Manifesto for Agile Software Development

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

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

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

Ken Schwaber and Jeff Sutherland

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

Purpose of the Scrum Guide

We developed Scrum in the early 1990s. We wrote the first version of the Scrum Guide in 2010 to help people worldwide understand Scrum. We have evolved the Guide since then through small, functional updates. Together, we stand behind it.

The Scrum Guide contains the definition of Scrum. Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum. Changing the core design or ideas of Scrum, leaving out elements, or not following the rules of Scrum, covers up problems and limits the benefits of Scrum, potentially even rendering it useless.

We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots. As Scrum's use spreads, developers, researchers, analysts, scientists, and other specialists do the work. We use the word "developers" in Scrum not to exclude, but to simplify. If you get value from Scrum, consider yourself included.

As Scrum is being used, patterns, processes, and insights that fit the Scrum framework as described in this document, may be found, applied and devised. Their description is beyond the purpose of the Scrum Guide because they are context sensitive and differ widely between Scrum uses. Such tactics for using within the Scrum framework vary widely and are described elsewhere.

Ken Schwaber & Jeff Sutherland November 2020

Scrum Definition

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. *Repeat*

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

Scrum Theory

Scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

Scrum employs an iterative, incremental approach to optimize predictability and to control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.

Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.

Transparency

The emergent process and work must be visible to those performing the work as well as those receiving the work. With Scrum, important decisions are based on the perceived state of its three formal artifacts. Artifacts that have low transparency can lead to decisions that diminish value and increase risk.

Transparency enables inspection. Inspection without transparency is misleading and wasteful.

Inspection

The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems. To help with inspection, Scrum provides cadence in the form of its five events.

Inspection enables adaptation. Inspection without adaptation is considered pointless. Scrum events are designed to provoke change.

Adaptation

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

Adaptation becomes more difficult when the people involved are not empowered or self-managing. A Scrum Team is expected to adapt the moment it learns anything new through inspection.

Scrum Values

Successful use of Scrum depends on people becoming more proficient in living five values:

Commitment, Focus, Openness, Respect, and Courage

The Scrum Team commits to achieving its goals and to supporting each other. Their primary focus is on the work of the Sprint to make the best possible progress toward these goals. The Scrum Team and its stakeholders are open about the work and the challenges. Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work. The Scrum Team members have the courage to do the right thing, to work on tough problems.

These values give direction to the Scrum Team with regard to their work, actions, and behavior. The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values, not diminish or undermine them. The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts. When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust.

Scrum Team

The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when, and how.

The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. In general, we have found that smaller teams communicate better and are more productive. If Scrum Teams become too large, they should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same product. Therefore, they should share the same Product Goal, Product Backlog, and Product Owner.

The Scrum Team is responsible for all product-related activities from stakeholder collaboration, verification, maintenance, operation, experimentation, research and development, and anything else that might be required. They are structured and empowered by the organization to manage their own work. Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.

The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. Scrum defines three specific accountabilities within the Scrum Team: the Developers, the Product Owner, and the Scrum Master.

Developers

Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.

The specific skills needed by the Developers are often broad and will vary with the domain of work. However, the Developers are always accountable for:

- Creating a plan for the Sprint, the Sprint Backlog;
- Instilling quality by adhering to a Definition of Done;
- Adapting their plan each day toward the Sprint Goal; and,
- Holding each other accountable as professionals.

Product Owner

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;
- Ordering Product Backlog items; and,
- Ensuring that the Product Backlog is transparent, visible and understood.

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Scrum Master

The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.

The Scrum Master is accountable for the Scrum Team’s effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.

Scrum Masters are true leaders who serve the Scrum Team and the larger organization.

The Scrum Master serves the Scrum Team in several ways, including:

- Coaching the team members in self-management and cross-functionality;
- Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
- Causing the removal of impediments to the Scrum Team’s progress; and,
- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

The Scrum Master serves the Product Owner in several ways, including:

- Helping find techniques for effective Product Goal definition and Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Helping establish empirical product planning for a complex environment; and,
- Facilitating stakeholder collaboration as requested or needed.

The Scrum Master serves the organization in several ways, including:

- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
- Removing barriers between stakeholders and Scrum Teams.

Scrum Events

The Sprint is a container for all other events. Each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the transparency required. Failure to operate any events as prescribed results in lost opportunities to inspect and adapt. Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. Optimally, all events are held at the same time and place to reduce complexity.

The Sprint

Sprints are the heartbeat of Scrum, where ideas are turned into value.

They are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality does not decrease;
- The Product Backlog is refined as needed; and,
- Scope may be clarified and renegotiated with the Product Owner as more is learned.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month. When a Sprint’s horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase. Shorter Sprints can be employed to generate more learning cycles and limit risk of cost and effort to a smaller time frame. Each Sprint may be considered a short project.

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making.

A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

Sprint Planning

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The Scrum Team may also invite other people to attend Sprint Planning to provide advice.

Sprint Planning addresses the following topics:

Topic One: Why is this Sprint valuable?

The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders. The Sprint Goal must be finalized prior to the end of Sprint Planning.

Topic Two: What can be Done this Sprint?

Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items during this process, which increases understanding and confidence.

Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

Topic Three: How will the chosen work get done?

For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.

Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Daily Scrum

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint's work.

Sprint Review

The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.

During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The Product Backlog may also be adjusted to meet new opportunities. The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.

The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Sprint Retrospective

The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. Inspected elements often vary with the domain of work. Assumptions that led them astray are identified and their origins explored. The Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.

The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Scrum Artifacts

Scrum's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

Each artifact contains a commitment to ensure it provides information that enhances transparency and focus against which progress can be measured:

- For the Product Backlog it is the Product Goal.
- For the Sprint Backlog it is the Sprint Goal.
- For the Increment it is the Definition of Done.

These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

Product Backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.

Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.

The Developers who will be doing the work are responsible for the sizing. The Product Owner may influence the Developers by helping them understand and select trade-offs.

Commitment: Product Goal

The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define “what” will fulfill the Product Goal.

> *A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.*

The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next.

Sprint Backlog

The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.

Commitment: Sprint Goal

The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog. As the Developers work during the Sprint, they keep the Sprint Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.

Increment

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Commitment: Definition of Done

The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

The moment a Product Backlog item meets the Definition of Done, an Increment is born.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum. If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.

The Developers are required to conform to the Definition of Done. If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.

End Note

Scrum is free and offered in this Guide. The Scrum framework, as outlined herein, is immutable. While implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

Acknowledgements

People

Of the thousands of people who have contributed to Scrum, we should single out those who were instrumental at the start: Jeff Sutherland worked with Jeff McKenna and John Scumniotales, and Ken Schwaber worked with Mike Smith and Chris Martin, and all of them worked together. Many others contributed in the ensuing years and without their help Scrum would not be refined as it is today.

Scrum Guide History

Ken Schwaber and Jeff Sutherland first co-presented Scrum at the OOPSLA Conference in 1995. It essentially documented the learning that Ken and Jeff gained over the previous few years and made public the first formal definition of Scrum.

The Scrum Guide documents Scrum as developed, evolved, and sustained for 30-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.

The complete history of Scrum is described elsewhere. To honor the first places where it was tried and proven, we recognize Individual Inc., Newspage, Fidelity Investments, and IDX (now GE Medical).

The Agile movement was born in 2001 and its initial purpose was to offer guiding principles and values that will help software teams to improve development processes, be more adaptive and responsive to the ever-changing customers' needs.

Indeed, the deeper idea of Agile was to get rid of heavyweight software development processes and suggest alternatives that will help teams manage their projects in a more flexible way. After this, different methods and frameworks were born or re-created led by the main Agile values and principles.

As Agile became a hot topic at the beginning of the 21st century, many frameworks took advantage of the rising hype and rapidly became famous (Scrum, SAFe, etc.). However, many companies seeking true business agility realized that highly prescriptive frameworks and agility are kinds of opposite things. This is why many organizations today are looking into methodologies that create and support a stable workflow and tailor the processes to their own needs, instead of adopting highly prescriptive frameworks.

Today, the most popular Agile methods or frameworks for project management are Kanban, Scrum, and Scrumban.

So, let's start with Kanban.

