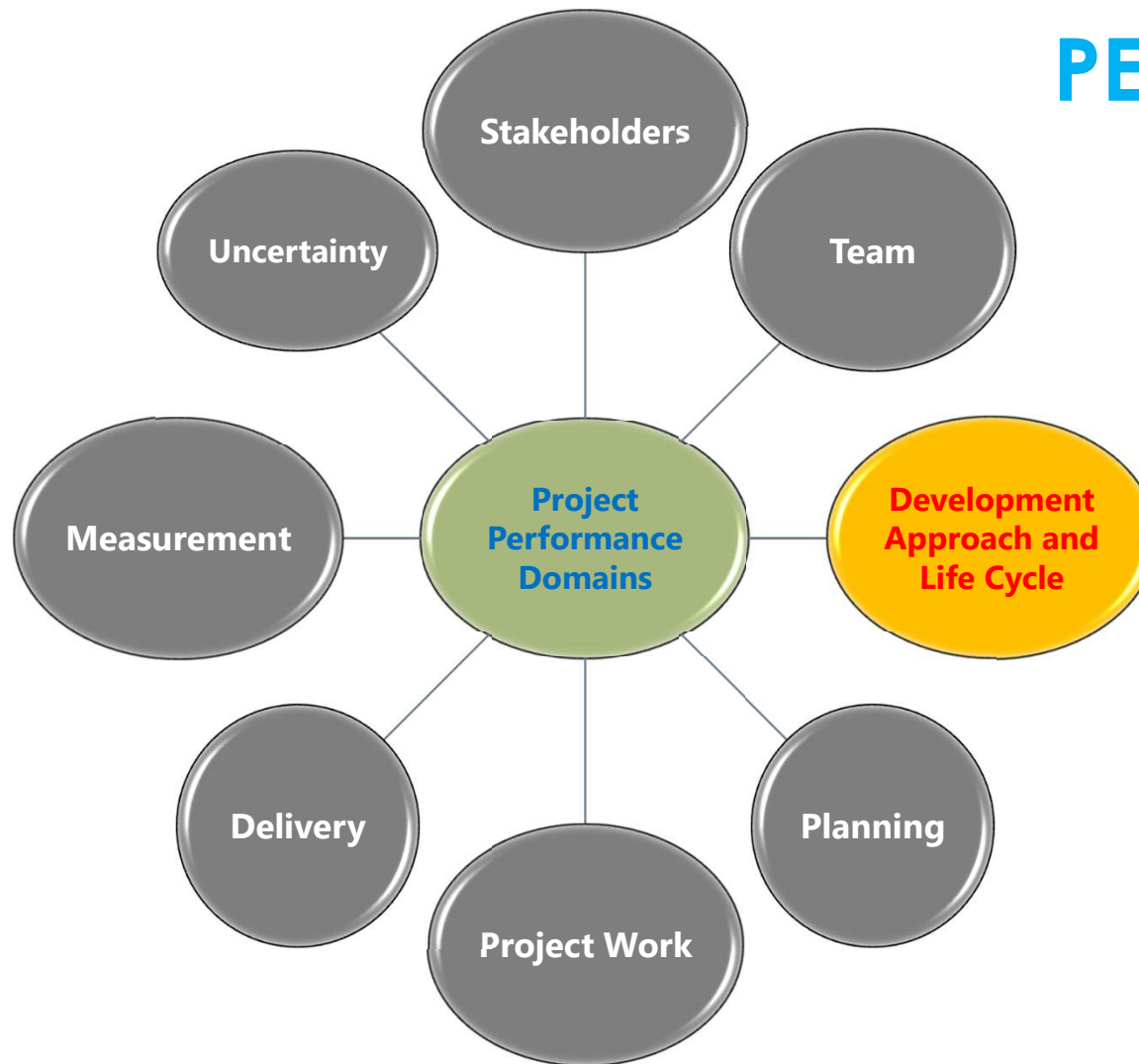


Development Approach & Life Cycle

PERFORMANCE DOMAIN

PROJECT PERFORMANCE DOMAINS



DEVELOPMENT APPROACH & LIFE CYCLE APPROACH PERFORMANCE DOMAIN

DEVELOPMENT APPROACH & LIFE CYCLE APPROACH PERFORMANCE DOMAIN

The Development Approach and Life Cycle Performance Domain addresses activities and functions associated with the development approach, cadence, and life cycle phases of the project.

Effective execution of this performance domain results in the following desired outcomes:

- Development approaches that are consistent with project deliverables.
- A project life cycle consisting of phases that connect the delivery of business and stakeholder value from the beginning to the end of the project.
- A project life cycle consisting of phases that facilitate the delivery cadence and development approach required to produce the project deliverables.

This performance domain entails establishing the development approach, delivery cadence, and project life cycle needed to optimize project outcomes.

