

Waterfall v Agile: How Should I Approach My Software Development Project?

By Duncan Haughey, PMP

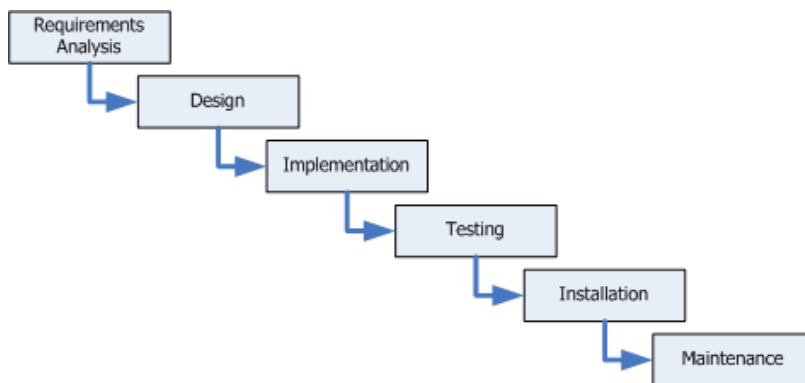
Software development projects are usually approached using one of two methods, Waterfall or Agile. Both have pros and cons and each have their exponents who espouse the value of their chosen approach. In this article, I look at both methods and try to understand which is best and under what circumstances answering the question, "How should I approach my software development project?"



Waterfall Method

The waterfall method has been around since the 1970s and remains in common use. It says you should follow a distinct set of steps through the life cycle of your project. The steps are usually similar to this:

- Requirements Analysis
- Design
- Implementation
- Testing
- Installation
- Maintenance



The reason it is called the waterfall method is that each stage follows from the previous one when it has been completed, cascading down like a waterfall. Like a waterfall it flows down and never up. This method is often seen as "safer."

The Waterfall method works well with commercial arrangements where contracts are signed and money is paid. However, when working for internal customers it's more difficult to be hard-nosed about last-minute changes when it's people in your own organisation asking who often have backing from senior management.

Pros

- Detailed documentation.
- Agreed and signed off requirements.

Cons

- Slow start.
- Fixed requirements difficult to change.

Pros

Can be delivered using developers with a lower skill set.

Reduced number of defects through thorough design planning.

Defined start and end point for each phase, allowing progress to be easily measured.

Cons

No customer visibility of software until the development has been completed.

Lack of flexibility making it difficult to change direction.

Customers often unclear about their requirements initially.

Agile Method

The Agile method refers to an iterative approach to projects, where requirements and solutions evolve through collaboration among customers and development teams. The term was coined in 2001 when the "Agile Manifesto" was put together.

If your manager is worried about you making mistakes in front of customers, this method may not be for you. The Agile method means that you create deliverables early and refine them through several iterations with the customer. However, you may find that some stakeholders view this negatively.

During a recent project to deliver a video homepage for a customer's website, the first iteration was close to the example provided by the design agency. Testing this iteration threw up some items that needed attention, most trivial in terms of development effort needed to fix them.

My manager saw this as having failed to deliver the customers' requirements and an embarrassment for the team. My view was that our delivery was 90 per cent right and the final 10 per cent would only take a few days of development effort. This first iteration allowed us to understand fully the customer requirements and for the customer to be clear about their expectations.

After being suitably rebuked I was left with no doubt of my managers feelings. I wasn't to allow this to happen again, and all future work should have a fully agreed and signed off specification before a single-line of code is written.

Interestingly enough, the customer was happy with the Agile approach as it enabled them to see work in progress and comment on it during development.

So does the Agile method lead to a faster delivery? That's debatable. Just because you start coding early doesn't necessarily mean you'll finish sooner. However, it does ensure the final deliverable meets the customers' needs, mainly because they provide useful input throughout the development phase.

Pros

Quick start, incremental releases and regular customer reviews and feedback.

Evolution of requirements over time.

Ability to respond to change quickly.

Less rework, achieved through continual testing and customer involvement.

Real-time communication among the development team and customer.

Cons

Can be misinterpreted as unplanned or undisciplined.

Needs a high-quality, customer facing development team.

Needs a high-level of customer involvement.

Lack of long-term detailed plans.

Produces a lower level of documentation.

Conclusion

So which method is best, Waterfall or Agile?

Neither...

The method you use depends on several factors, some of which are:

- **Type of work:** is there a clearly defined outcome? Website development is creative and therefore lends itself to the Agile method. Transactional system development, where there is a clearly defined outcome, is better suited to the Waterfall method.
- **Type of customer:** are they willing to work in an iterative way, do they have enough time to review and comment on regular iterations?
- **View of your manager:** does he or she favour one method over the other. How adaptable is he or she?
- **View of IT within the organisation:** is IT seen as a valued partner or a necessary evil? If the latter then use the Waterfall method.
- **Internal or external customer:** is your customer internal or external? The Waterfall method works well with external customers where you can hold them to a signed contract, as opposed to internal customers who can force you into changes by gaining the backing of senior management.

It's important as project manager to select the method that best satisfies your customers' needs. The problem comes when there's a difference of opinion. In these cases choose the method that will deliver the best result, and manage anyone who disagrees.

In the end both methods can deliver your project. It's all about managing the expectations of your customer and delivering a quality product. Let's face it, once you reach journey's end, nobody worries how you got there.

© Project Smart 2000-2010. All rights reserved.