Modern Project Management

PAPERS

Rev. 1.1.0 –March 12, 2023

Modern PM

an Agile Constellation Star

Summary

1. Overview 4

2. The Modern PM 5

2.1 Mindset 6

2.2 Governance 7

2.3 Guided Continuous Improvement 8

2.4 Leadership 10

3. Roles 13

3.1 The Project Manager 14

3.2 The Product Owner 15

3.3 The Scrum Master 16

3.4 The Team Leader 18

4. ModernPM Lifecycle Toolkit 19

4.1 Predictive Lifecycle 21

4.2 Iterative and Incremental Lifecycle 24

4.3 Agile lifecycle 25

4.4 Lean lifecycle 28

4.5 Lifecycle Exploratory 32

4.6 Lifecycle Program 33

4.7 Hybrid lifecycle 36

5 Modern PM Fast Prototyping 37

5.1 Vision 37

5.2 Scope definition 39

5.2.1 Work Breakdown Structure 39

5.2.2 Story Mapping 40

5.3 Complexity 41

5.3.1 Scope Definition Accuracy 43

5.3.2 Uncertainty 44

5.3.3 Modern PM Decision Tree 46

5.4 Lifecycles Bouquet 47

6 Call back 48

6.1 PMI Disciplined Agile 48

6.2 PMI PMBOK 51

7 Conclusion 53

1. Overview

This guide is intended to provide a working approach to project management and execution, hereinafter referred to as Modern Project Management (ModernPM, MPM).

Based on consolidated solutions, in particular on the PMBOK® and the PMI Disciplined Agile® (DA™) toolkit, a concrete strategy is described to analyze the specific context of each project and identify the most suitable lifecycle mix for its management and execution.

The various topics will be treated at a medium-high level, leaving it to the reader to study and research any insights he may deem necessary. This approach was chosen to avoid transforming this guide into a "tome", avoiding reporting elements that are easily found in the literature, but focusing on putting together a flexible and immediate-use tool that puts the reader in a position to be immediately operational.

2. The Modern PM

The keyword of the iterative nature of the ModernPM is Adapt: after having created the project vision, and identified the relative context, we proceed to the phase of definition of the deliverables (objectives) and the choice of the most suitable lifecycles to rapidly generate value for the customer and collect feedback.

In this way it is possible to realign the activities according to the real value generated for the stakeholders, reducing the risk of having to make massive interventions in the final phases of the project where the costs could no longer be sustainable.

As visible in the Poster below, the ModernPM is based on 4 Pillars which make explicit its founding elements:

- **Mindset**, the cultural aspect to be developed;
- **Governance**, how to manage a project operationally and what skills are required;
- **Guided Continuous Improvement**, the ability to continuously improve in a pragmatic way;
- **Leadership**, the need to stimulate the emergence of natural leaders in their own context.

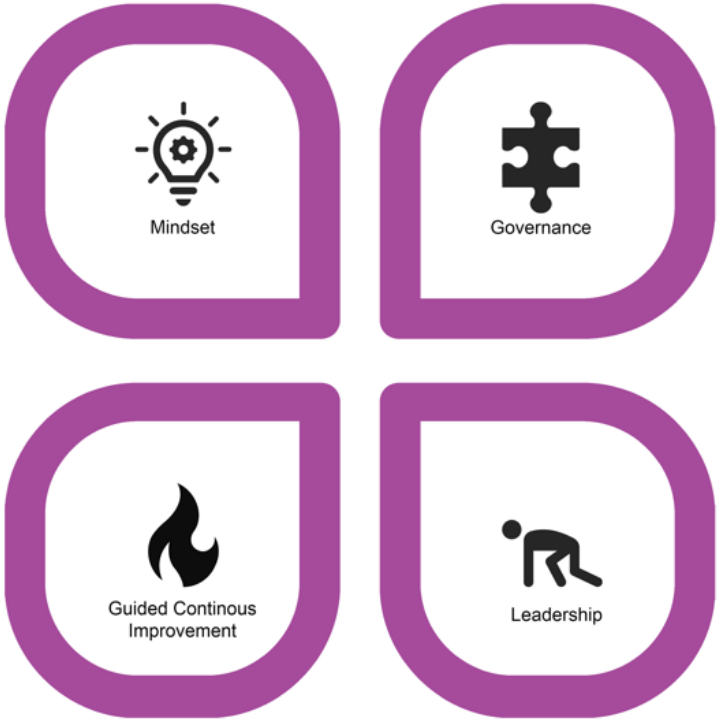


Figure 1-Modern PM Poster

2.1 Mindset

At the basis of the ModernPM we find the Philosophy, Principles and Practices of the AgileConstellation Manifesto[1], of which ModernPM is a Star, defining specific practices and principles related to the reference domain.

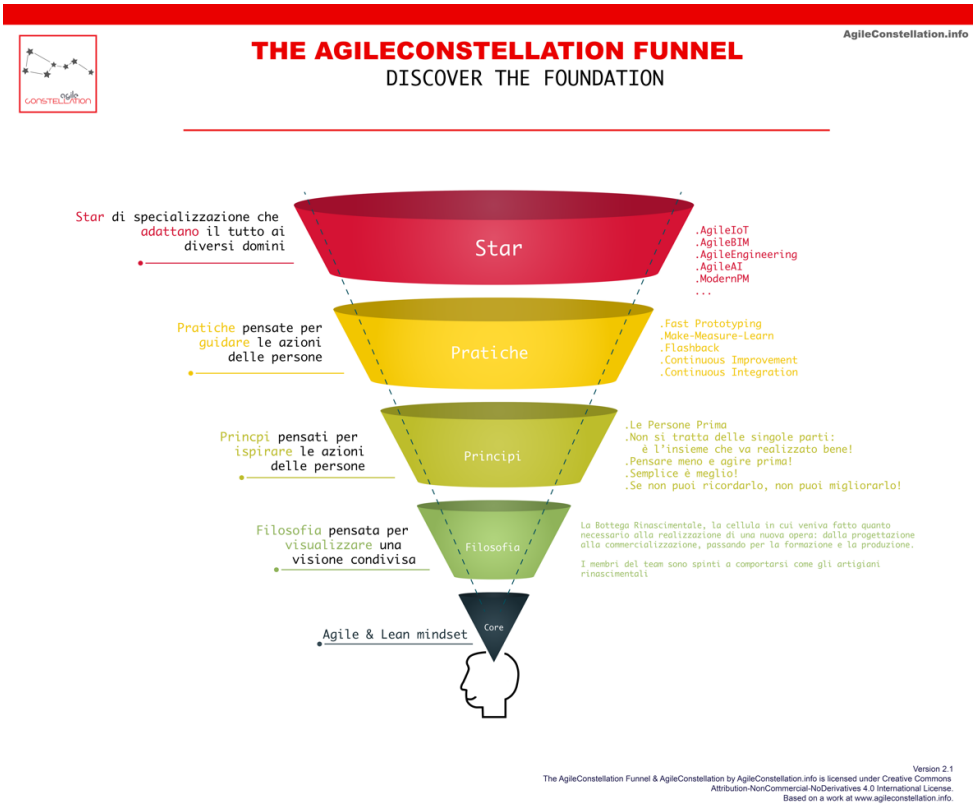


Figure 2- AgileConstellation funnel

We therefore have:

- **Philosophy**, inspired by the Renaissance workshop, i.e. the cell that performs what is necessary for the creation of a new work: from design to construction and marketing.
- **Principles (core):**
  - *It's not about the individual parts: it's the whole that must be done well!*
  - *Think less and act sooner!*
  - *Simple is better!*
  - *If you can't remember it, you can't improve it!*
- **Practices (core):**
  - *Fast prototyping*, validate the sustainability of the solution

[1] www.agileconstellation.info

- *Make-Measure-Learn*, quickly experiment with the different hypotheses and the different assumptions
- *Flashback*, rapid alignment in which the observer goes to the work desk
- *Continuous improvement*, constantly improve every aspect

Continuous Integration, constantly integrate the different souls of the solution

2.2 Governance

One of the common mistakes is to confuse Project and Product, even using the two terms as synonyms. However, there are profound differences between them:

- A Project has a well-defined beginning and end, also responding to the characteristic of uniqueness, as it is not routine but composed of a specific set of activities designed to achieve a single goal, often through people who do not work permanently together. At the end of a project, the team (or teams) is usually disbanded and members assigned to new projects.
- A Product is a service, platform, application, system, that is created, maintained and supported to provide benefits to specific customer and business needs. Products tend to be maintained by a stable team (or even several stable teams) working continuously.

These specific perimeters are reflected directly on how projects and products are performed and managed, on the tools used and on the figures (roles) that accompany them.

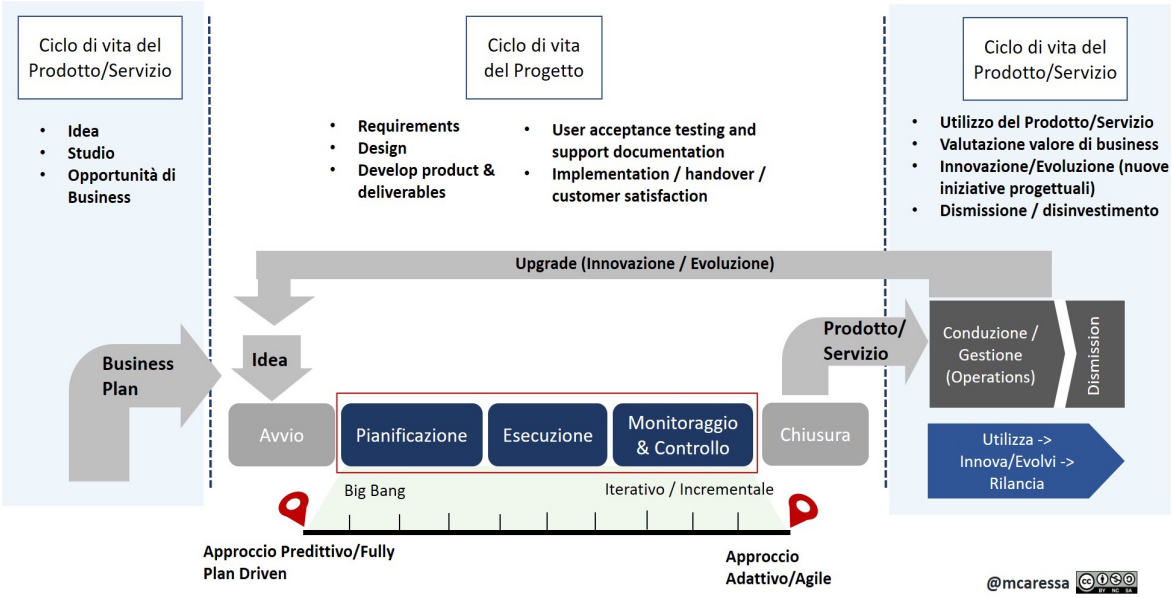


Figure 3-Project & Product Lifecycle

2.3 Guided Continuous Improvement

Guided Continuous Improvement[2] (GCI) extends the basic principle (AgileConstellation) of Continuous Improvement, and is based on two fundamental aspects:

- the Kaizen loop, i.e. the approach in which the team progressively experiences small changes in its Way of Working (WoW). After each experiment, if what has been tested has given the desired results, the team adopts it in a structured way, vice versa it will choose a different option which it will experiment again.
- the Continuous Improvement,or the act of applying a series of Kaizen loops to improve one's WoW over time.

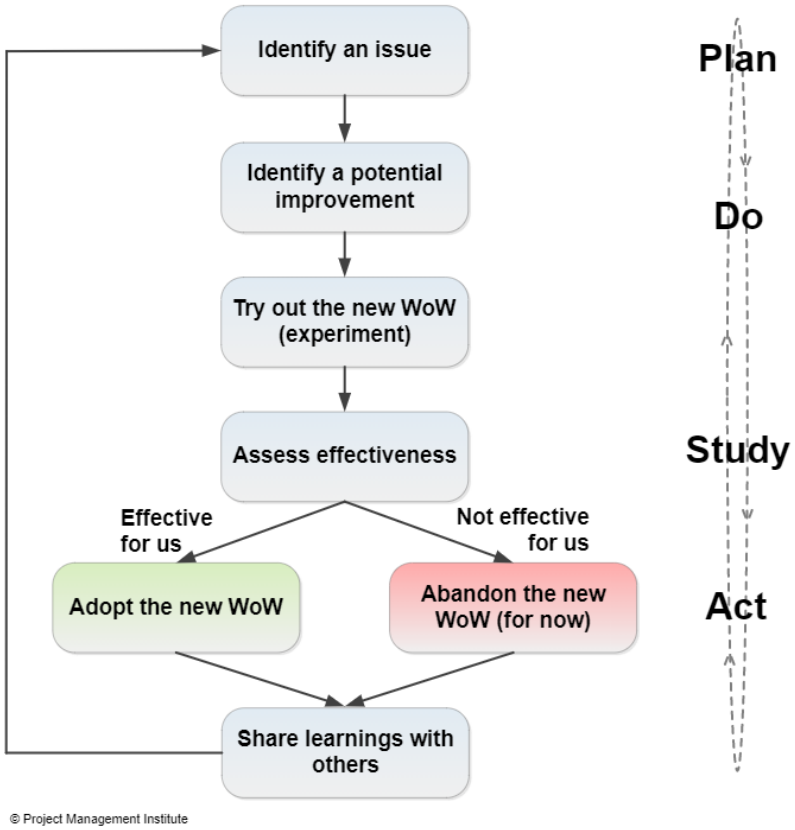


Figure 4- Kaizen Loops and Continuous Improvement

Guided Continuous Improvement is a technique (suggested by the Disciplined Agile toolkit) that reinforces Continuous Improvement, suggesting to the team a bouquet of options to pragmatically address the "method prison" problem[3]:

[2]https://www.pmi.org/disciplined-agile/gci/guided-continuous-improvement
[3]https://essence.ivarjacobson.com/publications/white-papers/tear-down-method-prisons-set-free-practices

"The only option that development organizations and teams see themselves as having is to adopt a method and to reject all others—whereas, in fact, what is needed for organizations and teams to be free is to select the professional practices that they need, from wherever these may be defined, and use them in whatever permutations and combinations are appropriate to meet the exact set of circumstances and challenges they face".

The impacts of getting caught up in an overly prescriptive framework (or method) were further highlighted by Donald J. Reifer[4]and represented in the following figure:

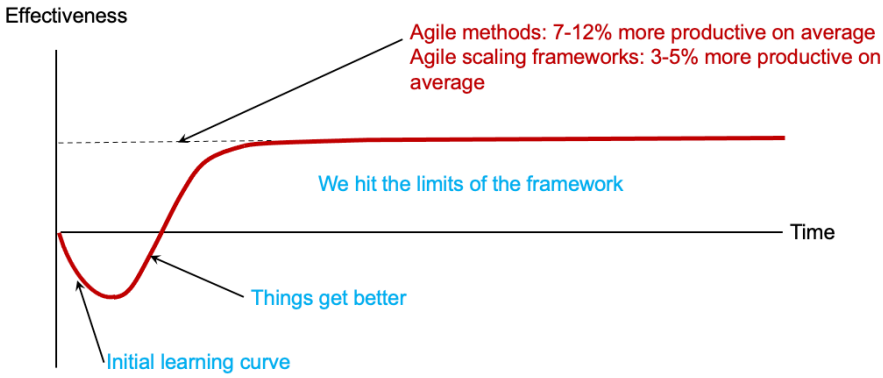


Figure 5- Source: Reifer, D. Quantitative Analysis of Agile Methods Survey (2017): Twelve Major Findings

The basic problem with such approaches is that too often we forget thatthey are models, often linked to well-defined contexts. If this is combined with the claim to use them indiscriminately for each project in a dogmatic way, and not in a critical and contextualized way, we find ourselves imprisoned in a mechanism that does little to help develop the ability to adapt according to the objectives to be achieved.

