



When, Why, and How to Use an Agile-Waterfall Hybrid Methodology



What is the Agile-Waterfall Hybrid methodology?



The adoption of Agile is steadily growing across all verticals, caused largely by the documented benefits it could bring to software development teams and companies. However, making the transition to Agile is a complicated process. While some imagine it as simply flipping a switch to instantly realize true 100% Agile, that's not how it works in practice. When product developers and manufacturers realize that they are too invested in a legacy development methodology (typically some Waterfall process), but sense that Agile could bring them benefits, oftentimes they are forced to choose an alternative route to becoming more Agile.

This is typically achieved through the addition of certain Agile elements, techniques and processes, and adapting them to a preexisting (or partly updated) lifecycle or methodology. While such solutions are not regarded as "true" Agile, they can help a team or company get started on the road to implementing Agile. These custom solutions are examples of an Agile-Waterfall Hybrid methodology, the goal of which is to provide a custom, tailor-made process for the individual needs of the company. After all, some claim that's what Agile is all about: using whatever works to increase the efficiency of your development efforts.

Agile-Waterfall is not easy to get right. For developers and managers used to Waterfall processes, the Agile was is counter-intuitive and sometimes downright scary. However with adequate understanding of all processes, planning, and controlled execution, Agile-Waterfall Hybrid solutions can be highly effective.

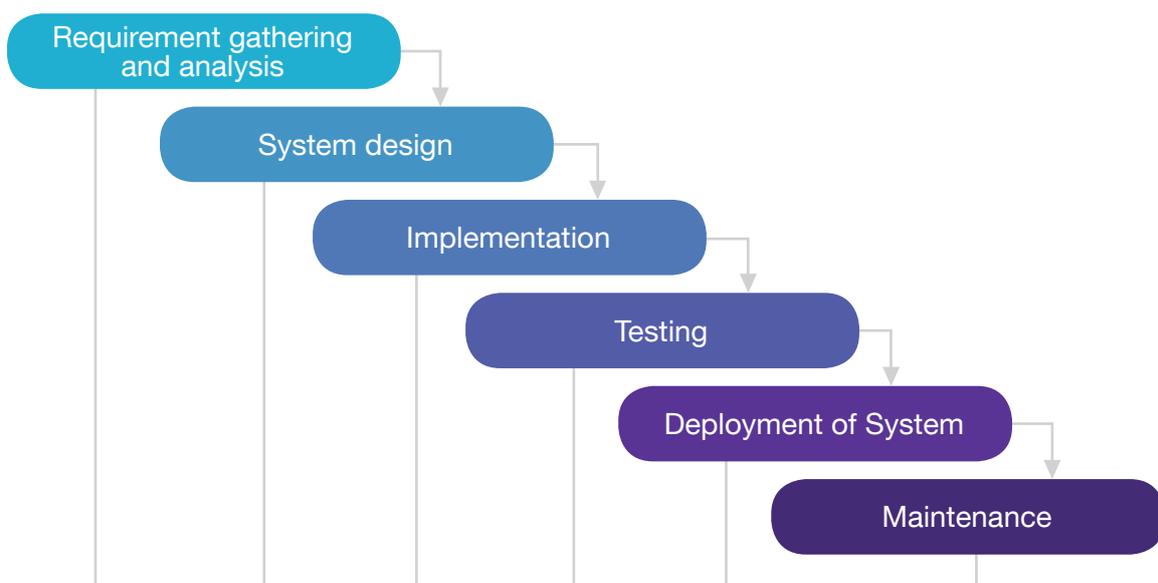


Introduction to Waterfall

**What is Waterfall software development?
Why is it important, and what are its
benefits?**

The Waterfall model of development emerged from the manufacturing and construction industries – in other words, from companies developing and manufacturing hardware products. Waterfall software development is an adaption of this method, and is today mainly used for the development of software embedded in hardware products. Due to the prohibitive costs associated with unplanned changes to hardware products, the Waterfall development model evolved in such a way as to avoid changes later in the development lifecycle.

An extensive planning phase is followed by a sequential order of lifecycle stages, with product delivery at the end of the cycle. Each phase must be completed before the next downstream phase of the process can start, which is where the Waterfall name originates (see the below Waterfall diagram). That's why it is referred to as a linear-sequential lifecycle model.



