The Ins and Outs of Iterative, Incremental Development

Kathlika Thomas, PMP
Head Writer, NuWave IT Project Blog
NuWave Technologies

Kathlika Thomas is a certified project manager with over a decade of business analysis and project management experience. She has worked in Big 4 consulting on domestic and international engagements and has also developed several workshops and training programs helping managers and analysts alike sharpen their skills at all levels of project delivery. Kathlika now serves as the main contributor to the NuWave Technologies IT Project Blog.

In the past several years, project leaders across different industries have decided, along with their teams, to choose over other methodologies an iterative, incremental approach to development of their IT enhancements and new product development. The inclination may be to consider this iterative approach to be a fairly novel idea compared to other software development models. For example, the iterative, incremental approach came as a response to the waterfall model, first formally cited in a 1970 article "Managing the Development of Large Software Systems" by W. Royce, a twentieth century leader in conceptualizing and building out software development processes. The more accurate truth is that before this article was published, iterative development was implemented and practiced as a means for quality improvement as early as four decades before this article was written. At the time, methods varied in terms of the length of each iteration and in terms of application across industries. By and large though, the concept was well entrenched in company methodologies, especially in the military and communications industries.

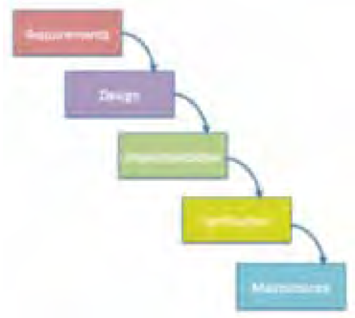
To add a little more color around the development landscape of these software methodologies, here's a quick overview of a few popular models, the existing flaws in management that they sought to address, and the pros and cons of each:

The Waterfall Model – the application of this method is fairly intuitive, given the process name. It was first developed as a means to address a limited amount of attention placed on requirements gathering and analysis and also the existence of defects in late project phases that should have been addressed early on in the project lifecycle. In the waterfall model, project phase progression from the identification of requirements through to design, implementation, verification, and maintenance is perfectly sequential. As when a waterfall flows from high to low ground, once progress has been made from one phase to the next, there is no revisiting a previous phase. Care is taken to make sure each individual stage is well-managed and the relevant tasks are completed without error.

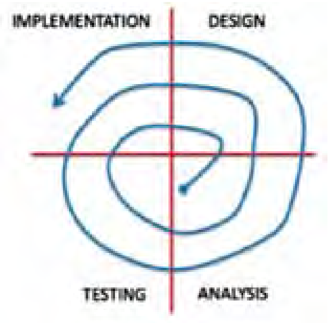
In **the Spiral Model**, an iterative approach is taken to develop a useful system prototype by taking care to spend a lot of effort upfront in perfecting requirements and then using them to conceptualize several different prototype models, identifying the positives and negatives of each, and understanding the risks involved depending on which solution is chosen. A series of prototypes are developed, each one improving upon the one prior to it by addressing its weaknesses and risks. Of course, the major advantage that the Spiral Model has over the Waterfall is that there is room to revisit the work done in previous phases and to limit defects in this way, through repeated and more accurate designs of the final product.

Finally, the **Agile Model** was introduced as a way to provide a finished product quickly through rapid iterations of smaller project deliverables. The goal is to

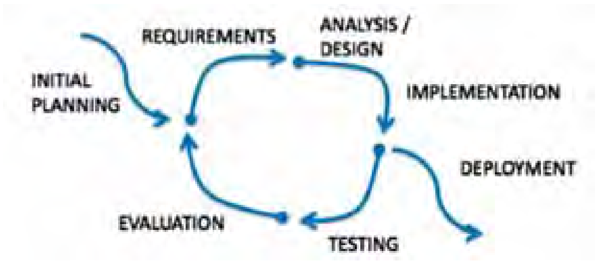
Waterfall Method



Spiral Method



Agile Method



heighten client engagement and satisfaction by delivering to them smaller packages of the finished product in quick succession. Because work schedules are compressed and finished products have to be quickly developed, agile teams are usually cross-functional and self-sufficient.

