

SOFTWARE CONSTRUCTION

The purpose of this report is to provide some insights for the software construction process, such that we will provide how software construction is useful and a crucial step in the field of software development cycle. In this phase the design specifications are converted down to the actual useful code stuff.

The term software construction refers to the detailed creation of working software through a combination of coding, verification, unit testing, integration testing, and debugging.

SOFTWARE CONSTRUCTION FUNDAMENTALS

There are main 5 major fundamentals of the software construction such that:

- Minimizing complexity
- Anticipating change
- Constructing for verification
- Reuse
- Standards in construction.

Minimizing Complexity

- Minimize code complexity for better optimization.
- Achieve optimization through coding standards.
- These standards enhance code readability.
- Benefits include reduced errors and easier maintenance.
- Prioritize readability over overly clever code.
- Optimization scales across the project.
- Efficient development and lower maintenance costs result from optimized, readable code.

Anticipating The Change

Most Softwares are built today such that, when the platform changes , in short when the system changes then our software should withstand those changes to the fullest leading to no system or software failure.

Constructing for the Verification

This scenario means that the faults can be found by the software engineers in an easy way by the testers and the coders when they are individually searching building up the test scenarios.

This actually involves restricting the complex to hard language structure, organizing the code to support the automated testing.

Reuse

It basically means that , whatever existing features or assets are there , using them to build a unique element.

In software construction, typical assets that are reused include libraries, modules, components, source code, and commercial off-the-shelf (COTS) assets.

Reuse has literally two facets , *construction with reuse* and the *construction for reuse*.

Which means that, constructing down the software using the existing set of technologies, certain code standards and all whereas constructing anything which is totally new in nature so that it could be reused again and again in the other software production cycles that would be there in future.

Standards In Construction

There are certain aspects which come while the designing of the software, these standards are used in almost every domain of software construction leading to some standards that are developed already.

First of all, coding standards play a key role in the software development lifecycle. Just like some classes, some libraries should be written in standard format.

Secondly, Platforms also play an important role just like on which scale the meetings are being conducted and all.

Third, which programming language we are working on.

MANAGING CONSTRUCTION

Construction In Life Cycle Models

Some models are sort of linear in nature such that they are linear from the construction point of view, such as the waterfall and the staged-delivery life cycle models.

These sort of models actually treat this construction as an activity that occurs only after certain work has been completed—including detailed requirements work, extensive design work, and detailed planning

The more linear approaches tend to emphasize the activities that precede construction (requirements and design) and to create more distinct separations between activities. In these models, the main emphasis of construction may be coding.

Construction Planning

Construction method plays a key role in the construction planning activities because these actually lay the foundation of the software development cycle. Construction prerequisites are affected by the choice of the construction method, in which order they are being formed and also in which order they were performed or initialized.

PRACTICAL CONSIDERATIONS

Construction Design

Some of the projects are allocated considerable design activities to the construction, such that others allocate the design to a phase explicitly, Regardless of the exact allocation. At the construction level, some of the design work will occur.

Construction Languages

So in the software construction process, there is a kind of communication by which a human can actually execute the problem solution. These construction languages actually affect quality attributes of performance, reliability etc. Simplest type of language is the configuration language, under this option the software engineers choose from the set of attributes to work on it.

The files which are text based used up in the windows as well as in the linux are the perfect example of this.

Scripting languages are the perfect example of the languages which are used up in the application programming languages.

Similarly, programming languages are also the best example and yet the most flexible languages which are part of the construction language, as they contain least amount of the data about the particular domain set so these are quite simple in nature.

Coding

Following considerations apply to the software construction coding activity:

- Techniques for creating the code structure understandable is very important, including the source code optimisation.
- Use of classes, enumerated types, variables, named constants, and other similar entities
- Use of control structures
- Handling up the errors

Construction for Reuse

Creating software with the intention of reusing it in the future, either for the current project or other projects, is called "construction for reuse." This approach involves designing and analyzing the software to make it versatile across different systems. To prevent issues like duplicating code, it's a good idea to organize reusable code parts into well-organized libraries or components.

PROJECT PERFORMANCE DOMAIN

Project Performance Domain refers to a specific area within a project where performance is assessed, measured and managed. The Project performance domains form a unified whole and are independent of each other which results in the successful completion of the project and its intended outcomes.