DEFINITIONS

The following definitions are relevant to the Development Approach and Life Cycle Performance Domain:

Deliverable. Any unique and verifiable product, result, or capability to perform a service that is required to be produced to complete a process, phase, or project.

Development Approach. A method used to create and evolve the product, service, or result during the project life cycle, such as a predictive, iterative, incremental, adaptive, or hybrid method.

Cadence. A rhythm of activities conducted throughout the project

DEFINITIONS

The following definitions are relevant to the Development Approach and Life Cycle Performance Domain:

Project Phase. A collection of logically related project activities that culminates in the completion of one or more deliverables.

Project Life Cycle. The series of phases that a project passes through from its start to its completion.

DEVELOPMENT, CADENCE, AND LIFE CYCLE RELATIONSHIP

- The type of project deliverables(s) determines how it can be **developed**.
- The type of deliverable(s) and the development approach influence the number of **cadence** for project deliveries.
- The deliverable approach and the desired delivery cadence determine the **project life cycle and its phases**.

DELIVERY CADENCE

Delivery cadence refers to the timing and frequency of project deliverables. Projects can have a single delivery, multiple deliveries, or periodic deliveries.

- **Single delivery.** Projects that have a single delivery deliver at the end of the project. For example, a process reengineering project may not have any deliveries until near the end of the project when the new process is rolled out.
- **Multiple deliveries.** Some projects have multiple deliveries. A project may have multiple components that are delivered at different times throughout the project. A project to develop a new drug may have multiple deliveries, such as preclinical submissions, phase 1 trial results, phase 2 trials result, phase 3 trial results, registration, and then launch.
- **Periodic deliveries.** Periodic deliveries are like multiple deliveries, but they are on a fixed delivery schedule, such as monthly or bimonthly. A new software application may have internal deliveries every two weeks, and then periodically release the deliveries into the market.

DELIVERY CADENCE

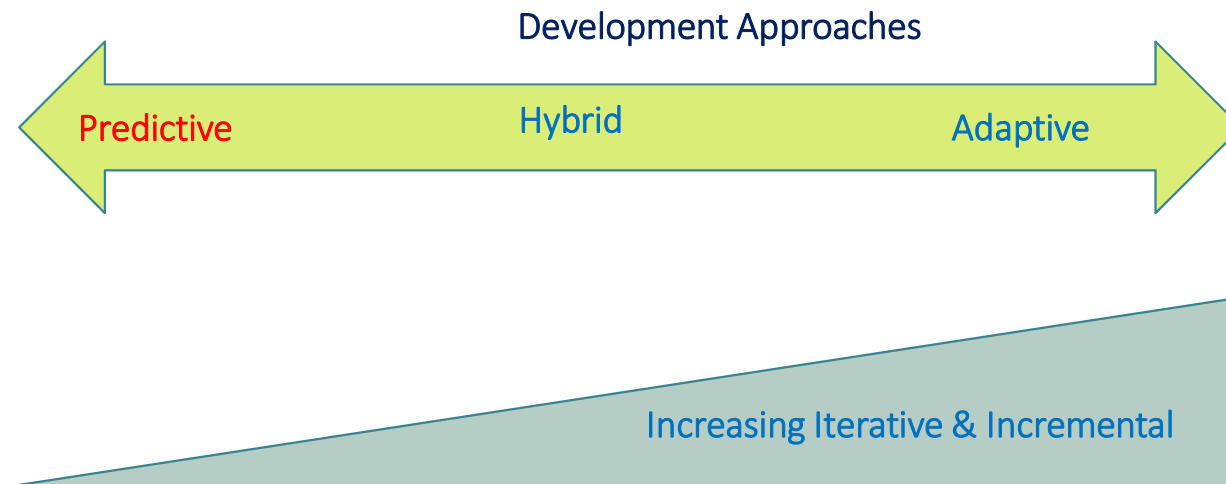
Another delivery option is called continuous delivery. **Continuous delivery** is the practice of delivering feature increments immediately to customers, often through the use of small batches of work and automation technology.

Continuous delivery can be used for digital products. From the product management perspective, the emphasis is on delivering benefits and value throughout the product life cycle. Similar to a project, there are aspects that are development oriented.

However, similar to a program, there can be many development cycle as well as maintenance activities. This type of undertaking works better with project teams that are stable and remain intact.

