

## Software Development Methodologies

If you are running a software project, one of the main questions you are likely to come across is which development methodology to use. There are as many opinions on the subject as there are project managers and developers, with some keenly advocating an agile approach and others sticking to the traditional waterfall approach.

Below, we will look at two fundamentally different development methodologies and also discuss what the middle ground looks like. We will evaluate the pros and cons, and assess how you best go about determining which approach (or mix of approaches) to choose for your project.

### Test Yourself ...

- How would you describe the difference between a waterfall, an iterative and an agile software development approach?
- What are the disadvantages of a traditional waterfall approach?
- Under which circumstances would you favor using an agile development approach?

### **The traditional approach: waterfall**

The traditional way of running a software project is very structured and controlled and is often referred to as the waterfall approach. It proposes that you follow a distinct set of steps through the lifecycle of your project from requirements analysis, design, build and test to implementation and maintenance. The project team takes a serial approach to development and the project normally has a long and complicated test phase and a big bang implementation.

In these types of projects you typically develop a detailed plan and budget for the entire project very early on based on a comprehensive analysis phase of the requirements. The project manager will then execute and track the plan until the project is delivered. Some organizations allow the plan and budget to be refined over time, whereas others assume that they are set in stone and will hold the project manager to them. Changes to requirements are firmly controlled as the goal is to minimize changes so that the project can be delivered on time and budget.

The advantages of the traditional waterfall approach are that it gives some degree of transparency and a feeling of predictability to senior management. The project is executed sequentially with clear sign off points and has definite start and end points. It tends to be extensively documented and it produces solid (albeit assumption-based) plans and estimates which can be tracked and reported on.

The disadvantages of this approach are that it does not always provide the level of predictability which senior management thinks it gets and that the customer only sees a working version of the product after it has been fully developed and released. Another

disadvantage is that project plans can be very rigid and do not cope well with changes. As a result the project may end up producing functions and products which are not fit for purpose or, if changes are incorporated, the project may be delivered late or over budget.

### **The agile approach**

Agile is an iterative and incremental development approach which is performed in a highly collaborative manner. It uses a different and in some ways more flexible method to produce the end product than the waterfall methodology. With the agile approach, fixed parameters such as cost, time and scope are often less important than efficiency, feedback and quality. The emphasis is on face to face communication and on building the right product. Plans and budgets evolve over time and change as the project progresses. The agile philosophy is based on the fact that requirements change and that they must be easily incorporated into the project. The team will continuously check what the highest priority requirements are, and focus on analyzing, developing and implementing those. The team will iterate through these deliverables in close cooperation with the end users, until a level of satisfaction is achieved. It is about constantly managing priorities and change in order to provide ongoing and maximum benefit to the users and stakeholders. The users receive working software on a regular basis and in the order of priority which they have defined.

The advantages of agile are that features and benefits are delivered early and incrementally throughout the project, as opposed to at the very end, and that the risk associated with producing an incorrect outcome is minimized. Changes are easy to incorporate and stakeholders have good visibility of how the project and its products are progressing. In addition, agile teams are often relatively dynamic and motivated due to the collaborative nature of the project and because they are empowered to make decisions.

For traditional organizations the disadvantages of agile may be that it has a lower level of predictability around cost, schedule and scope which can make it harder to define a business case and negotiate fixed price projects. As a result the approach is currently not that well accepted by senior management, although that may change. Many would however argue that a waterfall approach is no more predictable than agile as its plans and budget are based on changeable assumptions. On large projects with many dependencies, the changing priorities of the agile approach can be difficult to manage, especially if resources are scarce and shared amongst many projects. Another limitation may be the constant need for user feedback. The users or product owners need to be ready and available for prompt specification and testing of features, and although this helps ensure quality, it involves a higher time commitment than a waterfall approach.

### **The middle ground**

The differences between waterfall and agile can be significant and in most cases it would require some adjustment for a team to switch from one to the other. In between these two approaches are many variants which are more or less iterative in nature. The middle ground contains methodologies which are neither extremely agile nor extremely rigid. They are often iterative in nature as they aim to build the solution gradually. The team may start off creating a fully functional version of the core system and then iteratively add functionality until it is complete. These methodologies would not however be classified as agile because their iterations are likely to last for months rather than weeks and because their products may not be released to the customer on an incremental basis.

Compared to waterfall, these methodologies can seem flexible as they allow for change to take place from iteration to iteration. Compared to agile, however, they are likely to appear rigid and far less collaborative.