The advantages of the Waterfall model



Easy to manage due to the rigidity of the model with clear deadlines



Simple and easy to understand and use



Each phase is run through once, with no overlap, allowing thorough QA at the end of each phase



Works well on small projects with fixed requirements

Disadvantages of the Waterfall model



Difficult and prohibitively expensive to change the product once in testing



High level of risks (risk management done in the planning phase)



Not good for projects where requirements are expected to change



No completed product until late in the SDLC



Unsuitable for complex projects

Waterfall provides a structured, planned approach to development that works well in embedded software development, when the project is of short duration and the technology is well known. As soon as the duration is extended and changes start coming in, that's when you might feel the need for a new approach.



Introduction to Agile



How and why did Agile come about, and why is it becoming increasingly popular?

To be 'agile' in business is to be able to react to the unexpected, to be able to adapt to a changing market environment, and to respond to consumer needs quickly. While these qualities are valuable in all walks of the business life, they have gained special importance as software development has evolved.

Agile development has been devised as an alternative to the Waterfall methodology. As opposed to Waterfall's rigid structure and inflexible sequence of predefined development phases that follow one another, Agile has been created to offer more flexibility. With increased flexibility comes greater velocity (shorter time to market), increased collaboration, and ideally, the better fulfilment of requirements due to regular revisions.

Agile software development is an incremental approach to the delivery of software which has developers build the product in small pieces as opposed to the Waterfall approach, which delivers software as a final product at the end of development process. In Agile, the small pieces of the project are comprised of bits of user functionality. Each of these segments are developed, tested, and reviewed to see if they perfectly fit the end users' requirements. Then, the next short phase begins, either with adjustments to the previously completed piece of software functionality, or with the next one in the backlog.

Requirements management under Agile is greatly supported by user stories. A user story is a practical way to capture requirements, which makes it easy to define not only the required feature, but also what purpose it serves (e.g. the benefit the feature could bring to the user). User stories are prioritized and produced in iterations (in development cycles of approximately 2 weeks).



The short duration and size of user stories makes it easier to go back and change things if necessary – and since only a small portion of code is produced at a time, it's easier to make sure the end product perfectly suits the needs of the end user.

The greatest benefit of Agile development is that it enables project teams to update the plan constantly, without this negatively impacting the success or speed of development. Changes could occur, for example, in response to customer feedback on recently completed features.

After the user feedback is processed, the feature could be altered or updated within a few days or weeks. This reduces the need for an upfront and heavily documented planning period, contributing to reduced costs and speeding up the development lifecycle.

Benefits of Agile Software Development

Productivity

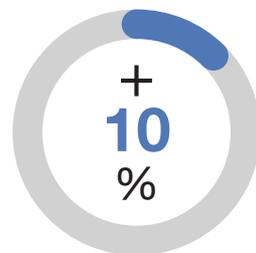


Project delivered on time and on budget



Reduced cost

Engagement

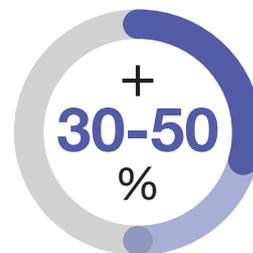


Better collaboration & communication



Satisfied project team and customers

Quality



Decrease product defects



Increased customer satisfaction

Time to market



More responsive for market changes and customer's demands



Shortened product delivery cycles, increased frequency

Agile processes address the lack of flexibility, enabling developers to manage changing requirements. With Agile, developers can produce and release features and then continue to add additional features thereafter and even go back and change features. Agile consists of or is closely related to several development methods such as Scrum or XP. Waterfall, and any of these Agile practices are increasingly being fused, creating custom hybrid methodologies, which are tailored to meet individual business requirements.

Success rate of working with Agile vs. Waterfall



Standish Group, an independent international IT research advisory firm, publishes its CHAOS reports every year. The study analyzes 50,000 software development projects around the world to assess the rate of successful outcomes, and learn more about the factors that could contribute to these projects' success.

According to the latest CHAOS report 2015, on the basis of all projects regardless of size, Agile projects are 350% more likely to be successful than Waterfall projects. This difference is minimal when running small projects: only 32%. But at the huge project end of the spectrum, Agile projects are 600% more likely to be successful.



