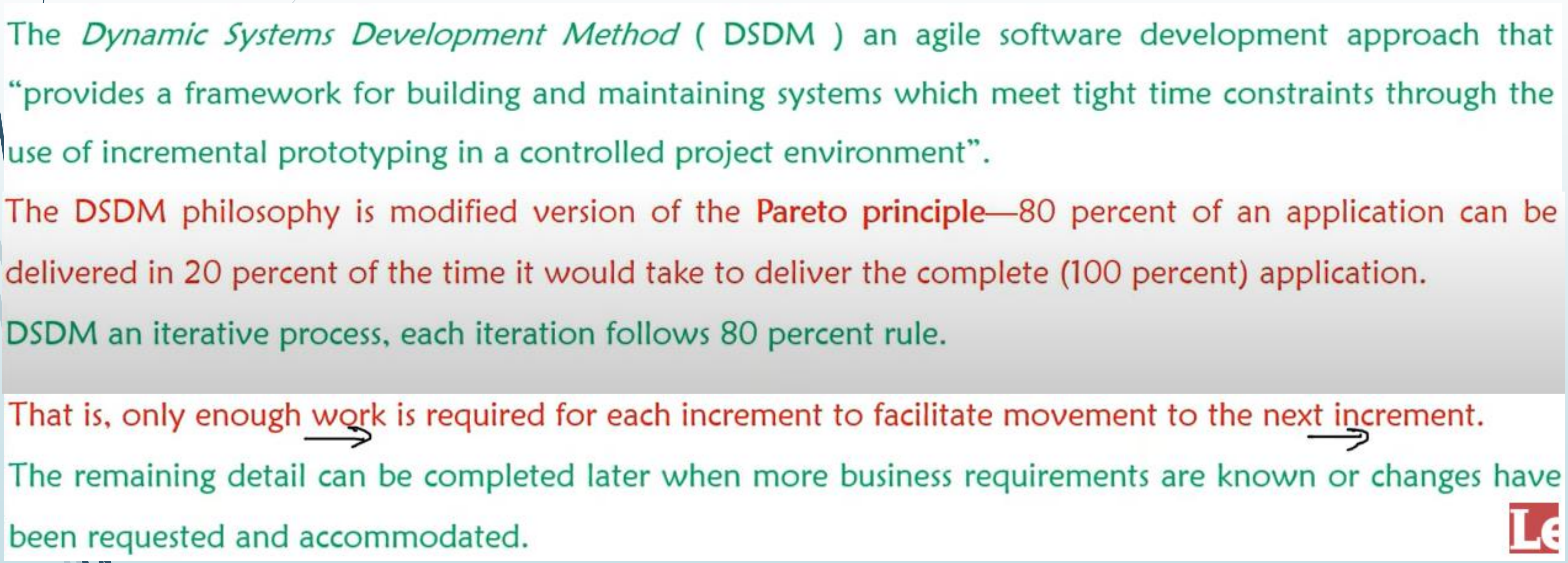
DSDM framework is categorized into three phases:

Pre-project phase--we pick out the projects to work on, register the budget or funds required (and available), and establish project commitment.

Project life cycle--Project lifecycle covers the stages of feasibility and market research, functional model and prototype iteration, design and structure iteration, and execution.

Post project phase--The objective of this phase is to ensure that the product is working efficiently and as per user and company's requirements. This stage requires maintenance, rectifications, and performance enhancement. Because of the iterative nature of the project lifecycle, one cycle is not enough to complete the product development. It undergoes retesting for the previous phases to refine the product.