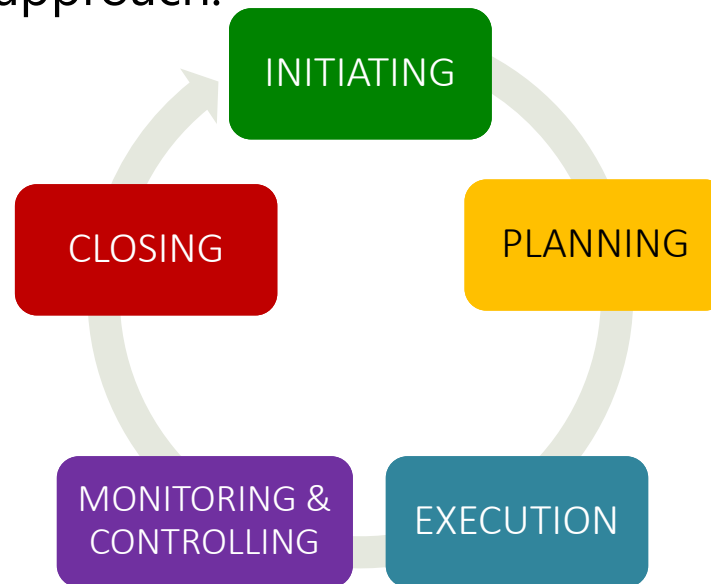
DEVELOPMENT APPROACH

A development approach is the means used to create and evolve the product, service, or result during the project life cycle. There are different development approaches and different industries may use different terms to refer to development approaches. Three commonly used approaches are predictive, hybrid, and adaptive. As shown in the below, these approaches are often viewed as a spectrum, from the predictive approach on one end of the spectrum, to the adaptive on the other end.



PREDICTIVE APPRAOCH

A predictive approach is useful when the project and product requirements can be defined, collected, and analyzed at the start of the project. This may also be referred to as a waterfall approach.



Project Phases are divisions within a project where extra control is needed to effectively manage the completion of a major deliverable

HYBRID APPROACH

A hybrid development approach is a combination of adaptive and predictive approaches. This means that some elements from a predictive approach are used and some from an adaptive approach are used. This development approach is useful when there is uncertainty or risk around the requirements. Hybrid is also useful when deliverables can be modularized, or when there are deliverables that can be developed by different project teams. A hybrid approach is more adaptive than a predictive approach, but less so than a purely adaptive approach.

An example of a **hybrid approach** could be using an adaptive approach to develop a product that has significant uncertainty associated with the requirements. However, the deployment of the product can be done using a predictive approach. Another example is a project with two main deliverables where one deliverable is developed using an adaptive approach and the other using a predictive approach.