Project size	Method	Successful %	Challenged %	Failed %
Large	Agile	18	59	23
	Waterfall	3	55	42
Medium	Agile	27	62	11
	Waterfall	7	68	25
Small	Agile	58	38	4
	Waterfall	44	45	11
All	Agile	39	52	9
	Waterfall	11	60	29

Does this mean that Agile methods are universal silver-bullet ways to success? Is Agile simply better than Waterfall?

The general answer is very simple: no, any methodology by itself is not a recipe for success. The question of Agile is not so black and white. Projects can be different in terms of duration, budget, complexity, number of stakeholders, etc. Agile is not a „one size fits all” approach, and the degree to which Agile is implemented also varies. Waterfall projects are better suited for certain types of projects, while Agile fits others better.

In general, we can say that smaller, less critical projects with more local scope and with more focus on the end user are better suited for Agile. That said, several global enterprises that operate in highly regulated safety-critical industries are also using scaled Agile. The world’s leading medical technology company, Medtronic is a prime example of this: Medtronic’s Neuromodulation department chose to implement scaled Agile using codeBeamer ALM. [Read our case study of this process here.](#)

Agile-Waterfall Hybrid: Smart Approach or Terrible Solution?

One of the most popular “mixed” methods in use today is the Agile-Waterfall Hybrid methodology as described by Erick Bergmann and Andy Hamilton.

This model is far from perfect and as with all Hybrid methods, it is a compromise. It has as many advantages as disadvantages, but it is widely considered a smart approach for adopting both methodologies without compromising too much – essentially, making the best of both worlds. The most important details to know about Agile-Waterfall Hybrid:



Tight, continuous integration between Waterfall and Agile development processes from product concept all the way through to validation and production. As with all product development methodologies, collaboration is key. The Agile-Waterfall Hybrid method enables teams to define, and adapt to changing requirements in a collaborative manner. The Agile-Waterfall Hybrid method supports feedback-driven development.



The Hybrid model is aptly suited for reusing software code, when dealing with a series of similar product variants, and when future products must also be considered. In such situations, a quick turnaround time may be needed to keep pace with fast product releases. Backlog management is a critical area for the successful adoption of this Hybrid model, and adoption is best assisted by software version release planning tools.

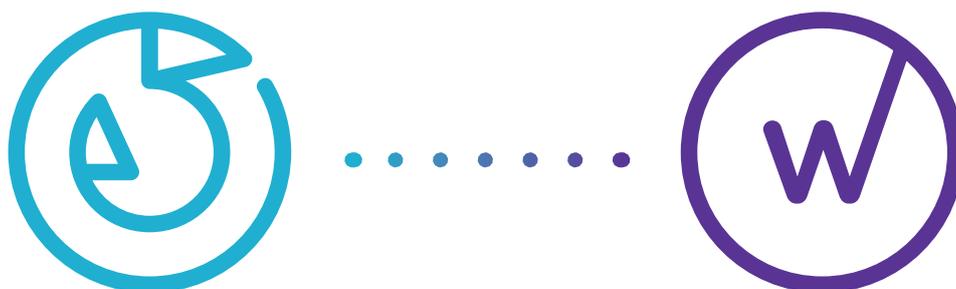
Implementing Agile-Waterfall Hybrid allows software teams to work Agile, while hardware development teams and product managers can keep using a traditional Waterfall approach. The use of Agile will be limited to certain areas, and these areas require a deeply rooted cultural transition to the new framework. ►

As with all Hybrid models, both sides must compromise. Waterfall development must give up some of the certainty of fixed expectations, for the flexibility and freedom of the Agile world. The Agile compromise is to be creative but with far less freedom, working against a fixed deadline with adequate planning to forecast costs, estimate effort, and assess risks.

On the upside, a Hybrid model overall allows larger teams to reap efficiency benefits, collaborate, and take advantage of these seemingly contradictory process frameworks.

Typically, those implementing and managing a project with the Agile-Waterfall Hybrid method are PMP-trained managers that are trying to make Agile fit the Waterfall mold. By no means does a manager's PMP certification automatically make them good candidates for Agile management. The right balance of compromise between Agile and Waterfall, and adopting Agile in a previously Waterfall-only environment requires a seasoned experts.