Dynamic System Development Method



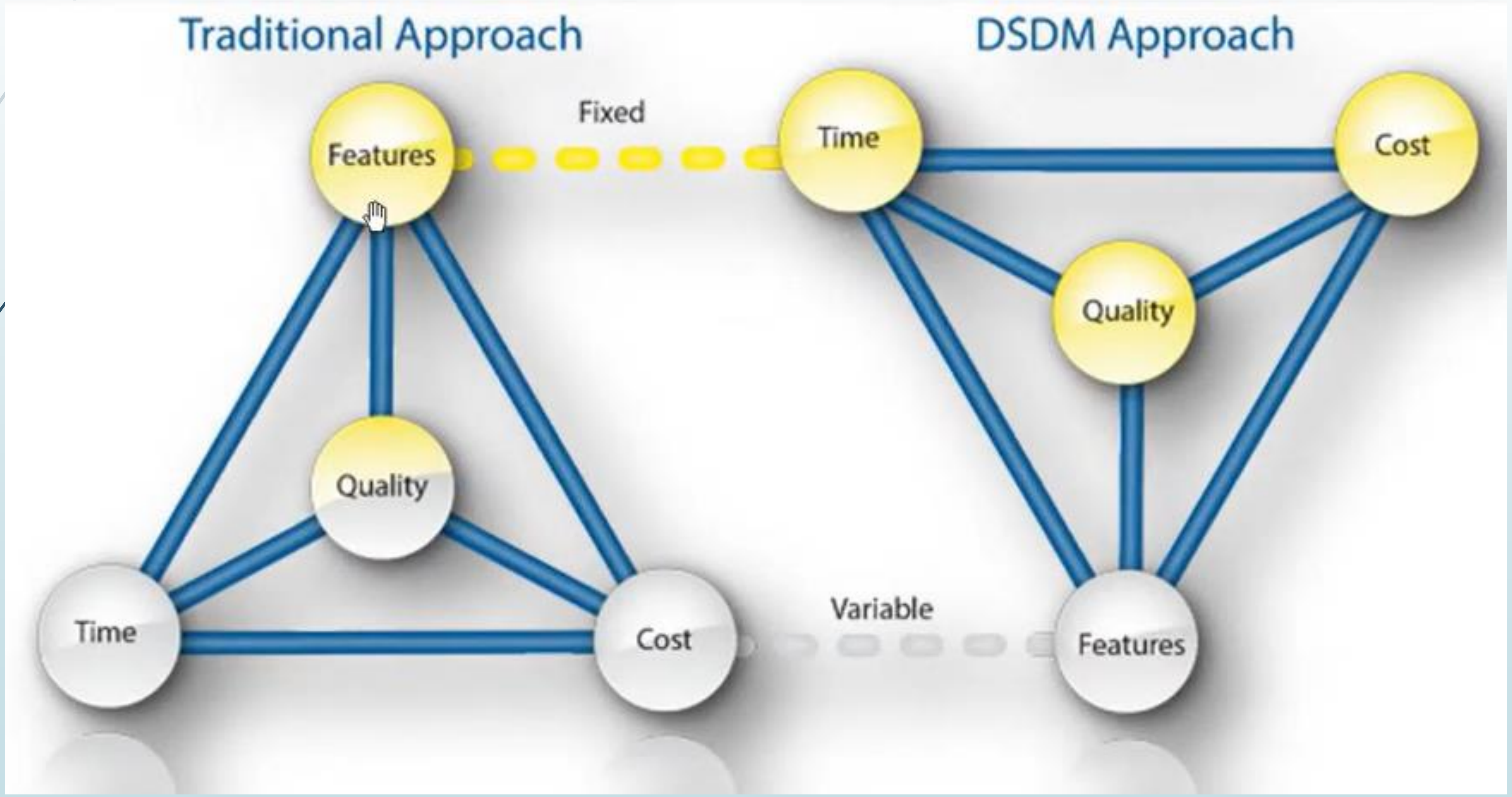
The *Dynamic Systems Development Method* (DSDM) an agile software development approach that "provides a framework for building and maintaining systems which meet tight time constraints through the use of incremental prototyping in a controlled project environment".

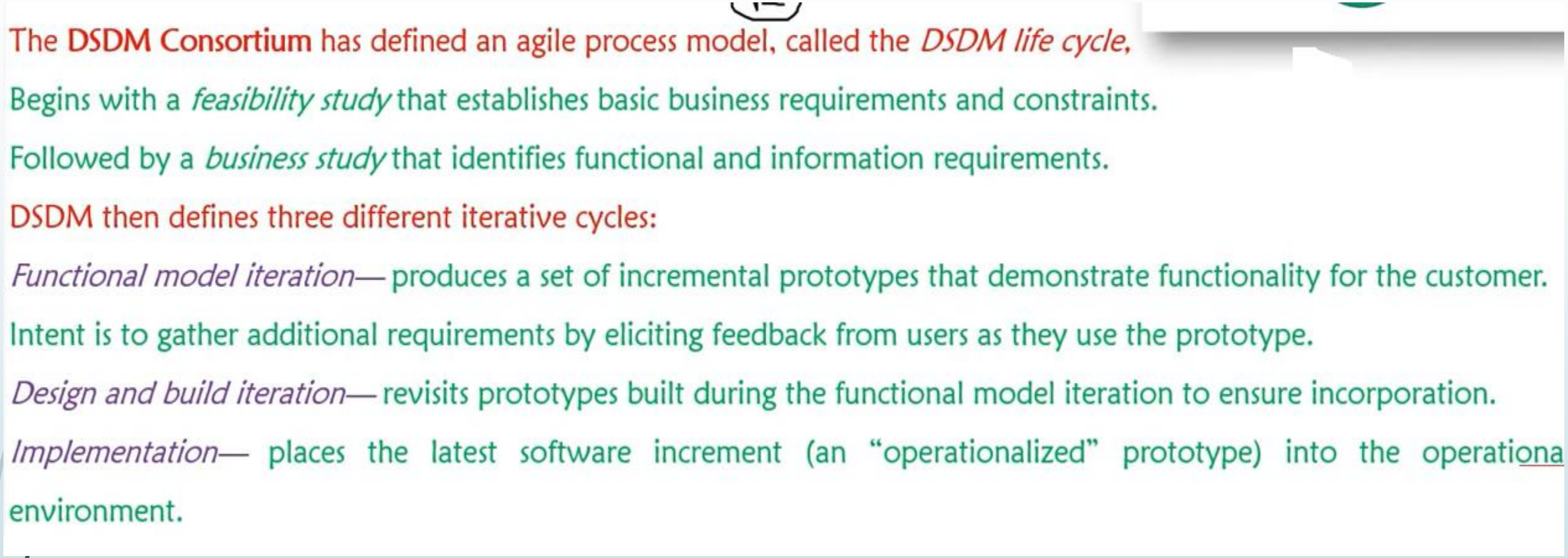
The DSDM philosophy is modified version of the **Pareto principle**—80 percent of an application can be delivered in 20 percent of the time it would take to deliver the complete (100 percent) application.

DSDM an iterative process, each iteration follows 80 percent rule.

That is, only enough work is required for each increment to facilitate movement to the next increment.

The remaining detail can be completed later when more business requirements are known or changes have been requested and accommodated.


Traditional Approach
DSDM Approach
Fixed
Features
Time
Cost
Quality
Time
Cost
Quality
Variable
Features



The **DSDM Consortium** has defined an agile process model, called the *DSDM life cycle*,

Begins with a *feasibility study* that establishes basic business requirements and constraints.

Followed by a *business study* that identifies functional and information requirements.

