

# The World Is Changing And So Can You By Using Agile

A Technical Paper prepared for SCTE by

**Aparna Srinivasan**

Manager Cybersecurity

Comcast Cable

Chennai, India

[Aparna\\_srinivasan@comcast.com](mailto:Aparna_srinivasan@comcast.com)

**Venkata Ramarao Sanka**

Development Engineer 4

Comcast Cable

Chennai, India

[venkataramarao\\_sanka@comcast.com](mailto:venkataramarao_sanka@comcast.com)

## Table of Contents

Title	Page Number
1. Introduction.....	4
2. Challenges .....	4
3. What Is Agile Management? .....	5
3.1. The Pillars of Agile .....	6
3.2. Agile mindset.....	6
3.3. Agile Values [10] .....	7
3.4. Agile Principles [9].....	7
4. Planning Agile Implementation.....	8
4.1.1. Understand the culture and needs of the organization.....	9
4.1.2. Identify what needs to be changed and what is not changing.....	9
5. Plan for Agile Implementation in a non-Agile Organization .....	10
5.1. Product Vision .....	11
5.2. Product Roadmap .....	11
5.3. Create a Release Plan .....	11
5.3.1. Prioritize Requirements.....	11
5.3.2. Create an Initial Design.....	11
5.3.3. Determine the Success Factor.....	12
5.4. Who Was Involved in the Plan? .....	12
5.5. Prerequisites for Iteration Planning .....	12
5.6. Iteration Planning Steps .....	13
6. How To Implement Agile in An Organization That Is Not Agile .....	13
6.1. Design Sprint (Sprint 0).....	13
6.2. Project Charter .....	13
6.3. Scrum Ceremonies .....	14
6.3.1. Sprint Planning.....	14
6.3.2. Daily Scrum/Stand up .....	14
6.3.3. Sprint Review .....	14
6.3.4. Sprint Retrospective.....	15
6.4. Metrics for Agile.....	15
6.4.1. Productivity metrics.....	15
6.4.2. Progress metrics .....	15
6.4.3. Quality metrics .....	16
7. The Benefits of Agile Methodology .....	17
8. Conclusion.....	17
Abbreviations .....	17
Bibliography & References.....	18

## List of Figures

Title	Page Number
Figure 1 – Agile methodology workflow .....	5
Figure 2 – Pillars of Agile .....	6
Figure 3 – What is Agile .....	7
Figure 4 – Difficulty of Agile adoption .....	8
Figure 5 – Understand organization culture.....	9
Figure 6 – Steps in Agile planning .....	10
Figure 7 – Create a release plan .....	11

Figure 8 – Scrum team..... 12  
Figure 9 – Scrum ceremonies..... 14  
Figure 10 – Sprint burndown..... 16

## 1. Introduction

In this changing business environment, it is crucial to take a proactive stance to be more agile and collaborative. One of the development teams in Comcast Cybersecurity adopted a waterfall methodology for their product development. They committed to understanding requirements, designing to those requirements, and asking teams to develop them once the requirement was signed off.

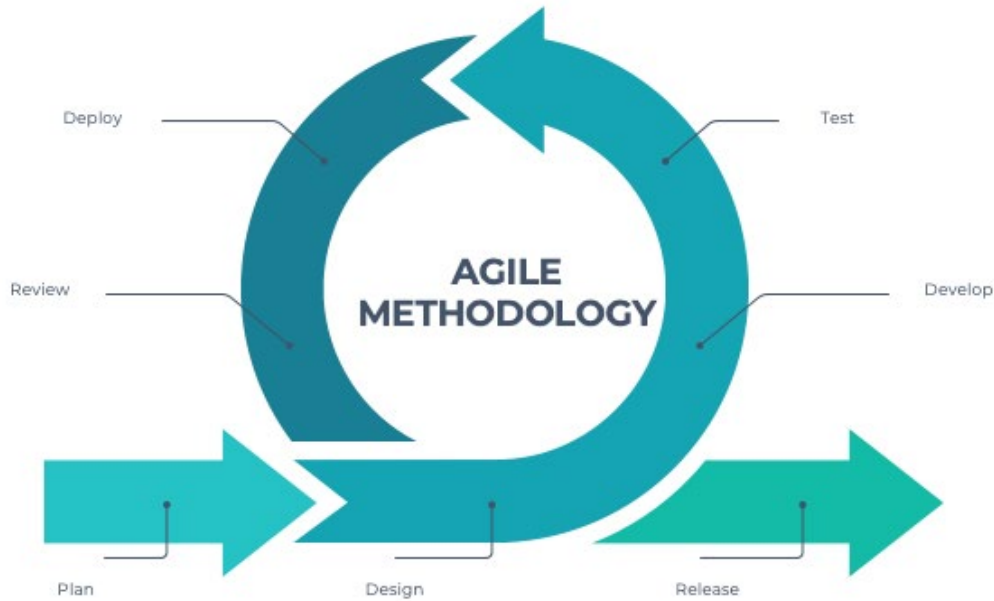
Agile is a popular methodology for developing products and services, but it hasn't been fully adopted in many organizations. More than ever, Agile has shown to be an effective way to create products and services with proper implementation. It is a culture that changes how teams work together, communicate with each other, and manage their work. It's more than just a new way of doing things. It is a whole new way of thinking about how to do things.