Following are the 8 Project Performance Domains:

- Stakeholders.
- Team.
- **Development approach and Life cycle.**
- Planning.
- Project work.
- Delivery.
- Performance.
- Uncertainty and ambiguity.

DEVELOPMENT APPROACH AND LIFE CYCLE PERFORMANCE DOMAIN

The Development Approach and Lifecycle Performance Domain is all about the '*big picture*' of how you deliver a project:

- The type of methodology you select
But not the specific methodology you'll use, nor the adaptations you make
- The general rhythm (or '*cadence*') of major events
But not the day-to-day scheduling of activities
- The broad stages or phases within your project
What each stage will focus on and how you will sequence them in time

Effective execution in a project is essential in achieving desired outcomes. There are three key components to consider when it comes to optimizing project outcomes: development approach, delivery cadence, and project life cycle.

WHAT IS DELIVERY CADENCE?

The type of deliveries of projects may change from case to case. Some projects may have a single delivery, some may have multiple deliveries, and some may have periodic deliveries. Delivery cadence refers to the timing and frequency of project deliveries. Maintaining the correct rhythm of activities is important here. Because value creation and project delivery are like a heartbeat. When the heartbeat rhythm is steady it means your project is healthy, if it has irregularities it may point out potential problems.

PROJECT LIFE CYCLE

The project life cycle is made up of five project stages: project initiation, project planning, project execution, monitoring & control and project closing. Each of these phases is necessary for the effective delivery of the project.

If you're managing projects, you'll need the right tools to make the process more effective and efficient.

DEVELOPMENT APPROACHES IN PROJECT MANAGEMENT

Different people may give different names to development approaches. Some organizations even may create their own approach. However the most common ones are predictive, adaptive and hybrid development approaches.

PREDICTIVE APPROACH: The predictive approach is the most familiar one as it was the focus of former PMBOK guides. The other name of it is the waterfall approach. It is based on planning as much as possible before performing the project activities. It is useful when we are able to define, collect and analyze requirements at the beginning of a project.

ADAPTIVE APPROACH: When uncertainty exists at the beginning of a project, then the adaptive approach is the most applicable one. In this type of project, requirements change frequently and the project needs to adapt itself to changing requirements. There are two iterative approaches as iterative and incremental.

HYBRID APPROACH: It is simply appointed somewhere in the middle of predictive and adaptive approaches. Even if it is more adaptive than being predictive most times, it still has some characteristics of a predictive approach.

CONSIDERATIONS FOR SELECTING A DEVELOPMENT APPROACH

The criteria can be divided into three categories: product, service, or result; project; and organization

Product, Service, or Result	Project	Organization
<ul style="list-style-type: none">• Degree of Innovation• Requirements Uncertainty• Scope Stability• Ease of Change• Delivery Options• Risk• Requirements and Regulations	<ul style="list-style-type: none">• Stakeholders• Schedule Constraints• Funding Availability	<ul style="list-style-type: none">• Organizational Structure• Culture• Organizational Capability• Project Team Size and Location

Product, Service, or Result

The factors influencing the product, service, or result consideration all have to do with the nature of a project's outcome, whether it is a product, a service, or another type of result.

Degree of Innovation

Deliverables that have a well-understood scope and requirements, that the project team has worked with before, and that allow for planning up front are well suited to the predictive approach. Deliverables involving a high degree of innovation or those with which the project team does not have experience are better suited to a more adaptive approach.

Requirements Certainty

A predictive approach works well when the requirements are well known and easy to define. When requirements are uncertain, volatile, or complex and are expected to evolve throughout the project, a more adaptive approach is a better fit.

Scope Stability

If the scope of the deliverable is stable and not likely to change, a predictive approach is practical. If the scope is expected to undergo many changes, an approach that is closer to the spectrum's adaptive side can be helpful.

Ease of Change

Related to requirements certainty and scope stability, if the nature of the deliverable makes it challenging to manage and incorporate changes, then a predictive approach is best. Deliverables that can quickly adapt to change can use an adaptive approach.

Delivery Options and Cadence

The nature of the deliverable, such as whether it can be delivered in components, influences the development approach. Products, services, or results that can be developed and delivered in pieces are aligned with incremental, iterative, or adaptive approaches.