HYBRID APPROACH

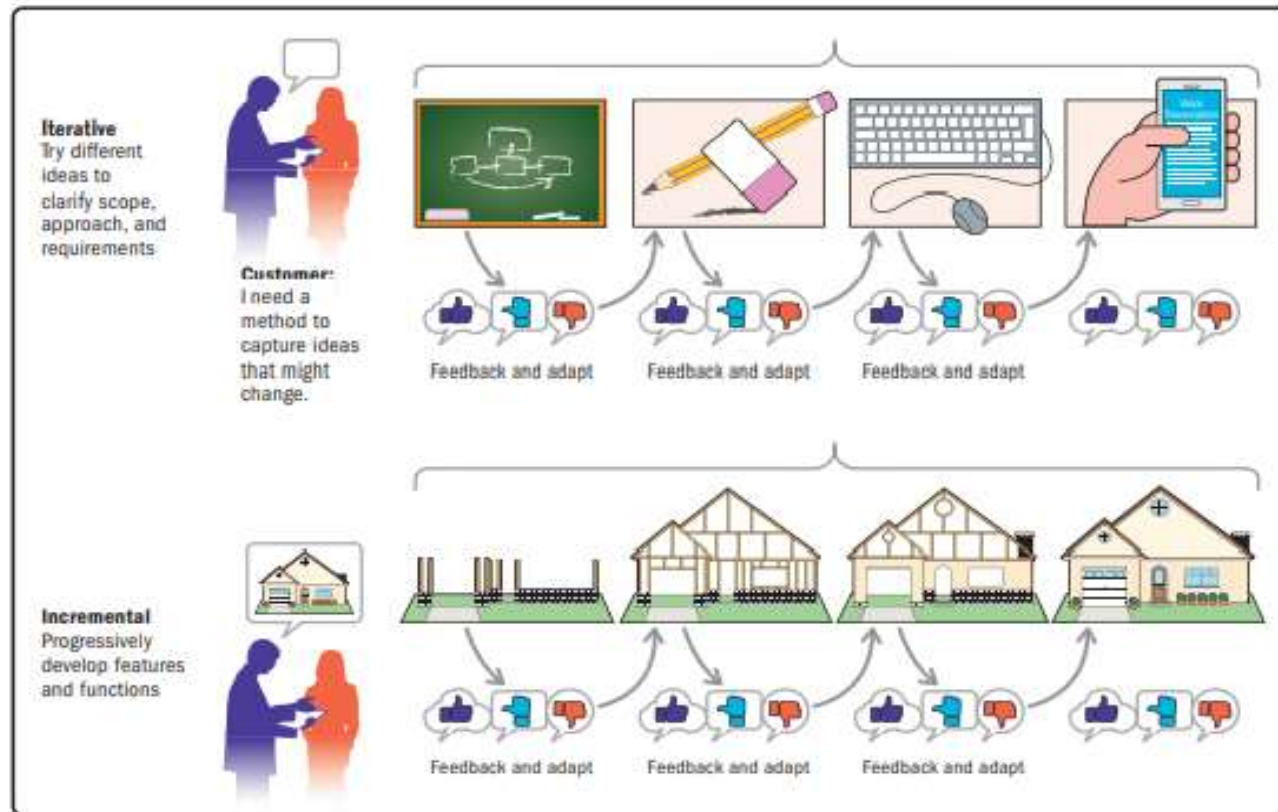


Figure 2-8. Iterative and Incremental Development

HYBRID APPROACH - Example

As part of the community center, a project to establish senior services could be developed and deployed **iteratively**. For example, the first iteration could be a Meal on Wheels program. This could be followed by a transportation service, then group outings and events, caregiver relies, adult day care, and so forth. Each service would be complete on its own and could be deployed when it was available. Each additional service would improve and increase the senior services for the community.

A project to establish training for community action patrol volunteers could use an **incremental** approach. The training, comprised basic training, logistics training, and patrol training, can be developed by different people. It can be developed at the same time in modules, or one module can be developed, feedback gathered, and then subsequent modules can be developed. However, the community action patrol training program will only be complete after all the modules are developed, integrated, and deployed.

ADAPTIVE APPROACH

Adaptive approaches are useful when requirements are subject to a high level of uncertainty and volatility and are likely to change throughout the project. A clear vision is established at the start of the project, and the initial known requirements are refined, detained, changed, or replaced in accordance with user feedback, the environment, or unexpected events.

Adaptive approaches use iterative and incremental approaches. However, on the far side of the adaptive methods, the iterations tend to get shorter and the product is more likely to evolve based on stakeholder feedback.

The project team is very engaged with the planning for each iteration. The project team will determine the scope they can achieve based on a prioritized backlog, estimate the work involved, and work collaboratively throughout the iteration to develop the scope.

ADAPTIVE APPROACH- Example

The community center will need a website so community members can access information from their home computer, phone, or tablet. The high level requirements, design, and page layout can be defined up front. An initial set of information can be deployed on the website. User feedback, new services, and internal stakeholder needs would provide content for a backlog.

The backlog information would be **prioritized**, and the web team would develop and deploy new content. As new requirements and new scope emerge, the estimates for the work would be developed, the work would be done, and once tested, it would be demonstrated for stakeholders. If approved, the work would be deployed to the website.

