

# **The Sprint to Scrum's Success: A Story of Strengths and Shortcomings**

**ELEC60020: Managing Engineering Projects 23/24**

**By Liam Browne**

**Word Count: 2972**



Photo: PierreSelim, Wikipedia

<b>Introduction:</b>	<b>3</b>
<b>Scrum Origins:</b>	<b>4</b>
<b>Scrum Overview:</b>	<b>6</b>
<b>The Scrum Team and Goal Definition:</b>	<b>7</b>
<b>The Sprint:</b>	<b>8</b>
<b>Agile vs Scrum:</b>	<b>9</b>
<b>Case Studies 1: Switching to Scrum</b>	<b>11</b>
<b>Case Studies 2: Scrum is no Panacea</b>	<b>13</b>
<b>Conclusion:</b>	<b>15</b>
<b>Works Cited:</b>	<b>16</b>

## **Introduction:**

In the ever-evolving landscape of project management, the quest for more efficient and adaptable methodologies has led to the emergence of various frameworks, among which Scrum stands out prominently. Scrum, an Agile framework, has revolutionised the way projects are handled, especially in dynamic environments where traditional methods falter. The following essay aims to investigate the benefits and potential pitfalls of using the Scrum framework. To achieve this, we briefly look at the developments that led to Scrum's creation, including its rugby origins. Subsequently, an overview of the Scrum framework will be provided, as well as an explanation as to how it differs from Agile. At this point, we will be equipped with the knowledge required to identify the successes and failures of using Scrum in various organisations.

## Scrum Origins:

The origins of the Scrum project management technique can be traced back to “The New New Product Development Game”, a 1986 article of the Harvard Business Review.<sup>1</sup> Authored by two Harvard Business School professors, Hirotaka Takeuchi and Ikujiro Nonaka, this article argued that traditional project management techniques that rely on slow and sequential development were dated. Instead, a fast and flexible technique was needed. This was justified with the observation that newer products were increasingly dominating company revenues. For instance, a 1981 survey of 700 U.S. companies found that new products accounted for over 30% of all profits in the 1980s, an increase from 20% in the 1970s.<sup>1</sup>

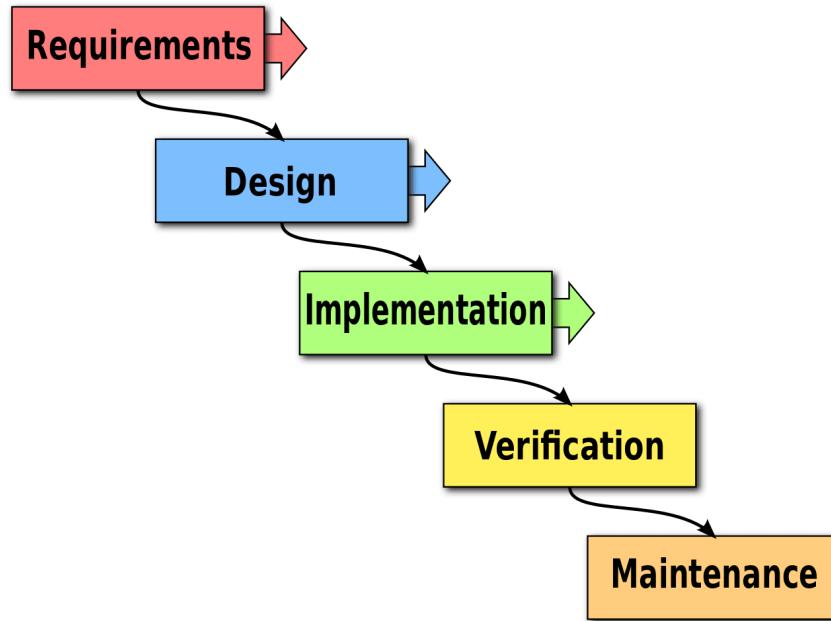


Figure 1: Phases of the Waterfall project management technique.<sup>2</sup>

One of the more traditional project management techniques is Waterfall. This approach consists of partitioning a project into phases, as depicted in Figure 1. Each consecutive phase only commences once the previous phase has been completed. Despite being effective, it is slow and inefficient.<sup>2</sup> Takeuchi and Nonaka likened such traditional techniques to a relay race, where one group would pass the baton on to the next group. As a result, a group remained unproductive until it was their turn. Not only was this slow, it also resulted in inflexibility. Making any major changes to the product in later phases would be difficult as it would have to be

passed back and go through the earlier phases again. To overcome this rigidity a rugby-style approach to project management was suggested.<sup>1</sup>