Risk

You can reduce risk by building products modularly and adapting the design and development based on learning to take advantage of emerging opportunities or reduce the exposure to threats. Adaptive approaches frequently mitigate high-risk requirements by addressing their viability first.

Safety Requirements and Regulations

Products that have rigorous safety requirements often use a predictive approach because significant up-front planning is needed to ensure that all the safety requirements are identified, planned for, created, integrated, and tested. Likewise, environments that are subject to significant regulatory oversight may need a predictive approach due to the required process, documentation, and demonstration needs.

Project

The factors in the project consideration column all have to do with aspects of the project, such as how it is structured, constrained, and funded.

Stakeholders

Specific stakeholders, such as product owners, play a substantial role in establishing requirements from the customer's perspective and prioritizing work. If such dedicated project team staff are available to support project work, adaptive methods are preferred.

Schedule Constraints

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

Funding Availability

Projects that work in an environment of funding uncertainty can benefit from an adaptive approach. A minimum viable product can be released with less investment than an elaborate product. This allows for market testing or market capture with minimum investment.

Organization

The factors in the organization consideration column all have to do with the organizational environment of the project, including the culture, structure, and complexity.

Organizational Structure

An organizational structure with many levels, a rigid reporting structure, and substantial bureaucracy is likely to use a predictive approach. In contrast, projects that use adaptive techniques are associated with a flat structure.

Culture

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

Organizational Capability

If organizational policies embrace attitudes and beliefs that support an agile mindset, then adaptive methods are likely to succeed.

Project Team Size and Location

Adaptive approaches often work best with project teams of five to nine individuals. Adaptive approaches also favor project teams that are located in the same physical space.

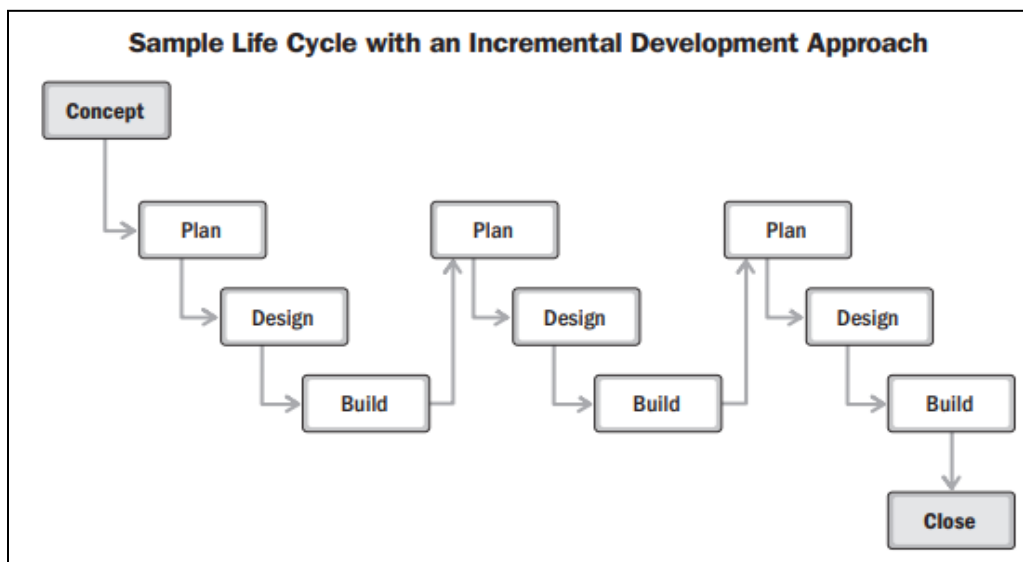
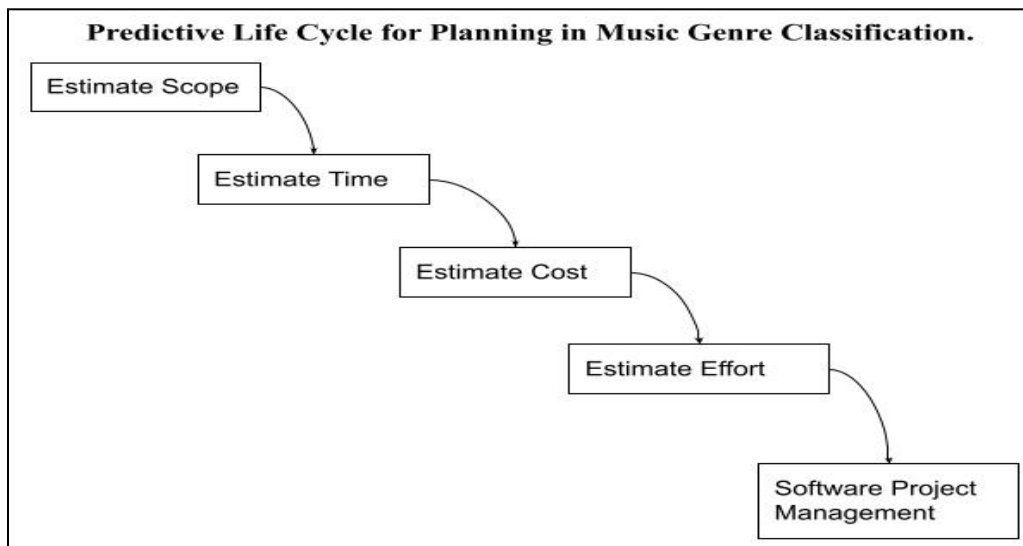
LIFE CYCLE AND PHASE DEFINITIONS