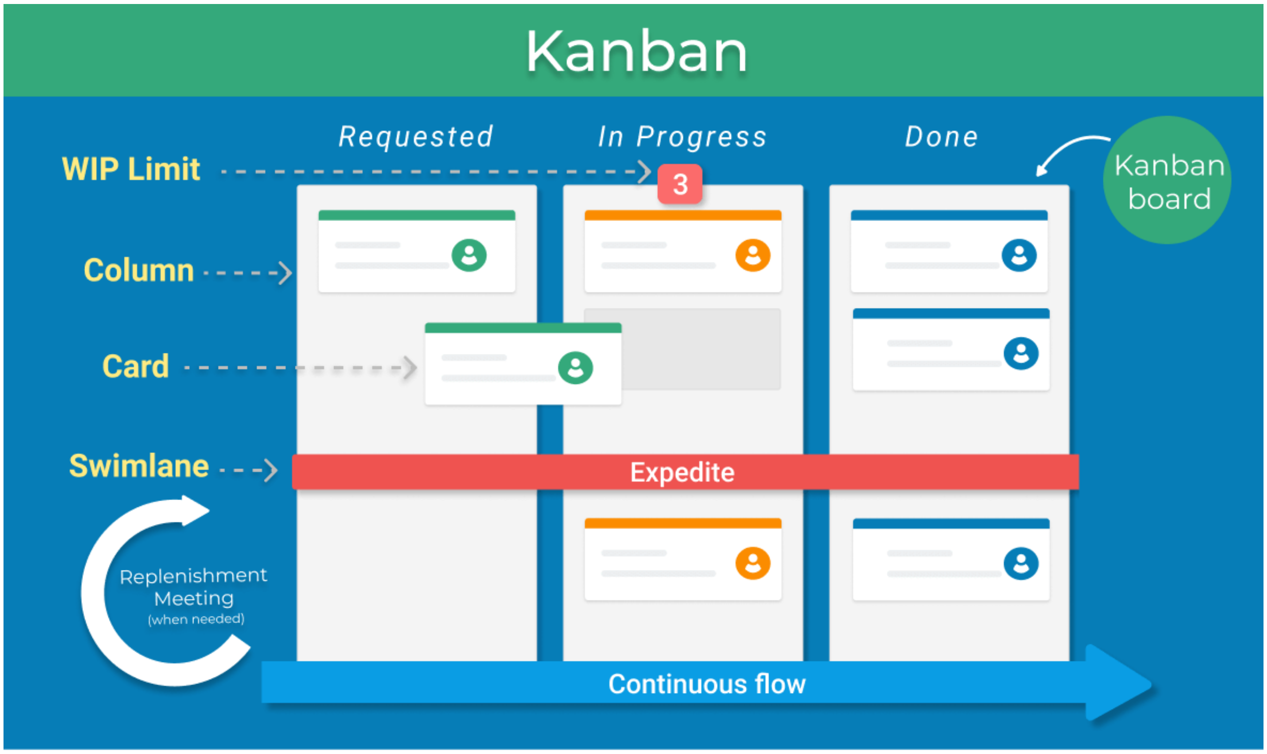
Kanban

Kanban is a method formulated a decade ago. It focuses on evolutionary change and continuous process improvements.

The method has 6 core practices:

- visualize work
- limit work-in-progress
- manage flow
- make process policies explicit
- implement feedback loops
- improve collaboratively

Teams visualize their work on a Kanban board that serves as a central information hub where all tasks should be placed. This will allow people to exchange information much faster and collaborate more effectively while working on different projects.



A Kanban board is divided into columns which represent different stages of the workflow. This helps project managers and teams organize and manage work much better, keep track of different projects and acquire a better overview of the whole process.

One of the most important Kanban practices is limiting work-in-progress. The WIP limit is the amount of work that's allowed to exist in each of the board columns. This is one of the most effective tools you can use to create a massive focus on your team. It will help your team focus on finishing work and improve overall efficiency.

On the other hand, we all know that projects, teams, and individuals are unique. Different teams have a different set of skills, level of experience, expertise. Different projects may have different scope, budget and so on.

This is why Kanban suggests that you should start with what you do now and evolve gradually. No drastic changes, no revolutions. This is what makes Kanban one of most adaptive Agile project management methods.

As a matter of fact, Kanban can be applied by any team in your organization from IT to marketing. The main reason is that Kanban:

- respects the current processes and roles
- doesn't require revolutionary, but evolutionary changes
- suggests you should pursue incremental, evolutionary change and try to improve continuously

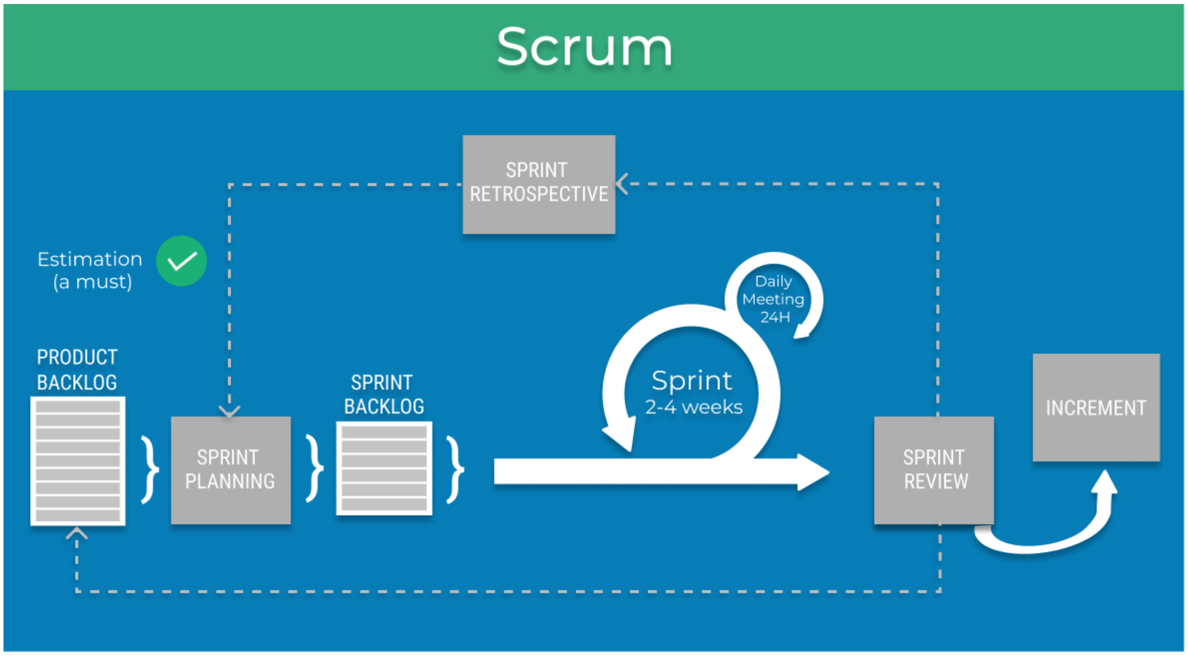
Scrum

Many believe that Scrum is a method, but actually, it is a prescriptive framework. By nature, it is an iterative approach that uses time-boxed intervals and split projects into fixed periods called sprints. The main purpose is to support teams to productively and creatively deliver products of the highest possible value.

There are 3 immutable roles:

- the Product Owner
- The Scrum Master
- the Team

The product owner represents customers and other stakeholders. He/she organizes and manages the product backlog, which is a prioritized task list of all the work items needed for the product. On the other hand, the Scrum master is a servant-leader of the team and helps everyone understand and apply the rules correctly.



The Sprint backlog should be filled with items selected from the product backlog until the capacity for the Sprint is reached. The work itself is done by a self-organizing team during the Sprint, which may be considered as projects with a fixed length of no more than one month.

There are 4 major Scrum events:

- Sprint Planning
- Daily Stand Up
- Sprint Review

- Sprint Retrospective

An interesting fact is that in the original paper that formulates the framework and later the Scrum guide, the authors never mention the usage of a task board.

However, nowadays you can observe that all teams or organizations use a task board while practicing Scrum, which is a borrowed practice from Kanban.

After all, the board increases transparency and supports the Agile project management values.

Scrumban

As Kanban was becoming more and more popular, some people of the Agile community saw an opportunity to develop a method which makes it easy for Scrum teams to move forward and focus on continuous improvement and evolutionary change. And this is how Scrumban was born.

An interesting fact is that 81% of the Scrum masters use Kanban along with Scrum.