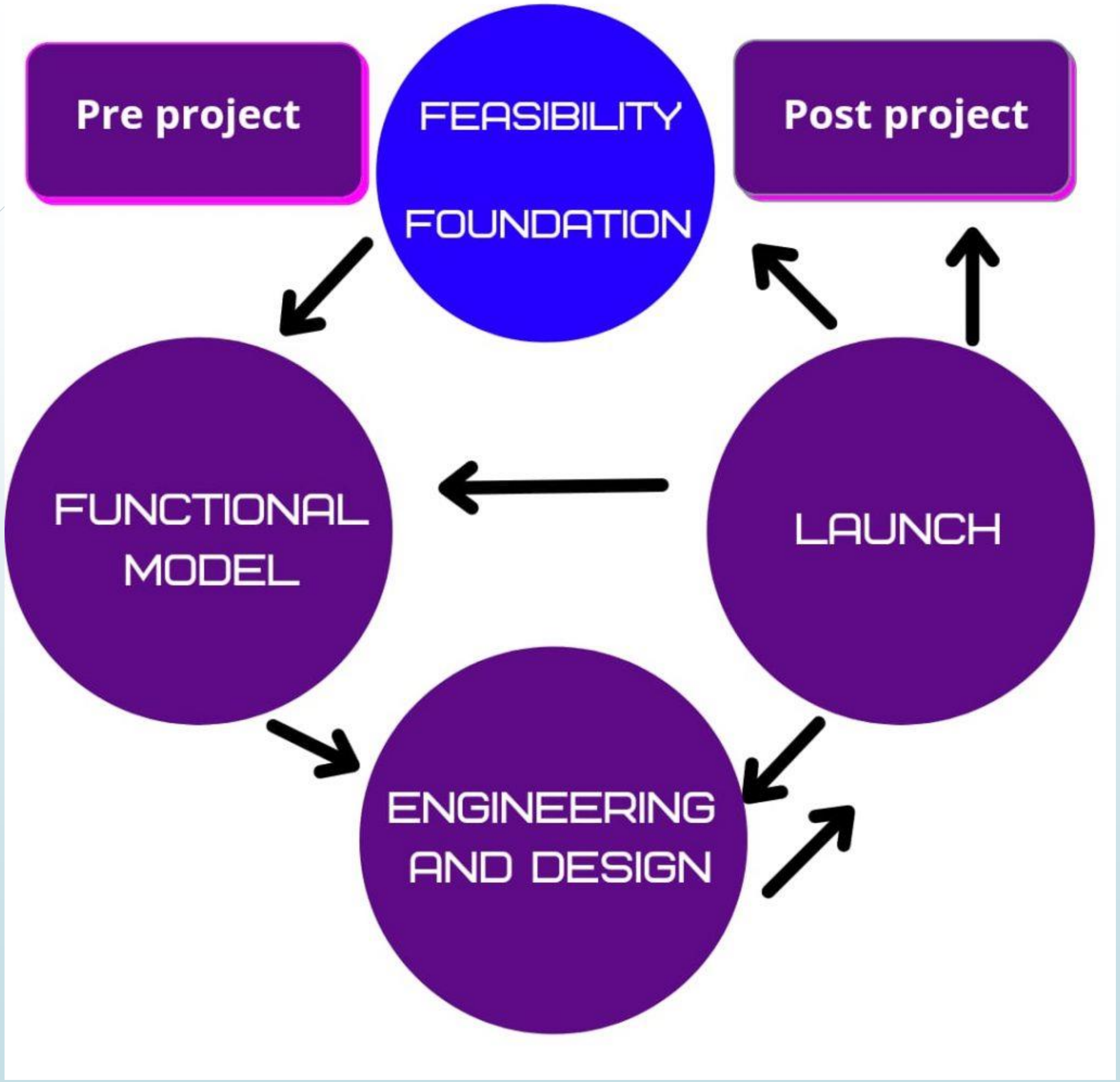
DSDM then defines three different iterative cycles:

Functional model iteration— produces a set of incremental prototypes that demonstrate functionality for the customer.

Intent is to gather additional requirements by eliciting feedback from users as they use the prototype.

Design and build iteration— revisits prototypes built during the functional model iteration to ensure incorporation.

Implementation— places the latest software increment (an "operationalized" prototype) into the operationa environment.


Pre project
FEASIBILITY
FOUNDATION
Post project
FUNCTIONAL MODEL
LAUNCH
ENGINEERING AND DESIGN

➢ Principles of DSDM

- ➢ Focus on market needs
- ➢ Deliver on time
- ➢ Collaboration
- ➢ Quality
- ➢ Build incrementally from firm foundations
- ➢ Develop iteratively
- ➢ Communicate continuously and clearly
- ➢ Demonstrate control

DSDM Tools and Techniques:

- The Timeboxing - One software development project will have multiple time boxes, and every time box is assigned one specific objective. Timeboxing helps in achieving the target in time. Usually, for a project duration of a time box can be from two weeks to a maximum of 6 weeks but not more than that.

  Example - Milestones
- The MoSCow prioritization- Must-Have: The requirements that are fundamental and must conform to the solution.

  Should Have: The things that are important for the business solution.

  Could Have: The requirements that are important but can more easily be left out for a short while.

  Won't Have: The requirements that can wait and include in later development.
- Facilitated workshops- team members and developers can work on a clear pre-set of deliverables.
- Iterative development- Its helps to ensure a solution good quality, ieration doen throught the project
- Modeling and prototyping techniques.- It helps visualize business domain & improves understanding.