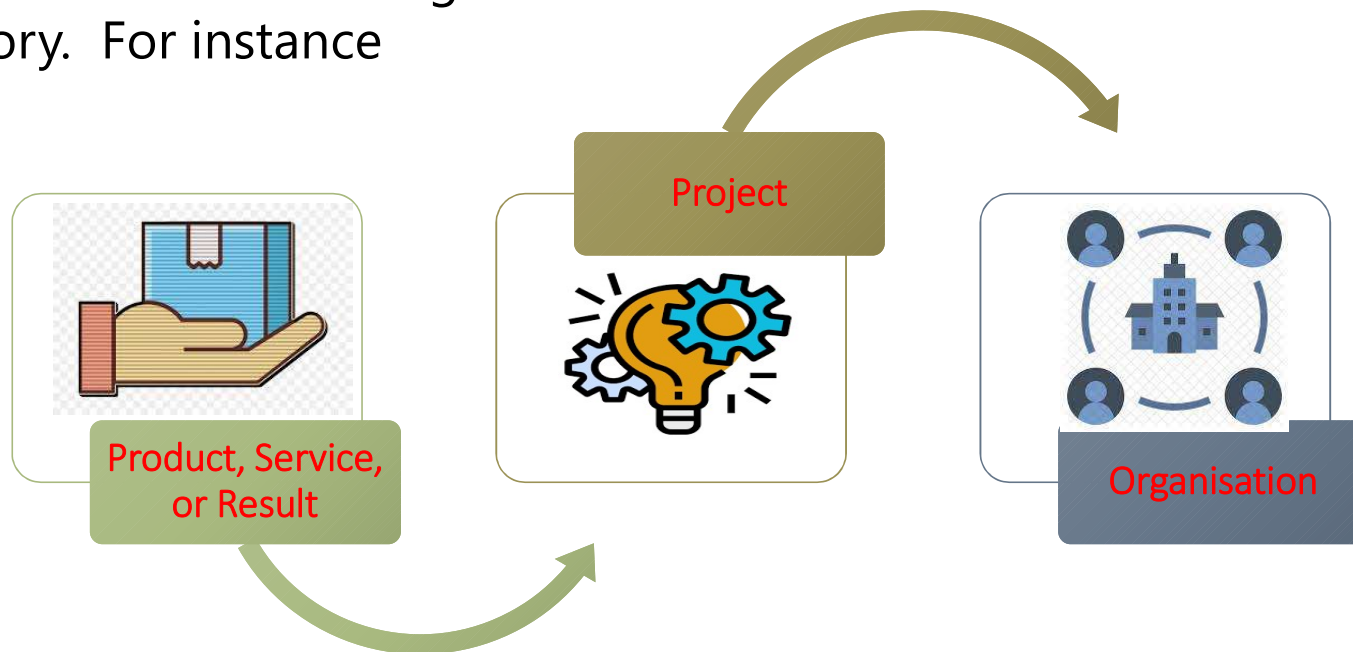
ADAPTIVE APPROACH- Example

The community center will need a website so community members can access information from their home computer, phone, or tablet. The high level requirements, design, and page layout can be defined up front. An initial set of information can be deployed on the website. User feedback, new services, and internal stakeholder needs would provide content for a backlog.

The backlog information would be **prioritized**, and the web team would develop and deploy new content. As new requirements and new scope emerge, the estimates for the work would be developed, the work would be done, and once tested, it would be demonstrated for stakeholders. If approved, the work would be deployed to the website.

CONSIDERATION FOR SELECTING A DEVELOPMENT APPROACH

There are several factors that influence the selection of a development approach. They can be divided into categories of the product, service, or result; the project; and the organization. The following subsections describe the variables associated with each category. For instance



CONSIDERATION FOR SELECTING A DEVELOPMENT APPROACH:

PRODUCT, SERVICE, OR RESULT

There are many variables associated with the nature of the product, service, or result that influence the development approach. The following list outlines some of the variables to consider when selecting the development approach.

Degree of innovation. Deliverables where the scope and requirements are well understood, that the project team has worked with before, and that allow for planning up front are well suited to a predictive approach.

Requirements certainty. When the requirements are well known and easy to define, a predictive approach fits well. When requirements are uncertain, volatile, or complex and are expected to evolve throughout the project, a more adaptive approach may be a better fit.

Scope stability. If the scope of the deliverable is stable and not likely to change, a predictive approach is useful. If the scope is expected to have many changes, an approach that is closer to the adaptive side of the spectrum can be useful.

CONSIDERATION FOR SELECTING A DEVELOPMENT APPROACH:

PRODUCT, SERVICE, OR RESULT

Ease of change. Related to the requirements certainty and the scope stability, if the nature of the deliverable makes it difficult to manage and incorporate changes, then a predictive approach is best.

Delivery options. As described on Delivery Cadence, the nature of the deliverable and whether it can be delivered in components influences the development approach. Products, services, or results that can be developed and/or delivered in pieces are aligned with incremental, iterative, or adaptive approaches.

Risk. Products that are inherently high risk require analysis before choosing the development approach.

Safety requirements. Products that have rigorous safety requirements often use a predictive approach as there is a need for significant up front planning to ensure that all the safety requirements are identified, planned for, created, integrated, and tested.

Regulations. Environments that have significant regulatory oversight may need to use a predictive approach due to the required process, documentation, and demonstration needs.

CONSIDERATION FOR SELECTING A DEVELOPMENT APPROACH:

PROJECT

Project variables that influence the development approach are centered around stakeholders, schedule constraints, and funding availability.

Stakeholders. Projects that use adaptive methods require significant stakeholder involvement throughout the process. Certain stakeholders, such as the product owner, play a substantial role in establishing and prioritizing work.

Schedule constraints. If there is a need to deliver something early, even if it is not a finished product, an iterative or adaptive approach is beneficial.

Funding availability. Projects that work in an environment of funding uncertainty can benefit from an adaptive or iterative approach. A minimum viable product can be released with less investment than an elaborate product.

CONSIDERATION FOR SELECTING A DEVELOPMENT APPROACH:

ORGANISATION

Organizational variable such as the structure, culture, capability, project team size, and location influence the development approach.

Organizational structure. An organizational structure that has many level, a rigid reporting structure, and substantial bureaucracy frequently uses a predictive approach. Project that use adaptive methods tend to have a flat structure and may operate with self organizing project teams.

Culture. A predictive approach fits better in an organization with a culture of managing and directing, where the work is planned out and progress is measured against baseline. Adaptive approaches fit better within an organization that emphasizes project team self management.

Organizational capability. Transitioning from predictive development approaches to adaptive approaches and then to using agile methods is more than just stating that the organization will now be agile.

Project team size and location. Adaptive approaches, especially agile methods, often work better with project teams. Adaptive approaches also favor project teams that are located in the same physical space large project teams and project teams that are mostly virtual may do better by using an approach that is closer to the predictive side of the spectrum.

LIFE CYCLE & PHASE DEFINITIONS

The type and number of project phases in a project life cycle depend upon many variables, chief among them the delivery cadence and the development approach, as described previously. Examples of phases in a life cycle include:

Feasibility. This phase determines if the business case is valid and if the organization has the capability to deliver the intended outcome.