One common example of points of friction is when developers used to working with Waterfall want to finish all the features before the deadline by simply sticking a bug fixing process at the end of the lifecycle (as in traditional Waterfall development), while testers and managers invested in Agile want the features to be fully tested by the end of each sprint. In this situation, the compromise has failed, and the resulting internal conflict can be seen as a failure to introduce the Agile working culture. It means that the alignment of processes and development strategies has failed – therefore, managers rather than developers are to blame.



When-Why-How to use Agile-Waterfall Hybrid



The Agile-Waterfall model aims to retain the dependency-tracking and clarity of Waterfall, while embracing the strengths of the Agile methodology, providing the flexibility and transparency necessary to adapt to the fast-changing requirements of stakeholders.

When should you use Agile-Waterfall Hybrid?

The Hybrid model can be used in two typical scenarios:

1. The product consists of hardware and software components with the same importance (neither of them being dominant)
2. The product is a piece of software that involves complex back-end and front-end technology (in which case the back-end would be developed using Waterfall).

The third, and possibly just as common scenario isn't based on the characteristics of the project itself. In most cases, software development projects aim to satisfy a customer's needs – and since they are paying for it, they have certain requirements not only for the end product itself, but also the project that aims to build the product. A fixed development schedule and cost need to be estimated during a thorough planning phase. In some cases, the customer will not interfere with how the project is carried out, so Agile may be used with adequate effort planning and estimation. For instance, the Waterfall approach may be used for application design, planning and requirements definition, and Agile (e.g. Scrum) for development and testing.



Why use Agile-Waterfall Hybrid?

“Plan with Waterfall, execute with Agile, and speed up the whole process.”

Waterfall is not a speedy methodology, as deliverable results can only be shown at the end of the whole cycle. Agile shortens the delivery time, and allows you to gather feedback in early stages to better suit changing requirements.

Therefore, the main reasons to use the Hybrid model are:

- Accelerates design, and development, but defines project frames including budget and time of delivery.
- Compliance with standards while shortening time to market can be achieved using a Hybrid framework
- Hybrid helps increase collaboration between teams – therefore, it can be seen as a first step towards “real” Agile
- Waterfall’s rigid structure doesn’t encourage cultural changes. Implementing a brand new framework helps adjust culture and align it with strategic goals.

One greatly important detail to remember is that Hybrid models needs strong collaboration between the two types of teams. Therefore, special communication forms and channels have to be implemented. **Usually those using the Hybrid model need training for both types of team to understand the fundamentals of the new approach.**



How to use Agile-Waterfall Hybrid?

As explained earlier, implementing a Hybrid approach requires experience as there's no one size fits all solution. Therefore, the tips below are merely further ideas on what to pay attention to. Overall, it's important to understand that when it comes to implementing an Agile-Waterfall Hybrid solution, there are no "rules". However, learning from the mistakes of others, there are a few best practices which you may be able to use as guidelines:

- Planning, design and requirements definition may be done with Waterfall techniques, although capturing requirements is a process that could greatly benefit from Agile practices. Development and testing should be conducted in an iterative, incremental manner (in short sprints) using Agile-Scrum.
- You'll have to devise and implement new types of communication channels, and most likely you will also have to put in use an adequate collaborative software solution to support data sharing, communication, and traceability.
- Involve Product Owners and clients in communication regularly – rather than inviting them to participate at certain stages, make sure all relevant stakeholders are deeply involved in the process of development throughout the lifecycle.
- Define standard compliance expectations and focus on them in the design-planning phase. There's no reason why you shouldn't accelerate product development in highly regulated industries, but defining and enforcing compliant processes throughout the lifecycle requires special attention, and software tools with powerful workflow management capabilities.



Hybrid in Practice: How to Make the Hybrid Approach Work?



As we've clarified in the sections before, it is generally considered a successful model to employ Waterfall for overall project planning and hardware development, while software development processes benefit from the flexibility of an Agile approach.

Even with software-only projects, injecting Agile practices into a Waterfall process could yield major advantages. Primarily, one of Waterfall's most loved characteristics, the initial planning phase itself can be done in an Agile way, using tasks and sprints. Next, development phases that can be handled simultaneously and should not have to wait for each other: while keeping Waterfall's "step-by-step" approach on a higher level, it is a good idea to incorporate the Agile principle of starting parts of projects as soon as possible.

