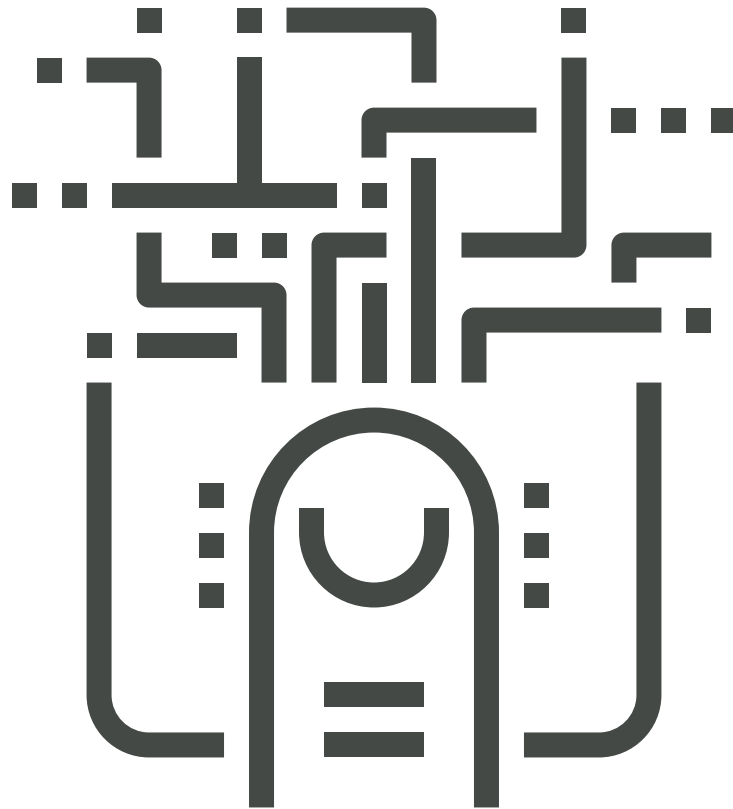


modis

Overcoming the waterfall.

The path to Agile.