Scrumban takes the Kanban philosophy and practices and lays them on top of Scrum as well as it eliminates some rules.

Let’s see what Scrumban takes from Kanban.

Visualize work. This is the first thing that Scrumban prescribes as mandatory. This is very important because Scrum doesn’t mandate a board, while in Kanban the board is a must.

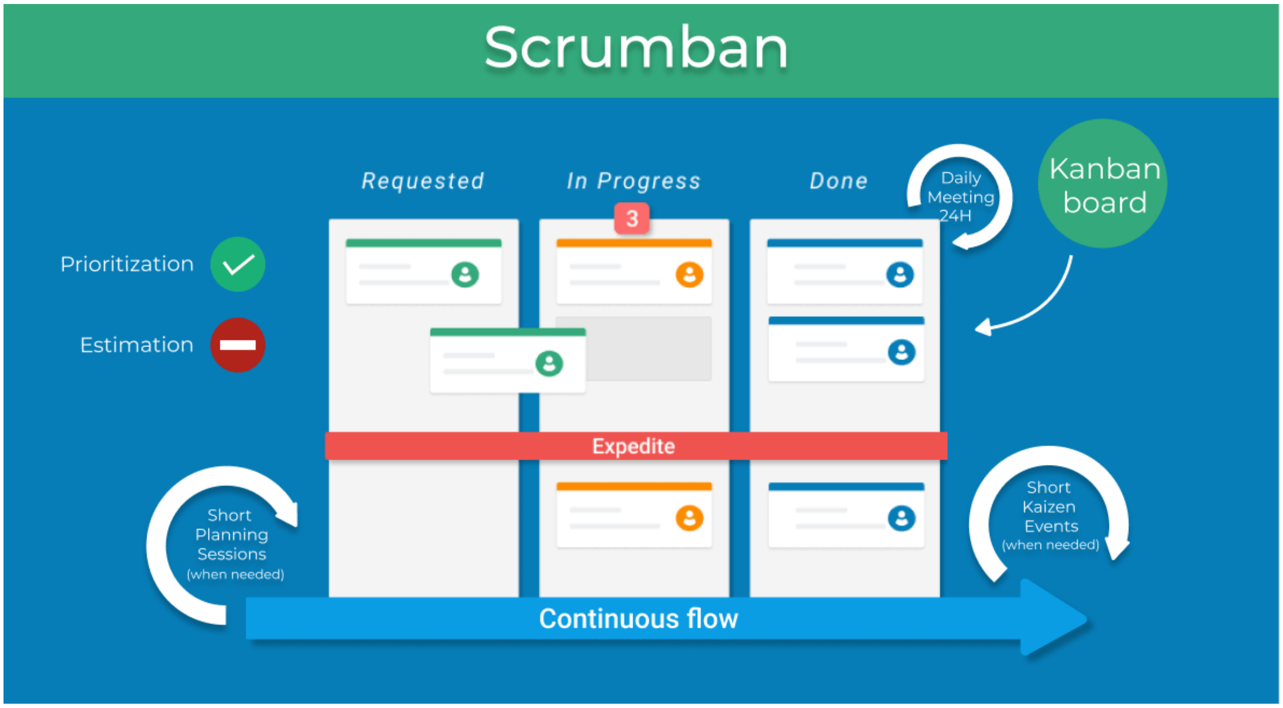
Limit Work in Progress (WIP). If you know anything about Kanban, you will know that limiting WIP is the game-changer. One of the most important Agile metrics. Scrumban takes this practice and applies it successfully as it makes it possible for teams to focus on finishing work. Limiting WIP is a good prerequisite for creating a pull system, where the tasks enter naturally into the workflow, instead of being pushed.

Extend the board. In other words, add more columns to the board. This is typical for Kanban and it is a great way to visualize the different workflow steps on the board. This way, your team can have a better overview of the process and it will help you discover where exactly in the process bottlenecks appear.

Prioritizing. Scrumban applies another technique from Kanban – prioritizing. It is pretty simple. You order the cards in the Requested (To Do) column and there is one simple rule: the top one is the most important. Having in mind this rule, the team starts pulling the cards one by one.

(Stop) Estimating. This is probably where Scrumban cheats on Scrum. Why is that? Scrumban states that you don’t need to estimate work. Here is the thing. According to Lean every activity which doesn’t add value to the end result is considered waste. In this sense, estimating is a wasteful activity. Here is why in Scrumban the planning sessions are relatively short and they are focused on prioritizing, instead of estimating.

Planning on demand. This is one of the main differences between Scrum and Scrumban. Scrumban eliminates Sprint planning in its initial form. Instead, the team plans if there is a need for this. In other words, the team pulls work items from the backlog until it gets empty, which is a trigger that the team should plan more tasks.



As you can see Scrumban takes Scrum to the next level by applying the Kanban principles and practices. This allows teams to increase output and decrease waste while providing visibility and higher productivity. It also lets teams apply Agile planning in its full potential.

Other Honorable Mentions

There are other Agile project management methods that had a positive impact on developing the Agile community, but during the years they were slowly pushed aside. This is why we won’t dedicate separate paragraphs to them. However, we have to mention that some of them are: XP, Crystal, FDD, DSDM, and others.

In the beginning, Scrum was accepted pretty well (and still is) and it became mainstream in the software development industry. However, through the years, Kanban, Scrumban and the usage of hybrid models grew in popularity and it helped Agile spread across various industries. After all, Kanban, Scrum, and Scrumban are the top 3 that successfully crossed the chasm and spread across other industries such as product development, architecture, marketing, financial services, healthcare, insurance, education, and others

The Origins of Lean Project Management

The story of Lean begins in post-World War II. Japan’s devastation during the war led to scarce equipment and resources, and manufacturers had to invent ways to thrive in a new economic environment. The United States sent consultants to Japan to help the country’s manufacturers rebuild their production capabilities. One of these experts was W. Edwards Deming, a management consultant whose ideas about quality control found more receptive audiences in Japan than they had in the United States.

It was from these consultants, as well as from visits to Ford and American supermarket chains, that Japanese manufacturers, and Toyota in particular, refined the concept of Just in Time (JIT). This technique aims to increase efficiency and decrease the amount of stocked inventory by moving materials into position just before they are needed for the next stage of the production process. JIT is not used solely in manufacturing - the technique applies in any situation where a supplier delivers materials using a timeline determined by customer demand. The success of JIT depends on the ability to synchronize and coordinate steps of the manufacturing process so that materials and products are where they need to be, when they need to be there.

In the 1950s, JIT, in combination with the Japanese manufacturing method of Jidoka or autonomation (automation with a human touch on an exceptions basis), would become the twin foundations of the Toyota Production System (TPS). Many consider Toyota engineer Taiichi Ohno the father of TPS and Lean. TPS was geared towards meeting the needs of the Japanese markets at the time, which called for smaller numbers of several different vehicle types. Its core principle was the systematic removal of waste in an ongoing effort to improve efficiency.

A couple of decades later, after the 1973 Arab oil embargo caused energy crises in the United States, Japan, Canada, UK, and the Netherlands, other Japanese companies began to study and imitate TPS. By now, the benefits of TPS were clear. It brought:

- Reduced lead times
- Lower inventories
- Decreased costs
- Improved productivity
- Higher profit margins
- Increased product quality
- Greater customer satisfaction

The concepts of Muda, Muri, and Mura (three types of waste that are known as the 3M) are central to the idea of eliminating waste. Muda refers to activities that consume resources without increasing the end value delivered to the customer. Muri refers to practices that involve overusing equipment or overworking employees beyond reasonable or practical limits - both of which increase costs and decrease efficiency and productivity in the long run. Mura describes operational “unevenness,” which can be thought of as the irregular performance of work that increases costs and possibly decreases efficiency over time.

Lean Migrates from Japan to the West

In the 1980s, Western manufacturers discovered that Japanese companies were outperforming them. They tried emulating TPS, employing it under such names as World Class Manufacturing, Stockless Production, and Continuous Flow Manufacturing. Manufacturers also began to implement some of the Lean manufacturing techniques, though in isolation from the overarching business management philosophy.

In 1988, a quality-engineer-turned-MBA-student named John Krafcik wrote an article that began a paradigm shift in American manufacturing. Krafcik, who had worked at New United Motor Manufacturing, Inc. (NUMMI), a car manufacturing company jointly owned by GM and Toyota, published *Triumph of the Lean Production System* based on his Master's thesis at MIT. Scholars at MIT's International Motor Vehicle Program continued his research into Lean production.