It's also advantageous to gather customer feedback every way possible early on the development process. Clickable wireframes can be used to showcase products even before development begins, to make sure a user-friendly application is created that matches the client's needs. After the prioritization of tasks, project planning and scheduling should be done in sprints – user stories, story points and burn-down charts help keep the project on track. Once development begins, it's time to rev the engines to those QA & testing processes. The Agile practice of incorporating testing in the development ensures that a quality product will be shipped to the client.

In addition to Kanban boards that help manage development processes, another Agile practice that even Waterfall projects can benefit from is **the retrospective meeting**. Basically, these meetings serve to clarify what activities the team needs to start doing, stop doing, or continue doing. What went well? What didn't go so well? What have I learned? What still puzzles me? – these are all questions that a retrospective meeting helps answer.

Making the Case for Agile Adoption

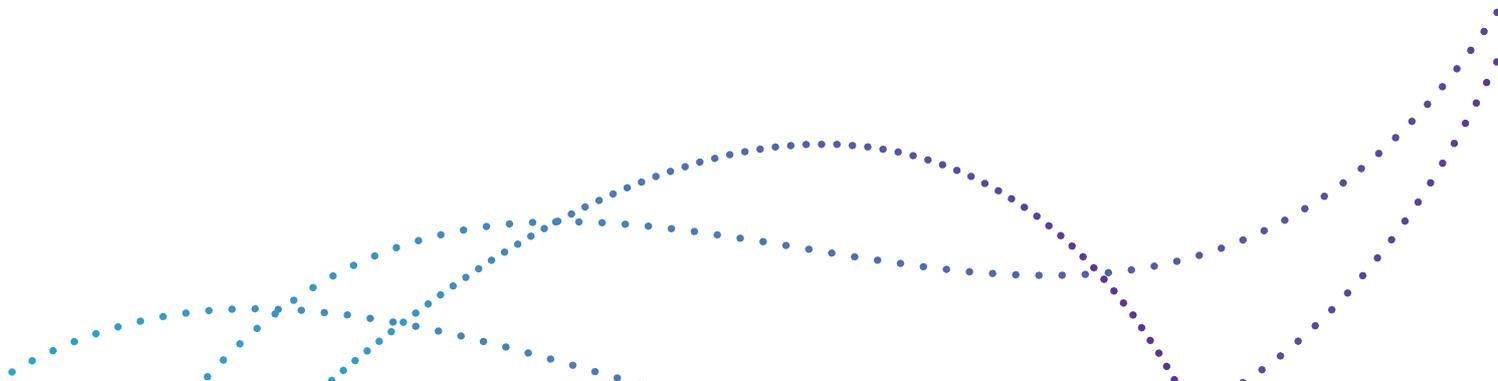


One of the less frequently mentioned requirements of Agile adoption is having the leadership onboard. It's not as trivial as you might think: while Agile definitely has its documented benefits, legacy processes proven to work well are difficult to change, even with the promise of improving them.

Many large enterprises still use some custom version of the Waterfall method, and are wary of replacing their tried and tested processes with new ones.

In such situations, getting the leadership on board can feel like an uphill struggle. That's because Agile inherently involves a cultural change – one that, in several ways, requires stakeholders to take a 180 degree turn and start thinking a whole different way. This is especially important where current methods work, and have been in use for years successfully. The fact that they're good doesn't mean they can't be improved. This section aims to give you guidance in making the case for Agile/Hybrid adoption.

Besides the fact that Waterfall can be so deeply entrenched, there may be other problems. Often, on a large project, individual development teams lack the power to make decisions on their own segment of the project, including decisions about development methodologies to use. There might also be some resistance due to the alleged difficulties of having one team working differently via Agile while other teams using Waterfall, which some claim could be disruptive to the project. This is especially true in the case of large, distributed projects, where multiple development teams, departments, and contractors might be involved.