To break out of this prison, the GCI is the winning weapon, providing direct support in identifying potential problems and, above all, proven guidance to help the team identify techniques that could work in the specific context.

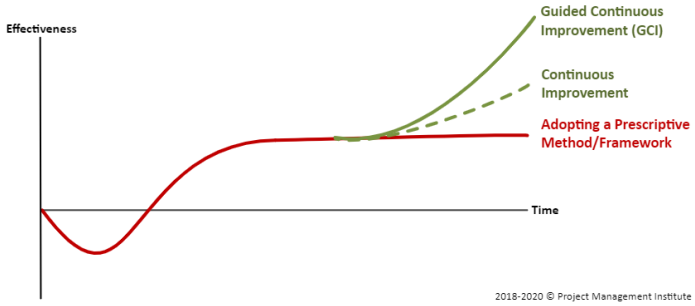


Figure6- CGI, CI & Methods/Framework

[4]https://www.infoq.com/articles/reifer-agile-study-2017/

Although each option should still be treated as an "experiment to be done", the advantage is that CGI allows you to increase the percentage of successful experiments and therefore to increase the overall rate of improvement.

Figure 7– GCI increases the chances of successful experiments

There are two key directions for adopting a GCI-based approach:

- *Make use of an expert (PM, Scrum Master, Coach, etc.) and, above all, listen to him.*It is not easy to find a true expert who can help the team and guide it in its growth, but once found, it is essential to have the time and patience to listen and follow him.
- *Make use of consolidated practices in the cotext*, in order to provide a series of reasoned options to pragmatically support the team. Good coaches have the humility to recognize that they don't know everything, and they, in turn, will leverage a mix of tools and solutions to help the team make better decisions about how to approach challenges, continuously improving themselves in an experimental way.

2.4 Leadership

Identifying the appropriate leadership style is critical to project success.
However, it is unlikely that a single style is sufficient, more realistically, it will be necessary to constantly adapt to the evolution of the activities and the associated development context.

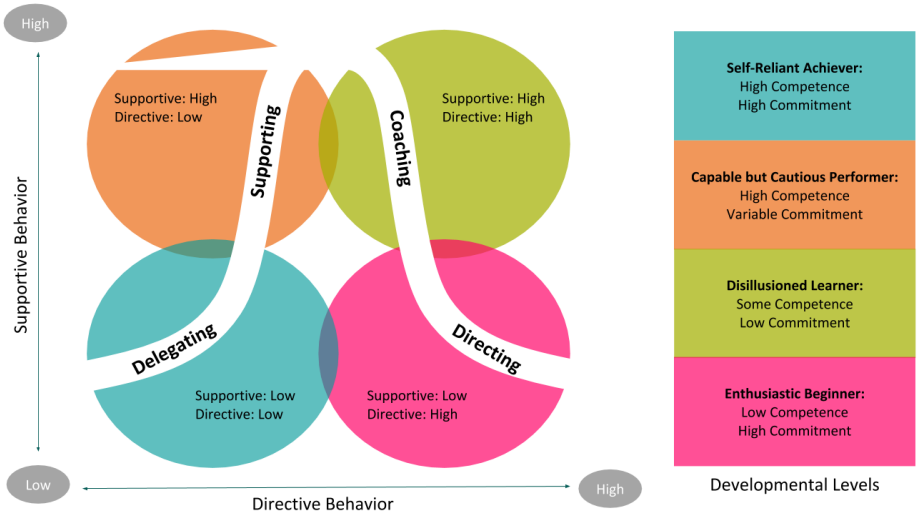
A help in this direction comes from Situation Theory[5], or from the studies and experience of Paul H. Hersey, Kenneth Blanchard and Dewey E Johnson, who describe Situational Leadership in the book Management of Organizational Behavior: Leading Human Resources:

> *"there is no "right" and unique way to be a leader, but it is necessary to choose the most appropriate style for different situations."*

Situational Leadership builds on the relationship between leaders, teams and stakeholders, providing a framework for analyzing different situations based on the maturity level of the team and that of the stakeholders.

[5]https://it.brainmain.net/10493030-the-situational-theory-of-leadership

Hersey-Blanchard Situational Leadership Theories



*This graphic combines elements of the Hersey-Blanchard Situational Leadership Theory & Blanchard's revised Situational Leadership Theory II

Figure 8- Situational Leadership (source: https://guntergroup.com/situational-leadership/)

Identifying the most appropriate leadership style for the level of maturity helps to have fewer conflicts and a broader motivational drive, generating greater cohesion in the work group. In fact, as the tasks and/or projects vary, the necessary skills that must be acquired by team members vary and, therefore, the leadership style necessary to achieve the objectives through the development and growth of one's people must vary.

Again according to Blanchard (see Leadership and the One Minute Manager and the SLII® model), there are three elements that support an effective leader:

- **Definition of Objectives**, the starting point of a situational leader, creating alignment on what needs to be done and when and thus motivating participants to take an initiative.
- **Maturity level of team members**, the leader co-evaluates the maturity (development) level of each team member based on their competence and commitment to a specific goal or task. The level of maturity is always related to the goal or task, and not to the person.
    - There are four levels of development characterized by different combinations of employee competence and commitment: D1 (Low Competence and High Commitment), D2 (Some Competence and Low Commitment), D3 (Moderate-High Competence and Variable Commitment) and D4 (High Competence and High Commitment).
- **Correspondence between leadership style and development of team members**, provides the most appropriate leadership style to guide and support the collaborator

according to his level of development. With this skill, leaders learn about different leadership styles and how to be more flexible.

- Just as there are four levels of development, there are also four leadership styles characterized by different combinations of directive behavior and supportive behavior: S1 (High Directives and Low Support), S2 (High Directives and High Support), S3 (Low Directives and High Support), S4 (Low Directives and Low Support).

The correspondence phase is the determining moment, with the aim of matching the leadership style with the level of maturity to identify the most appropriate correspondence to the context: the more leadership styles are aligned with the levels of maturity, the faster people will improve and will evolve from level to level.

Although it is difficult to identify all possible leadership styles, the following matrix tries to represent the most common in relation to the complexity of the activities and the experience of the work team.

Figure 9- Types of Situational Leadership

3. Roles

Within a project there may be many roles with specific responsibilities and objectives, but also with overlapping which can make it difficult to clearly identify the relative operational loss.

Among them it is possible to identify the so-called leadership roles:

Role	Description
Project manager	Oversees the project and focuses on stakeholders. Manages all inter-functional work, ensuring that the project develops on time and on budget.
Product Owner	Support the team tactically by prioritizing the Product Backlog. He represents the voice of the customer and deals with finding the right synergy with the customer, as well as the way to represent his needs in workable elements for the team.
Scrum Masters (Agile Master)	Helps the team in day-to-day operations, in particular in the effective adoption of Agile and in the development of the related mindset. He is responsible for facilitating events, stimulating communication and helping teams overcome problems and obstacles, all with a view to continuous improvement.
Team Leader	Coordinates the group to which he belongs in the development of a project and in the achievement of the associated objectives. It is not a question of a "boss", but rather of a "companion" who offers his support without influencing decisions too much.

Table 1: Leadership Roles

The competencies of these roles can be mapped to the PMBOK Performance Domains[6], or on the aspects of focus of a project leader:

- **Team**,addresses the activities and functions associated with the people responsible for the development and deliverables of the project.
- **Development Approach and Lifecycle**, addresses the activities and functions associated with the development mode, cadence, and phases of the project life cycle.
- **Measurement**,addresses the activities and functions associated with evaluating project performance and taking appropriate action to maintain reasonable performance.
- **Stakeholders**,addresses the activities and functions associated with stakeholder management.

[6]"A Project Performance Domain is defined as a group of related activities that are critical for the effective delivery of project outcomes."

- **Project Work**,addresses the activities and functions associated with defining project processes, managing physical resources, and promoting an environment conducive to learning.
- **Planning**,deals with the activities and functions associated with the aspects of initial planning, continuous review and evolution, the coordination necessary to deliver the expected results from the project.
- **Uncertainty**,addresses the activities and functions associated with risk and uncertainty.
- **Delivery**,addresses the activities and functions associated with the delivery of expected functionality and related quality aspects.

The different roles can cover all, or only part, of the overall spectrum of performance domains, in relation to their contribution to the project:

	Project manager	Product Owner	Scrum Masters	Team Leader
Team	x	x	x	x
Development Approach and Lifecycle	x		x	x
Measurement	x	x	x	x
Stakeholders	x	x		
Project Work	x		x	
Planning	x	x	x	
Uncertainty	x	x		x
Delivery	x	x		x

Table 2 - Performance Domains and Roles Matrix

3.1 The Project Manager

The Project Manager (PM) is accountable for the project, for the resources and for the scope of the annexed requirements, and as such has the overall governance of all the related activities, constantly interacting with the related stakeholders.

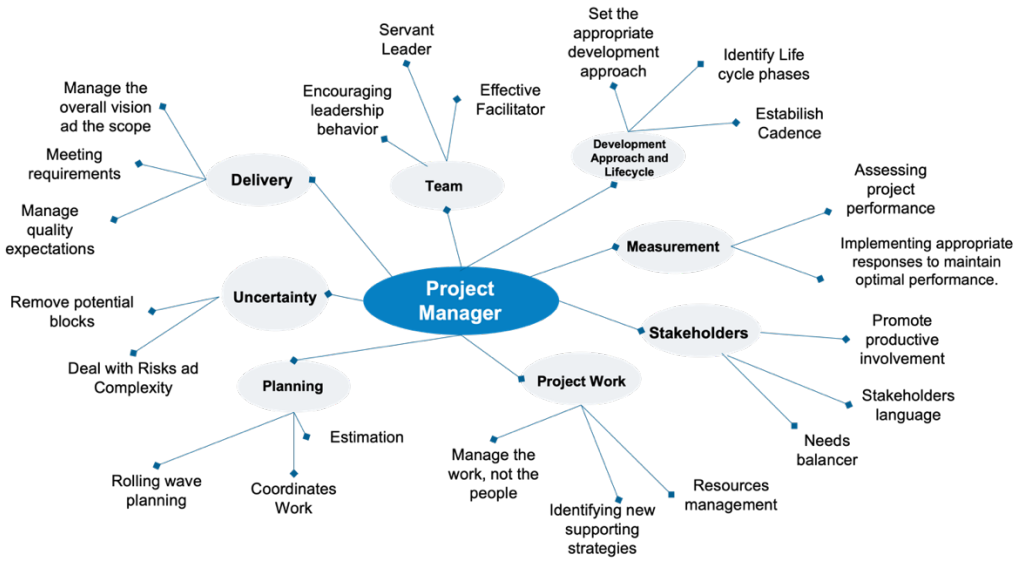


Figure 10-Project manager skills

To best carry out their activities, the project manager must possess a balanced mix of hard and soft skills, always proving to be an excellent communicator and an empathetic leader. This must be combined with organizational meticulousness, a resolution in making decisions, as well as flexibility and a spirit of adaptation.

Last but not least, it is essential to know how to deal calmly and appropriately with the various problems that arise during the entire life of the project, bringing in the people who can help bring it back into line with expectations.

3.2 The Product Owner

The Product Owner (PO), in the context of projects that follow an Agile lifecycle, is responsible for the value of a product and as such constantly interacts with the customer to validate its scope. Unlike the Project Manager, his focus is on the product, with respect to which he declines his activities and actions.

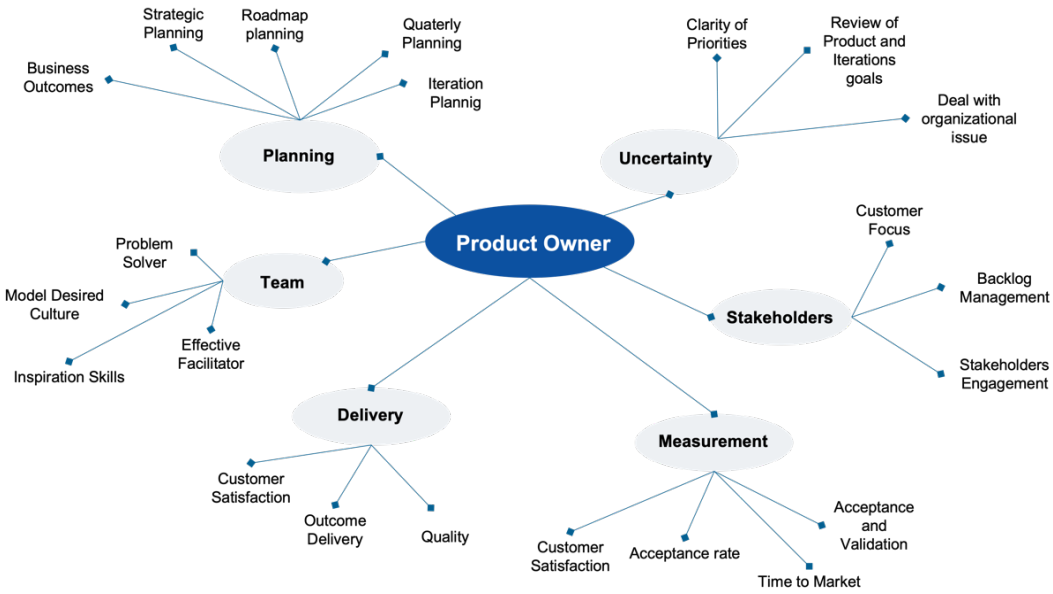


Figure 11-Product owner skills

The PO maintains and develops the product vision and has the primary objective of harmonizing the needs of the stakeholders in relation to the product. To achieve them, the OP:

- has exclusive governance of the Product Backlog[7];
- ensures that the team creates Value, accepting or rejecting what has been achieved according to the "Definition of Done";
- ensures that the Product Backlog is visible to all;
- participates in iteration planning;
- defines and verifies the Acceptance Criteria.

3.3 The Scrum Master

The tasks of the Scrum Master (SM), a figure present in the projects that follow an Agile lifecycle, are very varied, depending on the specific context and situation.

His focus is on facilitating the lifecycle that the team is using, supporting the latter thanks to his leadership and coaching skills, and helping him to constantly improve in achieving the objectives.

[7]For a description of the specific tools associated with the various lifecycles, refer to what is described below.