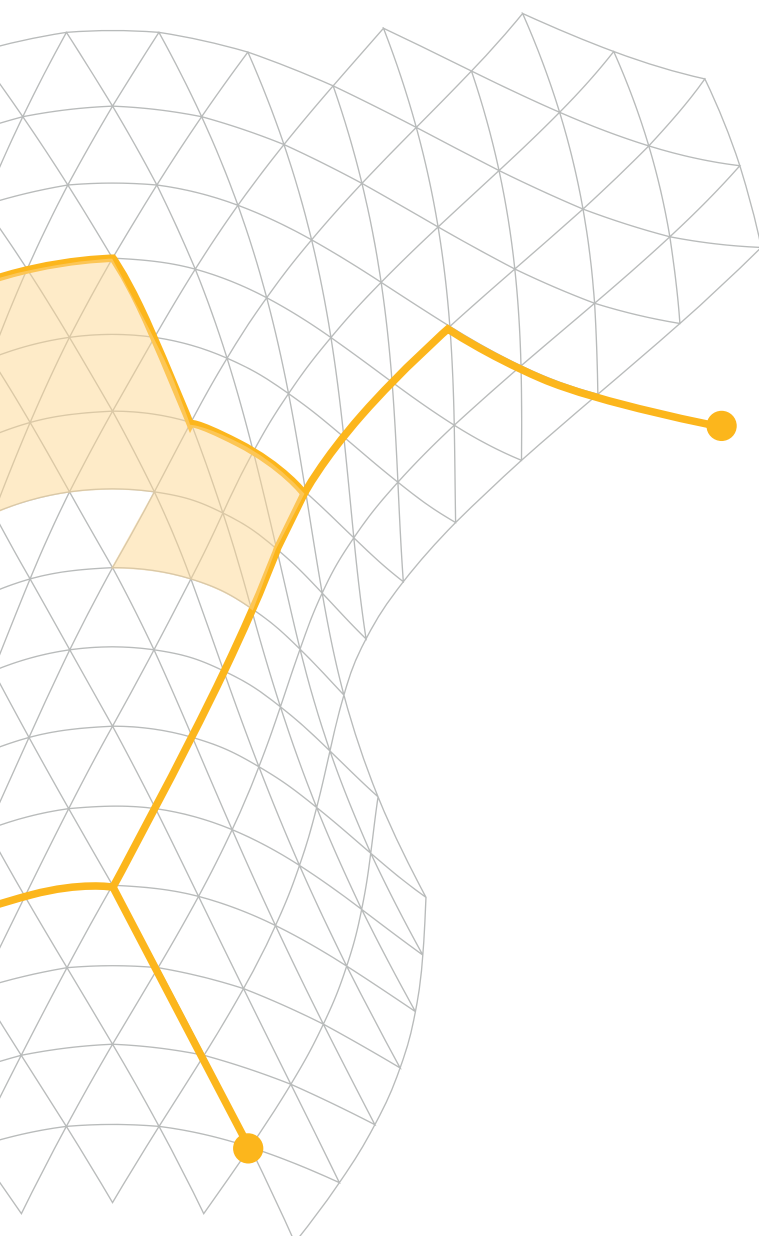




Waterfall based development methodologies are rapidly being replaced by Agile ideologies. Programmers and programming teams wonder what this means for developing their skill sets and tackling the vast array of projects coming down the pike with Agile methodologies. Businesses are wondering if they should build Agile teams to meet the challenges of the future.

Until recently, the waterfall methodology of development and deployment had taken a strategic hold in the majority of enterprises. Enterprises have moved into the world of Agile, where different concepts are leveraged and the development process is handled in a completely different fashion. Agile development ideologies are proving to be a better way to deal with rapidly changing objectives

and speeding the time-to-market for business applications, services and other customer (or internal) facing offerings. The ideas behind Agile extend beyond basic application development, and can benefit numerous projects by improving the process to achieve deliverables and garner improved productivity.



The birth of Agile.

Agile was created in February 2001, when 17 software developers met in Utah to discuss development methods.¹ These developers were bogged down by documentation driven software development processes and the silos created by these methods. They wanted to encourage collaboration and the sharing of ideas. The result of this gathering was the Manifesto for Agile Software Development.

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.²

To break this Manifesto down further:

- **Individuals and interactions:** self-organisation and motivation are important, as are interactions like co-location and pair programming.
- **Working software:** working software is more useful and welcome than just presenting documents to clients in meetings.
- **Customer collaboration:** requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.
- **Responding to change:** Agile methods are focused on quick responses to change and continuous development.

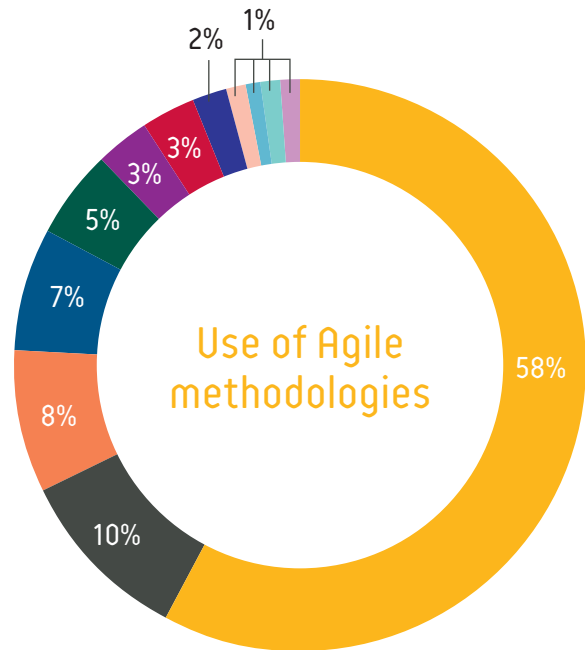
Agile defined further with methodologies

Without defined processes, methodologies and steps, Agile became difficult to implement. This led to the creation of more specific methods in the Agile movement.

