

EBOOK

Don't Go Chasing Waterfalls

pitss

Table of Contents

Introduction	3
What is Waterfall Development?	3
Top Issues with the Waterfall Methodology	4
<i>Going Over Budget</i>	4
<i>Risk of Project Failure</i>	4
How Agile Development Can Help Avoid Project Failure	4
What to Do First to Avoid Project Failure	5
<i>Assess How Familiar You Are with Agile</i>	5
<i>Identify Your Organization's Familiarity with Agile Development</i>	5
How Agile Development Can Help Salvage a Failed Waterfall Project	7
<i>Perform a Current State Assessment</i>	7
<i>Identify What's Been Done Wrong on the Technical Team Side</i>	7
Work with an Agile Partner	8

Introduction

In the last decade, the way that we develop software has shifted dramatically. The long term development of monolithic applications with annual release cycles has given way to rapid developments and deployments broken down into smaller, manageable sprints.

The business logic behind agile development is sound: development is driven by data and business needs, not by guesswork. The only way to see if the project is a success is to see it in action. And now, when a project fails, it can fail faster and better, moving on quickly to the fix.

So how do we use the agile methodology to avoid ultimate project failure? What if a project using the traditional waterfall methodology is already on its way to failure? Can we save it using agile? This ebook will help answer some of these questions, discuss the benefits of agile over waterfall development, and provide advice for what to do when met with the possibility of failure.

What is Waterfall Development?

Waterfall is a traditionally taught, old school method of software development. It involves a set process of analysis, design, implementation, testing, deployment over the entire project lifecycle. Waterfall is done in phases: each stage is completed at one time, then the team moves on to the next phase. All analysis is completed, then all design, etc.

This is the way things have been done for a very long time. It's stuck because the waterfall process is easy to remember. It's seen as the "easiest" or "most logical" way of doing things. Because waterfall puts one foot in front of the other, it follows a path which makes sense on the surface.

When you actually try to accomplish a project using the waterfall methodology, everything goes wrong. It's susceptible to many issues, but two of the top issues are budgetary and overall risk.

Top Issues with the Waterfall Methodology



Going Over Budget

When you spend a lot of time doing analysis and design, you're spending money. There are strict standards of documentation to follow using this method, meaning that your team will spend a long time doing spec work. When you finally reach the implementation stage, those developers haven't been involved in the analysis or design processes. So when it comes time to implement, the implementation team was too separate from the process. They haven't seen the designs, they weren't able to give input on how much effort or time it would take for the features at the analysis and design stages. The means the project is more likely to cost even more money as the implementation and testing stages fail.



Risk of Project Failure

Because your team spends so much time doing analysis, design, implementation, and testing, you have to wait through all phases over a long period of time just to produce any results. Stakeholders and decision makers don't get to see a functional version, they don't get to use it until the last phase has been completed. That duration of time can be huge. For a small project, the wait could be 6 months, but for a large project it could be years. If this is a project that will have major impact on the business and they end up with something that isn't useful or doesn't meet their needs, they've lost significant investment.

How Agile Development Can Help Avoid Project Failure



Is this you?

You want to create new opportunities, new revenues, streamline operations in your organization. You have ambition. You see a product that needs to be developed that would impact the business significantly and create new opportunities. You want to be the champion for change. Once you establish that your goal is to own this product, then the fun can begin.



What to Do First to Avoid Project Failure



Assess How Familiar You Are with Agile

When you spend a lot of time doing analysis and design, you're spending money. There are strict standards of documentation to follow using this method, meaning that your team will spend a long time doing spec work. When you finally reach the implementation stage, those developers haven't been involved in the analysis or design processes. So when it comes time to implement, the implementation team was too separate from the process. They haven't seen the designs, they weren't able to give input on how much effort or time it would take for the features at the analysis and design stages. This means the project is more likely to cost even more money as the implementation and testing stages fail.



Identify Your Organization's Familiarity with Agile Development

Because your team spends so much time doing analysis, design, implementation, and testing, you have to wait through all phases over a long period of time just to produce any results. Stakeholders and decision makers don't get to see a functional version, they don't get to use it until the last phase has been completed. That duration of time can be huge. For a small project, the wait could be 6 months, but for a large project it could be years. If this is a project that will have major impact on the business and they end up with something that isn't useful or doesn't meet their needs, they've lost significant investment.

It's easy to manage budgets on an agile project. You divide the work into manageable, ROI-driven sprints. By the nature of the agile methodology, you immediately know what you're setup to accomplish. Those sprints produce ready-to-use deliverables that the customer can see, interact with, critique, and provide direction.

It's almost like you're moving the ball in-by-inch in agile, versus waiting in the end zone for someone to make a Hail Mary touchdown pass. The risk of project failure is mitigated because you have a complete understanding of how the project works after each deliverable sprint. This is called delivering the minimum viable product (MVP), or the minimum value the business needs to get value from the system.