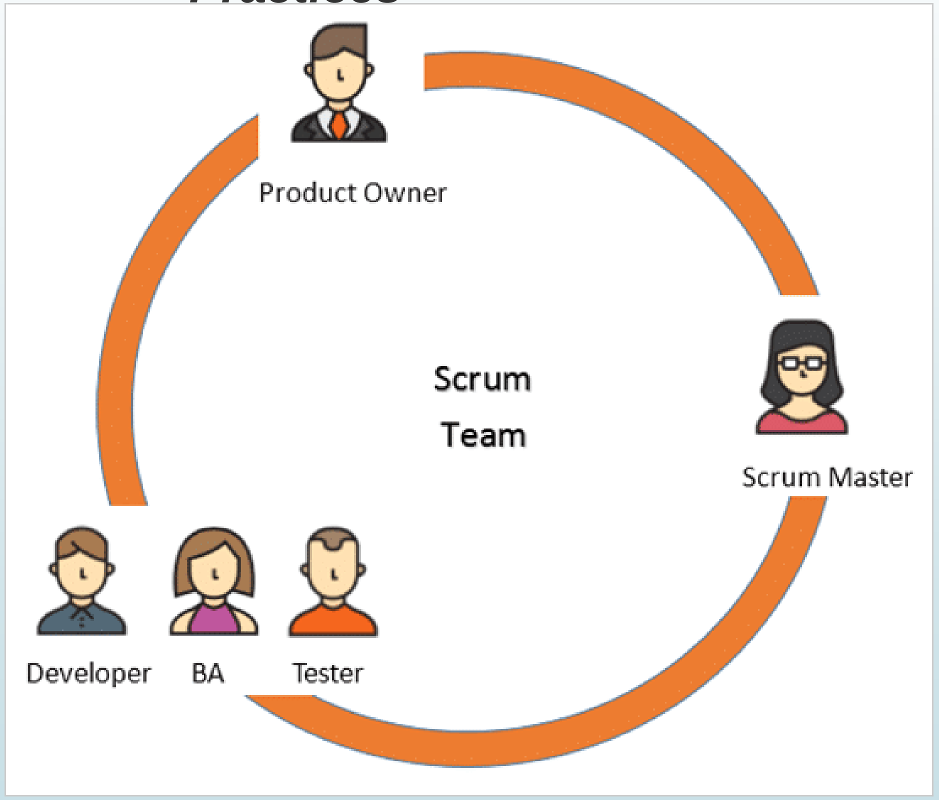
Advantages of DSDM

- It's a rapid process.
- Collaboration and communication are critical points of DSDM; the product designers can easily connect to the end-users for feedback.
- Adherence to deadlines.
-

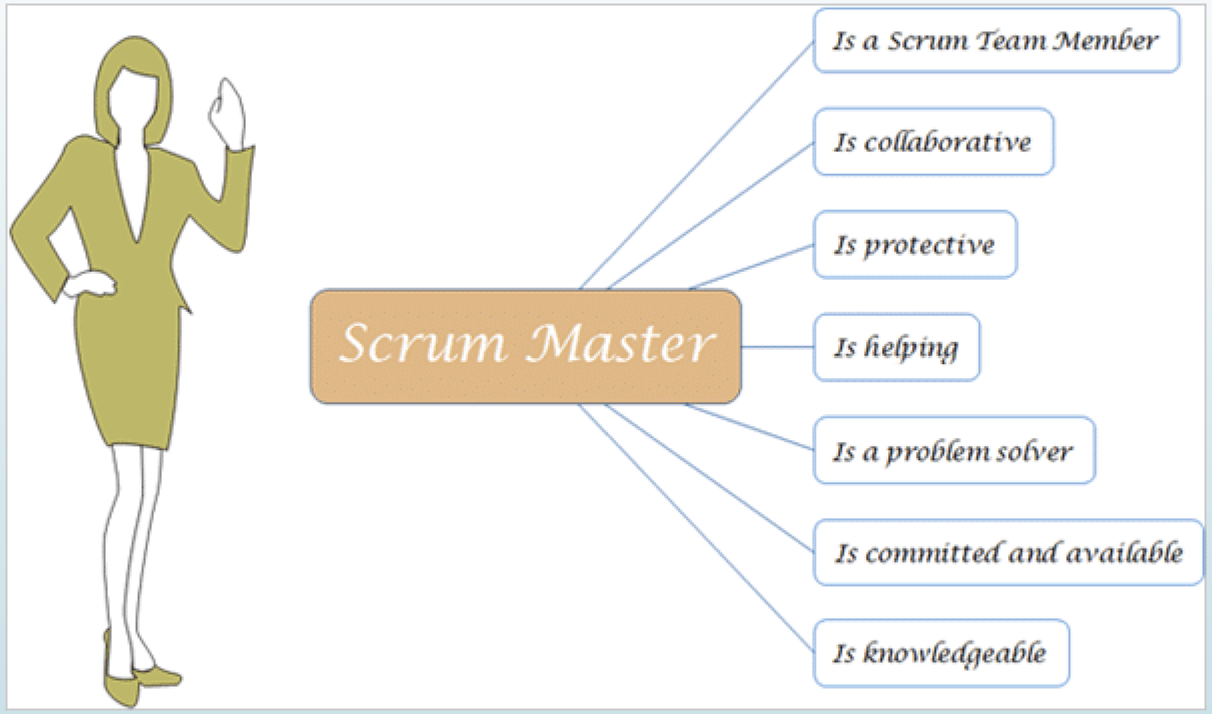
Scrum Team Size

Scrum Master is a Process Leader who helps the Scrum Team and the others outside the Scrum Team to understand Scrum Values, Principles, and Practices