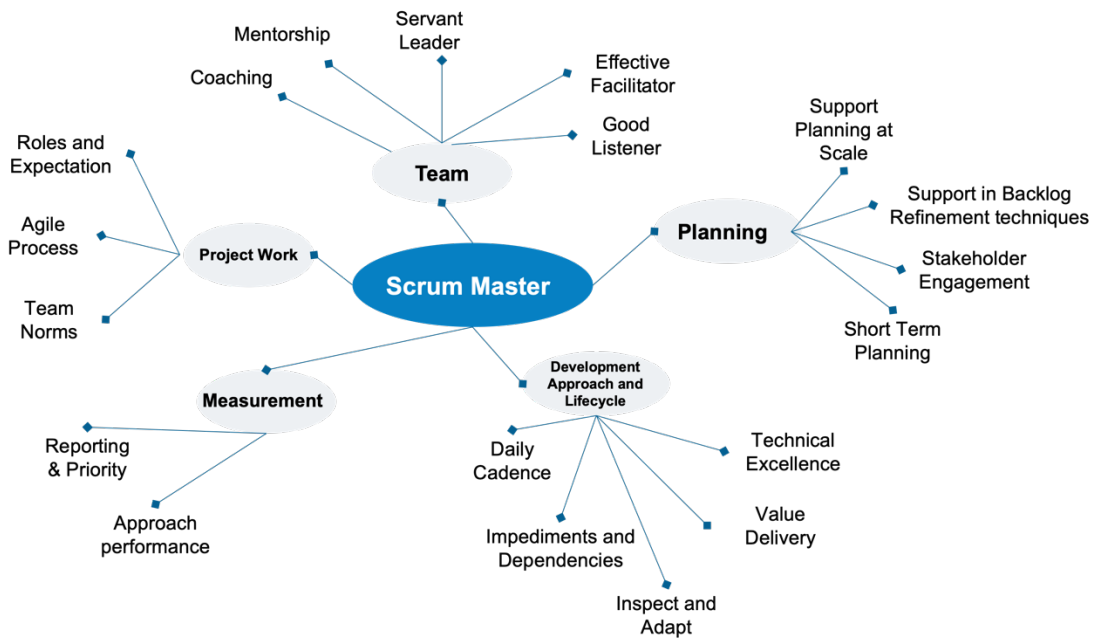


Figure 12- Scrum master skills

The Scrum Master plays an almost «consultant» role, supporting in particular:

- the Product Owner, in setting product objectives, in using the Product Backlog, in Iteration Planning and in the various meetings.
- the Team, in the adoption of the specific lifecycle toolkit (see below), in guaranteeing the correct implementation of the deliverables and in the search for actions for continuous improvement.

The SM focuses on the team development path:

- making sure that the team itself, and the organization, develops an Agile mindset;
- helping him, together with the organization, in adopting agile practices;
- supporting him in improving his productivity;
- placing the emphasis on the quality of what has been achieved;
- protecting it from external interference;
- removing the impediments.

From a growth perspective, a Scrum Master generally follows the path shown in the following figure:

Figure13- Scrum Master growth path

3.4 The Team Leader

The Team Leader (TL) actively guides the specific work team in relation to the objectives it has in charge.

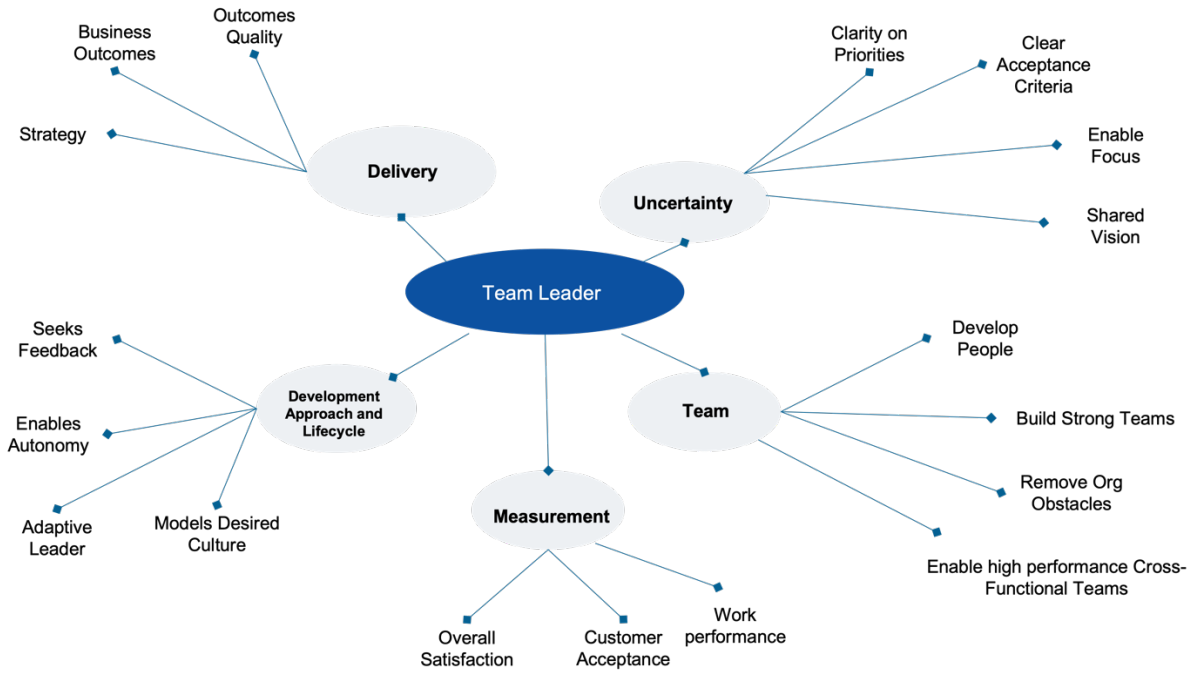


Figure 14-Team Leader skills

The figure of the team leader is often crucial for obtaining maximum involvement and, consequently, adequate team productivity. The TL supports team:

- in daily work;
- in solving problems related to specific activities;
- in the identification of possible solutions;
- in conflict mitigation;
- in professional growth.

4. ModernPM Lifecycle Toolkit

The way in which one decides to manage and execute a project leads to a bouquet of elements that can also be very different from project to project.

In general, it is possible to identify five families of lifecycles (or approaches):

approach	Requirements	Activities	Delivery	Goal
predictive	Fixed	Performed once for the entire project	Single delivery	Cost management
iterative	dynamic	Repeated until correct	Single delivery	Correctness of solution
Incrementally	dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback
Lean	dynamic	Performed once	Frequent small deliveries	Customer value via one-at-time delivery

Table 3 – Lifecycles, adapted from Project Management Institute, Agile Practice Guide, Project Management Institute

Each of these lifecycles uses a specific approach to managing requirements, planning, estimating and executing works, as summarized in the following image:

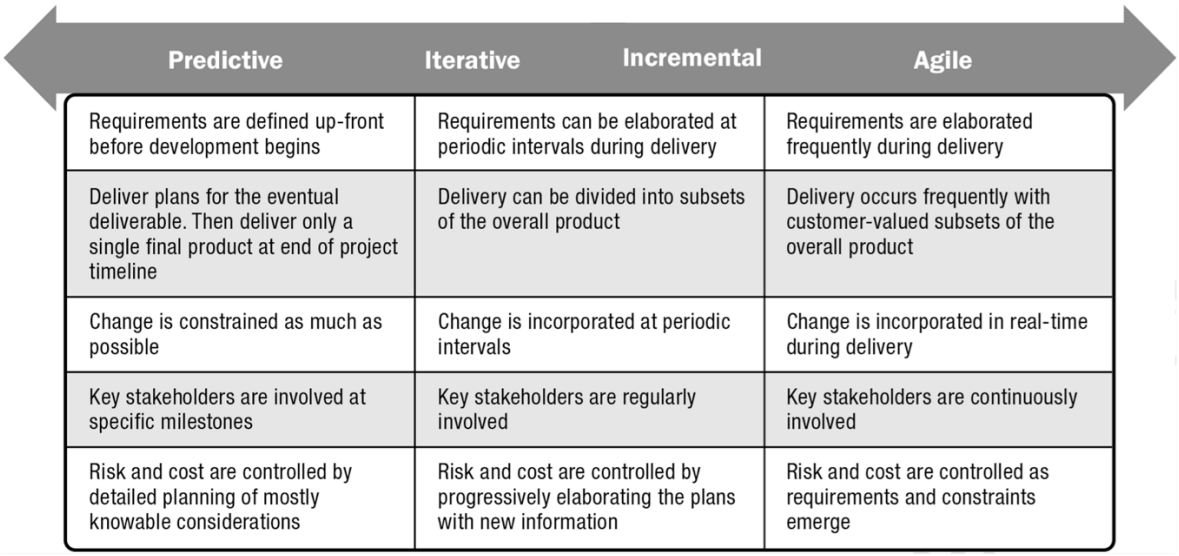


Figure 15– Lifecycles ranges - A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Sixth Edition

In the following, the different lifecycles are characterized in relation to five fundamental aspects: Context, Metrics, Roles, Events and Tools which, together, define the ModernPM Lifecycle Toolkit (MLT) for each of them.

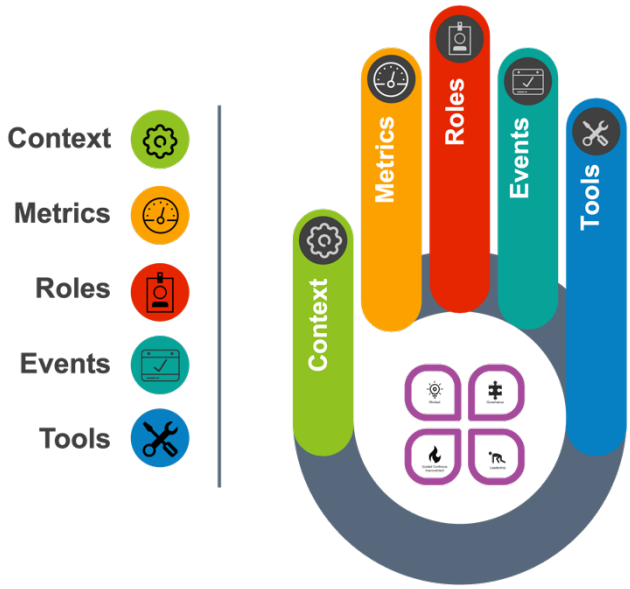


Figure 16– ModernPM Lifecycle Toolkit (MLT)

Specifically, therefore, we have:

1. **Context**: the context, in terms of project and organizational complexity, is the fundamental driver in the choice of the appropriate lifecycles.
2. **Metrics,** land the right metrics allow the working group to validate the improvements achieved.
3. **Roles,** depending on the lifecycles identified, it is essential to identify and support the related roles.
4. **Events**, the specific events help the working group to align itself in the various operational aspects and with the project stakeholders.
5. **Tools**, foreach lifecycle there are specific tools that help in the different phases of management and operation.

Each MLT explains what is operationally necessary to adequately implement the selected lifecycles and how to support their application.

In relation to the main tools and outputs to be obtained for each project phase, it is possible to define "*the Bignami of the ModernPM":*

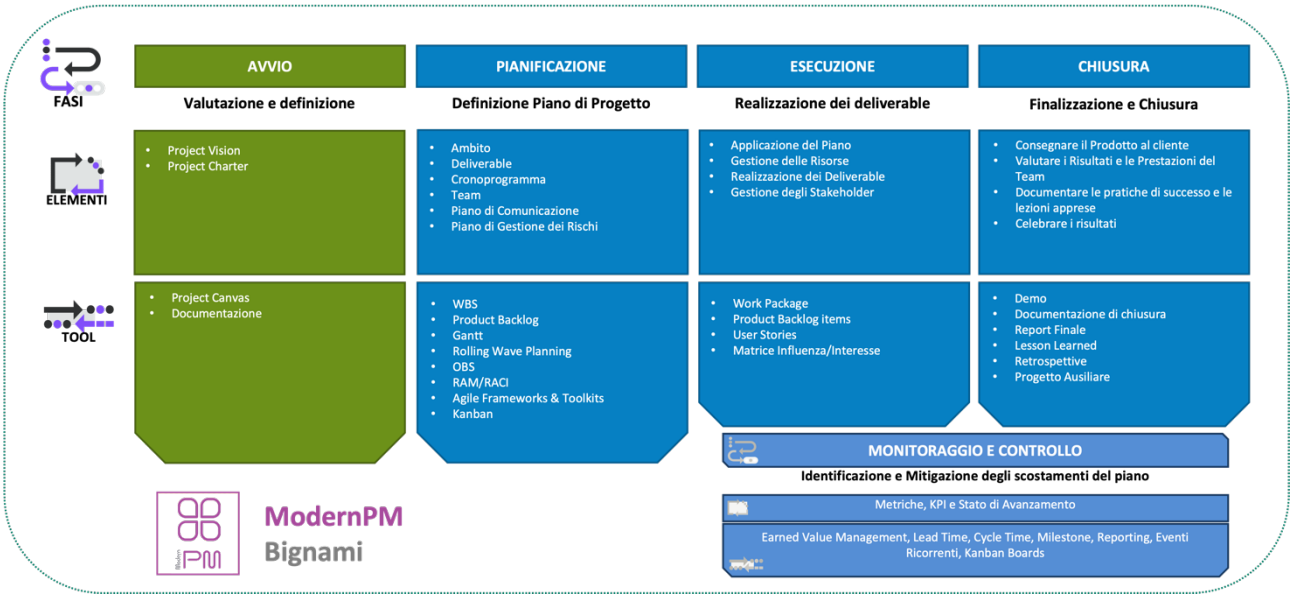


Figure 17- Bignami of the ModernPM

Let's now find out how all this specializes in relation to the different lifecycles.

4.1 Predictive Lifecycle

In the predictive approach, also known as waterfall or serial, the project scope, timing and associated costs are determined as early as possible.

The ability to develop detailed plans, before execution, allows the project manager to identify the reference phases and the sequence of activities that will be carried out in them, trying to optimize the related costs and the impact of the identified risks.

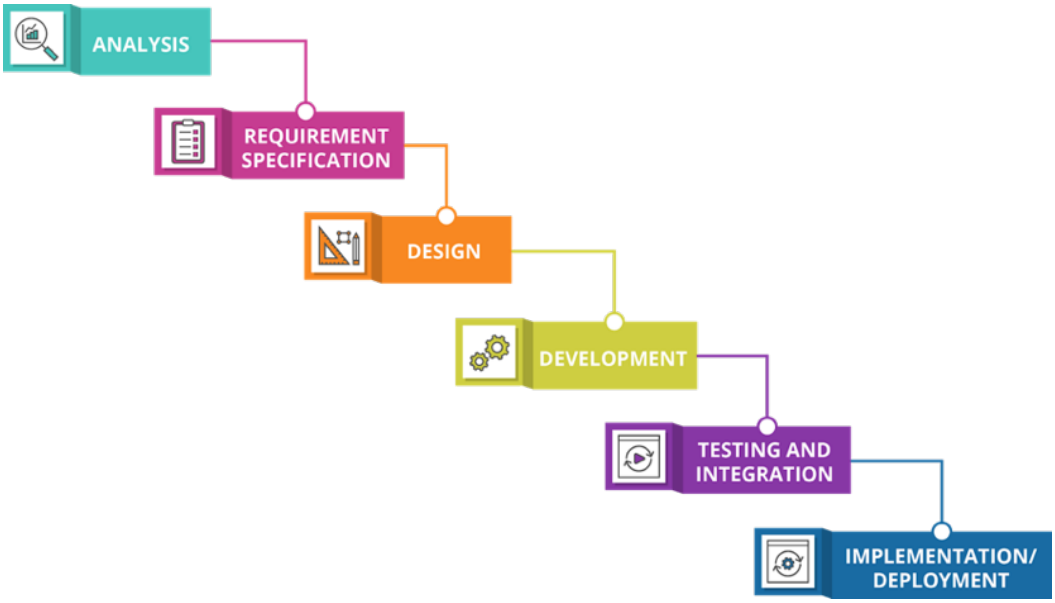


Figure 18- Predictive (Waterfall, Serial)

This approach is characterized by phases, each of which is focused on specific deliverables and in which different teams can be involved. The final result of each deliverable is generally not made available individually, but only in aggregate form at the end of the project.
Although the scope of the project is well defined at the beginning, it is not infrequently necessary to carry out actions to re-plan and review the budget of a project, with an often considerable effort.