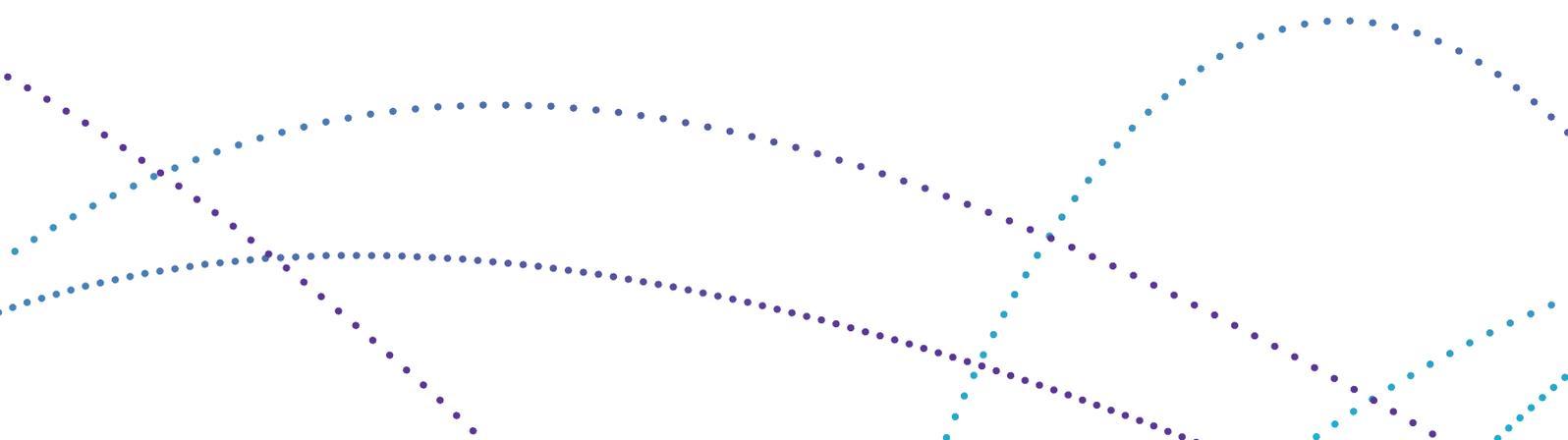


The next difficulty in convincing management is, paradoxically, a great argument for the adoption of Agile. Orchestrating the work of several teams using different methodologies is without doubt a challenge. Sure enough, managing complexity in this situation necessitates the use of powerful tools that have been designed exactly with this purpose in mind. From the management point of view, this might be an unnecessary extra investment.

Having to glue together separate ALM, PLM and single-point solutions used in the innovation, engineering, development and production of hardware, software and services is not only an immense task, it also seems pretty pointless. The purpose of adopting a Hybrid method is to unlock the benefits of Agile – limiting these benefits with clunky, manually integrated solutions doesn't make much sense.

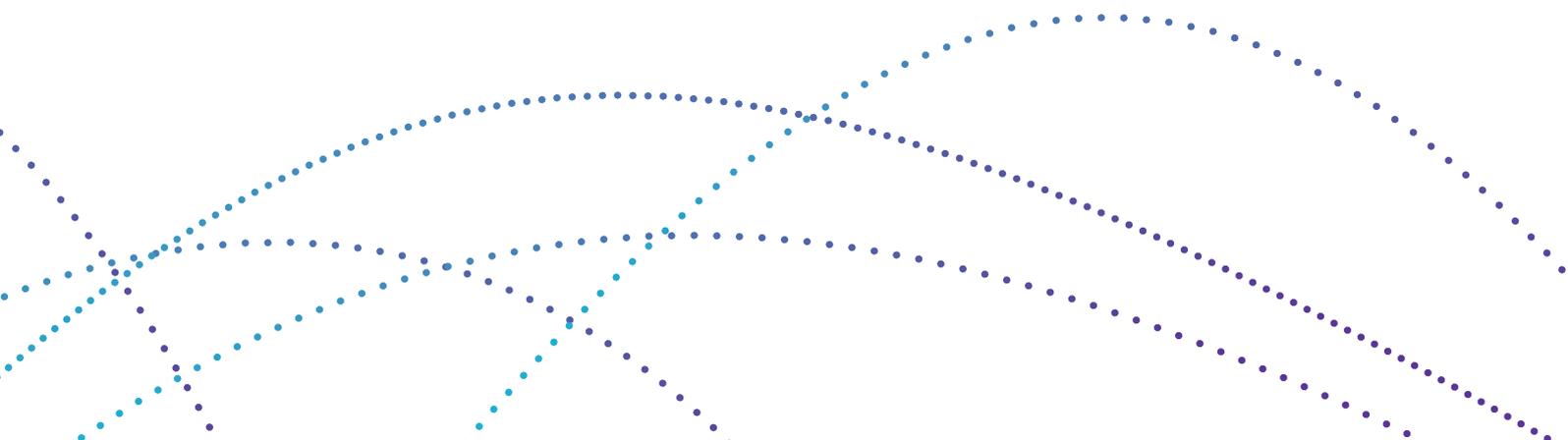
Therefore, fully integrated software tools such as codeBeamer ALM aren't just useful in the adoption of Hybrid lifecycle processes. Rather, they are instrumental in being able to efficiently implement such complex, custom methods of product development. What's more, the powerful features and organic integration that such tools provide can bring widespread benefits to the entire organization. Therefore, having to make this investment is actually on the "pros" list, rather than the "cons".