This paper and presentation will talk about the challenges faced by the team, provide a description of Agile, including the planning, implementation and benefits of using this approach. It will show how a transition to Agile delivered significant benefits to our customers.

## 2. Challenges

The cyber security landscape has been changing rapidly over the last few years, and it has become a constant challenge for organizations to keep up with these changes. There are often delays in the process of analysing customer requirements, often to the point of having them become obsolete before they are addressed. Market conditions often create changes to an initial requirement, which, if not addressed immediately will lead to dissatisfaction of the requesting party. Without an effective communication and feedback cycle, stakeholders are not included in initial discussions or informed of changes leading to a gap in development, stakeholder awareness, and further dissatisfaction. After development and during Testing or User Acceptance Testing can generate minor requirement changes; changes that can take a long time to implement, which might cause the real purpose of the requirement to be lost over time.

### 3. What Is Agile Management?



**Figure 1 – Agile methodology workflow**

Agile is an iterative approach to developing software products. It emphasizes the importance of early and continuous delivery of working software, which can be obtained through frequent releases, demonstrating progress early and frequently to stakeholders, and responding quickly to feedback (Figure 1 and Figure 2).

### 3.1. The Pillars of Agile

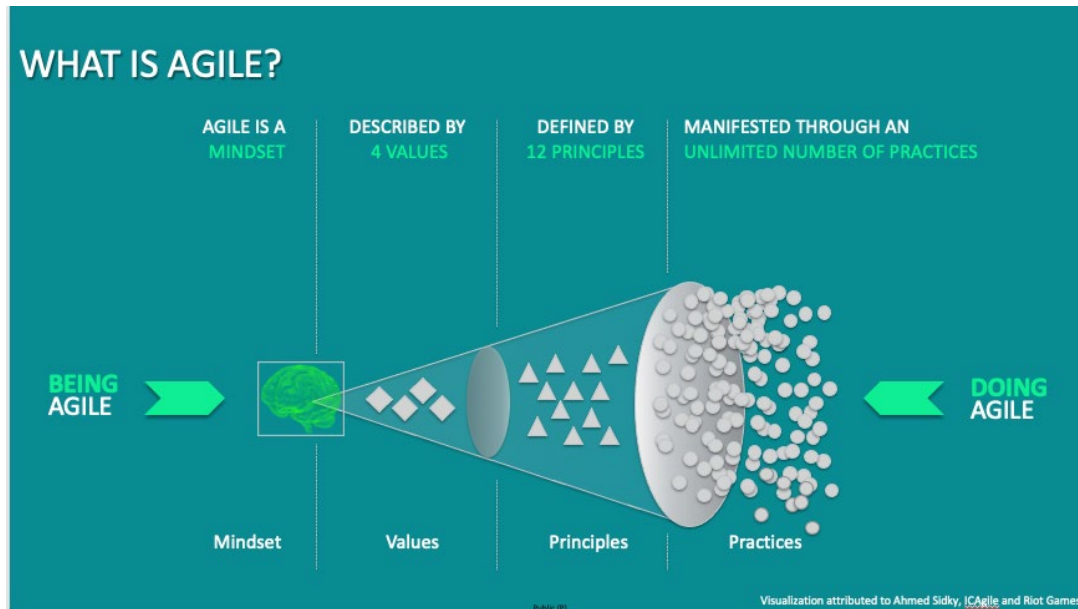


**Figure 2 – Pillars of Agile**

### 3.2. Agile mindset

To succeed with Agile methodology, teams must cultivate an Agile mindset. The Agile mindset is a thought process that involves understanding, collaborating, learning, and staying flexible to achieve high-performing results. [3] By combining the Agile mindset with methods and tools, teams can adapt to change and deliver incremental value to their customers. [3] Imagine a concert where each musician plays separate notes without coordination. This is not an impressive performance, and you will be disappointed by the end of the event. Take this case and apply the same scenario to your work environment. Concerts depend on melody, harmony, and rhythm to create a piece of soulful music. Similarly, software teams can successfully deliver the final product when they collaborate, help, share information, and work flexibly.