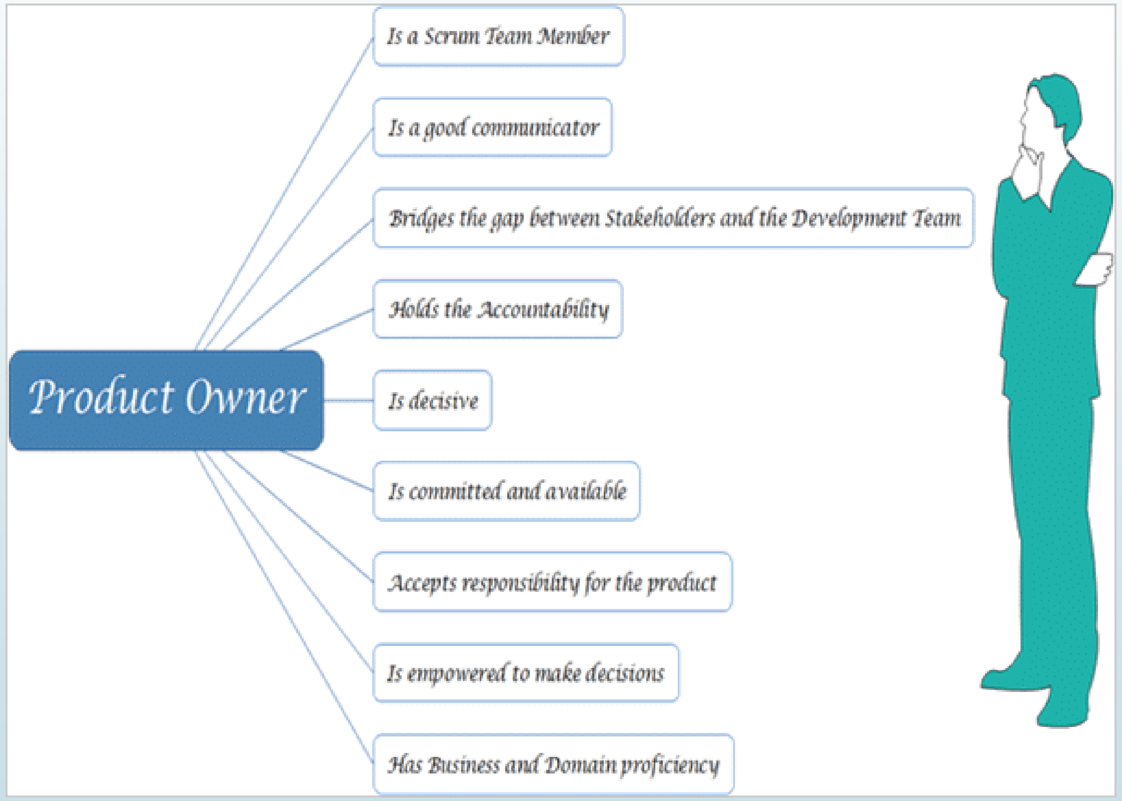
The recommended Development Team size in Scrum is 6+/- 3 i.e. from 3 to 9 members which do not include the Scrum Master and the Product Owner.



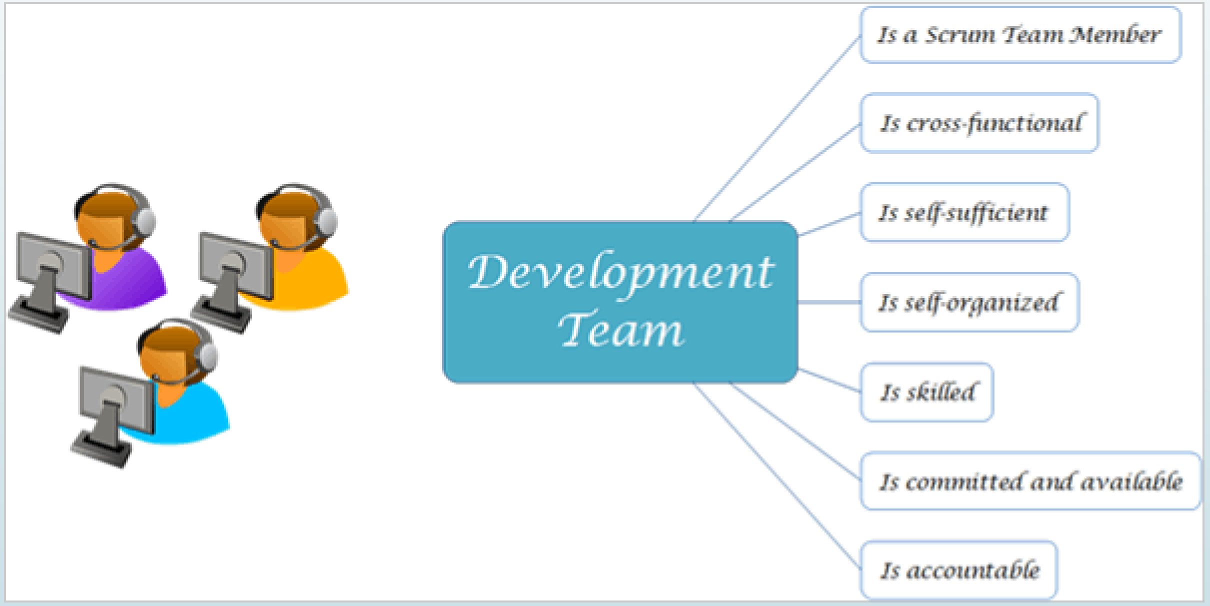
Roles and Responsibilities Scrum Master



Product Owner


Product Owner
Is a Scrum Team Member
Is a good communicator
Bridges the gap between Stakeholders and the Development Team
Holds the Accountability
Is decisive
Is committed and available
Accepts responsibility for the product
Is empowered to make decisions
Has Business and Domain proficiency