1

Find the MVP

Focus and sprint as hard as you can just to get to the MVP, which means your project is always delivering value that's crucial to the business. The business identifies what that MVP is at the onset, making it easier to reach the goal.

2

Become (or find) the Product Owner

Another way agile mitigates risk is that you have ownership. Product owners are responsible for working with the business unit to determine the MVP. The business unit buys into the product owner. This creates one channel of communication. In waterfall, you can have multiple analysts and designers feeding into the project, creating confusion and conflict.

3

Understand the Business Needs vs. Wants

A product owner should understand the business. They should be able to negotiate, and should determine the difference between a business want and a business need. A business want might be cool and interesting, but it simply doesn't provide enough value. A business need gives higher value by providing the business a way to generate more sales or streamline operations. It isn't just cool or new for the sake of being new.

For example: One of our clients wants a slick documentation viewer inside of their new application. The product owner identifies that the team could spend time developing this viewer, or we could spend time working on a feature that bills customers. As a compromise to the complicated and development time consuming viewer feature, the application instead was built to allow direct file download. It saved our client time to focus time on ROI-driving features that ultimately mattered more to the business.

How Agile Development Can Help Salvage a Failed Waterfall Project



Is this you?

You want to create new opportunities, new revenues, streamline operations in your organization. You have ambition. You see a product that needs to be developed that would impact the business significantly and create new opportunities. You want to be the champion for change. Once you establish that your goal is to own this product, then the fun can begin.

What to Do First to Salvage a Failed Waterfall Project



Perform a Current State Assessment

First, seek to understand the business objectives. Do you understand the business processes being developed and designed? Do you know the stakeholders and supporting business units? Get first-hand objectives straight from the business.



Identify What's Been Done Wrong on the Technical Team Side

What were they doing? What work has been completed that addressed the business's needs? What is the status, if you were going to translate business needs into a backlog of features. Take features to technical team and identify what work has been completed to meet the features.

So your organization's project is already failing. You have to try fix it. That's not a fun place to be. But there's still some hope. Take a deep breath and reset.

You have to re-think the entire project. Think through what has been done. Think about what the business actually needs. Correlate the needs with what has been done and what hasn't. Analyze your product with the MVP in mind: work with the business unit to vote on what features are most important to the end result. Then prioritize your features and follow these steps:

1

Start Over

Throw out failed project plans. Start with a fresh backlog. This would be creating all of the tasks that would need to be done to finish the project. An agile product owner will prioritize what needs to be completed and identify needs vs. wants, draw the red line, and excise unnecessary development.

2

Develop Your Team

It will take a capable team to turn what has been half-baked into something that can actually be useful to the business. Chances are if your organization has experienced a failed waterfall project, you don't have the talent needed to see the big picture. That's where an expert agile team can come in to facilitate the discussion.

3

Get to "Done"

Determine how do to get to the project to a done state. Define what is done, and how do we get there. Use the agile methodology for setting up project plans, defining the backlog, and setting up sprints. An agile vendor will provide a knowledgeable solutions architect to actually get these pieces developed and get the goals to done.

Work with an Agile Partner

Whether you're looking for a way to salvage a waterfall project failure or prevent failure altogether, an expert vendor can help. Enterprise development projects take specialized skills and knowledge to ensure that they're done correctly from start to finish. And most enterprises simply do not have access to this level of talent in-house.



Success

You'll want to see proof of their previous successes. Ask to see demos and case studies of similar systems, projects, or industries. You need to have confidence and trust in your partner so that you can make this long, and often expensive, journey together.



Support

Communication is key. Many development projects fail because of improper communication between internal and external teams. Good communication is at the heart of all successful process changes throughout the agile lifecycle, and is essential to a development project.



Increased Collaboration or Integration

The list doesn't stop there. Work with your leadership to identify the ROI metrics that are right for you. Then, work with your digital transformation partner to agree upon where each metric(s) will come into play and at which step.



Expertise

You'll need the right combination of skills and experience to manage both legacy IT and modern infrastructures. Remember to fill out roles that are lacking within your organization including enterprise architects, software engineers, and analysts. The right strategic partner can help fill in these gaps and drive your agile development initiatives.



Capabilities

Make sure that your chosen partner specializes in your unique systems. They have to know best practices for how to work with and update your systems, while also performing digital transformation using modern methods. They need the right combination of services, solutions, and tools to get the job done.

About PITSS

PITSS modernizes, customizes, migrates, and extends the life of valuable legacy systems. Using data, proprietary software, and deep-dive analysis, we reduce the cost and scope of your digital transformation effort by focusing on the highest ROI processes first. Then, we pave the complete road between you and your new horizons with confidence and expertise. Our full-stack team of experienced UI/UX designers, developers, engineers, technical leads and project managers can guide you through each step of a project. No one knows Oracle Forms like we do—we can take you from planning, to pilot, to the cloud and beyond.



*Meet our team to see how we can transform the systems
that matter most to your business.*

Contact us today.