In rugby, all players in the team must gradually move forward, together, passing the rugby ball back and forth in the process. In a project setting, this would be akin to all specialist groups in a project making progress at more or less the same time, passing the product between each other for each group to work on the product slightly and then passing it onto the next group. If a change has to be made, it can easily be passed back, since the other groups are at a similar development stage. This was faster than sequential project management techniques, since no group had to wait until the previous phases were completed. Rather, they would work on it at the same time as other groups, making progress homogeneously. This technique in rugby was called 'Scrum', giving rise to the name of the new project management approach.<sup>1</sup>

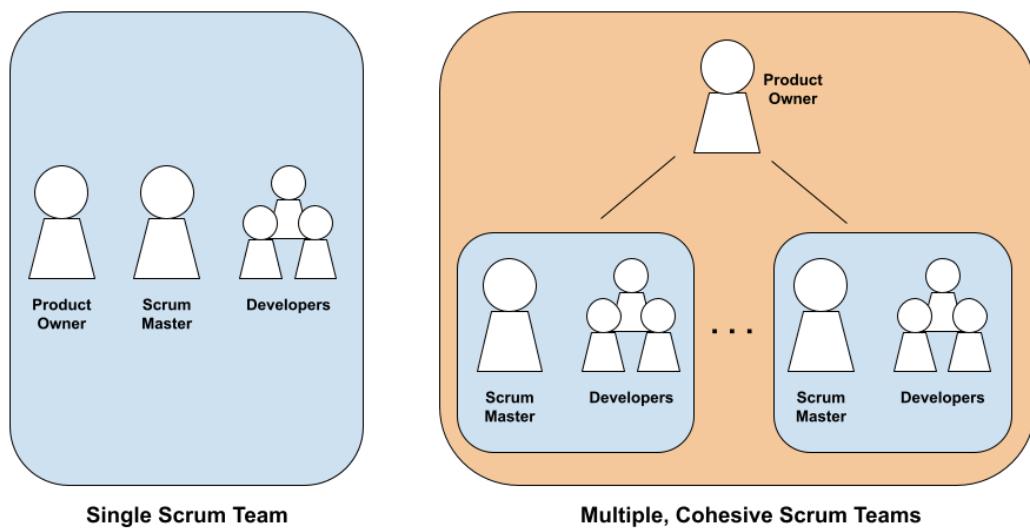
Scrum was later adapted to software development. This adaptation was achieved by Jeff Sutherland and his team at Easel Corporation in 1993, incorporating programming concepts such as iterative and incremental development into the Scrum framework.<sup>3</sup> Following this, software developer Ken Schwaber heavily consulted Sutherland on his approaches to Scrum, publishing the first academic paper on the Scrum framework in 1995.<sup>4</sup> This was the beginning of a decades long collaboration between Sutherland and Schwaber, who have published numerous books on Scrum since.<sup>3</sup> In 2010, they formalised Scrum and its terminology in 'The Scrum Guide', which is regularly updated.<sup>5</sup>

Scrum was adapted for software development early on. As a result, its process has predominantly been used in projects related to software. Nonetheless, due to its emphasis on rapid feedback cycles and promoting continuous improvement of a product, it remains a popular project management technique for any discipline where speed and flexibility are crucial.<sup>6</sup>

## Scrum Overview:

To analyse the effectiveness of a project management technique, one must first understand how it works. Although the underlying concepts of Scrum were first introduced in the mid 1980s, its process is dynamic and always undergoing change, as ways to make it more effective are developed. Its most recent form is summarised in The Official 2020 Scrum Guide, by Sutherland and Schwaber.<sup>7</sup> The following section will introduce modern Scrum terminology and detail its implementation.

## The Scrum Team and Goal Definition:



*Figure 2: Single Scrum Team vs Multiple Scrum Teams. Diagram created by Liam Browne.*

According to The 2020 Scrum Guide, Scrum is executed by the Scrum Team, which consists of a Product Owner, Scrum Master and Developers, depicted on the left in Figure 2. While these three groups may have different responsibilities, there is no hierarchy, such that they function as a group of professionals working towards achieving the Product Goal. The Product Goal is the Scrum Team's long-term objective. This is used to develop the Product Backlog, which is a list of items that must be completed in order to achieve the Product Goal. Scrum Teams tend to consist of 10 people or less to maximise communication. In the case of larger teams, one can