Development Team



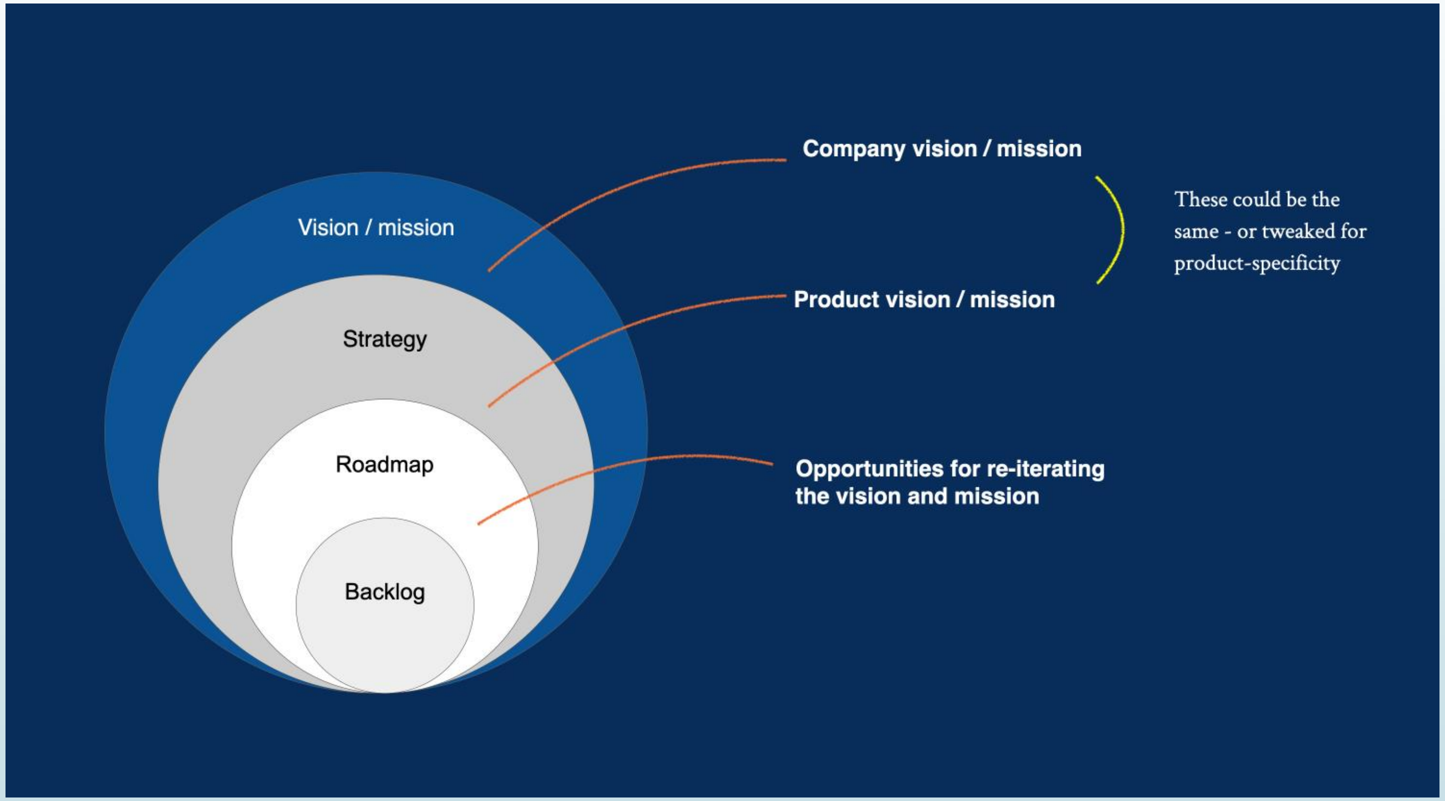
How to Distinguish Between Your Product Roadmap, Product Strategy, and Product Vision

- While you might "own" the product, your product's vision should be coming from the top of the house. It should be driving everything in your organization, not just product development. Sales, operations, technology… all of it should be working toward a common vision.
- **Product Vision**

A product vision is most simply known as describing the essence of your product. Used correctly a product vision can do a lot more.

Example

- LinkedIn: "To connect the world's professionals to make them more productive and successful."
- Amazon: "Our vision is to be earth's most customer centric company; to build a place where people can come to find and discover anything they might want to buy online."


Vision / mission
Strategy
Roadmap
Backlog
Company vision / mission
Product vision / mission
These could be the same - or tweaked for product-specificity
Opportunities for re-iterating the vision and mission

Product Strategy
Product strategies should be a little more specific than the vision, but not so specific that you're delving into the details of implementation.Your product strategy should cover the fundamentals:
Who are we trying to serve?
What problem(s) are we trying to address?
How are we planning to address the problem(s)?
How are we going to monetize our product?

Product Roadmap
In the context of this piece, the fundamental idea to grasp is that roadmaps are tactical. They're a set of plans and instructions that get you close to achieving the goals laid out in your product strategy. They're the only of these three items that should dictate specific behavior or tasks.