Design. Planning and analysis lead to the design of the project deliverable that will be developed.

Build. Construction of the deliverable with integrated quality assurance activities is conducted.

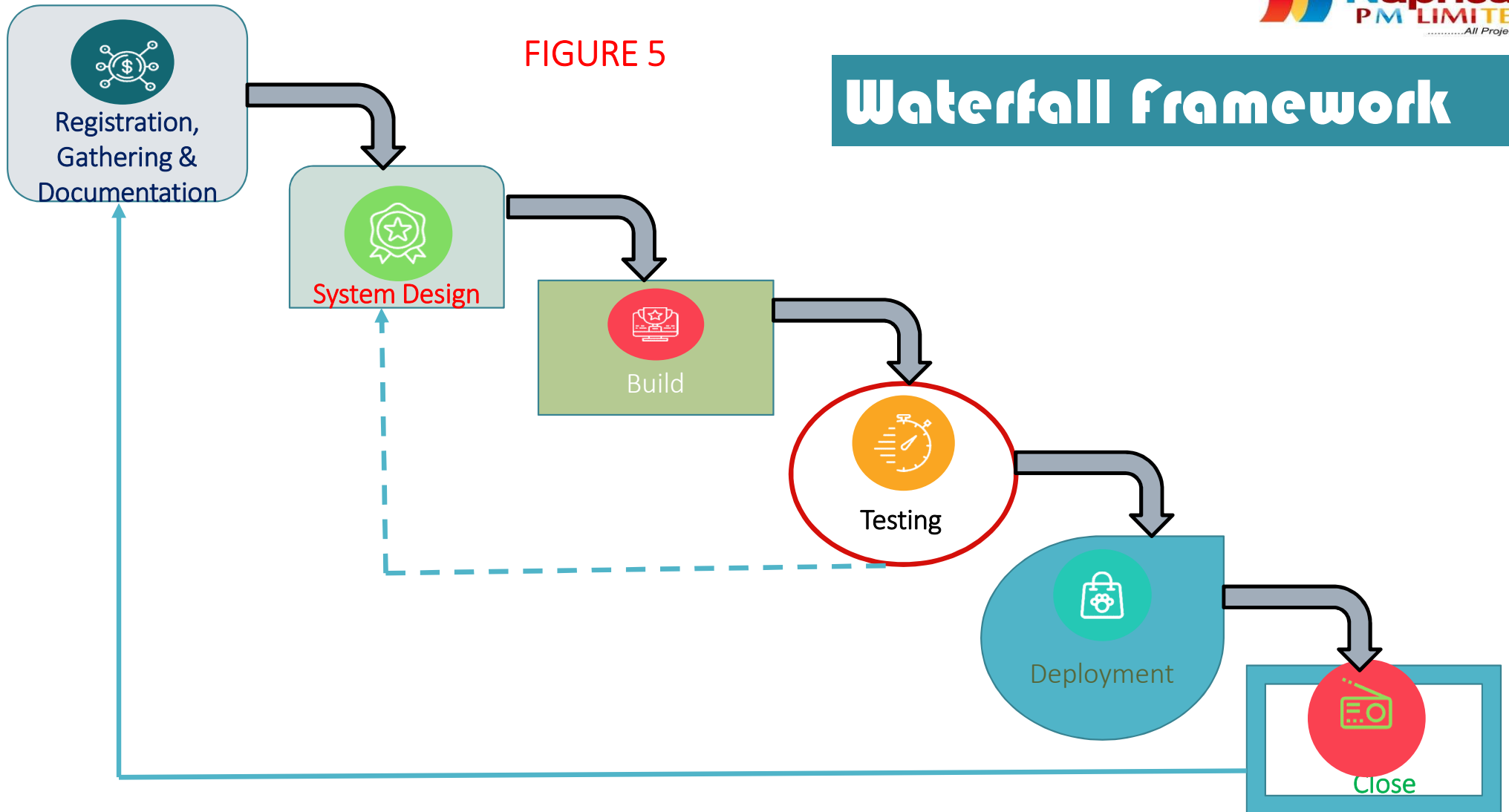
Test. Final quality review and inspection of deliverables are carried out before transition, go-live, or acceptance by the customer.

Deploy. Project deliverables are put into use and transitional activities required for sustainment, benefits realization, and organizational change management are completed.

Close. The project is closed, project knowledge and artifacts are archived, project team members are released, and contracts are closed.

FIGURE 5

Waterfall framework



LIFE CYCLE & PHASE DEFINITIONS

Project phases often have a phase gate review (also known as stage gate) to check that the desired outcomes or exit criteria for the phase have been achieved before proceeding to the next phase. Exit criteria may tie to acceptance criteria for deliverables, contractual obligations, meeting specific performance targets, or other tangible measures.