The Scrum Methodology is the most adopted methodology used to implement Agile development projects. Scrum emphasises collaboration, functioning software, team self-management, and the flexibility to adapt to emerging business realities.⁴ In this management framework for incremental product development, Scrum teams create product increments during fixed-length iterations called Sprints, which are typically 1-2 weeks.⁵ These increments and iterations allow feedback to be incorporated more quickly than it could be with waterfall development.

Other Agile methodologies include:

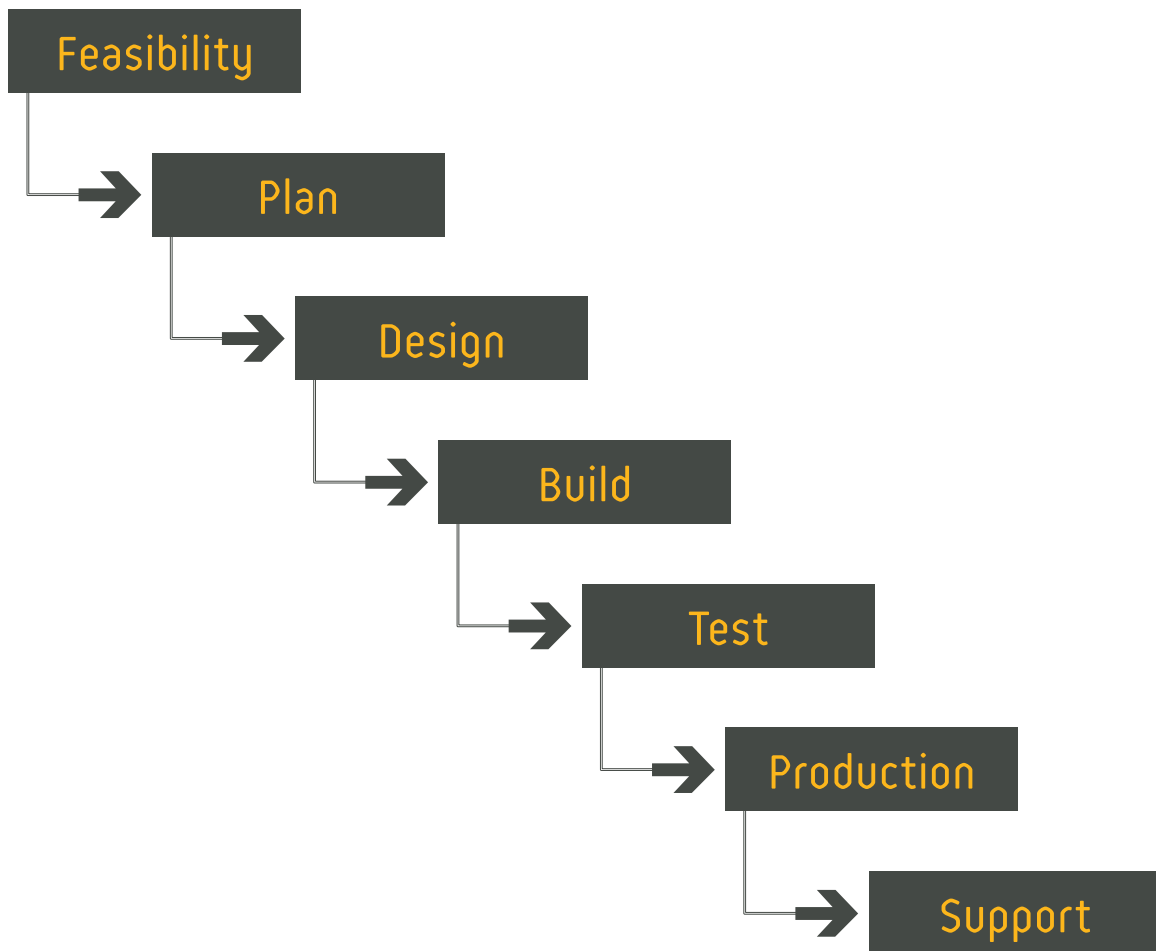
- **Extreme Programming (XP):** Encourages a workflow in which the customer works closely with the development team to prioritise needs (referred to as “User Stories”). Requires close teamwork and continuous testing to deliver software at frequent intervals, typically every 1-3 weeks.
- **Kanban:** Promotes continued collaboration by emphasising continual delivery while not overburdening the development team. This methodology emphasises use of the best possible team workflow, using visualisation to determine what this is.
- **Scrumban:** Combination of Scrum and Kanban that uses Scrum iteration and Kanban processes.⁶
- **Lean:** Focuses on delivering value to the customer and the efficiency of the “Value Stream” (mechanisms that deliver “Value”). Eliminates waste by focusing on feedback from small teams as opposed to a hierarchy and delivering prioritised features in small batches.
- **Feature Driven Development (FDD):** Model-driven, short-iteration process that works in two-week iterations to design each feature.
- **Dynamic Systems Development Method (DSDM):** Works off of requirements and prioritisation. Requirements are planned early in the project and are delivered in short iterations. Rework is a standard consideration and all development changes must be reversible.⁵