In 1990, three scholars, James P. Womack, Daniel T. Jones, and Daniel Roos, released an international bestseller, *The Machine that Changed the World*, that played a hugely important role in disseminating the concept of Lean manufacturing in the West. In that book and *Lean Thinking*, the principles of Lean were introduced in a way that allowed Western manufacturers to understand the full extent of its benefits.

Lean Thinking: The Spread of Lean's Influence

The principles of Lean manufacturing, now more broadly referred to as *Lean thinking*, have since been adopted outside of traditional manufacturing in fields like construction, healthcare, financial services, government, project management, and knowledge work. Using Lean for knowledge work has been met with some doubt and resistance by people who argue that because the field is essentially non-replicable and non-repetitive, it is not suited to standardization. Bradley Staats and David M. Upton argue in Harvard Business Review, however, that all companies specializing in knowledge work will perform non-knowledge-based activities that are suited to waste reduction efforts. Furthermore, you can even streamline sequences of core knowledge-based activities to achieve greater efficiencies.

For example, take Kanban, a Toyota practice that uses visual aids (such as signs, cards, or sticky notes) to match inventory with demand throughout the production life cycle. This makes process inefficiencies, bottlenecks, and other types of waste apparent. Kanban has been successfully used in software development, by visualizing the software development process as a production chain, identifying positions and situations that cause inefficiency in the production chain, and implementing solutions to increase the overall efficiency of the production chain.

Lean has had widespread influence. For example, the Lean Aerospace Initiative was a 1992 US Air Force-funded pilot project at MIT that examined the use of Lean techniques in manufacturing aerospace products. The project was renamed the Lean Advancement Initiative until it disbanded in 2012.

Healthcare, financial services, education, retail, construction, and other fields currently incorporate the principles of Lean based on the TPS. Because Lean is a paradigm that governs everything an organization does, rather than a single tactic or initiative aimed at a narrow outcome, it can be applied to a range of industry and organization-types.

Ultimately, Lean transformations today focus on shifting an organization's thinking so that it optimizes its purpose (providing greater value to the customer), process (maximizing workflow productivity) and people (how the team can best engage in continual improvement).

Applying Lean to Project Management

A Project Management Institute conference paper by Aziz Moujib describes Lean project management as the application of Lean manufacturing principles to the project management process. This is in an effort to achieve the same goal: maximizing value while minimizing waste. It draws from a set of five core principles identified in the book *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, which was written by two of the three MIT authors who wrote The M*achine that Changed the World*.

The concept of the value stream is central to Lean project management. This is the sequence of activities involved in delivering a project with an agreed-upon value (both the inputs and outputs). Value stream mapping, sometimes called business process mapping, an effort to understand how value and waste are created during the project lifecycle with the goal of optimizing the value stream. In doing so, Lean project management can help achieve a number of goals including:

- Improving the quality of the final product
- Completing the project on time, and reducing the time to completion
- Completing the project on budget and meeting project performance requirements
- Eliminating waste
- Reducing costs
- Adding value

Standardization is another critical aspect of Lean project management. Since most projects are novel (to some extent), standardizing tasks can both improve project performance in the short term and help improve efficiencies for projects with similar tasks in the long term. Improvement of tasks in the project lifecycle tends to be incremental, leading to gradual progress towards goals.

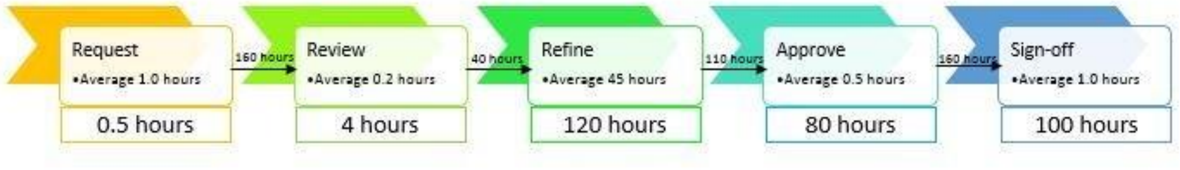
The Five Core Principles of Lean Thinking

The adoption of Lean thinking owes much to how it was presented to James P. Womack and Daniel T. Jones, the authors of *The Machine that Changed the World and Lean Thinking*. Womack and Jones defined five core principles of Lean thinking:

Understand Value: The first principle stresses understanding a product's (or service's) value in the eyes of the customer. The amount a customer is willing to pay for a product or service is directly related to how much they value it, so understanding the value of a product is the first step towards effective pricing and Lean management. Toyota, for example, adopted a top-down pricing approach defined by how much customers were willing to pay for a product with a certain value, and then focused on eliminating waste from their manufacturing processes in order to meet this price.

Map the Value Stream: The value stream is the complete sequence of activities involved in delivering an end-product with an agreed-upon value, and mapping the value stream means using visualization techniques such as Kanban, flowcharts, or spaghetti diagrams to represent this flow. Toyota pioneered the technique of value stream mapping, which allows business managers and strategists to identify parts of the value stream where waste occurs, and optimize the value stream to reduce waste. A spaghetti diagram is a great starting point because it visually documents the actual flow of product, paper, and people in a workplace or project workflow. Use the template for a spaghetti diagram below to make your own.

Experts recommend creating a value stream map with pencil and paper and documenting all the process steps that your product goes through, from supplier to your organization and finally to the customer.



Ensure that the Value Stream Flows: The ultimate goal of value stream mapping is the preservation and optimization of flow — the rate and "evenness" with which items and information proceed through the value stream. This is the principle of JIT manufacturing in action: because excess, early, or unexpected inventory creates waste, synchronization is the key to optimizing flow. Identifying and eliminating work that adds no value (either directly or indirectly) can also improve the flow of a value stream.

Employ a Pull Approach: Traditional manufacturing employed a push approach, where production targets are set based on an internally-determined schedule and production quota. This approach not very responsive to customer demand, and commonly led to production exceeding or failing to meet demand. In the first case, you would have to store the surplus product; in the second, you would have to increase the rate of production, possibly beyond optimum efficiency levels, to meet the demand. Either way, this approach creates a lot of unnecessary waste.

By contrast, a pull approach allows customer demand to determine production, so that nothing is created unless a customer asks for it. Done correctly, this eliminates waste caused by inventory costs and overwork. A pull approach is, however, difficult to implement effectively because it relies on accurate, effective assessment of the market and the ability to vary production quickly and on demand. Delivery must be speedy to ensure that customer demand still exists by the time the end-product is ready. Finally, a pull approach also requires highly effective coordination of information throughout the value stream, so that everyone is aware of production requirements and inefficiencies don’t arise because of confusion and mismatched expectations.

Pursue Continuous Improvement: At its heart, Lean management is an ongoing, incremental process. A waste-free system may be practically unattainable, but as a goal, it drives a need for constant improvement. The Japanese word Kaizen is often used to describe this practice in Lean. With Kaizen, the value stream is continually optimized, and defective processes are consistently improved or replaced in an effort to improve quality.

Other key principles in Lean software development include amplifying learning, deciding as late as possible, delivering as fast as possible, and empowering the team.

Lean Thinkers Obsess About Waste

As we’ve discussed, eliminating waste is the central focus of Lean. Waste in manufacturing or construction is easy to visualize: unused resources, unnecessary effort, perhaps refuse or byproducts. For work that doesn’t involve a physical end-product, however, waste can be a little harder to visualize. What sort of waste would you imagine from, say, a software development project?

As it turns out, the waste concept in Lean thinking stretches far beyond physical waste. Lean product development expert Ron Mascitelli describes waste as “anything the customer would not agree to pay for,” and Lean software developers Mary and Tom Poppendieck say waste is “anything that does not add customer value.”

TPS and traditional manufacturing identify seven types of waste (or muda in Japanese). Though these waste types were created with physical end-product manufacturing in mind, they translate well to non-physical projects, too. Let’s look at the seven types of waste, and show how they can be interpreted outside of traditional manufacturing.

Seven Areas of Waste in Lean

Overproduction: Used traditionally to refer to waste created by push manufacturing, this category covers surplus production and large inventories. Overproduction in software projects also refers to creating a product before establishing the demand for it. Overproduction may also refer to the mistake of providing functions, features, or services that the customer is not willing to pay for, which means that some of the work done on the project is unnecessary or redundant.