An Agile mindset (Figure 3) focuses on "being agile" as a foundation for success in "doing Agile." [3] It is defined by the four values and described by the twelve principles of the Agile Manifesto and then manifested through an unlimited number of practices and diverse ways of working. [3]



**Figure 3 – What is Agile**

### 3.3. Agile Values [10]

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

### 3.4. Agile Principles [9]

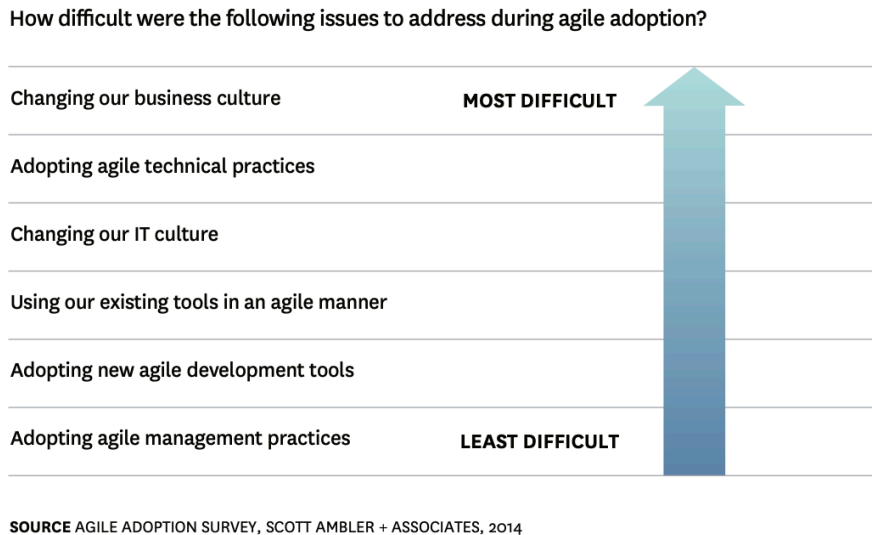
- 1) Satisfy Customers Through Early & Continuous Delivery
- 2) Welcome Changing Requirements Even Late in the Project
- 3) Deliver Value Frequently
- 4) Break the Silos of Your Project
- 5) Build Projects Around Motivated Individuals
- 6) The Most Effective Way of Communication is Face-to-face
- 7) Working Software is the Primary Measure of Progress
- 8) Maintain a Sustainable Working Pace
- 9) Continuous Excellence Enhances Agility
- 10) Simplicity is Essential
- 11) Self-organizing Teams Generate Most Value
- 12) Regularly Reflect and Adjust Your Way of Work to Boost Effectiveness

## 4. Planning Agile Implementation

*“Planning is everything, Plans are nothing”*

-Prussian Field Marshal Helmuth Graf von Moltke

Estimating and planning are critical to the success of any software development project of any size or consequences (Figure 4). Plans guide our investment decision. A plan helps us know who needs to be available to work on a project during a given period. [13]



**Figure 4 – Difficulty of Agile adoption**

Plans are documents or figures; they are snapshots of how we believe a project might unfold over an uncertain future. Planning is an activity. Agile planning shifts the emphasis from the plan to the planning.

Planning for Agile balances work and expense, ensuring we will change the plan throughout our project to improve the outcome. An Agile method is one in which we're not only willing to but are very keen to alter. We want to be flexible because change indicates we've learned something or avoided making an error. We might have discovered that customers want more or less of a specific feature, have determined it is more crucial to have usability than anticipated, or that programming in the new language takes longer than expected. As part of project planning, the team's Agile implementation, we identified and prioritized the requirements, created an initial design, and determined how we'll measure success.