In general, predictive lifecycles are more suitable for projects that have well-definable and well-detailed deliverables.

The Predictive Lifecycle is useful when the degree of uncertainty about what to do (scope) and how to do it is really minimal, combined with the fact that the value of the product in question can only be obtained at the end of production.	
Context	• **Work Package**(WP) detailable at a high level. • **People**also not totally dedicated to the project but involved in relation to the WPs. • **Few changes**in the specifications and in the surrounding conditions.
roles	• **Project Manager (PM)** • **Product Owner (PO)**, optional • **Scrum Master (SM)**, optional • **Architecture Owner (AO)** • **Developers**

	• **Specialists**
Events	• **Alignment meeting**, are held in relation to the provisions of the operational plan (Gantt) • **SAL**, periodic alignment event on the status of the workings. • **On Demand events**, are held when it is necessary to discuss specific issues
tools	• **Work Breakdown Structure (WBS)**, is the decomposition tool in the atomic activities of the product, project. • **Organizational Breakdown Structure (OBS)**, the hierarchical definition of the project team. • **RAM/RACI**, the responsibility assignment matrix. • **Gantt**, the most common operational plan visualization tool. • **Influence/Interest matrix**, for the classification of the stakeholders. • **Formal Documentation and Report**.
Metrics	• **Earned Value Management (EVM)**, allows active monitoring in relation to: costs incurred, value obtained and value planned. • **Milestone**evaluation of short, medium and long term objectives.

4.2 Iterative and Incremental Lifecycle

While they are often confused, there are specific differences between an iterative and an incremental lifecycle.

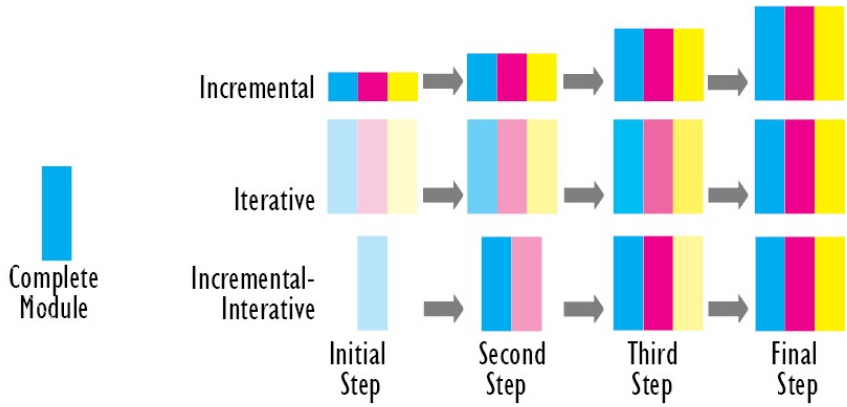


Figure 19– Iterative, Incremental and Iterative + Incremental

The goal of the iterative lifecycle is to achieve solution correctness.
Project scope is typically determined early in the project lifecycle, but time and cost estimates are regularly adjusted as the project team's understanding of the product increases.
The repetition (or iteration) of one or more project activities allows to deepen the understanding of the product by the project team (and possibly by the customer), allowing to progressively develop development and risk management plans. Iterations therefore allow deliverables to be developed through a series of repeated cycles, representing the key element used to drive value in the right direction. While there is the concept of “increments,” these are not distinct deliverables, but rather features or refinements of a single deliverable per phase.
A project with one or more prototypes, but with only one deliverable, is an example of an iterative approach

Dual to the iterative approach, the incremental one focuses on getting a solution quickly, developing its components incrementally. In an incremental project, new usable deliverables are continually created which produce an increase in product functionality.
Unlike a predictive design, having iterations that deliver finite elements allows requirements to be dynamic for those parts of the design that still need to be planned and executed.
An example of incremental development is the creation of the different sections of a web portal that can be used, significantly, independently.

In practice, iterative and incremental approaches are often used together, even implicitly, as there is significant overlap: an iterative lifecycle uses increments to some extent and an incremental lifecycle uses iterations to some extent.

In general, iterative and/or incremental lifecycles are preferred when:

1. The project is too complex to be planned from the beginning and the feedback (or lessons learned) are the main tool to improve the subsequent phases.
2. The scope and goals are subject to change as stakeholders learn more about the project itself.
3. Many of the decisions relate to the partial or incremental final results, unlike the predictive approach where the deliverables are available only at the end of the project.
4. The overall project may not yet be considered complete, but the increments provided so far can be used to deliver upfront to the customer.
5. It is decided to make concrete use of rolling wave planning, in which the next phase can take place while the project team is carrying out the current phase.

An iterative and incremental approach can be characterized by long-lasting iterations and detailed planning, specifically remembering a sequence of predictive cycles. When, on the other hand, an iterative and incremental approach is based on short iterations, often independent of each other and without extreme planning, one naturally passes to an Adaptive type lifecycle, also more commonly known as Agile.

4.3 Agile lifecycle

An Agile (or Adaptive) lifecycle is also simultaneously incremental and adaptive, but is characterized by short iterations (typically in the order of a few weeks) and the robust application of rolling wave planning. This allows this approach to address risks and review costs as the environment evolves.

The agile approach aims to make projects that involve high uncertainty and a high frequency of change more manageable, requiring a high degree of continuous involvement of the interested parties. At the heart of the action is the concept of time-boxing, i.e. the fact that each event/meeting/action normally has a well-defined time limit, also accompanied by a stable cadence.

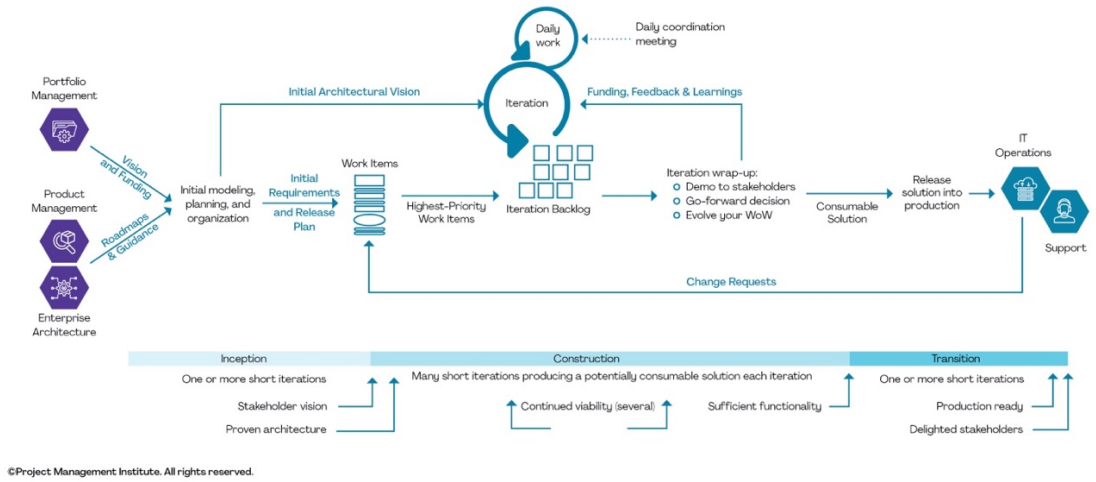


Figure 20- Agile Lifecycle (SME Disciplined Agile)

Each iteration is used to minimize the uncertainty of the requirements and have updated context plans, ending with an increment that is evaluated at its end. Related planning occurs at the beginning of the iteration itself, with the team determining how many of the higher priority requirements can be completed. Downstream of the iteration, the team identifies ideas to improve the process and any new features, going to affect the next iteration.

A project executed with an Agile approach can be seen as a series of short-term operational loops in which feedback is continuously collected, thus allowing for economically sustainable changes to be made, even at an advanced stage of project development.

Looking at the reference Lifecycle Toolkit, the following characterizing elements can be identified:

An Agile lifecycle is generally best for situations where the scope and requirements cannot be determined in advance in detail, or must remain flexible to adapt to changing factors characterizing the organization and the target market. It is indicated when it is possible to identify, prioritize and estimate the different functions to be implemented.	
Context	• Deliverables consist mainly of improvements or new features. • Deliverables can be identified, prioritized and estimated in a related planning phase. • The Team is mainly dedicated to one project.

roles	• **Project Manager (PM)** • **Product Owner (PO)** • **Scrum Master (SM)** • **Architecture Owner (AO)** • **Technical Experts and Developers**
Events	• **iteration**, defines the reference time window for developing a new increment. It is always accompanied by a reference goal. Iteration is also known as Sprint. • **Iteration Planning**, is the opening event of the iteration during which the deliverables to be created immediately are defined. • **Daily Stand-up**, is held every day, possibly at the same time, to allow the team to take stock of the situation. • **Iteration Review**, has the purpose of showing the product increment to the stakeholders and collecting their feedback. • **Retrospective iteration**, allows the team to question the validity of the process and how to improve starting from the next iteration. • **Refinement**, focuses on updating the Product Backlog so that it is constantly aligned with stakeholder expectations.
tools	• **Story Mapping**, allows you to illustrate the product features from the user's point of view and decompose them at the level of user stories to be included in the Backlog. • **Product Backlog**, is the prioritized list of deliverables that stakeholders expect from the product. It should always be associated with a product goal. • **Iteration Backlog**, is the set of user stories to build in the next iteration. It should always be associated with a specific goal. • **Iteration Board**, allows a graphical representation of the progress of the iteration activities. • **BurnDown/BunUp**, allows you to immediately and effectively view the progress of the activities during the execution of the iteration.

	• **Definition of Done**, is a sort of shortlist of things to do for a feature to be considered completed.
Metrics	• **Velocity**, a measure of the team's ability to develop what is foreseen in the Iteration Backlog. • **Story Points**, relative estimation method of the complexity of user stories. • **Minimum Time To Recovery (MTTR)**, records the earliest possible recovery for a particular problem.

When the targeted project deals with the evolution and stabilization of an already existing product, it is possible to make use of a specific declination of the Agile lifecycle from a Continuous Delivery perspective, defined by PMI DA as: "Continuous Delivery: Agile".

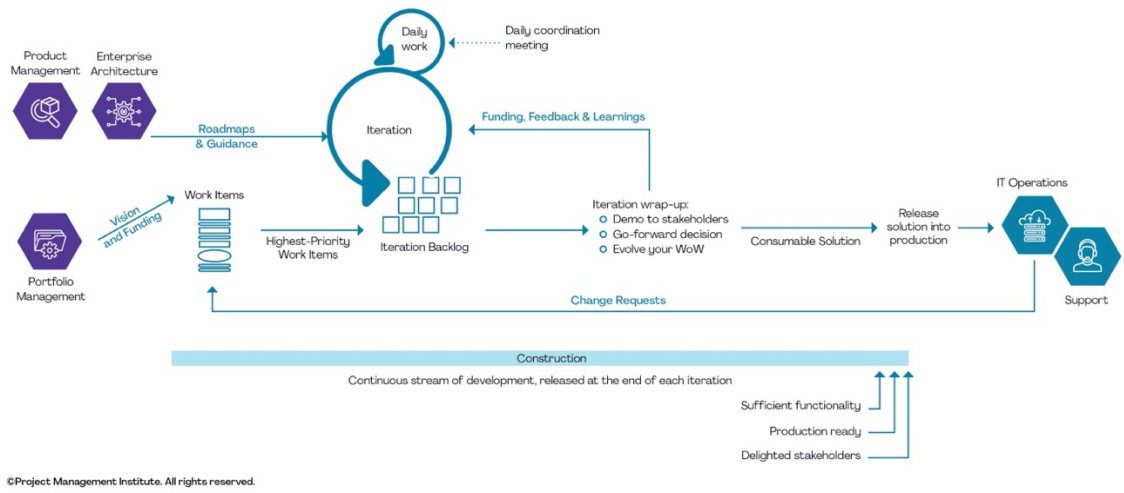


Figure 21- Continuous Delivery: Agile

While maintaining all the characteristics of the agile lifecycle, this specialization normally involves one-week iterations (maximum 2) and requires the automated delivery capability typical of a mature adoption of the DevOps philosophy[8], at each iteration.

As mentioned, a "Continuous Delivery: Agile" lifecycle is generally the best for situations where the goal is to stabilize a product from a specific quality point of view, without changing the features offered.

4.4 Lean lifecycle

Dually to the Agile approach, there is a second operating mode that falls under the adaptive hat, based on the management and optimization of the operating flow, i.e. Lean.

[8]https://www.pmi.org/disciplined-agile/process/disciplined-devops

While I share many key aspects of the Agile approach (such as, for example, rapid feedback and just-in-time plan definition), the concept of iteration is not explicitly present, instead the focus is on the continuous flow of value. When a new deliverable is requested, it can be immediately put into production, provided that the limit of concurrent activities for the team is respected, a limit defined as the Work-in-Progress limit (WIP limit). It goes without saying that even in the Lean approach, every opportunity is valid to discuss the progress of the process and the evolution of the project itself.

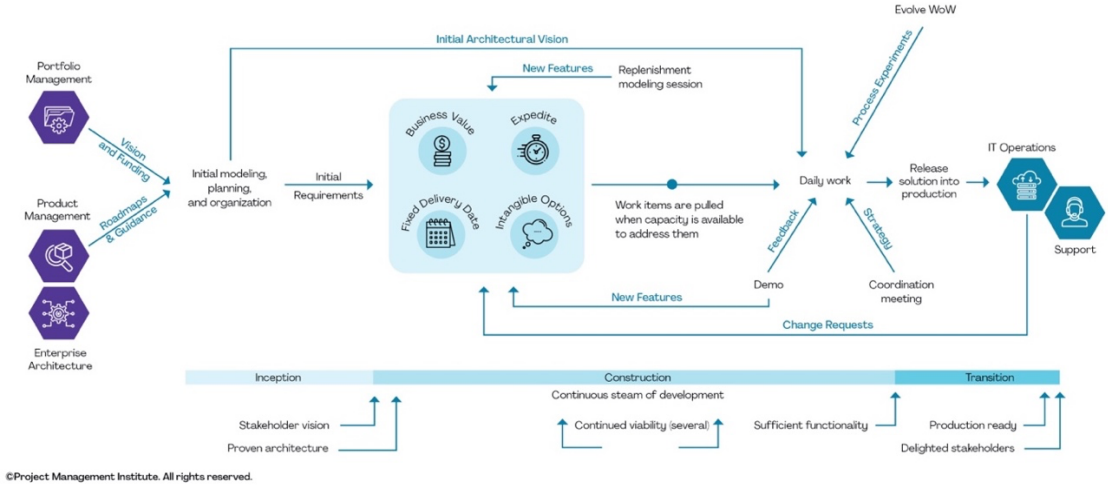


Figure 22– Lean Lifecycle (SME Disciplined Agile)

Unlike the Agile approach, there is therefore no iteration planning, but everything happens quickly (and organically) with respect to external events. Furthermore, the events/meetings do not have a predefined cadence, but I take place on "call" when necessary.