Waiting: This term traditionally referred to the time between a product being ready to move to the next stage in a production cycle and the product actually being moved to the next stage. In manufacturing, waiting occurs because of bottlenecked processes; in soft project management you can extend that definition to include the time that information required to proceed to the next stage is unavailable.

Transportation: This refers to the cost incurred and time spent physically moving a product from one place to another, especially as it is being produced. The potential costs of transportation extend beyond the time and money expended in the transportation itself, as transportation also raises the risks of damaging products. Inefficiency increases when production processes require goods to unnecessarily travel more around factories. Transportation waste is less of a problem in service projects, where communication is mostly digital and instantaneous. But inefficient paper trails and communication failures such as power outages or IT downtime are still problematic.

Over-processing: In manufacturing, over-processing refers to doing work that is not needed. This could be painting areas that won't be seen or tolerances that are tighter than required. This imposes costs related to labor, materials, and equipment wear. In service projects, over-processing takes the form of convoluted, redundant hierarchies and levels of approval, as well as the software development scenario of creating more iterations of a software product than actually needed to realize the value of the product.

Inventory: In manufacturing, a push approach may result in excess inventory, which raises transportation-related waste and can consume usable space. Inventory can also prevent the identification of problems with the workflow. For service projects, inventory costs tend to be mitigated, but the excessive stockpiling of information and difficulty in retrieving information when needed are analogous.

Motion: Motion-related wastes occur - in manufacturing, hard projects, and soft projects - when workers have to move too much to perform their tasks economically. Again, this is less of a problem with knowledge work, when you can quickly pull the required digital resources. Non-digital resources, however, can constitute a substantial proportion of resources used even in soft projects, and inaccessibility causes wasted motion.

Defects: The problem of defects is similar for all types of work, and entails reworking and using more resources than should have been necessary. The difference lies in how defects originate. In manufacturing, defects are typically caused by faulty equipment or operator errors, while in knowledge work they stem from poor design or from inaccurate estimation.

The Lean Six Sigma methodology also refers to an eighth waste: underutilized skills or brainpower. This type of waste is primarily associated with knowledge work and refers to the waste that occurs when not tapping into a worker's full mental potential. This can occur when companies hire overqualified employees or place workers in positions where they can't fully exercise their abilities.

How Lean Can Prevent Fatal Project Mistakes

Lean principles and the recognition of waste can help project managers avoid, mitigate, or control situations that might otherwise lead to project failure. Here's a list of common project pitfalls that Lean thinking can help avoid:

Failing to Establish Customer Value: Not understanding what a customer values in your project can cause you to misprice the project, and waste work and resources. When you understand the value your undertaking offers customers, you can more clearly establish project requirements, price the project according to what a customer is willing to pay, and revise work streams to meet this target price.

Scope Creep: Scope creep occurs when the value of a project is increased (usually due to customer requests), but corresponding changes in budgeting and pricing don't account for the increased value. You can avoid this problem by understanding and reevaluating value to the customer when scope changes occur, and ensuring that increased value is accompanied by a change in pricing.

Failing to Define the Value Stream: Value stream mapping is an excellent way to see how project activities create value, and is vital for trimming activities that don't create value for the project. If you do not map or define the value stream with an eye to optimizing it, non-value-creating activities may continue to strain the project budget and extend the project schedule.

Lack of Stakeholder Commitment: In a perfect world, projects would always finish on time and never exceed planned cost. In reality, most projects suffer due to cost and schedule overruns, so having the stakeholder's full backing is vital. A stakeholder who is not fully committed to the project may be less likely to extend support when a project needs to dip into its contingency reserves or request emergency funds, thus drastically exacerbating the waste creation problem. Learn more about securing and maintaining support from stakeholders in *The Definitive Guide to Stakeholder Management*.

Lack of a Communication Plan: An effective communication plan streamlines the flow of information between a project's stakeholders. Without effective, timely communication, projects run the risk of wasting time and resources on time-consuming approvals, delayed progress, and value mismatches.

Three Popular Lean Project Methodologies

Now that we understand the principles of Lean thinking, we can look at how three of the main Lean methodologies—the Deming Cycle, Six Sigma, and Kanban—approach project management. They all follow a disciplined approach to project management, stress optimization of the value stream, and map the value stream in its current state. After identifying inefficiencies and waste and making modifications, a map of the value stream's future state lays out the reengineered, optimized value stream with an improved flow. The future state can be thought of as an interim stage between the value stream as it currently exists and a hypothetical, ideal value stream. As such, value stream mapping is an ongoing process, and new measures to improve the value stream are regularly designed and implemented.

At a more granular level, each of the processes that constitute the value stream are made up of a sequence of steps; you use metrics to assess the performance of these sequences. Applying Lean management principles within value stream processes can improve performance on these metrics on a micro scale and reduce waste on a macro scale.

Remember that whichever methodology you use, successful Lean projects will all seek to continually improve flow through the value stream. To do this, they streamline the flow of information, examine the value stream and its constituent processes for redundancies, and aim to simplify and standardize to reduce waste. For development projects, they may adopt a lifecycle model that uses concurrent processing in an effort to mitigate waste caused by bottlenecks.

A Lean project will also adopt a pull (rather than a push) approach. This means that the customer recognizes the demand for the project before the project gets underway, and the project is completed in response to this demand. Since Lean thinking is such an overarching concept, an organization that embraces Lean management principles will usually have been successful building a culture of Lean thinking among its employees. Moreover, Lean management allocates decision-making responsibilities throughout the company hierarchy to minimize waste caused by unnecessary lengthy approvals and bureaucracy. Often, this means trusting lower-level employees and empowering them to make decisions for which they are qualified, without engaging in a wasteful review-and-approval process.

The Deming Cycle: A Method for Ongoing Quality Improvement

The Deming Cycle, also known as A3 problem solving because it was traditionally done on sheets of A3 paper, takes its name from W. Edwards Deming, the management consultant who helped Japan rebuild their manufacturing capabilities after World War II. Deming, who is sometimes referred to as "the father of quality control," created the Deming Cycle to facilitate the constant improvement of business processes. The Deming Cycle is also known as the PDCA Cycle or PDSA Cycle (for Plan, Do, Check or Study, and Act).

Planning entails conceptualizing and designing a plan to improve a process. *Doing* is enacting the plan and testing its results using performance metrics. *Checking* or *Studying* involves determining whether the improvement plan was successful, and *Acting* is the permanent implementation of the plan to improve the business process.

There’s some debate among quality gurus about whether PDCA and PDSA are the same thing, but generally any distinction is considered too minor for the average practitioner to worry about.

The Deming Cycle methodology is geared towards addressing process-related problems with a single — or at least a primary — underlying cause. This cause is called the root cause, and the team in charge of improving the process will design one or more possible solutions to address this root cause.

You can use this template to conduct your own root cause analysis.

Practitioners also identify the current situation, or current state of the process, to determine how to best address the root cause and to identify how to change the process to target the root cause. Once you identify and target the root cause, the team will prepare a problem statement that lays out what they are trying to achieve, and establishes the metric to measure the solutions. These solutions are evaluated during the Doing and Checking phases. After conducting a cost benefit analysis to determine the optimum solution, the team recommends a plan of action to the decision-maker.

The Deming Cycle: Project Example in Education

The American Society for Quality cites as a Deming Cycle project example the Pearl River NY school district, which uses the technique for curriculum and instruction design.

The school district’s planning involves the analysis of student needs to identify gaps or areas for improvement in instruction. Doing is a two-step process that involves first building a curriculum that both adheres to state and national standards and to the students’ own needs, and then actually providing the instruction. Checking involves the use of school assessments to evaluate student performance with the new mode of instruction. Finally, Acting is the implementation of curricula and instructional methods that successfully serve the students’ needs.

“Throughout the school year, if assessments show students are not learning as expected, mid-course corrections are made such as re-instruction, changing teaching methods and more direct teacher mentoring. Assessment data become input for the next step in the cycle,” the case study notes.