The following are some of the critical steps taken in the planning for Agile implementation in the Cybersecurity team:



**4.1.1. Understand the culture and needs of the organization.**



**Figure 5 – Understand organization culture**

Understanding the culture and needs of the team/organization is the key to success in Agile implementation (Figure 5). We realized that change and transformation are not easy. There will be some resistance, but it can be overcome with a proper understanding of the culture, needs, and goals.

The first step we took in understanding the culture of this was to learn about the different modules of their work on an elevated level (what they do), how they do their requirement gathering, design, development, testing (how they do it), identify their end customers, (for whom they do it), and why this development is needed (why do they do it). This helped all parties understand how Agile can fit into their existing framework.

The next step was to listen. Top management was open to hearing other perspectives, and they listened to what others wanted with agility and understood the impact. Once an Agile approach was approved, we identified points of leverage by asking how Agile could serve this project as effectively as possible.

**4.1.2. Identify what needs to be changed and what is not changing.**

For an organization/team to successfully transition from traditional project management techniques to Agile, it is essential to understand the difference between a change and a non-change, and how these factors affect how Agile implementation should be approached. The answer to this question is dependent on a few distinct factors.

In general, teams like ours which are not Agile need to make fundamental changes to their structure and how they work. We identified what processes need changing and how these processes will be improved.

Some of the key factors we considered when determining if the team needed to change its approach were the goals of the team, the percentage of available time the team spent on projects, and the number of people on the team.

It is essential to understand the different stakeholders' perspectives to communicate with stakeholders about the changes and challenges in Agile implementation. We began by identifying what our stakeholders wanted and expected from us. We then worked specifically on what they wanted, not what they didn't, and then began a process of open and continuous communication so they knew how their ideas were being implemented. This last item, communication, had been identified in our previous development model as an area that could be improved so it was important for us to address.

To ensure that we could identify all the stakeholders and involve them in the process, we created a plan to define the problem/opportunity and to identify the key stakeholders.

There are four main categories of people whom we involved in the change process:

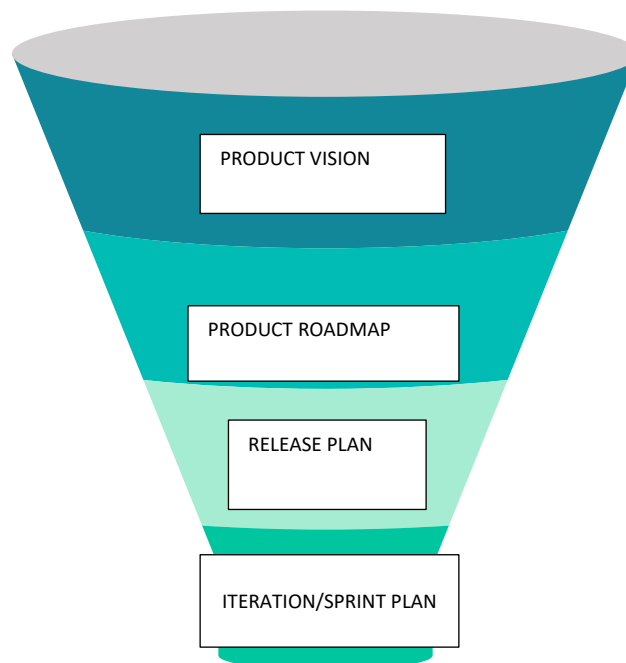
1. those whom the change will directly impact
2. those who need to support the change
3. those who will have a stake in the success of the change
4. those who must manage or implement it

The plan for involving them included getting their buy-in and commitment to participate, identifying what they will do and when they will do it, and identifying what resources they need.

We then developed a strategy for integrating Agile practices into existing organizational processes, policies, and procedures. This allowed us to efficiently integrate these behaviors into all levels of the organization while still preserving those aspects of the existing organizational culture that remain valuable or important to success.

## 5. Plan for Agile Implementation in a non-Agile Organization

We decided to go with the most popular Agile methodology, which is Scrum. Before planning for an Agile project, we needed the following details (Figure 6).



**Figure 6 – Steps in Agile planning**

## 5.1. Product Vision

Product vision is the product idea communicated to the team and stakeholders and is the driving force behind a product. We tried to answer the following questions in our product vision document.

1. What is the problem that this product solves?
2. What are we trying to achieve with this product?
3. Who will use this product, and what benefits will they get from it?
4. How does this product make people's lives better or easier?
5. How do we want users to feel when they use our product?
6. What are we going to build, why, and how are we going to do it?