Looking at the reference Lifecycle Toolkit, the following characterizing elements can be identified:

The Lean Lifecycle is oriented towards managing a continuous flow of activities, releasing increments (or individual deliverables) when they are ready, without having a specific reference window, and striving to minimize the time between two releases. A typical example of using a Lean approach is in the context of research projects, when the deliverables to be made change at a very high rate, daily or even in the order of hours.	
Context	• Deliverables divided into small work items and the same. • Constant and small number of competing activities. • Teams dedicated mainly to a project. • On-demand events, i.e. held when deemed necessary.

roles	• **Product Owner (PO)** • **Scrum Master (SM)** • **Architecture Owner (AO)** • **Developers**
Events	• **Demos**, is held when the team has developed one or more deliverables that it deems useful to show to stakeholders, and possibly to the customer, to continue its activities, collect feedback or release into production. • **Retrospective**, allows the team to question the validity of the process and how to improve starting from the next iteration. • **Daily Stand-up**, is held every day, possibly at the same time, to allow the team to take stock of the situation. • **3 Buckets planning**, the "three buckets technique", allows for a subsequent refinement of the deliverables, up to the degree of detail that allows the weighing in load.
tools	• **Kanban board**, is the visual management tool of the work progress. It is based on a series of sequential processing states that have limited capacity (WIP limits). • **Classes of Services**, each new deliverable is categorized in relation to a specific "Class of Service". Overall, classes of service allow you to manage processing priorities and changes in development. • **Control Chart**, it is a graph that presents maximum/minimum values (calculated on a statistical basis) which allow to intercept any divergent work situations, especially when continuous. • **CFDs (Cumulative Flow Diagrams)**, allows you to view and analyze the progress of the processes in relation to the status of the Kanban Board and the reference metrics. • **Definition of Done**, is a sort of shortlist of things to do for a feature to be considered completed.
Metrics	• **WIP (Work in Progress)**, number of activities in concurrent processing.

	• **Lead Time**, total time for processing a request, from when it is accepted (it enters a Class of Service) to when it is made available to the customer. • **Cycle Time**, total processing time, i.e. from when it enters the "process in progress" state until it is marked as "completed". It is a sub range of the lead time. • **Ideal Days**, calculation of the time/effort attached to the development of a deliverable in an absolute sense, therefore not linked to the time/days. • **Minimum Time To Recovery (MTTR)**, records the earliest possible recovery for a particular problem.

Exactly as happens for the Agile lifecycle, also for the Lean one it is possible to identify a declination in a Continuous Delivery key, which provides for a continuous release into production of what has been created: "Continuous Delivery: Lean".

The goal is to minimize the timedelivery, even several times a day, using automation tools and standardized processes over time.

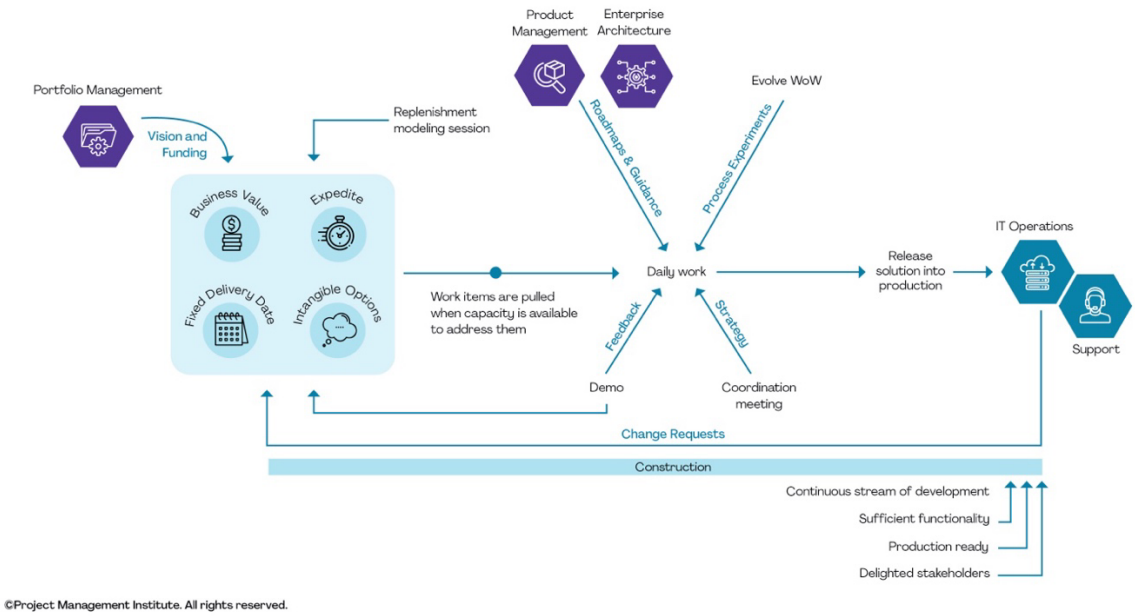


Figure 23– Continuous Delivery: Lean Lifecycle (SME Disciplined Agile)

A "Continuous Delivery: Lean" lifecycle is generally the best for situations in which the goal is to allow the customer, or stakeholders in general, an immediate response to their requests. Again, a mastery of DevOps practices and techniques is required.

4.5 Lifecycle Exploratory

The lifecycle**Exploratory**Andinspired by Lean Startup[9], designed to validate the sustainability of a new idea and/or hypothesis thanks to a series of Minimum Viable Products (MVP) and a specific implementation of the Build-Measure-Learn cycle.

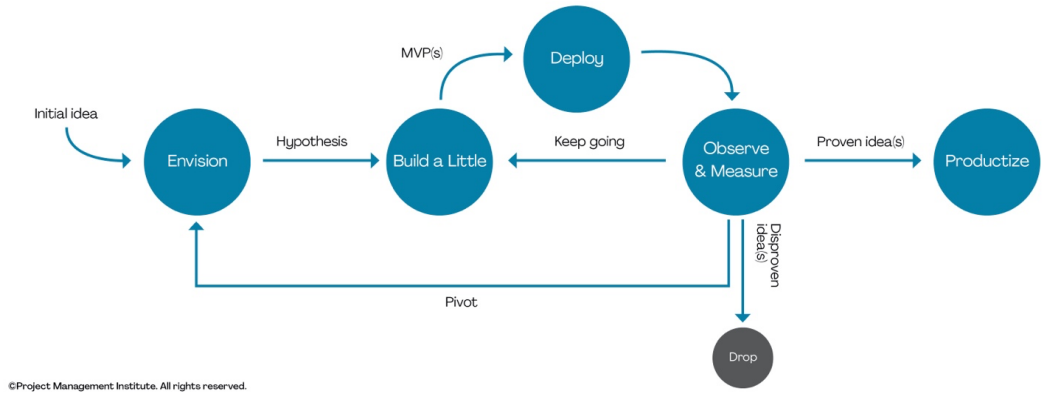


Figure 24- Exploratory (Lean Startup) Lifecycle

The goal is to reduce the range of uncertainty and think about the sustainability of the initiative through six reference actions:

- **Envision**, it is about exploring the new idea and formulating the related hypotheses.
- **Build a Little**, quickly develop what is necessary to validate your hypotheses: Minimum Viable Product (MVP). It is not important to focus on the technical quality, but to be extremely fast and able to highlight the peculiar characteristics.
- **deploy**, the MVP must be quickly tested in order to prove its value.
- **Observe & measure**, by observing the reactions of the people involved in the test, it is possible to realize if what has been hypothesized is in the right direction or if it is necessary to change.
- **Cancel**, if after having tested the main aspects and exploiting the pivots it is not possible to achieve the objective, it is advisable to consider the idea/hypothesis as "not validated" and discard it.
- **Productize**if, on the other hand, validation is achieved, it is possible to move on to the engineering phase through one of the other lifecycles presented.

The Exploratory Lifecycle is very useful when the project (or the deliverable) has a high uncertainty, such as, for example, a new unexplored market or a new product

[9]http://theleanstartup.com/principles

The goal is to gain maximum flexibility in adapting the solution as it is developed, testing the Assumptions and strategy with clear go/no-go criteria, experimenting the idea based on what is learned.	
Context	High general uncertainty about how and what.
roles	If you are within a Project, you inherit the roles of the reference lifecycle(s). In case you are in a real start-up action, you prefer to be on par in discovery. However, the PM can be present as a guide and facilitator of the team.
Events	Continuous review, alignment and feedback generation activities
tools	A Lean/Kanban style approach is used, inheriting all the tools previously described in the specific lifecycle.
Metrics	A Lean/Kanban style approach is used, inheriting all the metrics previously described in the specific lifecycle.

4.6 Lifecycle Program

The lifecycle**Program**(also known as Team of Teams) describes how to organize the development of a deliverable (or project) when there is a need for "horizontal scaling", i.e. when multiple teams work on it in parallel. The solution of having multiple teams in parallel is usually preferred over having one large team.

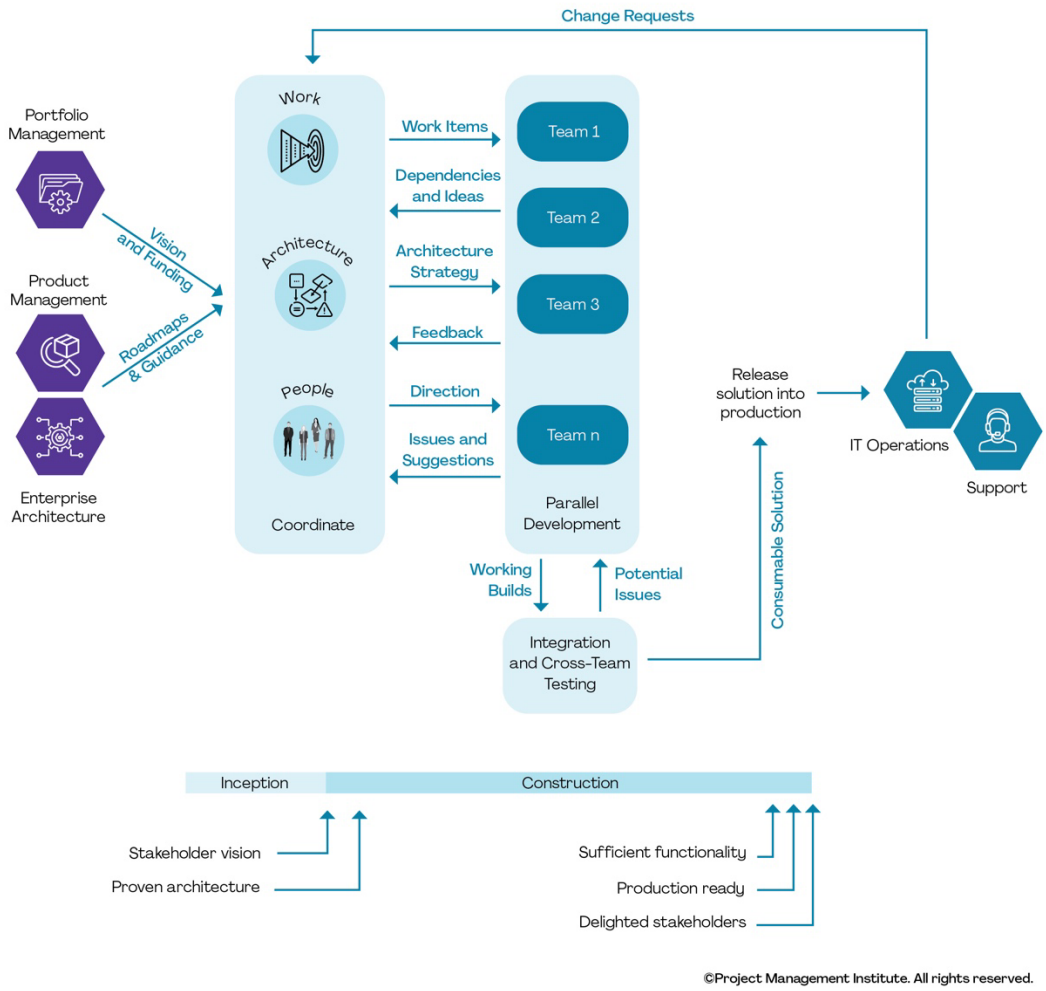


Figure 25- Program Lifecycle (SME Disciplined Agile)

Generally, the single team in turn can identify its own lifecycle by exploiting a bottom-up structure to align:

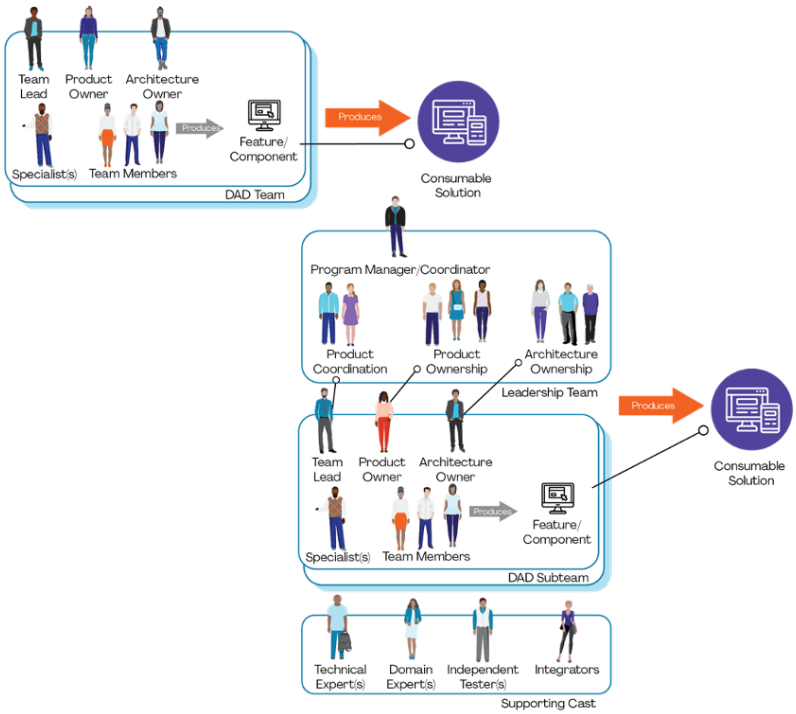


Figure26- Bottom-up alignment (Disciplined Agile)

The Lifecycle Program is ideal when there are multiple teams working on the same project/product, promoting a predominantly bottom-up organizational model in which agile or lean teams coordinate without weighing down the organizational flow.	
Context	One needs to have dozens of people on a project with the willingness to maintain the necessary operational agility.
roles	Depending on the type of lifecycle within the team, the roles of the Agile or Lean lifecycle are inherited and, optionally, Waterfall. In addition, 3 coordination leadership teams are created: • Product Coordination, formed by all MS • Product Ownership, formed by all POs • Architecture Ownership, formed by all AOs In addition you have a Program/Coordinator/Project Manager of reference.
Events	Each team uses those specific to the chosen lifecycle, to which are added: • **Integration and Cross-team testing**: they are the events, or rather the actions, of integration of the overall work.