Six Sigma: Data-Driven Method for Eliminating Defects

Six Sigma is a process improvement methodology that focuses on eliminating defects and minimizing variation in process outcome. It’s heavily data-driven. The name Six Sigma is a statistical reference to having six standard deviations fall between the process mean and the nearest specification limit, which effectively results in an error rate of 3.4 defects per million products or process outcomes. As such, Six Sigma’s primary goal is optimizing the consistency and precision of a process. You can read a complete guide to Six Sigma here.

Six Sigma was developed by engineers at Motorola in the mid-1980s, and Motorola later trademarked the name. The technique became a cornerstone of General Electric CEO Jack Welch's approach in the 1990s. It's important to note that while Six Sigma and Lean are not the same, the management philosophy and the methodology complement each other very well. Lean alters processes to remove waste, and Six Sigma alters processes to improve the quality and consistency of output. As such, they both play important roles in process reengineering. The Six Sigma methodology pursues process improvement through Six Sigma improvement projects, which adopt one of the Sig Sigma sub-methodologies. We'll discuss two of these sub-methodologies: DMAIC and DMEDI.

DMAIC: This acronym (Define, Measure, Analyze, Improve, Control) is used in Six Sigma projects that aim to revamp or improve an existing business process. The *define* phase involves defining the scope of the problem to be examined, establishing customer requirements, and setting goals for the project. *Measurement* is the evaluation of the process's current state through data collection. *Analysis* is the process of examining the data collected to identify the root cause. *Improvement* involves the use of process-improvement techniques to optimize the process, thus moving it to its future state. Finally, *control* involves monitoring the new, future-state process to ensure quality of output. You may repeat DMAIC until your reach the desired level of quality consistency.

DMEDI: This acronym for (Define, Measure, Explore, Develop, Implement) is used in Six Sigma projects that aim to design a new process. It's less data-driven and more creative than DMAIC. The *define* phase in DMEDI is similar to that in DMAIC: defining the process to be designed and the goals for the new process. *Measurement* is the identification of customers and their requirements. *Exploration* is the process of examining alternative process designs to evaluate which will best serve customer requirements. *Development* is the actual production of the design that is considered optimum for meeting consumer requirements. And lastly, *implementation* involves pilot-testing the new process to ensure that its output does indeed meet customer requirements.

Deciding whether to use DMAIC or DMEDI really depends on evaluating the state of a process, if it exists at all. Conventional thinking says that DMAIC is used to improve an existing process, and DMEDI to establish a new process. In some cases, however, an existing process might be so full of problems and root causes that it's easier to design a new process than to try and improve the existing one. If this is the case, DMEDI may be a better choice than DMAIC.

Six Sigma uses a martial-art-style belt system to indicate levels of certification. The belts, in order of increasing expertise, are: White Belt, Yellow Belt, Green Belt, Black Belt, and Master Black Belt. Black Belts and Master Black Belts lead Six Sigma problem-solving projects and train others seeking Six Sigma certification. Learn more on Six Sigma certification here.

Lean Six Sigma: Combining Focuses on Waste, Quality

Lean Six Sigma is a process-improvement methodology that marries the waste-reduction principles of Lean with the quality-improvement efforts of Six Sigma. Performance management practitioner Arun Hariharan explains that you can classify Lean Six Sigma projects into three broad categories:

- **Quality improvement projects** rely primarily on Six Sigma methods to improve the quality of a process output.
- **Revenue enhancing projects** rely equally on Six Sigma and Lean to improve quality, reduce waste (thus increasing speed), and thus to increase company revenues.
- **Cost savings projects** rely primarily on Lean principles to cut waste from a project, making it faster, more efficient, and thus cheaper.

Six Sigma: Project Example from Medical Manufacturing

Perhaps the most iconic Six Sigma project is one conducted at Motorola in the late 1980s, when Motorola engineers who wanted increased granularity in quality measurement switched from measuring defects per thousand opportunities to defects per million opportunities. This change in how to discuss and perceive quality would eventually lead to more than $16 billion in savings for the company via increased precision and conformance to quality requirements. Since then, Six Sigma and Lean Six Sigma methods have been implemented at thousands of organizations in the U.S., including Ford, General Electric, and the U.S. Army.

Another example of the benefits of Six Sigma is Orchid, a manufacturer of artificial joints used in hip and knee replacements among other products. Their process involves lost wax casting of molds that are used to form the artificial joint from cobalt. The company found that six percent of its molds would crack, wasting the cobalt.

A Six Sigma team including a plant receptionist began investigating the issue around 2010. They identified 200 variables in the manufacturing process that they speculated could contribute to the cracked molds. The receptionist noticed that there were two kinds of wax employed in the casting, virgin wax for the parts that would be in contact with surfaces implanted in the patient, and reclaimed wax for non-contact parts.

Testing by the Six Sigma team found that the waxes melted at different temperatures, which could account for the cracking. They sought to verify this with 100 castings using only virgin wax. In that test, not a single mold cracked. Sixteen years earlier, the company had started using the reclaimed wax to save 60 cents a pound over virgin wax, but the decision was causing waste of cobalt that cost $7.50 to $15 a pound. The move to all-virgin wax, a variable identified through Six Sigma methods, saved Orchid an estimated $2.1 million a year.

Kanban: Limit Work In Progress to Speed Completion

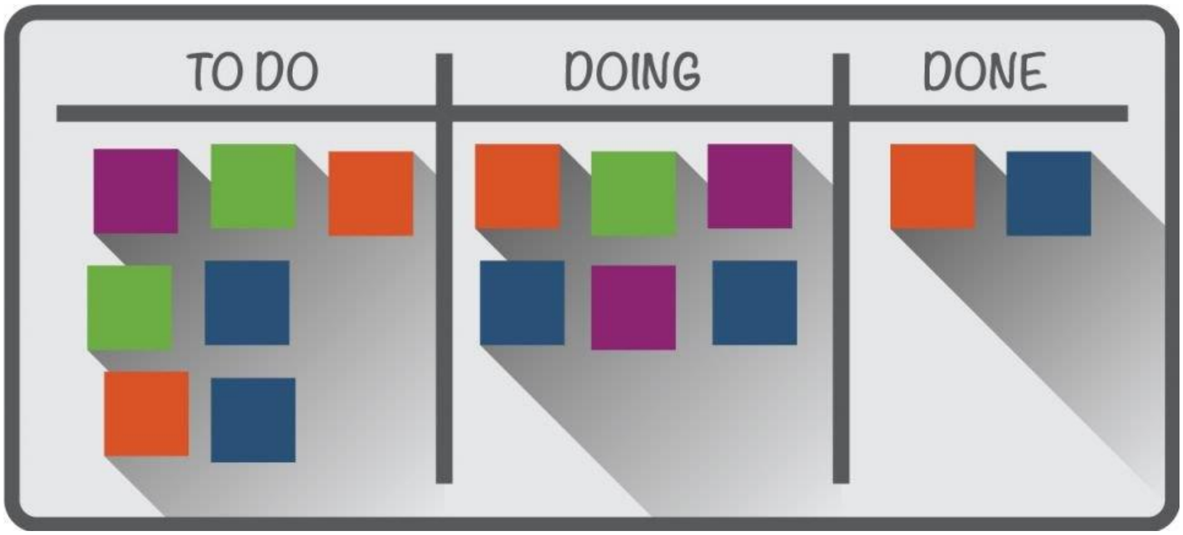
Kanban is another Lean method that originated at Toyota. This methodology focuses on eliminating backlogs of work in progress and keeping work flowing smoothly. To read how both manufacturing and software companies use Kanban, check out this reference.

In an ideal factory, work would proceed at a consistent pace — sometimes termed continuous flow — and would never be bottlenecked. In reality, however, some processes are slower than others, and if these processes are downstream of faster processes, they cause backlogs. Backlogs occupy space and cost money to store, and they can conceal problems such as quality defects. Read more about Kanban in inventory management here.

Toyota fixed this problem by implementing a system of visual cues — cards called Kanban cards — to indicate when a process was available to take on new work. As such, the pace of work is set by the slowest link in the production chain, which would “pull” production by using card signals to indicate when it was available to process work. As discussed earlier, the “pull” approach limits backlogging and controls bottlenecking to decrease waste. Kanban also limits the number of items being processed at any one time.

The classic Kanban system is a board divided up into sections, with a number of movable cards. Each section depicts a particular process, and individual cards represent work items that move through these processes. Visual cues makes it easy to spot inefficiencies and backlogs, so Kanban can be a great way to identify processes for improvement.