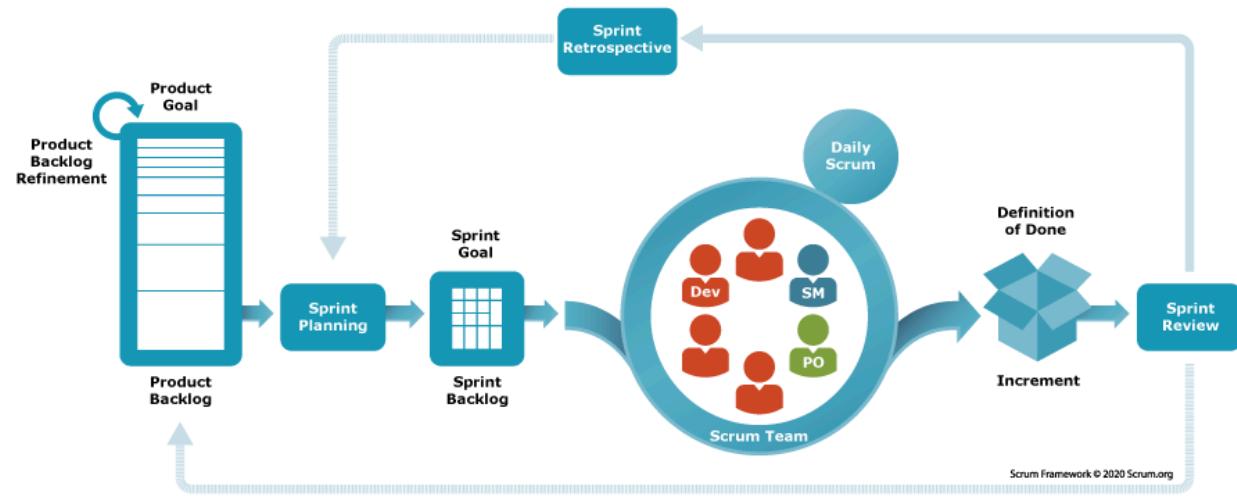
have multiple Scrum Teams, as seen on the right in Figure 2. However, these teams should be cohesive and have the same Product Goal, Product Backlog and Product Owner.

The Product Owner represents the needs of the stakeholders and is, therefore, in charge of developing the Product Goal. As a result, they also create the items in the Product Backlog and must ensure that this has been clearly communicated to the rest of the Scrum Team. The Product Owner is always an individual and never a committee. Their decision is final and they are accountable for how effectively Product Backlog is accomplished.<sup>7</sup>

The Scrum Master is in charge of establishing the Scrum framework and ensuring that all members of the Scrum Team understand how it works. In addition to this, they help the team prioritise goals, overcome obstacles, remain productive and stay within predefined timeframes. They also play an auxiliary role for the Product Owner, helping develop the Product Goal and Backlog, as well as establishing quantitative measures of progress.<sup>7</sup>

The Developers are the backbone of the Scrum Team. They are responsible for making progress towards the Product Goal by creating Increments, adhering to a Definition of Done, creating the Sprint Backlog and adapting their plans on a daily basis to achieve the Sprint Goal. Note, this terminology will be defined in the following section.<sup>7</sup>

## The Sprint:



*Figure 3: Diagram illustrating the repeated phases of the Sprint stage.*

Following the development of the Product Goal and Backlog, one enters the Sprint stage. Sprints are essentially short projects that aim to complete a subset of items from the Product Backlog, called the Sprint Backlog. Each Sprint also has its own Sprint Goal, which is to be achieved by the end of it. This adds flexibility, since the Sprint Backlog can be adjusted during the Sprint, given the approval of the Product Owner, as long as the Sprint Goal remains unaffected. The full Sprint process is visualised in Figure 3.<sup>7</sup>