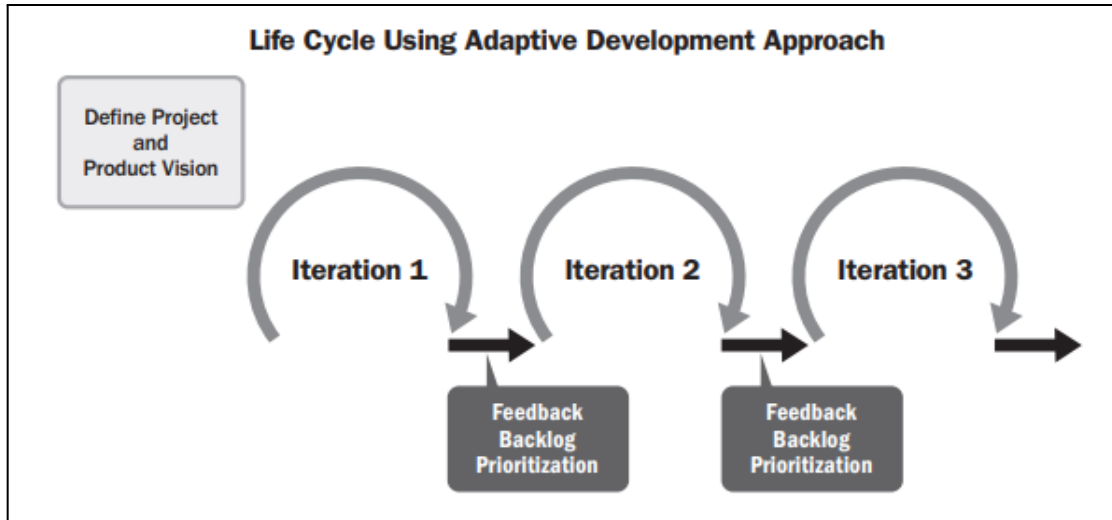
The type and number of the project's phases during its life cycle are dependent on a range of factors. The process of delivery and development approach, as described before, are the chief among them. The following examples of stages in the life cycle can be found:

- **Feasibility**
This stage will determine whether a business case can be proven and if an organization is capable of delivering the desired outcome.
- **Design**
The development of the project deliverables which will be drawn up is guided by planning and analysis.

- **Build**
Construction of the deliverable with integrated quality assurance activities is conducted.
- **Test**
Before the transition, go live, or acceptance by the customer, the final quality review and inspection of the deliverables shall be carried out.
- **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.

In order to check that the desired outcomes or exit criteria have been met before proceeding into the following phase, project stages are often accompanied by a stage gate review also known as Phase Gate Review. The exit criteria may relate to acceptance conditions for deliverables, contractual obligations and achieving specific performance objectives or other tangible measures.