	• **System Review**, in which what is produced by all the teams is discussed with the stakeholders. • **Release in Production**, i.e. the release (or delivery).
tools	Each team uses those specific to the chosen lifecycle, with the use of tools from the Lean/Kanban world for coordination and alignment.
Metrics	Each team uses those specific to the chosen lifecycle, with the use of metrics from the Lean/Kanban world for overall assessment.

4.7 Hybrid lifecycle

A hybrid lifecycle is a combination of several lifecycles.

It is up to the team to determine what the most appropriate mix is, ensuring they have the flexibility to best achieve their intended goals.

In a hybrid lifecycle, instead of associating a single lifecycle to the entire project, a mix of different lifecycles is created in relation to the characteristics of the expected deliverables, analyzing their specificities and the attached release strategy. In any case, deliverables are usually shared with the customer (and/or other stakeholders) to show progress and get valuable feedback.

The deliverables can also be developed sequentially and/or in parallel, and a project can have both cases. With sequentially developed deliverables, development of the next one begins only after the previous one has been completed and accepted. This approach is recommended when the deliverables are functional to each other, when there are strong dependencies and when there are no resources to implement parallel development.

In the case of deliverables developed in parallel, the processes can overlap (several teams) or be carried out simultaneously (a single team). This type of approach is recommended when the deliverables are independent of each other (or the relative dependency is very minimal) and there are sufficient teams and resources.

5 Modern PM Fast Prototyping

The ModernPM Fast Prototyping (practice inherited and extended from the AgileConstellation funnel) allows to quickly define the methodological aspects related to a specific project.

Figure 27- Modern PM Fast Prototyping

Compared to what is inherited from the basic practice[10], 4 new reference bubbles are added:

- **Vision**, i.e. the identification of the aspects characterizing the specific project;
- **Scope**, the scope of the project, i.e. the activities and deliverables to be implemented;
- **Complexity**), analysis of the complexity and uncertainty of the project;
- **Lifecycles Bouquet**, the construction of the reference lifecycle, which can be unique for the whole project or built from a hybrid model in relation to the deliverables.

5.1 Vision

The Vision is the starting point of every new initiative, and, in the case of a project, the relative definition allows us to answer the fundamental question: is this project worth starting and is it sustainable for the company?

[10]AgileConstellation Manifesto – agileconstellation.info

To work on this theme, the Project Canvas is used, which has the objective of "bringing to the table" all the primary players who revolve around a project, in order to discuss the related reference characteristics.

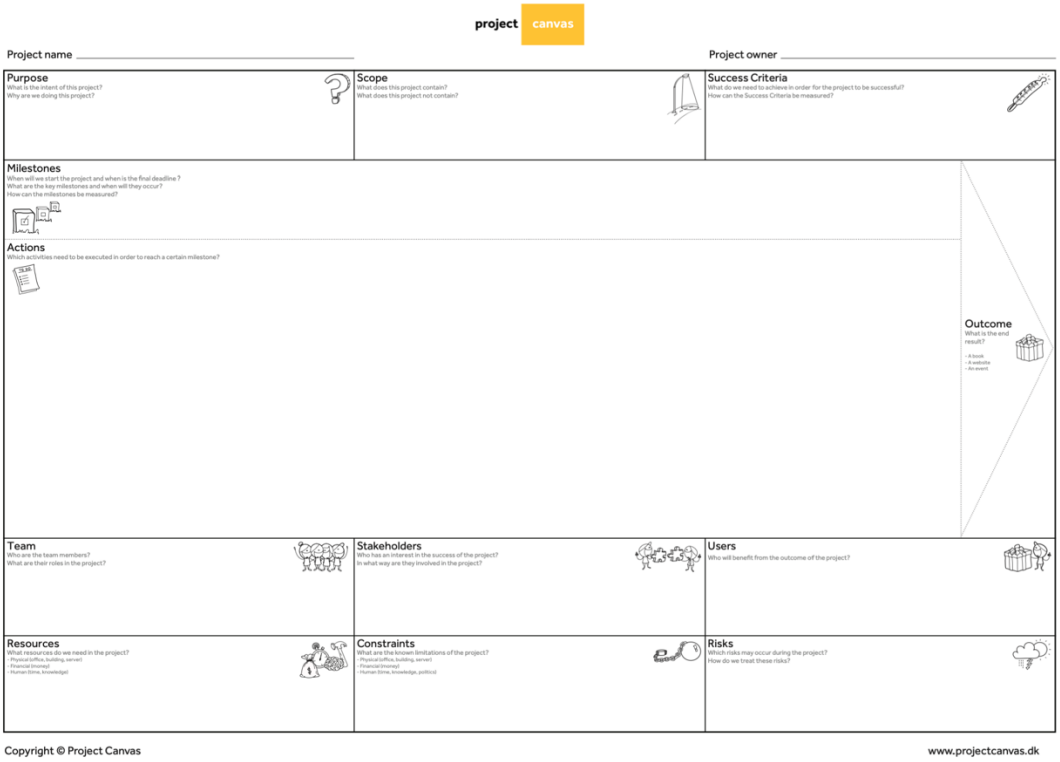


Figure 28- Project Canvas template (projectcanvas.dk)

In summary, it is a visual tool that allows you to focus the discussion on a set of fundamental elements, improving the communication of the project team itself:

- **Purposes:**the objectives and motivations of the project.
- **Scopes:**what the project must contain and what it must not contain, at a high level (in the following paragraph we will see how to detail this aspect).
- **Success Criteria:**how to establish the "success" of the project.
- **Milestones:**the moments of verification of the project.
- **Actions**: the actions necessary to achieve the goal.
- **Outcomes**: the expected benefit.
- **Team**: Project team members and their roles.
- **Stakeholders: the**project stakeholders.
- **Users:**the "consumers" of the project deliverables, or of the resulting product.

- **Resources:**the resources necessary for the development of the project: budget, people, equipment, software, etc.
- **Constraints: i**constraints, limitations and barriers of the project.
- **Risks: i**risks and related mitigation actions.

5.2 Scope definition

For the analysis of the project deliverables, two reference tools can be used: the Work Breakdown Structure (WBS), with the definition of the annexed reference dictionary, and Story Mapping.

5.2.1 Work Breakdown Structure

In the predictive approach (waterfall, serial) for the definition of activities, as well as the breakdown of a project/product into its components, the Work Breakdown Structure, WBS, is typically used.

The WBS allows you to set up the planning process, accompanying the whole project along the development and monitoring process.

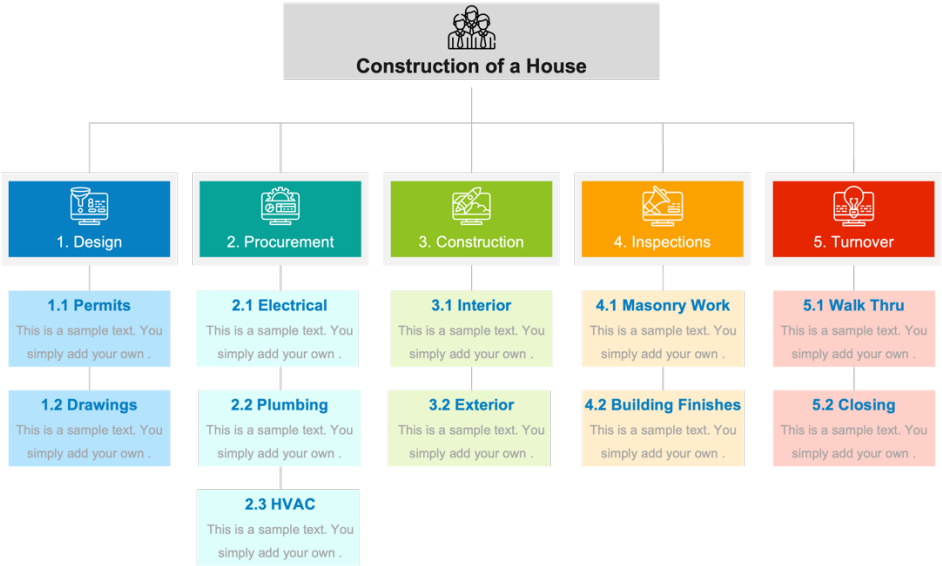


Figure 29- Example of WBS

Starting from the Vision, the WBS allows you to detail the scope of the project (and/or product) through its tree specialization (decomposition) which ends at the level of maximum detail with the Work Packages (WP).

The WPs are, therefore, the different elements that make up the project (product) and are accompanied by all the detailed information that allows it to be implemented, to which specific team members or even a single specialist can be linked.

Furthermore, the WBS allows for the structuring of the activity network, allowing for the scheduling and definition of the implementation times, as well as a specific correlation of the associated budget.

A constructive approach to the WBS is to consider it for what it is: not the plan, but a plan representation tool. In this world there is no risk of considering it as the "north star" but a common aid to the governance of the project.

5.2.2 Story Mapping

When moving towards a more adaptive (agile) approach, there is a tendency to focus more on features that can be quickly released to consumers, representing, atomically, a new “piece” of value. In this context, Stories (or User Stories) have somewhat become the de facto standard for establishing an agreement with the project team, replacing the Work Packages seen previously.

However, in order to identify the overall picture, similarly to the WBS, Story Mapping is often used which allows you to create a "map", easy to visualize, structured, shared and contextual of the stories.

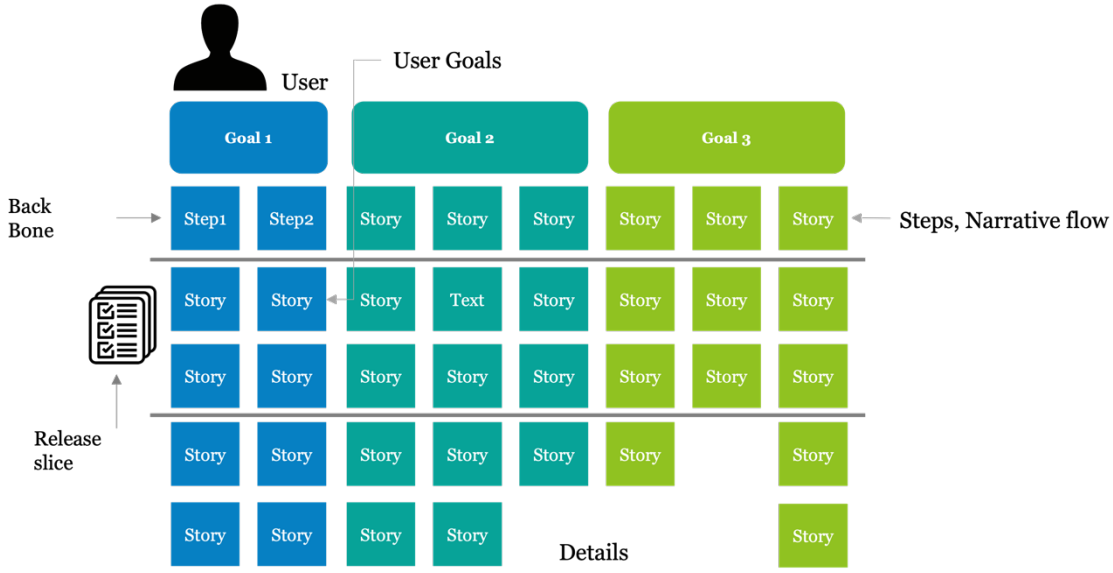


Figure 30- Example of Story Mapping

The map definition process can be summarized in three basic actions:

1. **Identify high-level features/activities**: This action can be performed by examining the user interaction (user journey) with the relevant part of the system for which you want to create the map. Activities are also known as Goals (also Objectives or Themes). It is essential to be able to identify how the user uses our system, in order to define the high-level

requirements and share them, for confirmation, with interested parties. Once the objectives have been defined, they must be arranged in chronological order, from left to right, as if describing the user's journey through the system.

2. **Document the steps**: it is about specifying the steps necessary to achieve the goal. Steps are often labeled “epic” as they tend to encompass related groups of user stories (or activities) and, collectively, form the backbone of the story map, providing much of its structure.
3. **Identify details**: At this point, the actual user stories are identified, as well as the required detail activities. Stories should always be developed in collaboration with stakeholders, enabling the team to implement the 'right thing' in the 'right way'.

Once the map has been created, it is possible to start identifying the different atomic areas of the product, applying the appropriate prioritization strategies for iterative and incremental development.

5.3 Complexity

Once the scope has been defined, the complexity of the project, or of a specific deliverable, is analyzed in order to identify the most effective lifecycle, using Stacey's Diagram.

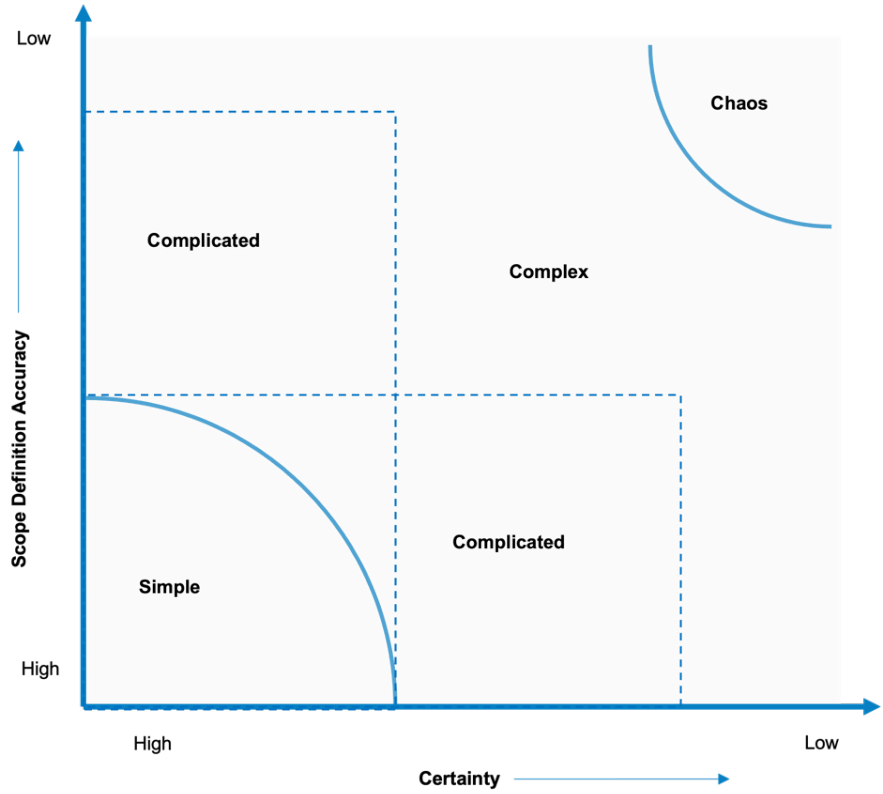


Figure 31- Stacey's Diagram

In it, on the Y axis, we have the level of confidence with the scope (Scope Definition Accuracy), while on the X axis, we have the uncertainty (Uncertainity), or the degree of uncertainty on the main aspects, such as: technology , skills, etc..