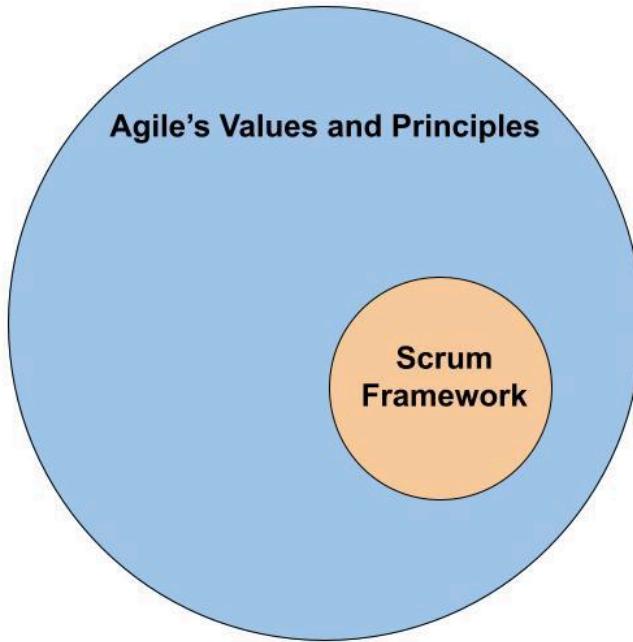
A measure of progress is an Increment. Each item in the Product Backlog has a Definition of Done, which is a set of criteria that must be met to consider that item completed. If an item's Definition of Done is completed, this is considered an Increment. Due to the Sprint Backlog consisting of items from the Product Backlog, multiple Increments can be achieved during a Sprint.<sup>7</sup>

Sprints are executed consecutively until all items on the Product Backlog are completed, thus achieving the Product Goal. Each Sprint usually lasts less than a month. This ensures that the progress toward the Product Goal and any adaptations that may need to be made to the Product Backlog can be identified and implemented at the end of every Sprint.<sup>7</sup>

As seen in Figure 3, each Sprint consists of Sprint Planning, Daily Scrums, Sprint Review and Sprint Retrospect. The Sprint Planning creates a plan for the Sprint by ensuring three vital questions are answered beforehand: Why is this Sprint valuable? What can be done this Sprint? How will the chosen work get done?<sup>7</sup>

During the Sprint, a Daily Scrum is held. This is a short 15 minute meeting that enables communication between the team and helps discuss any issues one may be having. After the Sprint period has concluded, a Sprint Review is held, where the Scrum Team presents their work and progress to the stakeholders. This is also the time to suggest additions or changes to items in the Product Backlog. Finally, in the Sprint Retrospect, the Scrum Team identifies methods to improve its productivity for subsequent Sprints.<sup>7</sup>

## Agile vs Scrum:



*Figure 4: The Scrum framework is a subset of Agile's Values and Principles. Diagram created by Liam Browne.*

Agile and Scrum are often used interchangeably. While they are related, they are not identical.<sup>6</sup> Agile is a set of values and principles, including, but not limited to, short and iterative development cycles, regular reflection, face-to-face communication and flexibility.<sup>8</sup> Despite these values being similar, if not identical, to those of Scrum, Scrum aims to implement these on a product development level. Agile, on the other hand, attempts to implement these values on a leadership and company culture level – which is crucial if Scrum is to be implemented in larger organisations.<sup>6</sup>

Although Scrum is a subset of Agile values, as seen in Figure 4, Scrum was developed 8 years before Agile. Due to the fact that Agile and Scrum are not mutually exclusive, they will often be found in conjunction in larger organisations.<sup>6</sup>

## **Case Studies 1: Switching to Scrum**

Many organisations were using traditional project management techniques before switching to more modern ones, such as Scrum. As a result, many employees within organisations who were familiar with traditional management methods did not always welcome such change. To understand the reasons behind this internal resistance in organisations, we will investigate British Telecom's (BT) and IBM's challenges when transitioning to Scrum, as well as the reasons behind transitioning in the first place.

In 2004, BT found itself in a predicament while using traditional Waterfall project management techniques. Despite successfully deploying a number of large scale IT solutions, it came at the cost of lengthy development periods, ranging from 12 to 18 months. Using Waterfall techniques meant a product's development was inflexible, with proposed changes in later development phases being difficult to implement. In an increasingly competitive market where products with shorter development cycles were being favoured, BT decided to switch to Scrum-inspired management techniques. This came at the cost of retraining their teams and restructuring the roles of their 8000 IT employees. Following its implementation, BT's development period reduced from up to 18 months to two or three 90 day cycles. The short cycles allowed for the scope of the project to be revisited regularly and any requirement changes induced by the environment to be accounted for. In essence, it meant greater flexibility and quicker deployment. BT's implementation of Scrum was successful.<sup>9</sup>

When IBM made its transition from Waterfall to Scrum, it faced a variety of challenges. Not only did they have to invest in the retraining of employees to understand the Scrum framework, they had to restructure entire teams to accommodate Scrum's prerequisite of smaller groups. This meant employees had to reevaluate their responsibilities and adjust accordingly, which was not taken lightly and required an adjustment period. Despite the challenges faced during IBM's transition to Scrum, its implementation resulted in increased employee satisfaction, autonomous teams and shortened production times – without compromising quality.<sup>10</sup>