ALIGNING OF DELIVERY CADENCE, DEVELOPMENT APPROACH, AND LIFE CYCLE

Delivery Cadence and 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

Based on this information, a potential life cycle might be:

1. Start Up

- Entry Criteria
Business case approval and project charter signing are required entry criteria. These criteria signify the formal initiation of the project.
- Activities in the Initiation Phase
A high-level project roadmap is developed.
Initial funding requirements are specified.
The project team and necessary resources are identified.
- Completion of Deliverables
Before concluding the initiation phase, specific deliverables must be completed. These deliverables typically include the development of the project roadmap, defining initial funding needs, forming the project team, and establishing a progress schedule.
- Exit Criteria and Origination Phase Gate Review
Exit criteria are the conditions that must be met before proceeding from one project phase to the next. In this case, exit criteria are reviewed at an origination phase gate review

2. Plan

This stage involves the decomposition of building's complex information into concrete plans. A detailed design document for the CAP training is completed. Together with a gap analysis, an analysis of the senior services offering is carried out. The first frame of the website is created. By the end of the planning phase, these outputs should have been completed. The exit criteria shall be revised as part of a gate review at the planning stage.

3. Development

Since the deliverables are varied in terms of their delivery timetable and different approaches, this phase will be accompanied by test and deployment phases. Early deliveries will be made on the website, giving information to the public about progress at the Community Centre. Before the opening of the Community Centre, certain senior functions and CAP training may start to take place. Before going to the test stage, each delivery can have its own review.

4. Test

Overlaps with development and deploy phases.

The type of testing depends on the specific deliverable.

Includes inspections for building, beta delivery of CAP courses, small-scale trials for senior services, and operating in a test environment for website releases.

Each deliverable undergoes relevant testing before moving to the deploy phase.

5. Deploy

Overlaps with development and test phases.

Initial website deployment may occur early in the project.

Activities evolve as more deliverables become available.

Final deployment is the opening of the community center.

Ongoing updates for the website and senior services become part of operations after the community center opens.

6. Close.

Occurs periodically as deliverables are completed.

Upon initial website deployment, project personnel (including contractors) are released, and retrospectives or lessons learned are conducted.

At project completion, phase gate reviews and an overall evaluation of project performance compared to baselines take place.

Prior to final closeout, the project charter and business case are reviewed to determine if intended benefits and value have been achieved.

INTERACTIONS WITH OTHER PERFORMANCE DOMAINS

Interaction with Stakeholder Domain

- The Development Approach and Life Cycle Performance Domain interact with the Stakeholder Domain.
- The chosen life cycle impacts planning.
- Predictive life cycles involve extensive upfront planning and use rolling wave planning and progressive elaboration.
- Plans are updated as threats and opportunities arise.

Interaction with Uncertainty Domain

- Development approach and delivery cadence help reduce uncertainty in projects.
- For deliverables with regulatory risk, a predictive approach may be chosen to incorporate extra testing, documentation, and robust processes.
- For stakeholder acceptance risk, an iterative approach with a minimum viable product is chosen for feedback before further development.

Interaction with Delivery Performance Domain

- There is significant overlap with the Delivery Performance Domain, particularly concerning delivery cadence and development approach.
- Delivery cadence is crucial for delivering value in alignment with the business case and benefits realization plans.
- Product and quality requirements from the Delivery Performance Domain influence the development approach.

Interaction with Team Performance Domain

- Interaction occurs regarding project team capabilities and leadership skills.
- The development approach influences the project team's way of working and the project manager's style.
- A predictive approach emphasizes upfront planning, measurement, and control.
- An adaptive approach, especially with agile methods, requires a servant leadership style and may involve self-managing project teams.

MEASURING OUTCOMES

Checking Outcomes—Development Approach and Life Cycle Performance Domain

1. Developing methods that correspond to the project deliverables

The development approach for deliverables (predictive, hybrid, or adaptive) reflects the product variables and is appropriate given the project and organizational variables.

2. The project's life cycle is divided into stages in which the supply of business and stakeholder value takes place from beginning to end of a project.

In the project phases, work from inception to closure shall be taken into account. Appropriate exit criteria are included in the phases.

3. Phases of the project's life cycle which facilitate delivery pace and development in order to produce Project deliverables.

The development, testing and deployment times are indicated in the life cycle stages. The overlaps or repetitions of phases shall be used to represent projects with a number of deliverables that vary in delivery schedules and development methods, where appropriate.