The Figure in immediate previous slide (Fig 5) shows a life cycle where one phase finishes before the next one begins. This type of life cycle would fit well with a predictive development approach since each phase is only performed once, and each phase focuses on a particular type of work.

LIFE CYCLE & PHASE DEFINITIONS

The figure (6) below shows a life cycle with an incremental development approach. There are three iterations of plan, design, and build shown in this example. Each subsequent build would add functionality to the initial build.

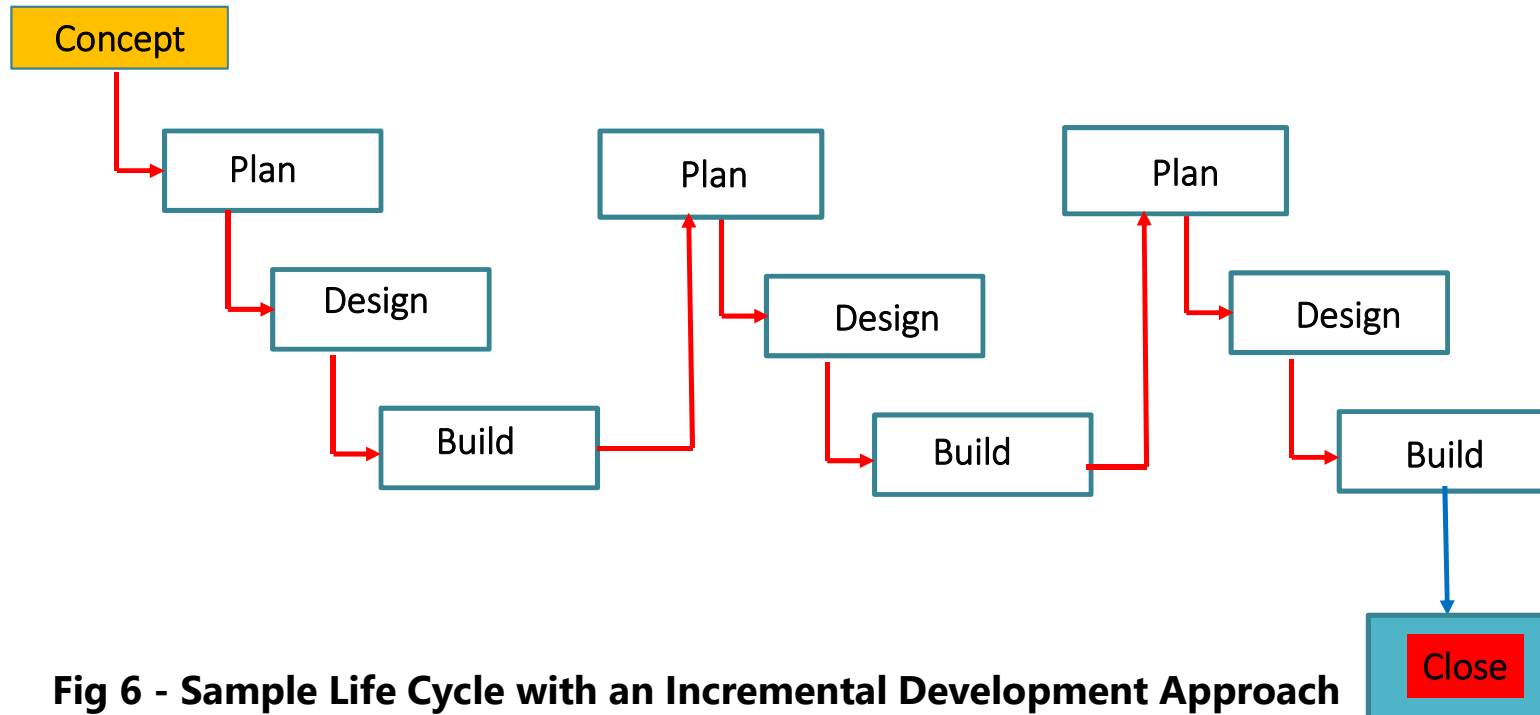


Fig 6 - Sample Life Cycle with an Incremental Development Approach

LIFE CYCLE & PHASE DEFINITIONS

Figure 7 below shows a life cycle using an adaptive development approach. At the end of each iteration (sometimes known as a sprint), the customer reviews a functional deliverable. At the review, the key stakeholders provide feedback, and the project team updates the project backlog of features and functions to prioritize for the next iteration. This approach can be modified for use in continuous delivery situations