- Scrum: 58%
- Scrum/XP Hybrid: 10%
- Custom Hybrid: 8%
(Multiple Methodologies)
- Scrumban: 7%
- Kanban: 5%
- Interactive Development: 3%
- Other: 3%
- Lean Development: 2%
- Agile Modeling: 1%
- Feature Driven Development (FDD): 1%
- DSDM/Atern: 1%
- XP: 1%

There are several different Agile methodologies, including Scrum, XP, Kanban, Crystal and a few others, with Scrum having the lionshare of the market.³

Waterfall development cycle



The waterfall development cycle follows precise steps in a defined order.

Waterfall vs. Agile

The waterfall development model uses an approach that relies on a defined sequence of accomplishments to derive an endpoint. As the name implies, a development product's life cycle flows steadily towards completion, much like how water travels down a waterfall.

The waterfall development method has its roots in industries, ranging from construction to assembly lines, in which sequential processes are the key to success.

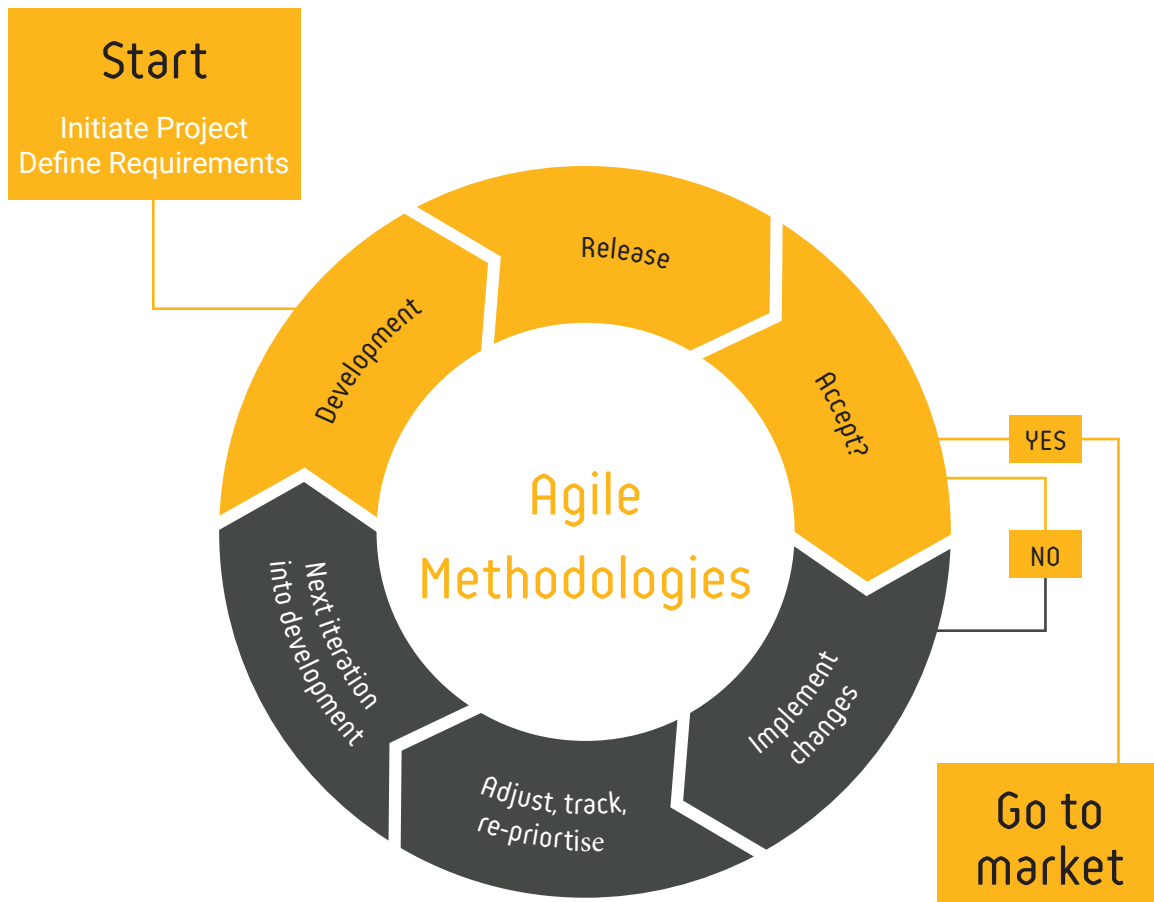
Waterfall methodologies also took hold in the early days of programming, where changes during development projects were costly or impractical.

Applying a waterfall methodology to a programming project meant that all the requirements gathering and design work was done before any coding took place.

Although the ideology behind waterfall isn't old, it was first described in the software development realm by Winston Walker Royce, who wrote a paper in 1970 explaining the methodology applied to software development projects. Waterfall has been the leading methodology for software development projects since its introduction and in most cases, adheres to the seven-step process outlined in Royce's 1970 publication.

The Agile development methodology eschews a linear, sequence-based approach, and uses an incremental, iterative approach. Agile-based projects do not start with extensive planning and design, which means changing requirements can be incorporated into the development process.