As you can see, there is a vast and varied array of methodologies in project management. More and more projects across industries are being approached as incremental and iterative endeavors, more in line with the principles governing the agile approach. Chopping a project into modular, manageable pieces of work that have a clear definition of success is an approach that has become more attractive to organizations.

So what's the appeal of iterative, incremental development? Foremost, each recurring iteration has a definite deliverable, timeframe, and even budget. As with the usual end-to-end software development lifecycle, each iteration has its own phases for requirements gathering, design, testing, et cetera. But, in the case of iterations, because development pieces are smaller, risk can be reduced, giving managers and sponsors more control of their overall budget, a primary concern in today's corporate financial climate.

While a shorter development window usually means a reduced budget, over time you may not have to pay resources as much if they're only engaged for a limited period and if they're not sitting idle during development of pieces for which they're not directly responsible. What about the potential for defects given such short, fast-paced development windows? Actually, the agile method has led to less defects in the final product. Why? Testing and verification is completed at each step of the way, allowing less high severity defects to slip through past go-live. Finally, another benefit to be expected is a product that will meet client needs. The agile method in particular emphasizes a spirit of collaboration, one that will lead to fewer "surprises" in the end because of constant communication between client and service provider.

Let's consider a case study from NuWave Technologies of an intricate and complicated project that was simplified because the team took on a phased, iterative approach. In this project, the goal was to assist a government agency to reinvent the way they track equipment maintenance in 23 different facilities using a Maintenance Management System (MMS). The problems to be tackled were many:

First, each facility had its own disparate database. That means that common information in the database had to be manually updated on a regular basis. Disparate databases gave no universal view of all facility maintenance. That lack of a broad lens led to limited BI in the business team's approach to maintenance. The solution called for a consolidation of all databases into a single and centralized datastore that could be used by all facilities and a newly developed crystal reporting solution to enhance intelligence and the business's ability to make smart decisions in their maintenance group. The development solution was phased in this way:

Phase 1: Consolidate – The project management team employed a third-party data extractor to replicate each of the 23 separate databases into a single consolidated database on the HP NonStop server. This included setting up online replication to capture ongoing updates to each database. Since the maintenance databases weren't designed for consolidation, some data transformation was required to prevent collisions in the consolidated database. Another problem was resolving

continued on page 35

The Ins and Outs of Iterative, Incremental Development

continued from pg. 13

conflicts in data that had been manually replicated across the 23 sites. So to deal with this issue, algorithms were developed to identify and resolve the conflicts and rules were developed to prevent conflict reoccurrence.

Phase 2: Replicate – The goal of this phase was to replicate the new database on a MS SQL server hosted on a clustered Windows system. This task required further data transformations to map data from the non-normalized records to a relation schema. We also created some monitoring tools to ensure continuous operation.

Phase 3: Convert – Over 50 COBOL language programs needed to be converted to Crystal Report definitions. This proved to be the greatest challenge because of the lack of documentation for the thousands of lines of report generation codes and the complex business logic that was encoded in them. After successfully completing this task and finishing the project, monitoring and operations was turned over to maintenance support staff.

What was the outcome? Not only did the operational datastore succeed in keeping the agency organized throughout its 23 facilities, but the users were thrilled with the overall experience and the constant communication between the client and the project management team, as is typical of an iterative development approach. Now end users can more easily track past and upcoming maintenance, allowing the agency to budget and plan more efficiently.

So while an iterative approach may seem more complex because of the dynamic nature of the work, it ends up being easier because the planning required is short-term with many opportunities to adjust as time progresses. Also, the frequent face-to-face communication with clients that is typical of an iterative project helps keep stakeholders and end users engaged to ensure satisfaction with the final deliverable.

What else can be said of iterative, incremental development? In the agile method specifically, each work package is delivered in a matter of weeks and each one builds upon the functionality of the previous one so that, at project end, the client has been fully involved in the release of the entire software solution from start to finish. However, since functioning software is the main measure of progress on this type of project, documentation often falls by the wayside, a potential risk for planning future projects and for post-project retrospectives.

As it stands, iterative development is not a novel concept. The phased approach to development and implementation has long been used to boost project management efficiency and the speed of project completion. Be aware of the advantages and disadvantages so that you can make the best of this approach when you decide to try it on your own projects. You'll find that, when used in an organization that's motivated and open to change, it can be a great tool in your project management arsenal.