This affects the discussion with management in several ways. First, of course, there's the question of how much this tool investment will cost the company. However, when talking about investment, it's important to make the difference between price and costs: price is what you pay for the software tool you're planning to use, but calculating the cost takes into consideration the ROI and benefits the tool could bring. Focusing on return on investment, efficiency gains, and collaborative synergies helps make a case for the adoption of Agile in a previously Waterfall-only environment.



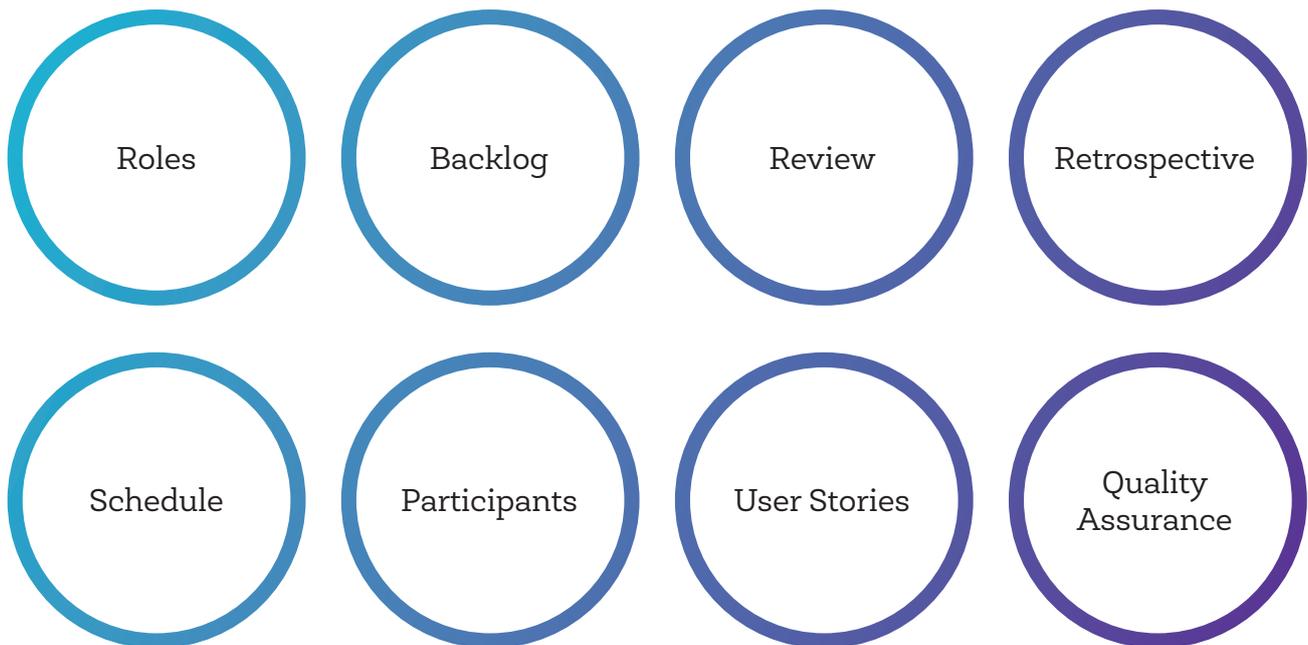
Typical arguments for the adoption of Agile on a Waterfall project