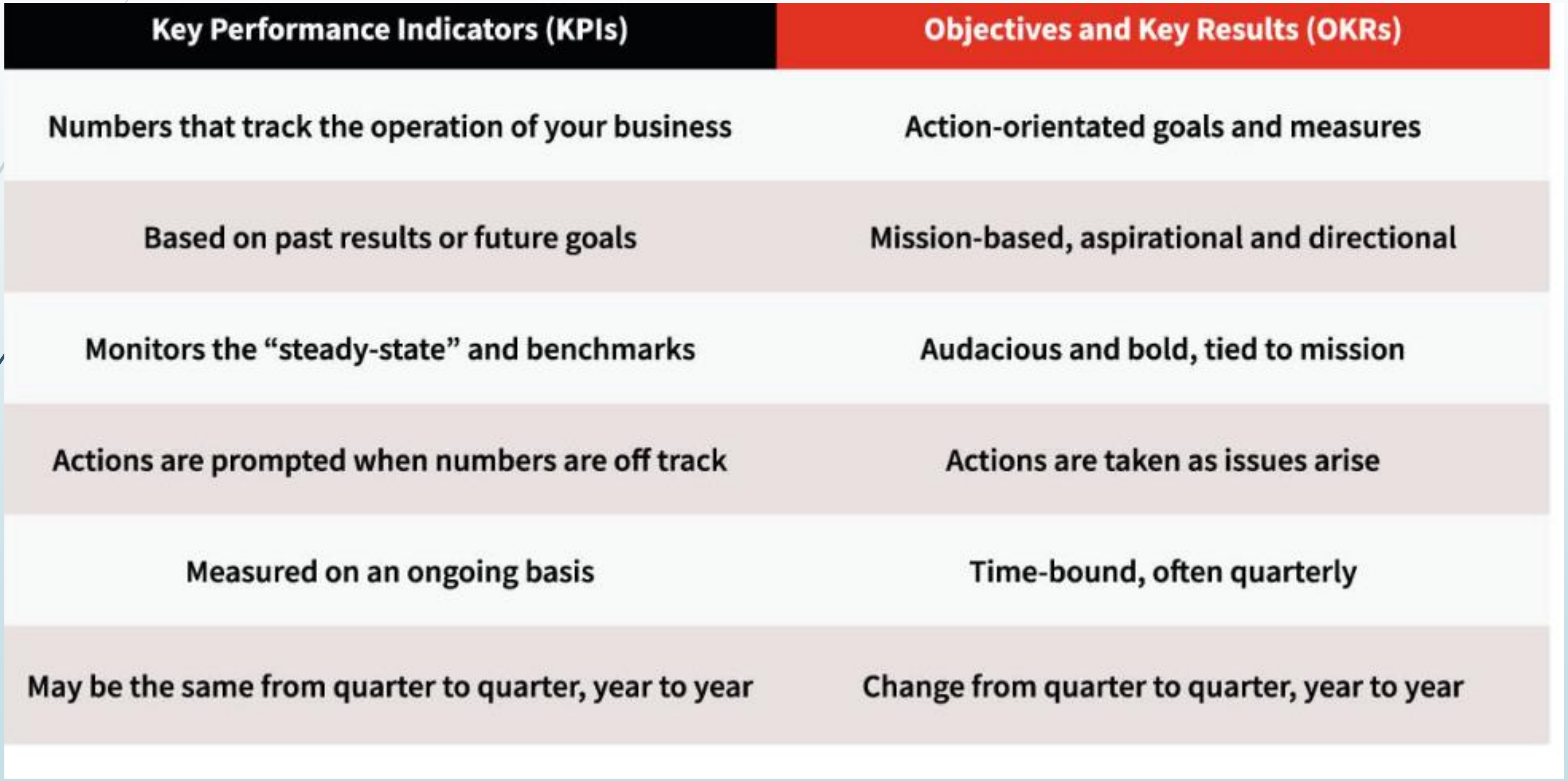
Keep Product Strategy changes to a minimum