Given how easy and effective it is to use, Kanban is widely adopted by organizations and teams of all sizes that run multi-process production lines. Kanban is also ideal for knowledge work such as content publishing, which involves multiple processes (writing, editing, proofreading, and typesetting and printing). By using cards to symbolize individual pieces of content, you can track progress through the editorial and design chain, and allocate human resources to backlogs when they develop. You display the cards are on a Kanban board - this was traditionally a physical bulletin board but today is often created online.



Kanban has become popular in services and knowledge work. When applied to these projects, the main principles are:

- Visualize work
- Limit work in progress
- Teams pull work when they have completed existing tasks
- No sprints (time-based work intervals)

Kanban Project Example: Website Reaps Efficiencies

An impressive example of Kanban's success comes from Moneysupermarket.com, a top U.K. comparison-shopping website that serves more than 120 million users a year. The development team was suffering from high demand, constantly shifting priorities, poor morale, and low throughput. The use of the Kanban board highlighted obvious bottlenecks and blockers, and made it clear where the team needed to focus to address problems.

After implementing Kanban, the team's lead time dropped from 120 days to 25 days, and throughput soared. Developers cleared the backlog of 469 jobs in five months.

Implementing Lean: Tips for Making Lean Work for You

When implementing Lean, the most important thing to remember is that it is best employed as a long-term philosophy, rather than a quick fix to waste-related problems. Waste reduction is an ongoing process and developing a philosophy of Lean thinking and consistent performance measurement ensures greater, sustained benefits over time. It also makes the application of Lean principles easier, as workers learn to recognize waste through practice.

One of the ultimate goals of Lean is creating and maintaining a continuous flow of processes — where work moves through sub-processes without stopping and creating waste. To do this, organizations will implement "pull" approaches, thus drastically reducing the costs of inventory, storage, and maintenance by producing end-products on demand. In the long term, this "pull" approach leads to a more even distribution of work, which can mitigate the problems associated with overwork — for both machines and people.

Of course, Lean also recognizes that flow may be interrupted not only by production processes, but also by the flow of information. Fostering a culture of strong, rapid communication is vital to the success of Lean management. Implementing tools that facilitate communication, such as Kanban, is often a necessary supplement to streamlining and improving workflows.

To incorporate principles of Lean management, start with small, well-defined projects delivered to short deadlines. This is helpful for a number of reasons. For one, it teaches people to recognize waste and decide what to do about it. Additionally, it has the potential to provide quick gains through waste reduction. Since these tend to be more immediate for short projects, this can do wonders for project managers trying to develop a culture of Lean thinking in a project team.

Project managers should also champion the core Lean principle of continuous improvement (Kaizen) with regard to their people and teams. Encourage continued training and learning, and project managers can obtain certifications in Lean project management, such as those offered by Villanova University and the Management and Strategy Institute.

The Best Tools for Lean Project Managers

Lean project managers have developed some tools to aid in project management and organizational transformation. Some of these tools are conceptual or process frameworks, while others exist as software and systems.

Value Stream Mapping: As we've discussed, value stream mapping plots the flow of materials and information involved in the creation of a product. Use a value stream map to analyze the current state of a value stream and to design improved future states that remove waste and create value according to customer demands. Value stream mapping was traditionally done on paper, but online tools are now available, too.

Work Cells: In manufacturing, a work cell is a strategic arrangement of resources designed to improve the flow of a process and decrease waste. You may create work cell arrangements using either physical equipment or human resources, and often a combination of both. A cross-functional team is an example of a work cell based primarily on human resources, while manufacturing workspaces are an example of work cells centered on physical equipment.

One-piece Flow: One-piece flow is the practice of moving work items through a work cell one piece at a time (instead of in batches), and is done mainly to decrease work in progress. Processing one work item at a time is quicker than having each worker produce batches of work before moving them to the next step (as batches take longer than single items to process, and work cannot move downstream until the entire batch is complete).

Kaizen: Named for a Japanese word that roughly translates to "good change," Kaizen is an approach to work that emphasizes incremental improvements in processes and work streams. The end goals of Kaizen are improved efficiency and higher quality. The concept encourages an organization to welcome small, easily implemented improvements that, taken together, provide major benefits in the long term. A popular way of implementing Kaizen is to start with a Kaizen Event, a short-term project (around one week) with a single, specific improvement goal. A common plan for a Kaizen Event (also sometimes called a Kaizen Blitz) assigns a function for each day such as current state documentation, current state evaluation, characterizing future state, implementing future state, and operationalizing future state.

5S: A workspace organization method that organizes work-related resources in a manner that facilitates efficient, effective work. The name 5S comes from five Japanese words which, translated to English, mean *sort, set in order, shine, standardize,* and *sustain*. 5S is commonly used in Lean Six Sigma as a methodology to organize workplaces and use visual cues to achieve more consistent results. The steps involved are remove items not needed for current operations, label and arrange items so they are easy to use, keep everything tidy every day, standardize a system for keeping things in order, and avoid backsliding. While this system started in factories, it is equally relevant to offices. Try this 5S checklist in your workplace.

Poka-yoke: A mechanism for error- or mistake-proofing. It helps human operators identify, prevent, or correct mistakes as they occur. The idea behind Poka-yoke is accounting for defects as they occur, minimizing waste that would otherwise be caused if defects proceeded down the production chain. Typically, Poka-yoke are divided among control and warning mechanisms. One example of a control Poka-yoke are electrical connectors designed so that they will only fit together the correct way. In a broader sense, Poka-yoke can be any technique that safety-proofs a process so it can't be screwed up.

Gemba Walk: The Gemba walk is an application of the principle of observing work as it occurs, where it occurs. It is a supplement to the Kaizen process, and is based on the idea that work processes are best observed, and thus best improved, in real time and in the actual work environment. Sometimes described as "management by walking around," Gemba is more than a manager ambling around the office and overseeing work. In Lean, the purpose of Gemba is to observe, engage, and improve, and should occur where the most critical activity happens (the production line in manufacturing, the classroom in education, etc.). Your interactions are designed to engage the people and processes in Kaizen or continuous, incremental improvement.

Obeya Room: The Obeya room, or "war room," is a large physical space used during the development of a new product or process to facilitate interdepartmental thinking and communication. All individuals involved in the development process meet in the Obeya room to communicate and make decisions about a specific project. The Obeya room is typically furnished to facilitate discussion and problem solving, and speed up decision making.

Visual Cues: The basic premise of Kanban is that it's easier to understand and run processes visually than solely by words or numbers. As such, the use of Kanban-style visual cues (sticky notes, colored golf balls, visual control charts, or software tools that represent work items visually) is a proven way of simplifying and speeding up communication, as well as of facilitating understanding of a process.

Documenting Metrics and Progress: Since Lean thinking is ultimately concerned with decreasing waste and improving flow (and typically seeks to do so incrementally), it's vital to establish metrics that measure flow, and to consistently record performance on these metrics. You can assess flow, for example, using metrics such as work in progress, lead time, queue time, and throughput. It's best to measure these metrics are using software tools, which can quickly determine and visualize performance via graphs or summary statistics.

How Lean Relates to Agile Methods

In software development, there's a tendency to conflate the principles of Lean thinking with the methodologies of the Agile manifesto. So exactly how similar are they?

Agile methodologies are a set of iterative development approaches designed specifically to meet ever-changing customer requirements in software development projects. The signature characteristic of an Agile project is its flexible scope, meaning that Agile methodologies are designed to easily accept and implement changes in requirements. Agile software development consists of a series of iterations, and Agile software development teams target incremental improvements in each iteration.

There are some similarities between Agile and Lean project management. Both prioritize customer satisfaction - Agile through extensive customer feedback and iterations and Lean by identifying value through the eyes of the customer. Both also focus on incremental improvement, rather than big, one-time fixes.

Lean and Agile project management are similar in other aspects, too:

- Adopting a culture of blame-free employee involvement that ensures buy-in to the Lean philosophy and contributes to efficiency for Agile methodologies
- The role for a strong facilitator or project leader to ensure the project stays on track and effectively applies the principles of Lean or Agile
- Elimination of waste or redundant work, and the replacement or re-engineering of inefficient processes
- The practice of pipelining projects to ensure ongoing project delivery

In other fundamental ways, however, Lean is quite different from Agile:

- Lean is an extensive, far-reaching business philosophy that is designed to improve the efficiency of processes while eliminating waste. It results in process improvements that last for long periods and that will benefit future projects. By contrast, Agile is simply a method to ensure that a customer's requirements for a discrete project are met quickly and efficiently.
- Lean principles work best when applied throughout an organization, encouraging overall efficiency and improving entire systems of processes. Agile, by contrast, targets good, quick decision making within development projects, and it is not applied outside of project work.