## 5.2. Product Roadmap

Product roadmaps are a visual representation of a product's future. Creating a product roadmap allowed our teams to visualize the progress of their work and adjust as needed. We used the roadmap to track progress, identify risks, and adjust according to sprint feedback. A product roadmap also helped us set expectations with our customers about what they can expect from our team in the future, which led to increased customer satisfaction and loyalty.

## 5.3. Create a Release Plan



**Figure 7 – Create a release plan**

As we did our requirement gathering, we evaluated the time needed for the project along with the constraints of technology, materials, and other considerations. We determined what would be accomplished through technological means or manual labour (Figure 7).

### 5.3.1. Prioritize Requirements

Agile projects are a way of working that helps teams focus on the priority requirements and deliver the most critical features first. Prioritizing requirements benefited us by allowing our teams to focus on what was essential, allowing us to not get bogged down in less significant details. Prioritization helped us reduce risk as we were able to focus on delivering value quickly, and it encouraged collaboration as we provided a transparent prioritization process.

Prioritizing requirements in our Agile project was a challenging task, and it required our teams to clearly understand the business, product, and product vision. The team also identified the essential requirements to be satisfied first. We prioritized requirements in an agile project using a Moscow (Must Have, Should Have, Could Have, Won't have) analysis.

### 5.3.2. Create an Initial Design

We developed a design that was easy to understand, aesthetically pleasing, and within the budget. The team worked together to include customers, stakeholders, and Agile team members in building the design

strategy. We had different perspectives on the problem and did not depend on documentation to express our ideas.

### 5.3.3. Determine the Success Factor

The success of our project was measured by how well we met our customer's and team's needs during the planning phase. To measure our success, we used criteria such as having a clear understanding of what needs to be delivered to meet customer expectations, a clear understanding of who is going to do each task and how this task will help in achieving the goal. We relied on excellent quality requirements, specification documents, and understood clearly how much time and money it will take to deliver each requirement.

### 5.4. Who Was Involved in the Plan?

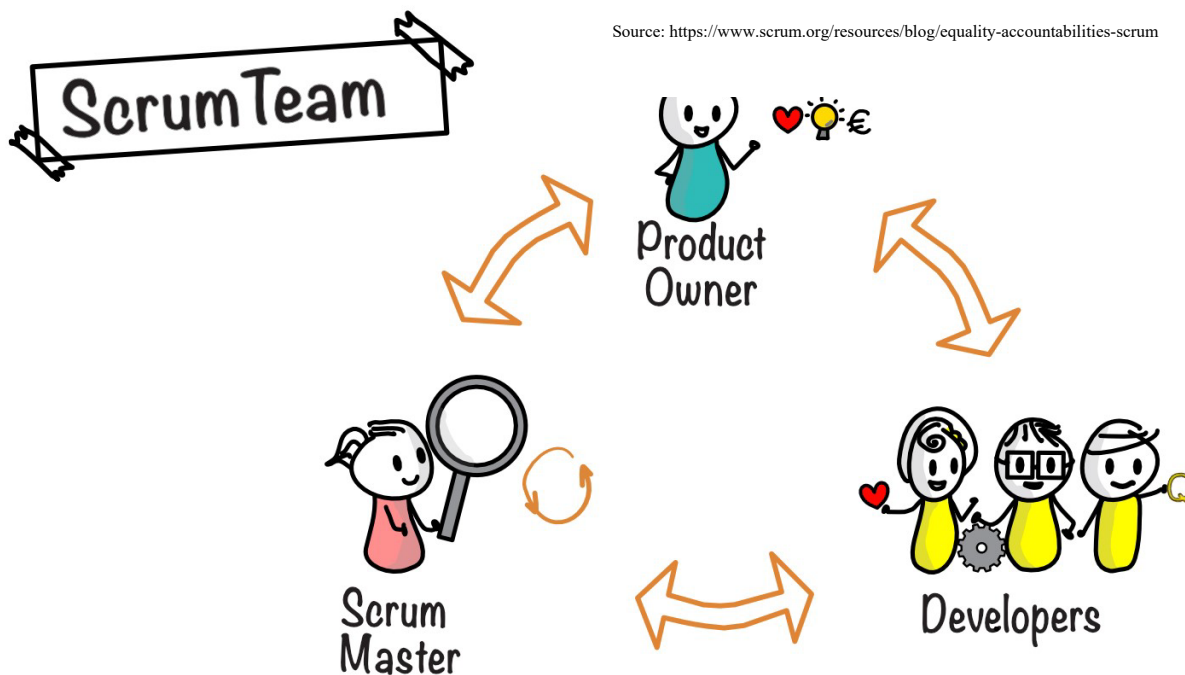


Figure 8 – Scrum team

The Scrum Master is a leader for the team and a facilities provider. He assists the team members in adopting Agile practices to ensure they fulfil their obligations and meet customers' needs. The Product owner provides a full view of the backlog of product requirements and the criteria for acceptance, while the Development Team defines the roles and carries out the work (Figure 8).