- 
- Waterfall equals overly long planning phases, which may be unnecessary for the particular part of the project in question.
 - Due to its sequential nature, Waterfall also rules out parallel processes, which could mean that some teams have to sit idly while others finish their share of the work.
 - Agile is better at keeping up with changing requirements – some examples from the past where this has caused problems might come in handy here if you want to convince management.
 - High pressure on development, with frequent changes of priority negatively impacting quality. Agile can help resolve this.
 - Agile can help accelerate development to shorten time to market, which will likely lead to increased profitability. Therefore, in certain cases, adopting Agile practices isn't really a technical or methodological question, but a high-priority business issue.



Post-Proposal for Agile Adoption

After a proposal for Agile adoption is made, it is important to build momentum. Don't just sit back and wait for a response, but rather take the initiative and start the proposed plan for Agile integration locally. We suggest implementing Scrum, applying those changes that directly affect your development team. A few of the fundamental aspects, techniques and practices to keep in mind:



The aim is to lead by example and to demonstrate Agile's effectiveness on a small scale, without negatively impacting the rest of the project. Having a proposal with no relevance to your enterprise is very different to showcasing a working example with visible benefits. Approaching leadership armed with facts and figures is often a game changer.

What about the Scaling Agile for Enterprise use?

It can be difficult to get management enthusiastic about implementing Agile even for a single team or department. Once they see the benefits, getting management on board for scaling Agile shouldn't be too difficult – but scaling your custom methodology does pose something of a technical, cultural, and methodological challenge.



There are several “schools” with regards to scaling Agile. Some claim that Agile scales just fine, so you don’t need to worry too much about it – just “spread” Agile across your teams, and you’ll see synergies and increasing benefits. Other experts factor in that large enterprises want to stay in control of all their teams and processes, and therefore, a more structured approach is necessary. While there are several solutions out there on the market to implement this structured transition to Agile, the Scaled Agile Framework (SAFe®) is perhaps the most widely used methodology that enables the scaling of Agile in large, geographically dispersed organizations. The framework’s latest version 4.0 allows the combining of hardware and software processes, and is therefore cut out for the sort of Hybrid approach that this e-book outlines.

You should consider well in advance if there’s any chance of scaling your Agile-Waterfall Hybrid development pattern to the entire organization later down the road. You don’t want to invest thousands of dollars in a tool that isn’t “future-proof”; one that only supports your current processes, but won’t stay as effective as your organization evolves.

Using Process-Agnostic Lifecycle Management Platforms



As outlined above, while somewhat contrary to what the Agile manifesto states, tools are actually quite important when it comes to implementing an Agile-Waterfall Hybrid development method.

That’s because orchestrating parallel lifecycles adds such complexity to your projects that the chance of error is significantly increased if certain processes are not automated.



Process control, enhanced collaboration, and complete transparency are the main benefits of using an integrated platform to manage the Agile and Waterfall lifecycles of a Hybrid framework.

As a tool used by market-leading global organizations, codeBeamer ALM is ideal for Hybrid software development. It helps extend the benefits of Agile tracking & management to encompass Hybrid development lifecycles. Due to its flexibility and process-agnostic nature, codeBeamer can be adapted to any custom process. Its capabilities are best leveraged using Intland Software's Agile-Waterfall Hybrid Template, available free of charge, which has been designed specifically for businesses currently using Waterfall, but wishing to adopt Agile elements in their processes.

In addition, codeBeamer ALM also takes care of your future-proofing worries: as the first implementation of SAFe in the Agile world, it offers several use cases of SAFe-adoption by large organizations. Should you decide to make the transition to full-blown scaled Agile in the future, codeBeamer not only offers the relevant capabilities and preconfigured artifacts to support you, but also brings Intland Software's knowledge of these processes.

To find out more about implementing an Agile-Waterfall Hybrid methodology in practice, feel free to:

Contact us for some support

**Ready to get started with Agile-Waterfall Hybrid?
Start a free trial of codeBeamer ALM to support
any development process.**

<http://intland.com/download-codebeamer/>