In relation to the possible intersections obtainable, it is possible to outline the following classification areas of the deliverables (or, possibly, of the project as a whole) and identify the most suitable lifecycle(s):

- **Simple (simple)**, bottom left: you know pretty much what to do and how to do it. In this case the predictive approach is the most suitable.
- **Complicated (complicated)**, in this case there are 2 cases: at the top left there are unclear requirements and the recommended approach is the iterative one. If, on the other hand, you are positioned at the bottom right, therefore the operational uncertainty increases, it is preferable to resort to an incremental lifecycle.
- **complex**(complex), medium-upper area. In this case there is a strong uncertainty about the requirements, as well as how to carry out the related activities, for which it is preferable to resort to an Agile or Lean lifecycle.

As repeatedly underlined, a project can be characterized by several lifecycles during its execution, depending on its typology, time, estimated resources and other characterizing factors.

Going to superimpose the lifecycle application spectrum with the Diagram ofStacey, therefore, the following scenario is obtained:

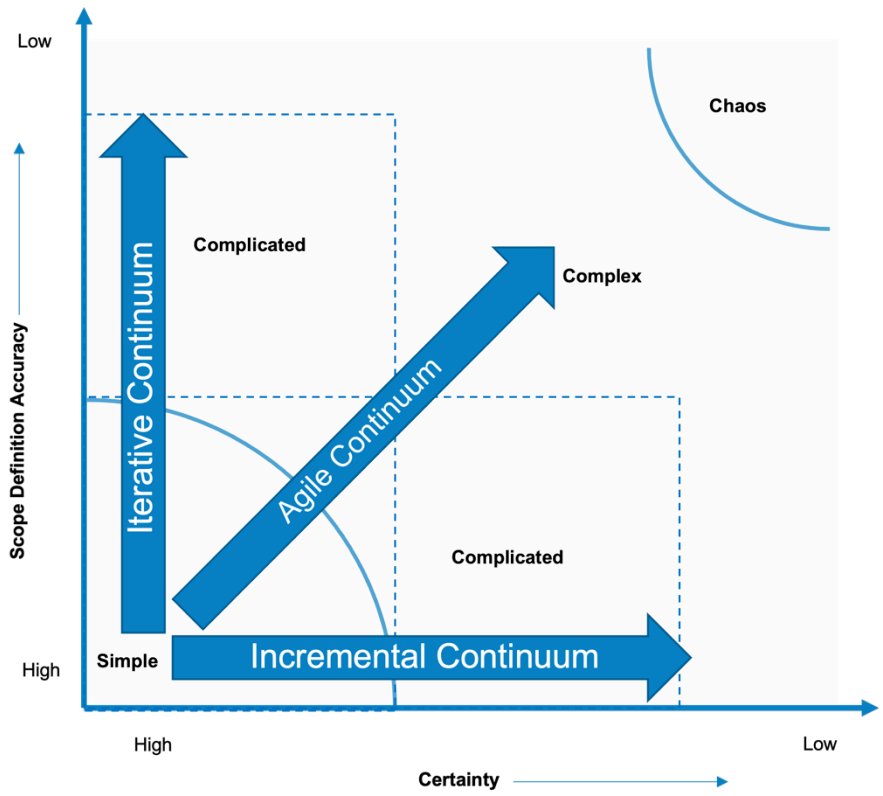


Figure 32- Stacey's Diagram and lifecycles

To make the most appropriate choice, it is therefore necessary to analyze the two dimensions using evaluation factors that are as objective as possible.

5.3.1 Scope Definition Accuracy

In relation to the various deliverables identified (Work Packages or Epics/Stories), or, in the extreme case, for the entire project, specific factors will have to be considered such as:

- **Numbers of Stakeholders**. The number of stakeholders involved in the project, both internal and external to the organization.
- **Stakeholders Engagement level**. The level of stakeholder engagement in terms of interest/power.
- **Assumed Frequency of Change**. The presumed stability of the scope, i.e. how stable the requirements are expected to be over time.
- **Project Duration**. The duration of the Project in terms of time.

These factors will be lost to obtain a specific overall rate of confidence level with the specifications of the deliverables:

Factors	Levels (1…5)
Numbers of Stakeholders	

Stakeholders Engagement level	
Assumed Frequency of Change	
Project Duration	
average rate	
Rates: from 1 to 2, Low level of Agreement	
installments: 3, Medium level of Agreement	
installments: from 4 to 5, High level of Agreement	

5.3.2 Uncertainty

For the analysis of the uncertainty characterizing the deliverables (here too it is possible to refer, if appropriate, to the entire project), reference can be made to the tactical scaling factors identified by Disciplined Agile[11]:

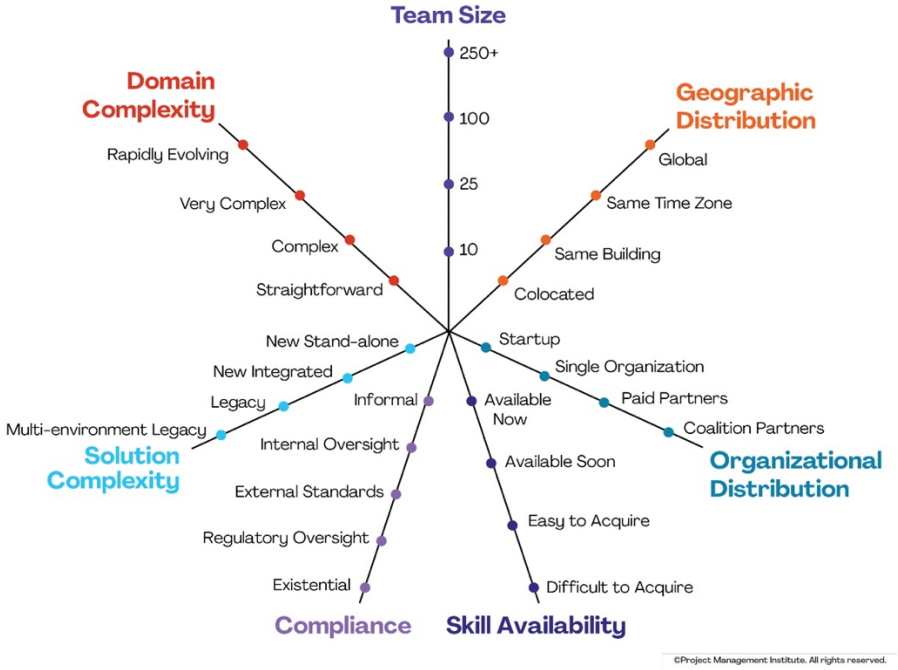


Figure 33- Disciplined Agile Factors for Tactical Scaling

[11]https://www.pmi.org/disciplined-agile/agility-at-scale/tactical-agility-at-scale/scaling-factors

- **Team size**. Teams can range in size from two people to up to several hundred. Larger teams (also referred to as teams of teams) are generally formed to tackle more complex problems and challenges of greater impact and duration over time.
- **geographic distribution**. Teams can be co-located, with team members and key stakeholders sharing the same workspace. They can be on the same floor in a single building, across multiple floors, some of the team members can work in different buildings, some can work from home and some can even work in different countries.
- **Organizational Distribution**.The origin and methods of involvement of the people functional to the project can be different. The simplest situation is to have all team members from the same group/division within a single organization, which happens, for example, in a startup or new product team within an enterprise.
- **Skill Availability**.The team needs the right people with the right skills to deliver the results you hired. At the ideal end of the spectrum you have skilled people "sitting on the bench" waiting to move forward, at the other end it could take many months, and potentially a lot of money, to build the team you need.
- **Compliance**.There are generally two types of compliance: self-imposed, such as choosing to be CMMI or ITIL compliant, and regulatory, which is potentially more difficult to meet.
- **Domain Complexity**.Domain complexity (also problem space) can vary widely depending on the goals. An information website, for example, is generally simple, while an e-commerce site is more complicated, just as an air traffic control system is decidedly more complex.
- **Solution Complexity**.The solution can be characterized by different levels of intrinsic complexity. At the "simple" end of the spectrum we can identify the new standalone solutions, built with new technologies. Things become more difficult if, for example, you need to leverage what was previously in place, such as legacy software and pre-existing business services.

Also in this case, in relation to the level of relative complexity of the various scaling factors, an overall deliverable (or project) indicator will be obtained which will allow to identify the relative level on the diagram, as well as the positioning in a specific quadrant thanks to the intersection with the previous level of accuracy.

Factors	Levels (1…5)
Team size	
geographic distribution	
Organizational Distribution	

Skill Availability	
Compliance	
Domain Complexity	
Solution Complexity	
average rate	
Rate: from 1 to 2, Low level of Uncertainty	
installments: 3, Medium level of Uncertainty	
installments: from 4 to 5, High level of Uncertainty	

5.3.3 Modern PM Decision Tree

Once the levels of the two dimensions of the deliverable under analysis have been identified, it is possible to use the ModernPM Decision Tree to choose the most suitable lifecycle for its development:

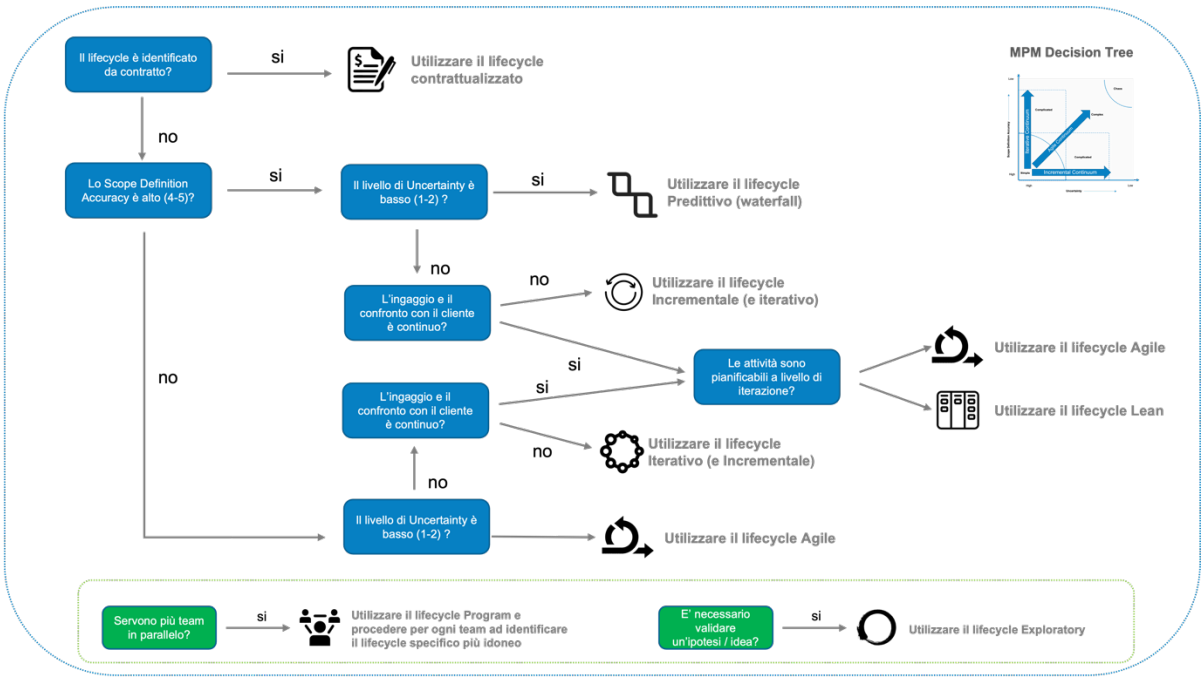


Figure 34-Modern PM Decision Tree

If the decision tree is used at the project level, the path followed leads to the choice of the single project lifecycle. If, on the other hand, the tree is retraced for each deliverable, once the analysis

is complete, a hybrid model will be obtained, generally falling within one of the cases shown in the following figure:

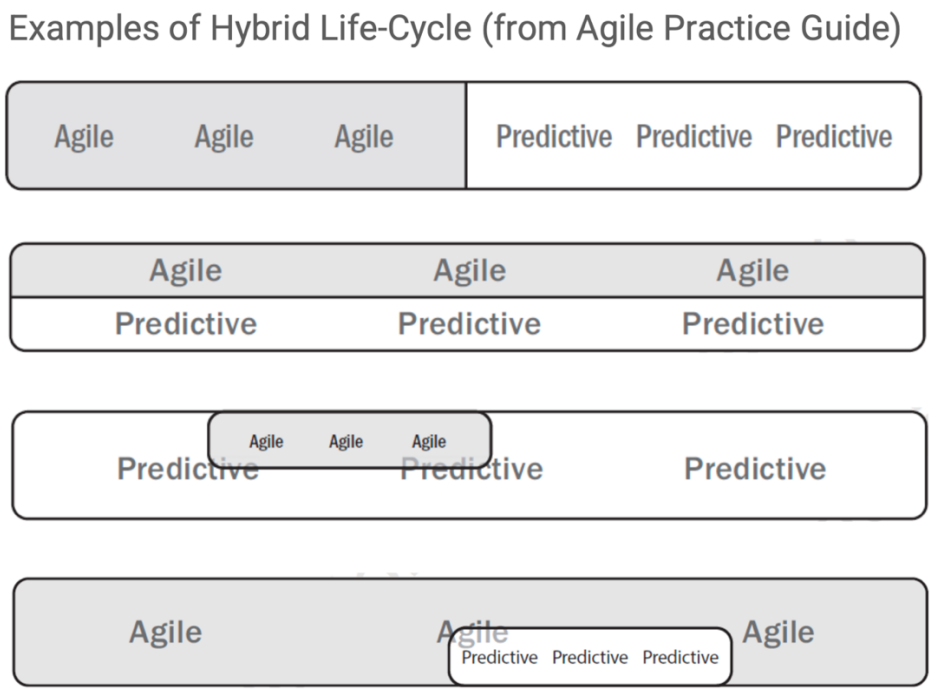


Figure 35– Examples of Hybrid Lifecycle (Agile Practice Guide)

5.4 Lifecycles Bouquet

Once the context analysis through the three bubbles has been completed, it is possible to proceed to validate and formalize the overall approach for the management and execution of the project, i.e. the Lifecycles Bouquet.

This model is structured in the terms already seen in the description of the individual lifecycles, appropriately mixing them together:

- **Lifecycles**, chosen as indicated above. You can have only one lifecycle per project, or a mix of them (hybrid situation) in relation to the characteristics of the deliverables to be produced.
- **roles**, i.e. the roles needed in relation to the selected lifecycles.
- **tools**, the specific toolbox.
- **Metrics**, the metrics necessary to measure operational performance and apply the necessary continuous improvement activities.