### 5.5. Prerequisites for Iteration Planning

Before we started the Iteration planning, we did the following:

- Ensured that the items in the backlog of the product were sizeable

- Ranked the epics created. Epics are large bodies of work that can be broken down into a number of smaller tasks (called stories) [11]
- Documented the acceptance criteria for each user story. User Stories are short requirements or requests written from the perspective of an end user [11]
- Calculated the number of user stories that could be accommodated into an iteration
- Broke these user stories down into tasks [4] and assigned each to an owner
- Assigned each task an approximate duration

The team members were assigned tasks based on their capacity or speed to ensure that the team member was not overwhelmed.

## 5.6. Iteration Planning Steps

Iterations are the basic building block of Agile development. Each iteration is a standard, fixed-length timebox, where Agile teams deliver incremental value in the form of working, tested software and systems. [12]

The Product Owner for our product picked up the top item in the existing backlog of items. The team members within explained the tasks needed to complete each user story as they were picked up and were accountable for tasks assigned to them.

The planning poker method was used to determine the complexity of each user story. Those with more story points required more time to develop and test compared to others. Team members were given the flexibility to determine the amount of time it would take them to complete each task.

These steps were then repeated for each item in the iteration process. If any team member was overloaded with work, duties could be re-allocated to others.

## 6. How To Implement Agile in An Organization That Is Not Agile

The fundamentals of Scrum are simple. To tackle an opportunity, the organization forms and empowers a small team, usually three to nine people, most of whom are assigned full-time. [5] The team is cross-functional and includes all the skills necessary to complete its tasks, and it manages itself and is strictly accountable for every aspect of the work. It is essential to train the team on the Agile framework if they have not previously worked in an Agile environment (Figure 9).

### 6.1. Design Sprint (Sprint 0)

The design process in an Agile project is iterative, and the process of designing in an Agile project is a continuous cycle of user feedback and design iteration. One of the best practices in Agile Design is to plan for a design sprint.

We started the design sprint by broadly exploring the problem, understanding its scope, and identifying potential solutions. Then, we generated ideas for solutions before focusing on one or two ideas to test initially. Finally, we presented our findings to stakeholders and then chose which idea would be implemented in more detail.

### 6.2. Project Charter

We kick-started the project by creating a project charter with the team's participation, which included a definition of what "done" meant. This way, when anyone in the team states that they are done with their task, everyone had the same understanding and meaning. We then defined the rules the team would adhere to and be accountable for.

### 6.3. Scrum Ceremonies



**Figure 9 – Scrum ceremonies**

#### **6.3.1. Sprint Planning**

Stories are committed to developers until they can fill the team's needs and are in line with the priorities of the owner of the product/stakeholders. In the following sprint, the team will utilize average velocity to help them determine the amount they should commit to in any given sprint.

The process began by finding the capacity based on the hours of each team member and subtracting the amount of time required to hold the celebrations during the sprint. For instance, four developers work 40 hours weekly, equal to 80 hours for each 2-week sprint. We subtracted hours for the daily stand-ups and backlog grooming, sprint planning reviews, sprint planning, and the time for sprint retrospectives to calculate the actual hours required to finish stories. Subtract 12 hours for the ceremony from each developer. Thus, their current availability is just 68. Multiplying 68 by 4, the capacity total is 272.

#### **6.3.2. Daily Scrum/Stand up**

Four days a week, team members met virtually via a team call and discussed what they accomplished the previous day and the next thing they planned to do the following day. These meetings were less about status and were utilized more for collaboration in the event of dependence and coordination within the team. We also ensured that no one was telling others how to proceed and that all are self-organized to complete the user story at the end of the sprint.

#### **6.3.3. Sprint Review**

At the end of the sprint, we conducted a sprint review meeting where the team reviewed/ demonstrated the stories to the product owners. We documented feedback from the product owners and evaluated its

impact on the development process. We used the test environment to highlight the demo where we created records using dummy data instead of actual production data to explain the feature.

