LifecycleStep
Project Lifecycle Process

LifecycleStep

Project Lifecycle Process™

Executive Summary

LifecycleStep Overview

Projects are the way that most new work gets delivered. All projects have certain characteristics in common. They all have a beginning and an end. In other words, they do not continue on forever. Projects result in the creation of one or more deliverables. Projects also have assigned resources - either full time, part time or both. There are other characteristics as well. All organizations can have projects. Projects can include building a house or office building, planning and executing a marketing campaign, upgrading desktop operating systems, installing a new phone system, developing an IT business application, etc.

Projects can be managed using a common set of project management processes. In fact, a similar set of project management processes can be utilized regardless of the type of project. All projects should be defined and planned and all projects should manage scope, risk, quality, status, etc. Project management, however, defines the overall management and control processes for the project. Project management does not actually result in the project execution. At some point, you still need to define the actual activities necessary to build the house, execute the marketing campaign, develop the IT business application and upgrade the desktop operating systems. These activities are referred to as the project lifecycle and the project lifecycle is the focus of LifecycleStep.

The typical waterfall lifecycle model is described first in detail. The steps are as follows:

Analysis

The Analysis Phase is where the project lifecycle begins. The only exception is a situation where you have broken a large project down into smaller components. In that case, you may have an entire project that is only focused on the Analysis Phase, while the next project may start at the Design Phase. The Analysis Phase is where you break down the high-level Project Definition into the more detailed business requirements. The Analysis Phase is also the part of the project where you identify the overall direction that the project will take through the creation of the project strategy documents.

Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps.

Design

The project team should have a nice set of requirements to work from, a set of direction-setting strategies and a Conceptual Systems Design. There might be some on the project team that would say that this is enough to start the Construct Phase. However, it is not. The design process comes next. Even if the project was small and the requirements were simple, there is still a mental design process that occurs in between understanding the requirements and starting to construct. Design becomes more and more important as the project becomes larger, more complex and impacts more and more people. Once you complete the requirements, you will typically see a myriad of alternatives for construction. These alternatives include the tools and technology you will you utilize, the scalability of the solution, and the structure of the components you will build. In essence, the Design Phase is where you look at the many potential solutions and narrow down the choices to determine the most effective and efficient way to construct the solution. The Design Phase answers the questions about "how" you will build the best solution for your organization and your environment.

At the end of the Design Phase, you will have a logical solution defined. The solution is "logical" because it exists on paper or in a design tool. This logical solution is then passed to the Construct Phase, where the logical solution is turned into a physical solution. However, the people that specialize in constructing the solution will not have to worry about the myriad of possibilities. That guidance will be provided to them through the work in the Design Phase. The people working to construct the solution can use their talents to build the solution based on the deliverables produced during the Design Phase.

Construct

Depending on your point of view, the Construct Phase is where the "rubber meets the road". This is the point in the project when we actually start to construct the solution. In an IT development project this is the time to start writing program code. If you followed the LifecycleStep process so far, your construct team has a lot of guidance. They have a complete set of design specifications showing how the application should be structured and organized. They have design specifications for all screens, reports and programs. At this point, then, the project manager should be able to start handing out the various design specifications and the construct team can start to build the solution.

The Construct Phase is also where we will create support documentation such as a Disaster Recovery Manual and Users Manual. In essence these documents are also being "constructed" and so the logical place to create them is the Construction Phase. Depending on the type of document, the support material can also be tested in the Test Phase and then executed or implemented in the Implementation Phase.

Test

We test our work for two main reasons. First, we want to ensure that the solution meets the business requirements. For instance, if a specific business transaction follows a logical path from A -> B -> C -> D, we want our testing to prove that this, in fact, happens. Second, we test to catch errors and defects. These may be programming errors, or errors that were introduced earlier in the development lifecycle. In other words, we need to prove that our solution is correct, while at the same time proving that there are no errors or defects. You may not catch many errors, but you still need to prove that the solution meets the business requirements.

Regardless of the type of project you are on, you want to catch as many errors as early in the lifecycle as possible. There are projects that purposely blast through the analysis, design and coding phases with a philosophy that they will fix all the problems in testing. This is usually disastrous. It’s not so bad to correct programming errors that you discover in your testing. It is much more difficult when you realize that a design error resulted in information missing from your database. It is even harder if you discover that your analysis missed an entire business process.

At the same time, you cannot ensure that your software is perfect. Even with the best tools, it is impossible to test every logic path of every program. Therefore, you cannot ever be sure that your code is bug-free. There are times when software that has been stable for ten years suddenly breaks through an unlikely combination of circumstances that never occurred before. (These are “subtle” bugs.)

Implement

Implementation refers to the final process of moving the solution from development status to production status. Depending on your project, this process is often called deployment, go-live, rollout and installation. For the purpose of LifecycleStep, all of these terms are synonymous with "implementation".

There is no single way to implement an application. It depends on the characteristics of your project and the solution. Some implementations are as easy as saying “we are now live.” This type of implementation can work when the solution is brand new and you are developing and testing in what will become the production environment. In these cases, implementation is just a state of mind. One day the solution is in development and the next day it is in production. What could be easier?

When we think about implementation, we should always start by understanding the level of complexity involved. If the implementation is relatively straightforward, then there is no reason for elaborate implementation processes. However, most projects have a number of implementation events to plan for and execute successfully.

Other Lifecycle Models

In addition to the classic waterfall model described above, LifecycleStep also describes other popular lifecycle models, including Rapid Application Development, enhancements, Agile and Research and Development. Although the waterfall model can be used on all projects, other models may be more effective given the nature of the project and your organizational environment.