### Questions ...

Think about the software projects you have been involved in to date and which development methodologies you used.

- What determined the choice of methodology?
- Which lessons did you learn?
- If you were to start up a new software project tomorrow, which factors would you take into account in order to decide on which methodology to use?

When deciding on a development methodology, think about how much flexibility it should contain and which aspects you can employ from different types of approaches to best suit your needs. Examine the nature of your project, the make-up of your team and organization and what you want to achieve. Then piece together the methodology and tailor it to your specific needs.

### Exercise ...

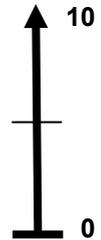
In order to help you decide how to select your software development methodology, consider each of the below aspects and questions.

As you read through them, think about your current project, and seek to evaluate each aspect as honestly as possible. Use the arrow to the right to indicate with a cross how high or low you believe the score should be. The higher your score, the higher the chance that a traditional waterfall methodology would work best. The lower your score, the higher the chance that a flexible or agile approach would work best.

As always, bear in mind that all projects and circumstances are different. In some situations it may even be that one of the below aspects could work both for and against a certain choice of methodology.

**1 How clear are the project's objectives, goals and requirements?**

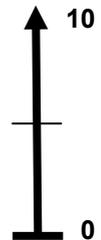
In some cases the project sponsor and end users know exactly what they are looking to achieve and what the success criteria of the project are. In other instances, it can be difficult to pinpoint what the project needs to achieve or what the requirements are. Imagine a project that sets out to build a new order management system for a newly established company which does not quite know their business processes yet, or a customer who wants a new user-friendly interface without being able to define what user-friendly means. When the requirements are difficult to define, or likely to change significantly during the course of the project, you need an approach which is relatively experimental and flexible; an approach which allows you to speedily demonstrate and prototype ideas to the customer and incorporate changing requirements.



Provide a high score on the arrow the clearer your project's goals, objectives and requirements are.

**2 How much general consensus is there around requirements?**

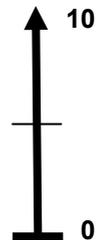
When looking at your stakeholders and user groups, how much consensus would you say there is with regards to the requirements? Is there broad agreement and would it be relatively easy to specify the requirements up front? Or is there very little agreement and would you need to gradually shape the product before consensus is obtained?



Provide a high score the broader consensus there is.

**3 How clear and well-defined is the solution?**

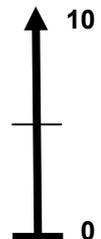
You may find yourself in a situation where the desired outcome of the project is clear, but where the solution for achieving that outcome is not. When the team is faced with difficult and risky choices around technology, design and implementation, your software development methodology needs to be flexible enough to cater for this and iterate through various stages of prototyping, development and roll-out.



Provide a high score on the arrow the more certain your proposed solution, design and choice of technology are. Uncertainty equals high risk and should be reflected with a low score.

**4 How difficult is it to access the end users (or user representatives)?**

Are the users, domain experts and product owners likely to be available to provide feedback and help test deliverables at short notice, or do you anticipate that they could be difficult to get hold of? Would it be easier to engage them for a pre-planned test cycle where lots of deliverables can be tested at once? Would an agile approach be difficult to sell to the users?

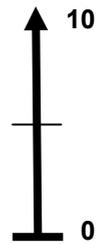


The more constraints there are around your user's availability, the higher your score should be.

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

Is your project on the larger side or is it reasonably small and lean? The larger the project, the more difficult it might be to incorporate lots of changes at short notice and refocus the team in a different direction. Small tight-knitted teams which work closely with the customer and end users are more flexible and better geared to quickly incorporate changes.

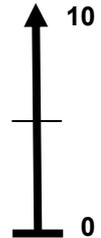
Provide a high score the larger the project.



**6 How dispersed is your project team?**

Do you have several project teams situated in different geographic locations or is the team more homogenous and located together? When the team is geographically spread out it can be difficult to quickly coordinate new work and adapt to changing priorities. Teams which sit together, and even share the same room, are much better placed to working in an agile environment with changing priorities.

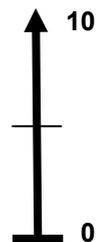
The more dispersed your team is, the higher your score on the arrow should be.



**7 What is the team's and stakeholder's experience with these methodologies?**

Are your team and stakeholders already accustomed to working with a specific methodology? If so, bear in mind that it takes time to change a team's working practices. If your project is time critical, be wary of making too many drastic adjustments to the existing methodology as it may jeopardize the timescales and success of your project.