Cross-functional teams become an important piece of the Agile method puzzle. They incorporate planners, designers, developers and testers working in concert to create success iterations of the product. Those iterations are completed over fixed time periods (referred to as timeboxes) with the goal of creating a working product, which can be demonstrated to stakeholders for feedback, comments and change requests to introduce in future iterations.



Agile uses an iterative approach based upon a constant development cycle.

Why Agile methodologies work.

To weigh the differences between waterfall and Agile methodologies, it is critically important to know how their characteristics can be interpreted based upon an individual's point of view. For example, a software developer may find the idea of flexibility and reduced test cycles a pro, while a project manager may consider those two elements a con.

The most common pain points of waterfall methodologies include: long waits for releases, inability to make changes throughout the development process, and having to completely restart the development process due to changes in needs. Fortunately, these can be resolved with Agile methodologies.



The top three reasons businesses are choosing Agile³

23%

Accelerate
time to market

16%

Manage changing
priorities

15%

Better align IT/business
objectives

Waterfall pain point

Software product consumers have little input into the development process and often lack the knowledge to understand what is possible and impossible with the software design tools at hand. Waterfall eschews end-user input during the development phase and relies on everything discovered during the design phase to set expectations. Completed products often fail to meet end-user needs because needs may change mid-cycle.

Designers often lack the insight to foresee problems that may arise during development or implementation of the design.

Changes to requirements (e.g. those resulting from new technologies, changes in a market, or changes to business goals) can't easily be incorporated with the waterfall method and there are often laborious change control procedures to go through when this happens.