Switching from traditional project management techniques to more iterative ones, such as Scrum, can increase flexibility, efficiency and transparency in an organisation. This often will not occur without resistance in an organisation, as many employees may be comfortable with more

traditional methods. This results in costly investments to retrain employees and familiarise them with Scrum. Furthermore, many traditional team structures do not accommodate the Scrum framework, since they are often too large. Consequently, organisations must undergo restructuring before Scrum can be implemented, often facing aversion from employees. However, although these initial transitioning costs may be high, they are more than compensated for by increased efficiency and flexibility, as was seen at BT and IBM.

## **Case Studies 2: Scrum is no Panacea**

Scrum's success is only as good as its implementation. The successes seen by BT and IBM were a result of training and internal restructuring to accommodate Scrum – its teams understood its principles and the restructuring meant it could work effectively. However, if implemented poorly, Scrum can result in budget overshoots, delays and even late-stage project termination.

The iterative nature of Scrum allows for flexibility, such that changes can be made to the scope of the project at the end of every Sprint. Having said that, if too much change is made to the scope after every Sprint, the project's scope may eventually grow too large, straying the project from its original focus. This is known as scope creep and was an underlying issue that resulted in the demise of the 2005 SIREN project. SIREN was undertaken by the Surrey police, aiming to create a new ICT suite for its police force, focusing on crime, intelligence and custody. However, due to the management team, as well as stakeholders, lacking proper familiarity with Scrum, the project was subject to scope creep. In 2013, the project was discontinued due to its scope being too large, despite having cost £15 million. SIREN's failure may have been averted by using a Waterfall technique, since management may have been more familiar with this method and it would have kept its scope more rigid. Hence, Scrum should not always be used. Instead, a project management technique that is familiar to management and that matches the project's requirements should be chosen.<sup>11</sup>

Communication is central to Scrum and is the reason for Scrum's requirement of small teams. Effective communication fosters ideas, productivity and ensures everyone is on the same page. This is not exclusive to the Scrum Team itself, but also applies to communication between the stakeholders and the Scrum Team. The consequences of lacking communication are made evident by a 2008 project aiming to develop a scheduling system for an energy-sector organisation. Due to being a large-scale project, numerous smaller Scrum Teams were organised. However, these Teams were located in various locations across Canada, the U.S. and Europe. Effective communication between teams was obstructed by time zone differences and a lack of live communication software at the time, such as Teams or Zoom. Thus, email communication predominated, with teams rarely calling each other over the phone. This resulted in little to no real-time information being communicated, preventing the Scrum Teams

from working cohesively towards the Product Goal. The situation was exacerbated by there existing only a single Scrum Master for all of North America rather than having one for each Scrum Team. Due to this, the Scrum Master was not properly involved in any Scrum Team and therefore not able to fulfil their duties effectively. Furthermore, while issues encountered during the Sprint were discussed in the Sprint Retrospect, no improvements were ever implemented. As a result, unproductive habits persisted in subsequent Sprints. Eventually, Sprint Retrospects and Sprint Reviews were cancelled by the executives of the client company, because they were deemed redundant. In the end, the one year project took over three years to complete, costing five times the original budget; the Scrum framework was used as a scapegoat for the project's failure. The execution of this project reveals how a lack of real-time communication throughout a project can make the Scrum Teams incohesive and therefore lose efficiency. Furthermore, as Sprint Retrospects and Sprint Reviews were gradually cancelled, subsequent Sprints lost their feedback mechanism. In essence, Scrum's core values were stripped away, resulting in great losses of efficiency that led to the project's delay and budget overshoot.<sup>12</sup>

## **Conclusion:**

Implementing Scrum can come at a high initial cost of retraining teams and restructuring organisations to accommodate it, as seen at BT and IBM. However, if correctly implemented, Scrum's framework gives organisations increased flexibility and efficiency over more traditional, sequential project management techniques. Its iterative nature and responsiveness to feedback also allow projects to adjust to new information learned throughout development. Furthermore, Scrum's Sprint Reviews allow for progress to be clearly communicated and discussed with stakeholders, including them in the development process. Also, Sprint Retrospects provide an opportunity to analyse the effectiveness of approaches used in the last Sprint, such that they can be improved upon for subsequent Sprints.