6 Call back

6.1 PMI Disciplined Agile

The PMI Disciplined Agile toolkit (PMI DA toolkit) looks at the entire organization, supporting its journey towards Business Agility. The areas addressed by the toolkit (defined Blades) are shown in the following poster:

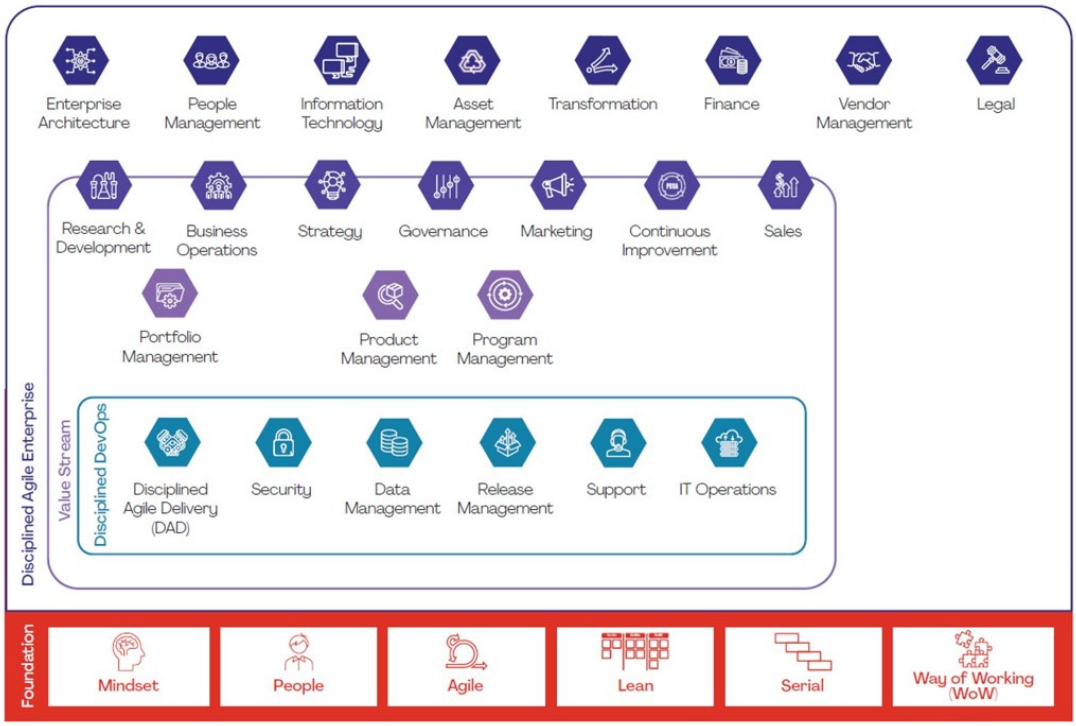


Figure 36- The Disciplined Agile Toolkit

As you can see, there are 4 layers that impact the entire organization: Foundation, Disciplined DevOps, Value Stream and Disciplined Agile Enterprise. In particular, the Foundation layer brings together all those aspects that are the beating heart of the toolkit, aligning it with the reference mindset.

Each layer encompasses more“Blade” (areas),presenting a coherent collection of options (especially practices and strategies) to be applied in relation to the specific context. Each Blade addresses an "atomic" area, such as, for example: Data Management, Continuous Delivery, or Portfolio Management. The choice of the term "Blade" is conceptually linked to the counterpart of the "Server Blade" to highlight the possibility of updating, changing or even replacing it with extreme agility, in relation to the evolution of the context in which one is operating.

Particular attention should be paid to the Value Stream layer, which canò be seen as the dynamic soul of DA, as it embroiders operations on the needs of stakeholders, or customers, embracing a

clearly Lean mindset that suggests focusing on workflow management and not on how to manage people or the micro-activities.

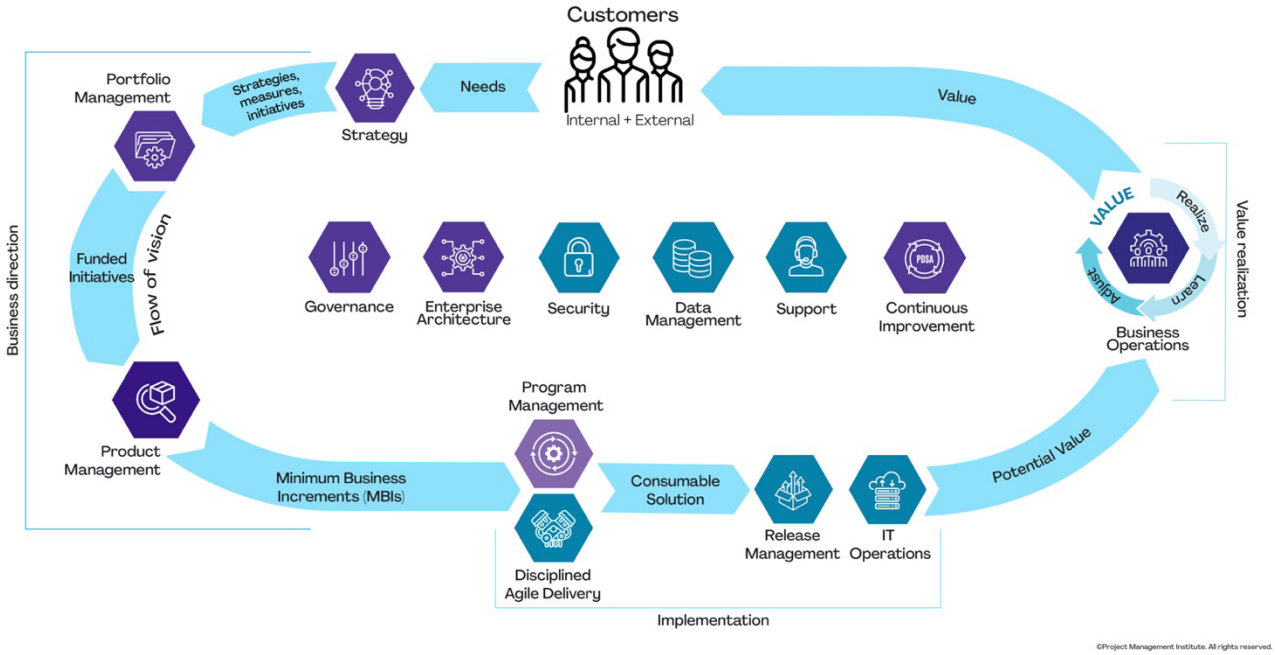


Figure 37- DA Value Stream

The Value Stream focuses on the development of MBI (Minimum Business Increment), ie the minimum amount of value to be built, distributed and used so that the initiative actually makes sense from a business point of view.

Looking at the more operational side of project management and development, the previously presented lifecycles are proposed in the Blade Disciplined Agile Delivery (DAD), characterized by three ideal progress phases:

- **Inception**, the initialization phase of the project, in which the fundamental elements for its success take shape and aspects such as Risk and Sustainability are evaluated;
- **Construction**, the implementation phase, i.e. where the solution is actually implemented;
- **transition**, the phase that deals with the aspects of Deployment.

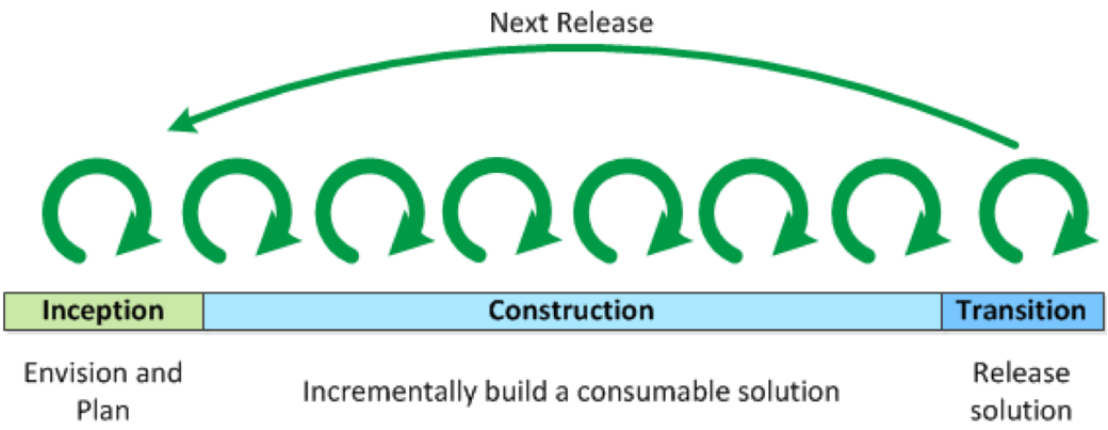


Figure 38- Stages of DAD

If the use of stages may seem something "strange" within the Agile context, it must be highlighted that they are not intended as stage-and-gate, but as "purpose containers" that do not limit any review of the decisions made .

Their goal is, therefore, to focus, at different times, on specific process goals that are typically required in the development of a product (project), goals that are made explicit by the attached process goal diagrams.

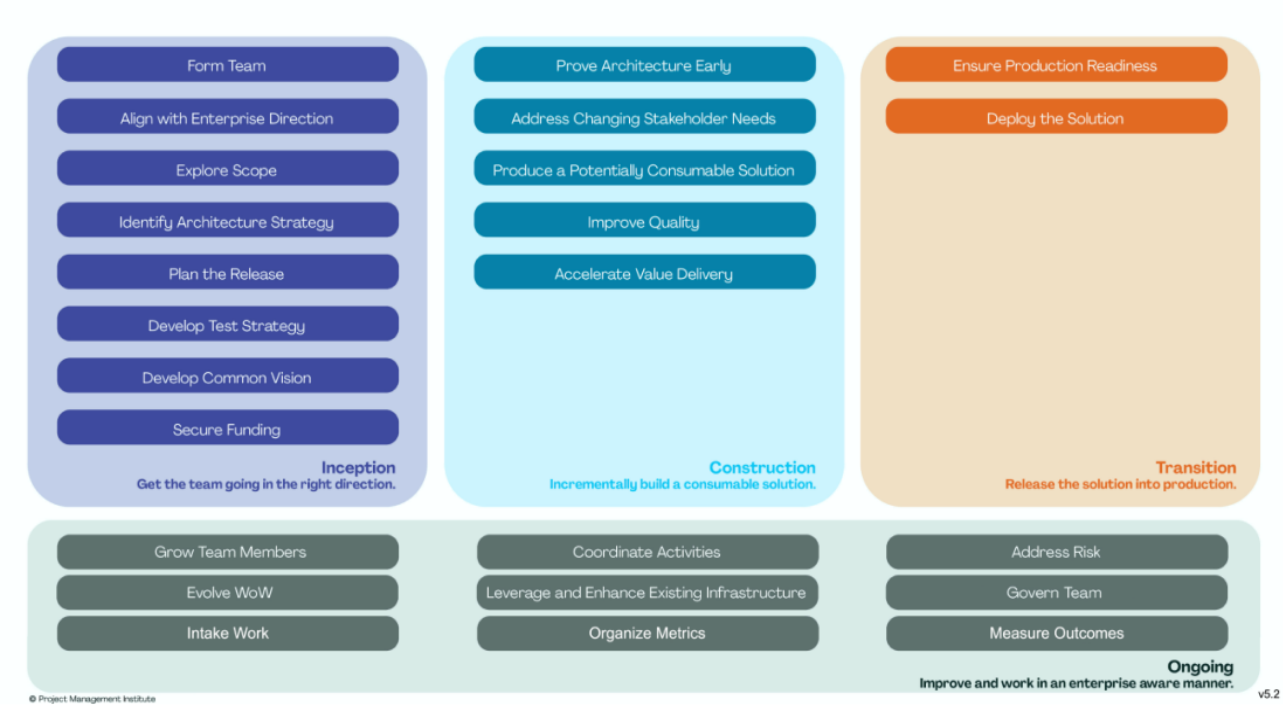


Figure 39- The process goals of Disciplined Agile Delivery (DAD)

The previous figure also shows the goals grouped in a "virtual phase" called Ongoing: these are transversal objectives that accompany the team throughout the entire creation of the product.

6.2 PMI PMBOK

Starting from PMBOK 7, the most appreciated project management body of knowledge has undertaken an evolution in strong discontinuity with the past.

The most evident change is the new approach based on principles rather than on processes, which gives life to a new basic philosophy: project management standards must focus on outcomes and therefore on the generation of tangible value for the customer.

The operational logic that distinguishes this new edition is that of: "Begin with the End in Mind"[12], which highlights how it is always necessary to ask yourself why you want to achieve something, and not focus exclusively on managing the operational aspects and on generating outputs which may not generate true value.

The changes made compared to the previous editions are really many and deserve the appropriate insights that go beyond this brief description. In order to make a quick summary, however, some key elements can be identified:

	PMBOK ® Guide Sixth Edition	PMBOK ® Guide Seventh Edition
General approach	• Prescriptive, not descriptive • Emphasis on the how, not the what or why	• Principles to guide the mindset, actions and behaviors reflected in the bodyof knowledge for project delivery, agile, lean, customer-centered structure, etc…
Basis for design	• Specific processes convert input and output using tools and techniques • Process focus and more compliance driven orientation	• Domains of interdependent and related areas of activity, both with performance results and with an overview of the most used tools, techniques, artifacts and frameworks.
Project environment	• Project environment, internal and external	• Internal and external project environment
Reference targets	• Mainly project managers	• Anyone involved in a project, with specific focus on team members and roles, including project leader, sponsor, and product owner
Levels of change	• Incremental revision based on previous editions	• Principles-based review to reflect the entire value delivery scenario
Customization guide	• Tailoring references, but no specific reference guide	• Specific guide to adaptation

[12]Stephen Covey - 1989

The focus therefore passes explicitly from the outputs (the what) to the outcomes (the why) of the project, effectively making the processes (the how) tools for achieving the ultimate goal of creating value for the stakeholders and not considering them the end of the project.

Furthermore, the processes are no longer part of the core of the PMBOK, but are instead part of the PMI Standards + Platform, allowing to organically develop the duality between a predictive approach and an adaptive approach, thanks to the focus on agile project management methods and approaches , which have been adopted by many organizations in project management over the years.

7 Conclusion

As we hope has emerged from reading the paper, the objective of its content is to present a structured path, and a series of operational tools, to discover how Project Management today is capable of accompanying the most complex challenges in an adaptive logic.

If you are interested in deepening and in the concrete application, you can contact us at the e-mail address info@agileconstellation.info or through our social channels that you find listed on the official website.

AgileBIM is distributed under License Creative Commons Attribution - Noncommercial - No Derivatives 4.0 International.

The authors cannot be held responsible in any way for the use of what is reported in this document, in the annexes and in the relative digital channels. The user has total responsibility for his own work and frees them from any type of direct or indirect duty.

Based on work available on info@agileconstellation.info

Permissions beyond the scope of this license may be available at AgileConstellation.info