Waterfall-based development relies on scripted momentum, eliminating the feedback momentum created by the end customer.

Agile resolution

Input is gathered and implemented throughout the development process. Major changes don't derail the process. Usable, working software is created faster and continuously improved with each iteration, creating a culture of continuous improvement for live systems.

Constant testing allows for considerations to be realised and implemented during development. Since Agile fosters a culture of continuous improvement, considerations after launch can also be implemented with ease.

Changes to requirements can be incorporated at any point of the process using the Agile method, even late in development.

All stakeholders and team members are exposed to the process, making this methodology highly transparent. Agile improves collaboration between developers and the business.

The driving forces behind Agile methodology adoption.

Agile methodologies are becoming the go-to ideology to embrace productivity while allowing businesses to react to changes, market demands, and shifting priorities.

Agile methods offer more flexibility than waterfall derived projects. Agile is also customer facing, meaning that customer requests can be implemented on the fly, allowing the creation of software that adapts as needed and can change with each iteration to meet new challenges. For example, if a company using waterfall methodologies is developing software with a green background, then decides they need a blue background, this would be a major delay for the project. Using an Agile methodology, this would be a small change seamlessly integrated into development.

The Agile methodology offers several perceived benefits, which is why the adoption of it as a development practice is growing. The benefits include:

- **Reduced Waste:** Continuous improvement proves to be a faster process, reducing waste in the form of time spent on design, redesign and addressing change.
- **Increased Speed:** Agile's ability to bring forth rapid change, especially in the form of course corrections, means that deliverables can be constructed quicker.

- **Improved Productivity:** Agile is a more effective way for teams to work together, meaning that productivity will increase due to improvements in communications, as well as insight quickly garnered by cooperative goal setting.
- **Enriched Decision Making:** Agile teams share wisdom and help to drive faster decisions based upon needs and not original design goals.
- **Improved Confidence:** With better decisions and a proactive approach, Agile improves the confidence of an organisation. It makes it easier for the product to pivot, which allows the organisation to react better to changing market conditions.
- **Improved Trust and Safety:** Agile has a different approach toward failure: it emphasises learning from mistakes and leverages a blameless post-mortem, which improves the attitude towards failures. Ultimately, learning from failures and eliminating blame leads to improved trust among team members.

An HP online survey of 601 development and IT professionals shows that 51 percent of respondents are leaning towards Agile, while another 16 percent are already using Agile as their sole development method. Only 2 percent surveyed were pure waterfall environments, while only 7 percent were leaning towards waterfall.⁷ This is a clear indication that Agile is taking hold and is quickly becoming the new normal.

The path to Agile.

Adopting an Agile ideology is more than just claiming teams are Agile. Managers and decision makers should understand the components of what makes up an Agile ideology, especially when Scrum is the target ideology.

One of the biggest struggles businesses encounter when shifting to Agile is the cultural shift. While having an Agile evangelist is an effective first step, the need to permeate the culture requires several major shifts.

- Goals shift from making money to customer satisfaction.
- Structure shifts from management overseeing to management enabling self-organised teams to contribute all they are capable of.
- Workflow shifts from bureaucratic to cyclical with direct customer feedback.
- Values shift from efficiency and predictability to transparency and continuous improvement.
- Communication shifts from hierarchical to lateral.⁸

Scrum not only outlines the process, but also defines the roles of those involved.

A Scrum team may consist of:

Product Owner: Shares the vision of the product, prioritises the functionalities to be built, and makes key decisions on behalf of the team. The Product Owner is also responsible for maintaining the product backlog, bridging the gap between the developers and other stakeholders, managing the end-user (or customer) expectations, and managing the budget.

Scrum Master: Primarily responsible for solving problems team members may be facing while building the product. The Scrum Master does not have to completely understand all requirements, but must still be able to find solutions to situations. Ultimately, the Scrum Master is responsible for creating the best possible working conditions for team members so they can meet the goals of each sprint effectively.