On the other hand, if implemented poorly, using Scrum can incur delays and high costs for projects. This is made evident by the 2008 scheduling system project, where the organisation was unable to find effective ways to communicate real-time information between transatlantic teams – admittedly, they were limited by the technology available in 2008. However, they also lacked understanding of the Scrum framework to the point that they removed feedback mechanisms that lie at the core of Scrum's success. Furthermore, despite Scrum's popularity, it is not suitable for all projects – traditional project management techniques should still be considered where appropriate. For instance, the SIREN project might have avoided discontinuation had they kept their scope more rigid by using a Waterfall technique instead of Scrum.

In conclusion, Scrum is a modern and widely used product management technique that has Agile values at its core. If implemented correctly, it can increase project communication, efficiency and flexibility, saving organisations' time and money.

## Works Cited:

- [1] Takeuchi, Hirotaka, and Ikujiro Nonaka. Review of *The New New Product Development Game*, *The Scrum Master*, no. 1986, Jan. 1986, pp. 1–3,  
[www.thescrummy.co.uk/wp-content/uploads/2016/09/The-New-New-Product-Development-Game.pdf](http://www.thescrummy.co.uk/wp-content/uploads/2016/09/The-New-New-Product-Development-Game.pdf) Accessed 14 Feb. 2024.
- [2] “A Complete Guide to the Waterfall Methodology.” *Monday Blog*, 1 Jan. 2024,  
[monday.com/blog/project-management/waterfall-methodology/](https://monday.com/blog/project-management/waterfall-methodology/). Accessed 15 Feb. 2024.
- [3] Salimi, Sohrab. “The Origins of Scrum.” *Agile Academy*,  
[www.agile-academy.com/en-foundations/the-origins-of-scrum/](https://www.agile-academy.com/en-foundations/the-origins-of-scrum/). Accessed 16 Feb. 2024.
- [4] Schwaber, Ken. “Scrum Development Process.” *Business Object Design and Implementation*, 1995, pp. 10–18, doi:10.1007/978-1-4471-0947-1\_11.
- [5] Kneafsey, Simon. “A Short History of Scrum.” *TheScrumMaster.Co.Uk*, 26 Jan. 2022,  
[www.thescrummy.co.uk/scrum/short-history-scrum/](https://www.thescrummy.co.uk/scrum/short-history-scrum/). Accessed 15 Feb. 2024.
- [6] *Scrum Alliance*,  
[resources.scrumalliance.org/Article/differences-agile-scrum-differ-waterfall](https://resources.scrumalliance.org/Article/differences-agile-scrum-differ-waterfall). Accessed 14 Feb. 2024.
- [7] Schwaber, Ken, and Jeff Sutherland. “The 2020 Scrum GUIDETM.” *Scrum Guide | Scrum Guides*, 2020, [scrumguides.org/scrum-guide.html](https://scrumguides.org/scrum-guide.html). Accessed 16 Feb. 2024.
- [8] Beck, Kent, et al. “Manifesto for Agile Software Development .” *Principles behind the Agile Manifesto*, 2001, [agilemanifesto.org/principles.html](https://agilemanifesto.org/principles.html). Accessed 16 Feb. 2024.
- [9] Evans, Ian. “Agile Delivery at British Telecom.” *Methods and Tools*,  
[www.methodsandtools.com/archive/archive.php?id=43p3](https://www.methodsandtools.com/archive/archive.php?id=43p3). Accessed 15 Feb. 2024.
- [10] Yakub, Akbar. “IBM’s Agile Revolution: Pivot of the Big Blue.” *Scrum Master Prep*,  
[www.scrummasterprep.com/blog/ibms-agile-revolution-waterfall-to-scrum](https://www.scrummasterprep.com/blog/ibms-agile-revolution-waterfall-to-scrum). Accessed 16 Feb. 2024.
- [11] Thornton, Grant. 2012, *Termination of the SIREN ICT Project*,  
[surrey-pcc.gov.uk/wp-content/uploads/2014/06/SIREN-PUBLIC-INTEREST-REPORT.pdf](https://surrey-pcc.gov.uk/wp-content/uploads/2014/06/SIREN-PUBLIC-INTEREST-REPORT.pdf). Accessed 15 Feb. 2024.
- [12] Rajpal, Mark. “Lessons Learned from a Failed Attempt at Distributed Agile.” *Agile Processes, in Software Engineering, and Extreme Programming*, 2016, pp. 235–243,  
doi:10.1007/978-3-319-33515-5\_21.