Which Life Cycle Is Best for Your Project?

By ExecutiveBrief

When choosing a development life cycle, don't just trust your feelings. Decide based on factors that really matter.

Which life cycle will work best for your project? This is an important strategic question because making the wrong choice could lead to disastrous results of catastrophic proportions. Think about delayed deliveries, unhappy clients, project overruns, and cancelled projects.

During the 80s and early 90s, the waterfall model was the de-facto in project delivery. With the rapid pace in software development and popular use of the Internet, many companies started shifting to more flexible life cycles such as the iterative, incremental, spiral, and agile. These new life cycle methods provide more flexibility and support fast-paced development, giving companies the edge in delivering "the first" in the industry. To date, there are dozens of life cycle methods available to choose from, each having its own advantages and disadvantages.

Here are some of the more popular life cycles:

**Waterfall**

This traditional life cycle method has been around for decades and has proven its ability to deliver. In fact, the US Department of Defence was actively promoting the use of this method in all its projects when it published Standard 2167A in 1998.

Waterfall is defined as a sequential development model with clearly defined deliverables for every phase. Many industry practitioners are strict in performing audit reviews to ensure that the project has satisfied the input criteria before continuing to the next phase.

The standard phases of waterfall are shown in the diagram below:

![Waterfall Phases Diagram](image_url)
Iterative, Incremental

The main objective of iterative development is to build the system incrementally, starting from basic partial system features and gradually adding more features until the entire system is completed. Compared to waterfall, iterative development allows flexibility in accommodating new requirements or changes thereof. It also provides room for improvement in succeeding iterations based on lessons learned from previous iterations.

The diagram below, courtesy of Microsoft's MSF, clearly shows how iterations are scheduled and delivered:

Agile

Agile methodologies arose from the need to develop software applications that could accommodate the fast-paced evolution of the Internet. Agile is, in some way, a variant of iterative life cycle where deliverables are submitted in stages. The main difference is that agile cuts delivery time from months to weeks. Companies practicing agile are delivering software products and enhancements in weeks rather than in months. Moreover, the agile manifesto covered development concepts aside from the delivery life cycle, such as collaboration, documentation, and others.

The diagram from Microsoft MSF shows the various components of an agile life cycle:
Other Variants

There are more life cycle methods and methodologies being practiced including Test Driven Development, RUP, Cleanroom, and others. However, all these life cycles can be generally classified into waterfall, being sequential, with clear and strict cut-off between phases; as well as iterative or agile, being repetitive with flexible cut-off rules.

Here are some questions you need to get answers to before deciding on which life cycle method to use:

How stable are the requirements?

One of the biggest factors that dictate your choice of a life cycle method is the clarity and stability of the project requirements. Frequent changes in requirements after the project has started can derail your progress against the plan. In such cases, choose agile or iterative approach because each provides an opportunity for you to accommodate new requirements even after the project has started. On the other hand, if you are engaged in a more traditional project development where there is a stiff rule on ensuring complete set of requirements before going on to the next phase, waterfall would be your choice. However, such traditional projects are becoming less and less common as companies realise the benefits of using a more agile method of managing projects.

Who are the end-users of the system?

Spend some time to know the users and stakeholders. Who are they? Are they dispersed or controlled group? How can they influence the project? A controlled group of end-users who greatly influence the project can help you define requirements and manage changes. This means you can achieve stability on project requirements and allow you to use the waterfall approach.

On the other hand, if the end-users are dispersed, you are likely to have a wide range of requirements, which you can’t define until the end-users have used the system and started requesting new features. This situation is typical in product development. For example, Google started Gmail and all its products such as Google Docs, Calendar as BETA because they wanted to know the reactions of the end-users and improve the features based on their feedback. Microsoft, the developer of the world’s most popular software, Windows and Office, also applies agile in their development methodologies. Recently, the Microsoft Solution Framework (MSF) adopted the agile approach. According to MSF for Agile Software Development, "small iterations allow you to reduce the margin of error in your estimates and provide fast feedback about the accuracy of your project plans. Each iteration should result in a stable portion of the
overall system." Microsoft and Google choose to be more agile because they have a very dispersed group of end-users.

**Is the time line aggressive or conservative?**

Experienced managers will solve aggressive time lines by negotiating and cutting down project deliverables. An iterative approach helps achieve this by giving opportunities to deliver partial functionalities early. This gives an impression that the project is delivering despite an aggressive time line, generally referred to as "quick wins." While the overall project delivery is not shortened, there is an opportunity for you to satisfy your stakeholders by delivering key features that are necessary. If your project is not time sensitive and end-users can wait for the release of the system, waterfall would be a workable approach.

**What is the size of the project?**

Large enterprise projects generally require large number of project teams to work on clearly defined deliverables. The scale of the deliverables is proportional to the size of the project team assigned to do it. Thus, larger project teams are assigned larger set of deliverables which need to be clearly defined. With this kind of scenario, long iterations or waterfall would be more ideal.

**Where are the project teams located?**

If you have several project teams located in different geographic locations, co-ordination of work needs to be more detailed and stringent. Work assignments need to be well-defined to avoid confusion and redundancy of work. In such cases, Waterfall is likely more beneficial as it provides clear-cut deliverables and milestones. Applying the agile approach on geographically separate teams may introduce new challenges. As noted by Martin Fowler, a well-known agile evangelist, "Because agile development works best with close communication and an open culture, agilists working offshore feel the pain much more than those using plan-driven approaches."

**What are the critical resources?**

Some projects require involvement of unique, skilled resource or integration with highly specialised equipment. In cases where such resources are not immediately available and require planning, the project team must ensure that the resource is fully utilised during its scheduled use. Moreover, tests must be performed on all possible scenarios during the resource's available time. Otherwise, requesting for another schedule of the resource may entail project delays. In such cases, waterfall may be a better approach where each
milestone must be completed before proceeding from one stage to the next and you are assured that the critical resource is well utilised.

ExecutiveBrief, the technology management resource for business leaders, offers proven tips, techniques, and action plans that companies can use to better manage people, processes and tools - the keys to improving their business performance. To learn more, please visit: http://www.executivebrief.com

© ExecutiveBrief 2008

© 2014 Project Smart. All rights reserved.