Insights from *Scrum: The Art of Doing Twice the Work in Half the Time* by Jeff Sutherland

"At its root, Scrum is based on a simple idea: whenever you start a project, why not regularly check in, see if what you're doing is heading in the right direction, and if it's actually what people want? And question whether there are any ways to improve how you're doing what you're doing, any ways of doing it better and faster, and what might be keeping you from doing that." *– Jeff Sutherland*

All major projects require cycles of execution and improvement called Scrum Sprints. Scrum Sprints are typically conducted bi-weekly, and they contain the following 7 steps:

1. LIST: Create/Update a Backlog List (list document)

- Quickly list or update ALL desirable outcomes for the project.
- Format these outcomes to represent specific points in the 'User's Experience Story': Who (X) + What (Y) + Why (Z) - "As an operator (X), I use the touch screen to start the motor (Y), so I can control the pump remotely (Z)."
- Ensure each item is testable (can be built and tested according to a clear set of pass-fail requirements).
- Sort items in order of their ability to validate critical assumptions AND provide immediate value.

2. ESTIMATE:

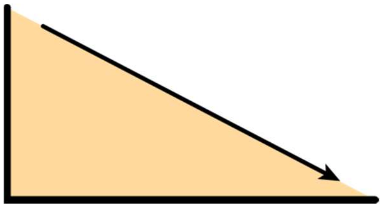
- Part 1: Refine and Estimate Backlog Items (list document with numbers and sections boxed off)
  1. Assign the longest duration item(s) with a 13 (highest Fibonacci number in the sequence: 1,2,3,5,8,13)
  2. Assign Fibonacci numbers 1,2,3,5,8,13 to all items, relative to the hardest item
- Part 2: Sprint Planning Session
  1. Set fixed Sprint duration (time till next evaluation - max 20% of the project duration)
  2. INITIAL SPRINT: estimate points to be completed within that time
  3. SUCCESSIVE SPRINTS: previous Sprint actual point total + 10%

3. POPULATE: Make Work Next Actions Visible

- Write all items that need to be completed during the current Sprint onto cards and put them in the 'DO' column of your Scrum Board (a Scrum board is a wall board with post-it notes OR software program with 3 lists: DO, DOING, DONE – I suggest using the software program Trello.com).
- Move the three top priority items into the 'DOING' column (never have more than 3 items in the DOING column).
- When you complete an item move it from the 'DOING' column to the 'DONE' column.

4. CHART: Make Work Progress Visible

- At the end of each day, take the total number of points in the 'DONE' list and subtract that that total from the Sprint total.
- Show the results on a 'Burndown Chart' (a line chart that reduces in value each day, with the y-axis representing the Sprint point total and the x-axis representing the number of day in the Sprint):



5. ASK: Conduct 15-minute Daily Stand-up Meetings

- What did I/we do yesterday to help the team finish the Sprint?
- What can I/we do today to help the team finish the Sprint?
- What upcoming obstacles might slow my/our progress?

6. DEMONSTRATE: Host a Sprint Demonstration

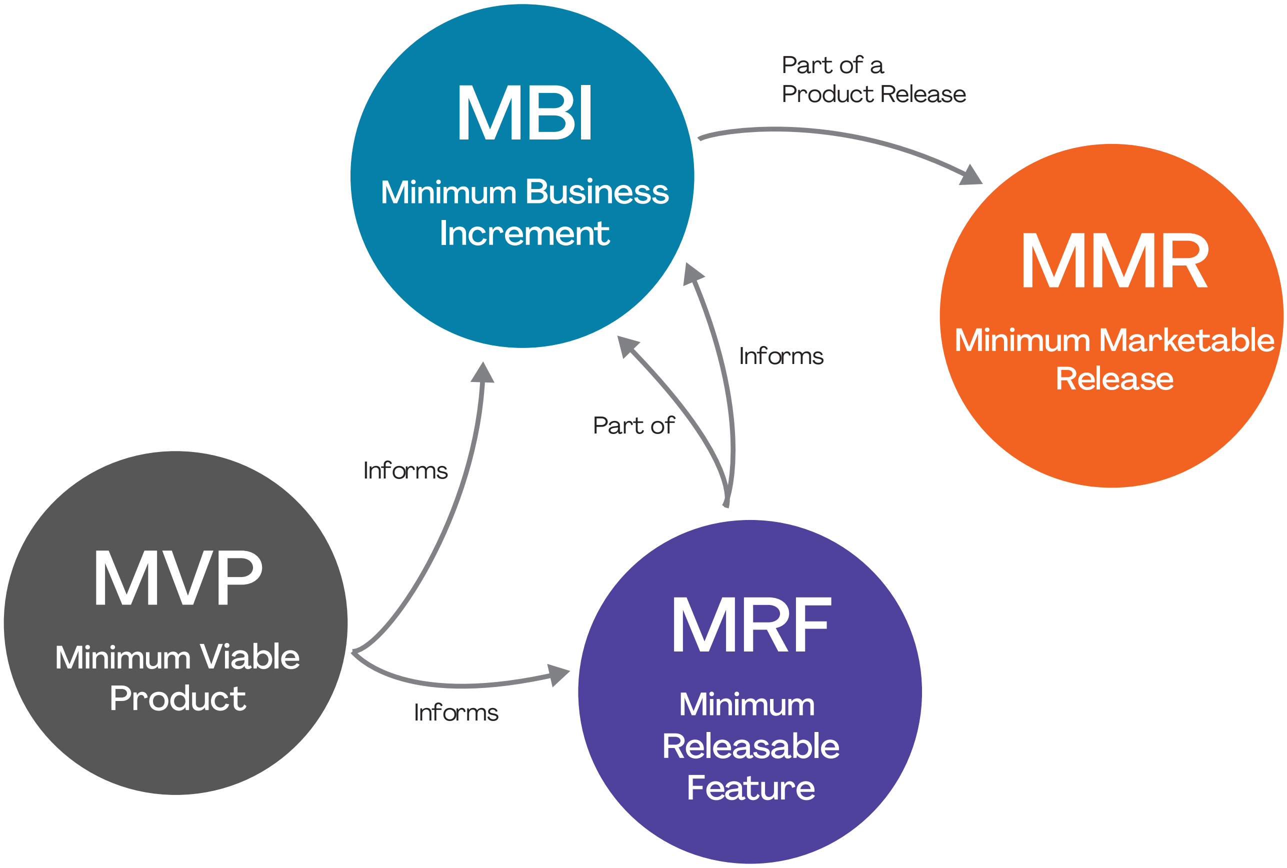
- Invite all project stakeholders (client, management, product owner or potential customer).
- Show the Sprint results (functional products only).
- Gather constructive feedback.

7. REFLECT: Conduct a Sprint Retrospective (lessons learned document)

- What went well?
- What could have been better?
- What can we do differently next Sprint?

Nathan Lozeron, @nlozeron
www.ProductivityGame.com

MVPs vs MBIs


MBI
Minimum Business Increment
Part of a Product Release
MMR
Minimum Marketable Release
Informs
Part of
Informs
MVP
Minimum Viable Product
Informs
MRF
Minimum Releasable Feature

Minimum Business Increment (MBI)

- The smallest, releasable chunk of value that makes sense from a business perspective
- Focused on the highest value, and quick realization of that value
- Targeted for a particular market segment
- Only artifact for which cost-of-delay makes sense

Minimum Marketable Release (MMR)

- A batch of one or more MBIs
- Sometimes called a Minimum Marketable Product (MMP)

Minimum Releasable Feature (MRF)

- The smallest feature that fits into an MBI or MVP
- A fully functional, single feature or function that provides real value to customers
- Could potentially be deployed on its own
- Sometimes called a Minimum Marketable Feature (MMF)

Minimum Viable Product (MVP)

- An investment in creating a new product or service
- Created to explore a hypothesis
- Often starts as a functional prototype where some functionality is simulated or performed manually
- Aim is to do just enough work to get something in front of potential customers to learn what they really want

Disciplined Agile