#### **6.3.4. Sprint Retrospective**

In the sprint retrospective calls, the team reviewed the tasks and efforts they contributed during the sprint. We recorded what went well, what could have gone better, and what did not go well for the duration of the sprint. This allowed us to determine what we would like to put aside, begin doing, and then continue to do. Retrospection at the end of every sprint offered us opportunities for continuous improvement. An enormous improvement in quality, management, and performance happened within just a few sprints (as short as four to six weeks).

As an example, in one of our sprint retrospectives, we discussed that lack of technical documentation led to extended development time as the developers had to spend significant time going through the code to understand and then pick up the enhancement. We started creating technical documentation as, and when, team members had additional time within the sprint which helped us increase the turnaround time. In another example, each team member created their own test framework for testing which led to inconsistencies. We decided to have a common framework on which team members will write their test cases. This helped us have uniformity across testing and test results.

As much feedback as possible was highly appreciated and played a significant role in the overall success of the entire team.

### **6.4. Metrics for Agile**

We categorized Agile metrics into productivity, progress, and quality types.

#### **6.4.1. Productivity metrics**

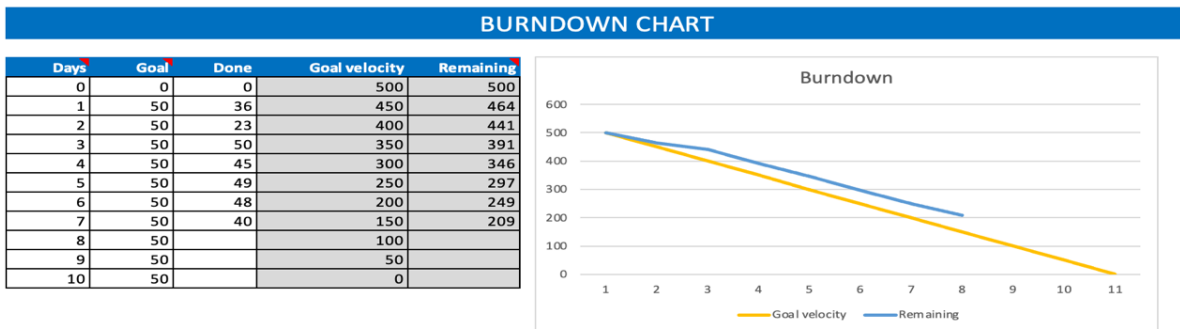
Productivity is a measure of how much work was completed. We used the Agile metrics below to understand how productive the team was.

- **Velocity** measured how much work was completed each time, which can be expressed as story points per day. We calculated sprint velocity based on the story points completed in the first few of sprints.
- **Lead time** measured how long it takes a team to complete their work from start to finish. [6] The team should always aim for short lead times because it's more efficient and cost-effective. [7] <https://kanbanize.com/kanban-resources/kanban-software/kanban-lead-cycle-time>
- We calculated the **average time** it took each team member to start and complete each task over the first four sprints. As an example, we found that Engineer 2 had a slightly longer lead time than Engineer 4 as they spent additional time finding solutions for some challenging problems. We used these metrics and conducted a brainstorming session to identify what we could do as a team to decrease the lead time.
- **Cycle time** is a measure that captures the time taken for the team to complete the sprint, i.e., time taken from In Development to Ready for testing.

#### **6.4.2. Progress metrics**

Progress metrics are a measure of how much work is completed.

The sprint burndown (Figure 10) represents actual scrum activities completed compared to the estimates of scrum-related tasks to assess a team's performance throughout a sprint.



**Figure 10 – Sprint burndown**

Epic and release burndowns are similar to sprint burndowns but epic burndown and release follow the progress across a more significant work body. These metrics are helpful when working with milestones. Also, the burndowns of releases monitor work progress when work is committed to a specific release.

### 6.4.3. Quality metrics

Quality metrics measure the degree to which the product meets its requirements. There are many quality metrics in an Agile scrum project, but the most important ones include the Quality of Backlog Items which determine how good the items in the backlog are. If they have high quality, they can be easily implemented, and their implementation will not take much effort. On the other hand, if they have low quality, they need more work, or they have a defect or deficiency.