Scrum Development Team: A cross-functional team that is responsible for developing the product. The team consists of developers, business analysts, testers and other liaisons. The goal of the team is to work together and in tandem while building the application. Team activities are aligned with the targets associated with the specific sprint. Team members must identify the complexity of assigned tasks and allocate the appropriate resources. Team members are expected to communicate the status of project and issues they are facing to Scrum Masters on a daily basis. Team members also are responsible for demonstrating tasks completed to product owners during sprint reviews.

Agile Terms and Ideologies



Product Backlog: A living document that contains the features desired in a product, as well as the bugs, technical work, and acquired knowledge. The product backlog is managed by the Product Owner, and is continually updated and prioritised.

Sprint: The developer team works in sprints, a classification used to describe an iteration task set. At the start of the sprint, the developers and the Product Owner agree on the items from the backlog they will complete during the sprint. That subset of items from the Product Backlog becomes the Sprint Backlog. During the sprint, the Product Owner is not allowed to change the items in the Sprint Backlog.

Daily Scrum: During a sprint, the developer team meets to coordinate work for completing the items in the Sprint Backlog. Ideally, the team will discuss who is working on what and whether any blocking issues have been discovered. Meetings are kept short and simple, consisting of what someone has done since the last meeting, what is planned for today, and whether there are any impediments. The idea is to share knowledge directly related to the project.

Stories and Tasks: Items in the Product Backlog or Sprint Backlog are referred to as stories. Stories are created by the Product Owner and represent a specific business need. Stories are often broken down into specific tasks so accomplishments can be measured. Without defined tasks, stories can become never-ending stories simply because there are no measurable milestones. Stories are owned by the Product Owner and are all about business value, while tasks are owned by the developer team and are all about implementation details. A story might take several days or weeks to complete; a task is something a developer can complete in less than a day.

Burndown Charts: In Scrum, Burndown charts are used to represent the remaining work on a project. A Release Burndown is built calculating the remaining number of uncompleted Story Points for the entire Product Backlog every day, while a Sprint Burndown chart is used to identify remaining work on a particular sprint.

Your path to Agile starts now.

It's time to begin your transition to the next stage of Agile implementation. At Modis, we help you make exceptional connections to find the Agile-versed talent you need. For more information about Agile or to find top tech talent for your company, contact your local Modis representative today.

Sources

Agile Manifesto. History: The Agile Manifesto. <http://Agilemanifesto.org/history.html>

Manifesto for Agile Software Development. <http://Agilemanifesto.org/>

Version One. The 10th Annual State of Agile™ Report. <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>

Scrum Methodology. <http://scrummethodology.com/>

Version One. Agile Methodologies. <https://www.versionone.com/Agile-101/Agile-methodologies/>

SolutionsIQ. What is Scrumban? <http://www.solutionsiq.com/what-is-scrumban/>

Hewlett-Packard. Agile is the new normal: Adopting Agile project management. <http://www8.hp.com/h20195/v2/getpdf.aspx/4AA5-7619ENW.pdf?ver=1.0>

Forbes. How To Make The Whole Organisation Agile. <http://www.forbes.com/sites/stevedenning/2015/07/22/how-to-make-the-whole-organization-agile/#55e4d665135b>

About Modis and Modis Managed Solutions.

As a global provider of IT staffing services, Modis connects the very best IT professionals to great opportunities at leading companies. And we do it every single day—for both temporary and permanent assignments and across virtually all industries. We use our insight, flexibility and expanded resources to anticipate the needs of leading tech companies and help them make exceptional connections with top tech talent.

Modis Managed Solutions (MMS) is a practice of outsourcing technology projects to a team of specialists that can manage the entire effort - providing thought leadership to assess business needs and deliver the solution. It's an approach of looking at the business as a whole, and not in separate transactions.

The world of technology is a complex place. But from our perspective, we can help you clearly see where you want to go.

To learn more about our services,
or the industries we serve, contact us today.

modis.co.uk

modis