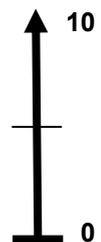
The more experienced your team is with waterfall methodologies, the higher your score and vice versa.



**8 What are the project's success criteria?**

Which is more important for your sponsor and stakeholders? Having a relatively firm estimate and schedule which must be adhered to, or ensuring that the end deliverables add value and match the user's needs and expectations? Is the steering committee ready to embrace a more flexible and quality-conscious approach at the expense of fixed costs and schedules?

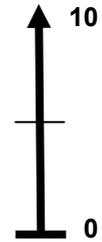
The more important fixed parameters such as scope, budget and time are, the higher your score should be.



**9 How much value would incremental feature driven development add?**

Does the project lend itself to being delivered in an incremental way and would this add real value to the stakeholders? Must the products and features be delivered at once in order to provide tangible benefits or would it be a real advantage to deliver them gradually?

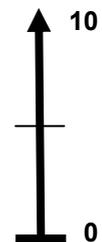
If iterative and feature driven development does not add a great deal of advantage to your project, you should provide a high score. If on the other hand there are numerous benefits associated with gradual releases, you should provide a low score.



**10 Is your customer internal or external?**

When working with external customers the need for fixed budgets and a locked down scope can be greater than when working with an internal customer. If your customer is internal you may also have better access to the end users – although not necessarily.

Provide a relatively higher score if your customer is external and the need for fixed budgets and locked down scope is greater.



**Questions ...**

- Look back through the above aspects and questions and evaluate your scores. Did you generally seem to place your cross above or below the middle line?
- What can you conclude from this exercise?
- Do you believe that your project's current development methodology is the best option for your project, customer and organization?
- If not, what can you do to make a positive change?

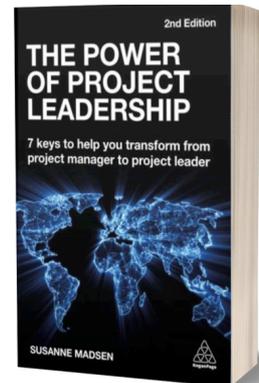


**Susanne Madsen** is an internationally recognized project leadership coach, trainer and consultant. She is the author of [The Project Management Coaching Workbook](#) (2012) and [The Power of Project Leadership](#) (2015, 2019).

Prior to setting up her own business, she worked for 17 years in the corporate sector leading large change programmes of up to \$30 million for organisations such as Standard Bank, Citigroup and JPMorgan Chase. She is a fully qualified Corporate and Executive coach, an NLP Practitioner, DISC accredited and a regular contributor to the Association for Project Management (APM).

Susanne specialises in helping managers improve their leadership skills so that they can gain control of their projects and fast-track their career. combination of training, coaching, mentoring and consulting.

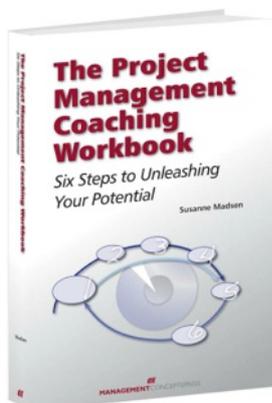
***"I loved this book the first time, and the revised edition is even better. Fully updated, this edition goes deep into strategies to help you lead yourself and your team more effectively. The book goes beyond theory - although you can tell the text is well-grounded in research - to give you practical tactics that you can use daily at work. When you combine powerful techniques and a willingness to do the work, you have a blend that will help you excel at delivering organisational strategy through projects. Susanne makes it easy by stepping you through the 'keys' to leadership, and you can pick and choose the approaches you think will make the most difference with your team."*** Elizabeth Harrin, The award-winning blogger behind [GirlsGuideToPM.com](#)



★★★★★ **Genuinely a Masterpiece on Project Management and GTD!**, July 4, 2012

By [Atif Hassan](#) **"The sweetest language..Bonjour!"** (Toronto, Canada) - [See all my reviews](#)

REAL NAME



***The project Management Coaching Workbook – Six Steps to Unleashing Your Potential.***

*"If I were to choose ONLY two books to get my job done or do a better job as a PM then the two books I'd hold onto would be the PMBOK and Susanne's Project Management Coaching Workbook. If PMBOK is the theoretical framework, then Susanne's book is the Real World Application.. this is where the rubber hits the road. This is by far the most well written book I have read not only on the subject of Project Management but also from a Get Things Done perspective. The way she explains and assists with the GTD approach is incredible."*

—Atif Hassan, Toronto, Canada