CDC UNIFIED PROCESS JOB AID

ITERATIVE DEVELOPMENT

Purpose

The purpose of this document is to provide guidance on the practice of **Iterative Development** and to describe the practice overview, requirements, best practices, activities, and key terms related to these requirements. In addition, templates relevant to this practice are provided at the end of this guide.

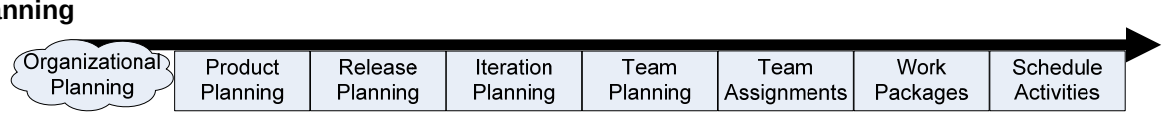
Practice Overview

Iterative development is best defined in terms of its processes that allow for dynamic development rather than any single defined method or approach. As a result, many types of iterative development methodologies exist, some of which include: Agile, Dynamic System Development, Extreme Programming, Feature Driven Development, Incremental, Spiral, etc. Regardless of name, commonalities exist across all iterative methods and include aspects such as:

- Development is done incrementally over multiple iterations
- Iterations include some component of planning, requirements analysis, design, development, testing, documentation, and implementation
- Emphasis is placed on person-to-person communication over written documents
- Software that works is the primary measure of progress and success
- Iterations build upon lessons learned in prior iterations – a continuous improvement methodology
- Teams are self-organized and granted authority to dynamically adapt to changing circumstances and are compiled in such a manner that allows for the delivery of completed, working code at the end of each iteration. Teams often include analysts, programmers, testers, writers, etc. Required facilities for deployment of the completed components are also made available as needed.

Iterative methods measure progress in terms of completely developed, tested, and working products. Iterative methods promote the delivery of the final product through refinement of requirements, design of usable components, and incremental delivery of those components over the life of the project. Each incremental cycle is referred to as an *iteration* and typically lasts ~4 to ~8 weeks. This iterative approach allows project teams to very quickly respond to changes in requirements, and other factors influencing project outcomes. The dynamic nature of this approach continuously improves and/or adds functionality, identifies and addresses risks, and delivers business value. These benefits occur incrementally throughout the life of the project, not just toward the end as in a traditional Waterfall model. An iterative approach is best used when the organization, product, or project must contend with changing requirements throughout the life of the project. Refer to the CDC Unified Process Job Aid entitled “CDC UP Guidance for Iterative Development Using EPLC” for additional information and examples.

Planning

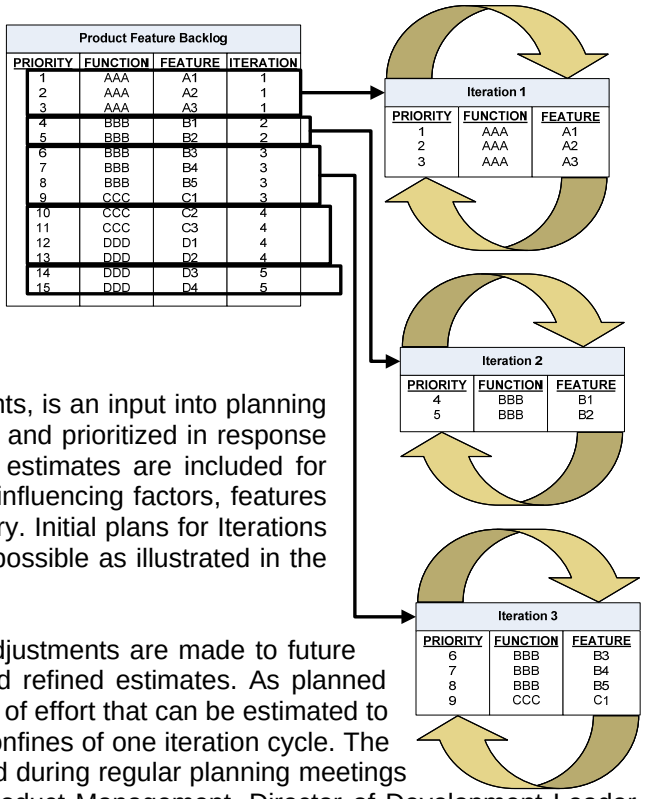


Just as there are many variants of iterative development, there can be many approaches to project planning. Commonalities exist across iterative methods; planning approaches share similarities as well. In almost all instances planning is confined by the constraints of scope, cost, and time as they influence the quality of the end product. Not accounting for organizational planning, planning projects often begins at a product level and then those plans are decomposed to a level appropriate for the methodology used. For example, in an iterative development approach, product planning may be decomposed into release plans, iteration plans, iteration teams, team assignments, work packages, and eventually scheduled activities.