Tie Your Product Roadmap to Your Product Strategy

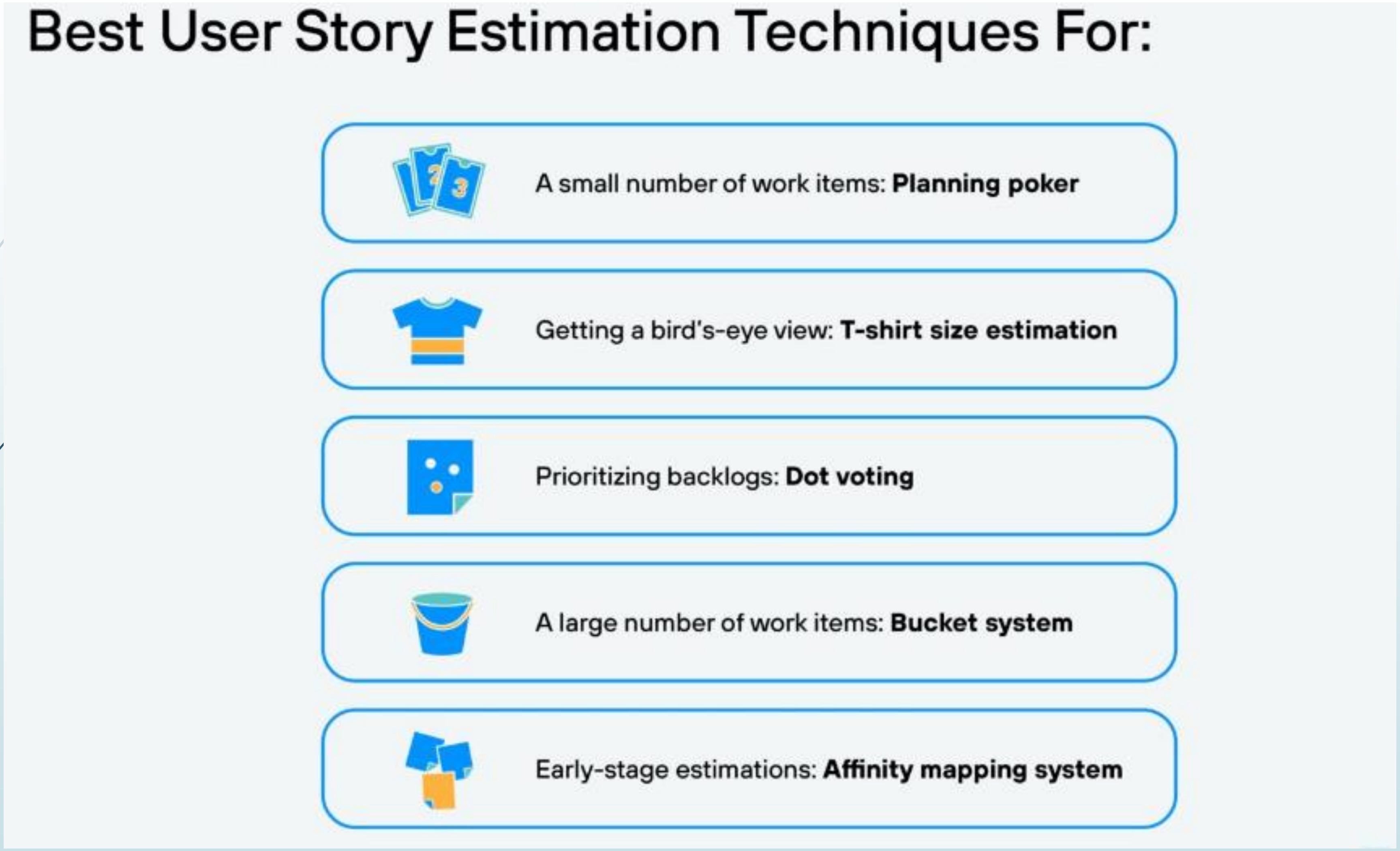
PROJECT OBJECTIVE AND KEY METRICS (OKR)

- Metric: A metric is a quantifiable unit of measurement that tracks and/orstatus of a particular process.
- KPI: KPI stands for Key Performance Indicator. A KPI is a metric of perfor determines if you are achieving success in an organization. KPIs ar indicators that look backwards in most contexts. KPIs can also be found in Result in an OKR.
- OKRS: OKR stands for Objectives and Key Results. It is a goal-setting frame helps create clarity, focus, transparency, engagement, and alignment by predetermined objectives and the key results or the metrics that sho towards the objective.
- Objective: The "O" in OKRS; the thing to be accomplished which shouldmeasurable, time-bound, and ideally quantifiable.
- KPOs: Where KPIs are a metric, a KPO (Key Performance Objective) re objective you hope to achieve. •
- Goal: Goals are longer-range and broader than objectives. They are typithe context of a 3-year or a 5-year aspiration. Goals are meant to betimeframe that extends beyond the period of one year. They can be broadand do not require quantification or a specific deadline.
- MBO- Mangement By objectives- Developed by faything of management "peter Drucker in 1950". It is collaboration of management/team in short term goals

https://kanbanize.com/okr-resources/okr/okr-vs-kpi



Key Performance Indicators (KPIs)	Objectives and Key Results (OKRs)
Numbers that track the operation of your business	Action-orientated goals and measures
Based on past results or future goals	Mission-based, aspirational and directional
Monitors the “steady-state” and benchmarks	Audacious and bold, tied to mission
Actions are prompted when numbers are off track	Actions are taken as issues arise
Measured on an ongoing basis	Time-bound, often quarterly
May be the same from quarter to quarter, year to year	Change from quarter to quarter, year to year


Best User Story Estimation Techniques For:

- A small number of work items: **Planning poker**
- Getting a bird's-eye view: **T-shirt size estimation**
- Prioritizing backlogs: **Dot voting**
- A large number of work items: **Bucket system**
- Early-stage estimations: **Affinity mapping system**

1. Planning Poker

This method uses the Fibonacci sequence where user story point values are presented as 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100 on playing cards, associated with different levels of complexity.

2. T-shirt Size Estimation

This agile estimation technique involves assigning each user story a T-shirt size (e.g., S, M, L.) The process helps team members achieve a big-picture understanding of the requirements.

3. Dot Voting

Using this estimation methodology, agile teams organize work items from the highest to the lowest priority to decide where to focus their time and efforts.

4. Bucket System

Start the estimation process by setting up a row of cards (i.e., buckets) with values in the Fibonacci sequence (i.e., 0, 1, 2, 3, 4, 5, 8, 13, 20, 30, 50, 100, 200). Then, team members will discuss a work item and place the user story in an appropriate bucket.

5. Affinity Mapping System

This method uses silent relative sizing. Start by placing two cards on opposite ends of a wall and giving team members a list of user stories (e.g., on sticky notes.)

Agile Release Plan

- Define Vision
- Rank the Product Plan
- Hold a release planning meeting
- Agenda of meeting
  - ✓ Review Roadmap
  - ✓ Review Architecture
  - ✓ Review velocity
  - ✓ Iteration Schedule
- Finalize and share product release calendar

Popular Prioritization Techniques

- MoSCoW agile Prioritization determines which priority you should give to each feature from a given project. It’s processed through the assignment of a list of requirements to four categories:
- Rice Scoring
- Kano Model
- Value Vs Complexity Quadrant
- ICE Scoring Model