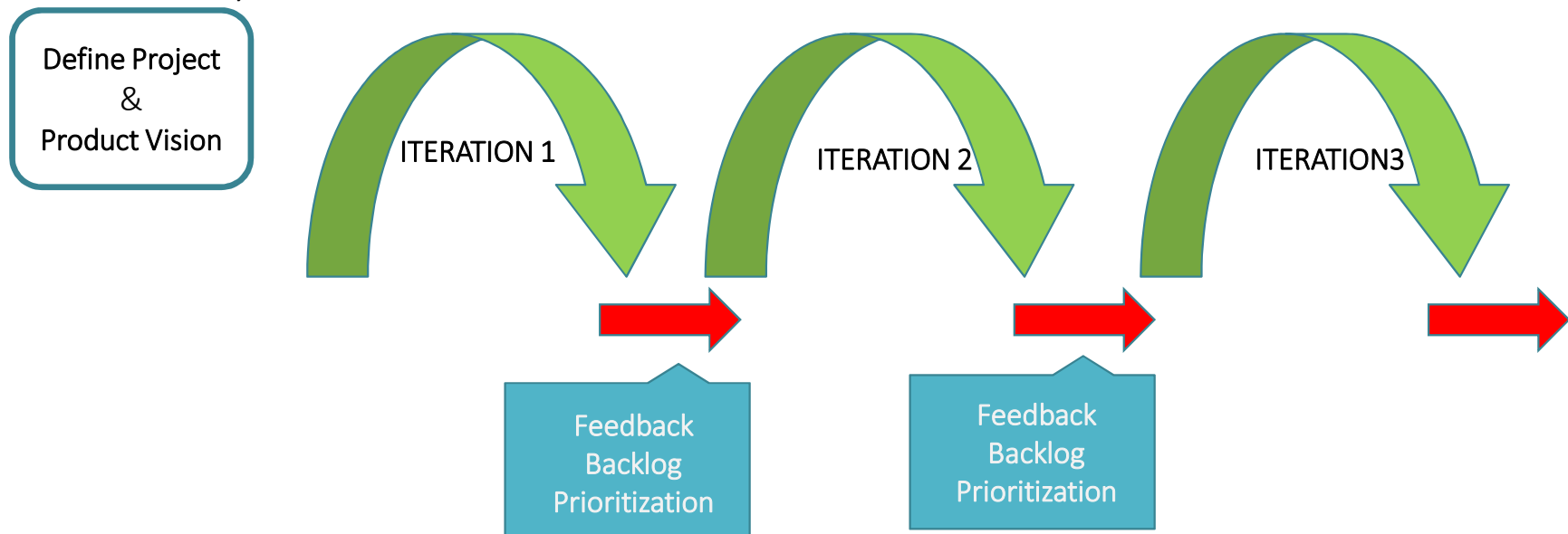


Fig 7 - Life Cycle using Adaptive Development Approach

LIFE CYCLE & PHASE DEFINITIONS

WORTH NOTING - Several adaptive methodologies, including agile, use **flow based scheduling**, which does not use a life cycle or phases. One goal is to optimize the flow of deliveries based on resource capacity, materials, and other inputs.

Another goal is to minimize time and resource waste and optimize the efficiency of processes and the throughput of deliverables. Projects that use these practices and methods usually adopt them from the Kanban scheduling system used in lean and just in time scheduling approaches.

Fig 7 - Life Cycle using Adaptive Development Approach

ALIGNING DELIVERY OF CADENCE, DEVELOPMENT APROACH, & LIFE CYCLE

The community center examples described in development approaches will be revisited to demonstrate how the delivery cadence, development approach, and life cycle fit together. In this example, there are four products and services; the building, the community action patrol (CAP) training, the senior services, and the website. The Table below describes the delivery cadence and the development approach.

Deliverable	Delivery Cadence	Development Approach
Building	Single delivery	Predictive
Senior services	Multiple deliveries	Iterative
Website	Periodic deliveries	Adaptive
Community action patrol training	Multiple deliveries	Incremental

Delivery Cadence & Development Approach

INTERACTION WITH OTHER PERFORMANCE DOMAIN

The Development Approach and Life Cycle Performance Domain interacts with the Stakeholder, Planning, Uncertainty, Delivery, Project Work, and Team Performance Domains. The life cycle selected impacts the way in which planning is undertaken.

Predictive life cycles undertake the bulk of the planning up front and then continue to re-plan by using rolling wave planning and progressive elaboration. Plans are also updated as threats and opportunities materialize.

MEASURING OUTCOMES

The Table below identifies the outcomes on the left and ways of checking them on the right. Checking Outcomes – Development Approach and Life Cycle Performance Domain

Outcome	Check
Development approaches that are consistent with project deliverables	The development approach for deliverables (predictive, hybrid, or adaptive) reflects the product variables and is appropriate given the project and organizational variables.
A project life cycle consisting of phases that connect the delivery of business and stakeholder value from the beginning to the end of the project.	Project work from launch to close is represented in the project phases. Phases include appropriate exit criteria.
Project life cycle phases that facilitate the delivery cadence and development approach required to produce the project deliverables.	The cadence for development, testing, and deploying is represented in the life cycle phase. Projects with multiple deliverables that have different delivery cadences and development methods are represented by overlapping phases or phase repetitions, as necessary.

Q & A?