A cumulative flow diagram (CFD) was used to see the status of various tasks - backlog, in-progress review, completed, and in-progress. The metric provided teams with an overview of all the activities in the workflow and the overall progression of the entire project.

The Quality of Team metric determined how good the team was. If they had high performance, the team quickly finished their tasks without much effort. On the other hand, if they had low performance, the team needed more time and effort to finish their tasks.

Escaped Defects allowed us to determine the quality of products received. This measure helped track the number of bugs discovered before a release is put into production.

Throughput measured the team's effectiveness by measuring the amount of work completed and handed over to the client within the period specified. This measure assisted us to determine the duration needed for the creation of software, and it let our end customers evaluate the team's degree of consistency.

The Quality of Product metric determined how good the product was in terms of value delivered (which uses value points to determine the quality of the work delivered to the client).

A control chart was linked to the cycle duration and was used to measure the cycle time of specific issues to assess the method's reliability, predictability, and stability. The control chart is a way to determine the success or failure of various processes within the project and is used to determine if there are any flaws.



## 7. The Benefits of Agile Methodology

1. Adopting Agile in our project involved teams creating and testing more frequently and delivering improvements to products and services more quickly to customers. Our customer satisfaction increased by 14% in 3 months since implementing this change.
2. The Agile development methodology allowed us to provide top-quality software in shorter release cycles that are more aligned with users' requirements.
3. Agile methodology helped us increase efficiency through more practical communication and was highly flexible to the ever-changing requests of customers. Our overall development-to-testing effort was reduced by over 30%. The number of back-and-forth communications using user acceptance testing decreased from around 5 - 6 rounds to no more than 2.
4. The ever-changing business environment of those customers requires flexibility and innovation on our part. As a team, we wanted to shift toward alignment between IT and business by ensuring that the process could be improved while considering the infrastructure and the technology.
5. We also focused on reducing time, efforts, costs, and expenditures in developing, documenting, testing, and deployment.
6. After Agile implementation in our project, developers and quality assurance (QA) teams were more prepared to work well together, allowing them to create software quickly while working closely with the client.
7. With an Agile approach to software delivery that focused on the set of measurements and metrics, teams were able to organize, plan, and deliver with sufficient certainty and a release-level commitment.
8. Measuring and tracking efficiency improvements cannot be accessed without transparency in the project. Teams responsible for software development and delivery understood the direction they were heading. Further development, testing, and operations teams were made aware of the present state of the project, its performance, and the goals of the project and the company.

## 8. Conclusion

Agile is a methodology of software development that focuses on the process of iterative and incremental delivery. It is a software development approach that emphasizes collaboration, simplicity, and communication. It also focuses on continuous testing and feedback to achieve higher quality products.

The Agile methodology has been gaining ground in various industries such as healthcare, finance, defence, education, etc. The ability to quickly adapt to changing requirements has made it popular among customers for its flexibility and ability to deliver results faster than traditional methods of project management.

## Abbreviations

CFD	Cumulative Flow Diagram
QA	Quality Assurance

## Bibliography & References

- [1] <https://www.agilealliance.org/agile101>
- [2] <https://habitation.com/what-is-agile-working>
- [3] <https://themelisting.com/how-to-make-decisions-with-an-agile-mindset>
- [4] <https://dokumen.tips/documents/gullvision-salisbury-university-authors-tim-stlauterburgcosc426finalreportexamplesgull.html>
- [5] <https://www.coursehero.com/file/55754463/5200-AlmetrixDrayton-Assignment1docx>
- [6] <https://industry-era.com/DevOps-Big-Data-and-Social-Media-10080.php>
- [7] [https://www.reddit.com/r/transit/comments/v8niy7/whats\\_the\\_best\\_solution\\_to\\_rush\\_hours\\_in\\_public](https://www.reddit.com/r/transit/comments/v8niy7/whats_the_best_solution_to_rush_hours_in_public)
- [8] [https://www.reddit.com/r/transit/comments/v8niy7/whats\\_the\\_best\\_solution\\_to\\_rush\\_hours\\_in\\_public](https://www.reddit.com/r/transit/comments/v8niy7/whats_the_best_solution_to_rush_hours_in_public)
- [9] <https://agilemanifesto.org/principles.html>
- [10] <https://agilemanifesto.org/>
- [11] <https://www.atlassian.com/agile/project-management/epics-stories-themes>
- [12] <https://www.scaledagileframework.com/iterations/>
- [13] Agile Estimating and Planning By Mike Cohn