CDC UNIFIED PROCESS JOB AID

ITERATIVE DEVELOPMENT

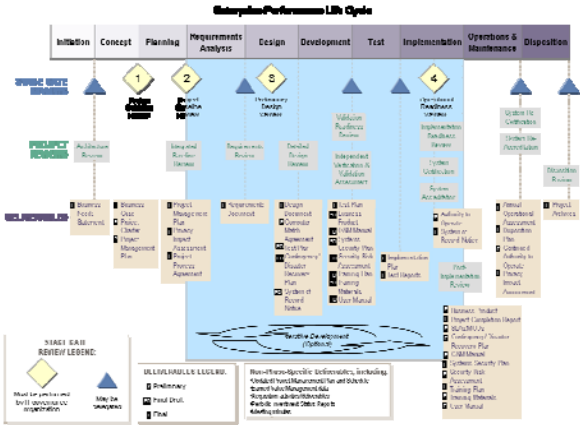
The product features that are identified and eventually incorporated into iterations come from some source of requirements gathering efforts in advance of beginning any iteration. Product requirements are also continuously gathered throughout the life of the project. As new requirements are gathered they are maintained in what can be described as a triage area, sometimes referred to as a requirements backlog, requirements inventory, feature list, etc for inclusion into future development iterations.



The feature list, comprised of customer requirements, is an input into planning future iterations. This list is continuously amended and prioritized in response to project influences and client demands. Rough estimates are included for each prioritized feature. Based on this, and other influencing factors, features are assigned an initial expected iteration for delivery. Initial plans for Iterations are documented as far into the future as feasibly possible as illustrated in the above image.

As the project progresses through its life cycle, adjustments are made to future planned iterations based on updated priorities and refined estimates. As planned iterations near, features are decomposed to a level of effort that can be estimated to include their design, build, and testing within the confines of one iteration cycle. The actual work assigned to each iteration is negotiated during regular planning meetings between stakeholders such as Product Owner, Product Management, Director of Development Leader, and eventually Functional Managers, etc. Work is allocated for upcoming iteration and assigned to iteration teams for delivery. Once an iteration begins, its planned work is frozen, not to be changed.

Managing



It's important to recognize that for any project there are multiple streams of management activities that interact to deliver a completed project. The HHS EPLC Framework image to the right illustrates how all activities associated with investment, project, and product development; capital planning, regulations, and governance interrelate throughout the ten EPLC life cycle phases.

Often officers, specialists, functional managers, etc, are responsible for managing these different aspects of the project. For example, the Project Officer, in collaboration with a CPIC Office(r), may be responsible for tracking the status of the financing and budgeting of the project. A Development Manager may be responsible for the actual product delivery. Each has distinct expectations that at the same time influence each other and the overall project's objective. The Project Manager is ultimately responsible for the coordination of these and other stakeholder expectations as well as overall project delivery. As a result, there are different levels of planning that must occur and each must take all of this, and more, into account. This level of planning is often done well before any development activities or iterations begin.

No project should ever begin without a project team's understanding and agreement upon the project's objectives and how they will be accomplished. In an iterative development environment, the project is initially planned at a high-level, from inception through eventual disposition. The high-level project

definition is presented to the appropriate governance body to obtain approval and funding. These actions relate specifically to some of the investment, project, and capital planning components of the EPLC Framework. Implied, inclusive of the presentation to governance bod(y/ies), are yet to be defined sub-plans, schedules, policies, processes, etc that will be used within the project, and project team, to organize and deliver project objectives. These sub-plans are often too granular for a governance review of any one project and may include items such as schedules for individual functional teams; resource pools management techniques, inter-team dynamics, etc.

To facilitate the oversight of all project efforts overarching documents are developed that define standards and approaches to manage the overall project, consistent with stakeholder expectations. These documents include plans to manage items such as risk, acquisition, change, configuration, communication, requirements, etc. and are used as inputs into planning the more detailed, day-to-day project activities such as product releases, iterations, iteration teams, daily communications, etc.

Executing & Controlling

The prior text identifies a level of effort required to plan for investment and project management, and capital planning activities, that should be completed before a project begins. In addition, an overarching effort is needed to baseline some assumptions as to how project management and product development work will be performed and managed. This overarching effort outlines, at a high-level, requirements, analysis, design, and development for the project as a whole and is completed before any development work begins. This is often the level of information that most projects should be presenting to governance committees, entering into and tracking within ProSight, using to calculate and track earned value, and reporting to stakeholders, executives, and project teams in status reports.

In addition to this higher-level planning, a level of detailed planning, requirements analysis, design, and development specific to each iteration, and how the iteration team will function, must happen before any development iteration can begin. This detailed planning effort defines standards for iteration teams and may include items such as:

- Definition of project objectives, architecture requirements, security requirements, etc
- Selection of program languages, operating system platform, coding styles, etc
- Definition of practices, policies, standards, etc for product development
- Estimation of resources, schedule, costs, duration, etc

This detailed-level is where project activities and actual work approaches are applied to building product functionality. At this level is where iterative development methods perform their iterative work and provide the most benefit to delivering upon stakeholder expectations. This level of management, although critical to project success, is often far too detailed to benefit executive level decision makers. Team leads are often accountable for managing and delivery at this level. The Project Manager does what is necessary to track progress of these activities and to ensure project work is progressing as planned. Project status is then reported by the Project Manager to stakeholders at a level appropriate for the interested audience. This is usually at a level that communicates the progress of specific deliverables and/or milestones.

The primary difference between these two levels of planning is how, and where, throughout the project's life cycle the architecture, requirements, and design is performed and reviewed by the appropriate level of governance. As it relates to EPLC, an iterative approach would deliver an overarching set of supporting documents at the appropriate Stage Gate Review. However, as the project progresses detailed architectures, requirements, and designs will evolve during each iterative cycle. Successfully progressing through planned deliverables and/or milestones would represent that the overarching investment and project planning, detailed iterative planning, and work performed, is progressing successfully. How the governance review of this evolution is handled will be defined by governance policies and in some instances may be exclusive to accommodate the uniqueness of a particular project.