AGILE MANUAL

Business
Explained

> “If you adopt only one agile practice, let it be retrospectives. Everything else will follow.”

Woody Zuill

Business
Explained

Disclaimer

All the information in this book is to be used for informational and educational purposes only. The author will not, in any way, account for any results that stem from the use of the contents herein. While conscious and creative attempts have been made to ensure that all information provided herein is as accurate and useful as possible, the author is not legally bound to be responsible for any damage caused by the accuracy as well as the use/ misuse of this information.

INTRODUCTION

There are many methodologies from which to choose while planning a project. Nonetheless, software development teams find that waterfall isn’t one of the most productive methodologies. The Agile methodology divides projects into phases to encourage continual improvement in response to shifting priorities and shifting customer needs.

Agile is an iterative methodology for managing projects and creating software that speeds up the process of satisfying clients. An agile team avoids risking everything on a single release by dividing the work into manageable chunks and releasing them in iterations. Continuous evaluation of requirements, plans, and results provides teams with an in-built mechanism for adapting rapidly to new circumstances.

The agile mindset is based on the ideas and concepts outlined in the Agile manifesto. These norms and standards point the way toward adapting to new circumstances and dealing with the unknown. When confronted with ambiguity, it’s best to try something out, see how it goes, and make any necessary adjustments based on the data you collect.

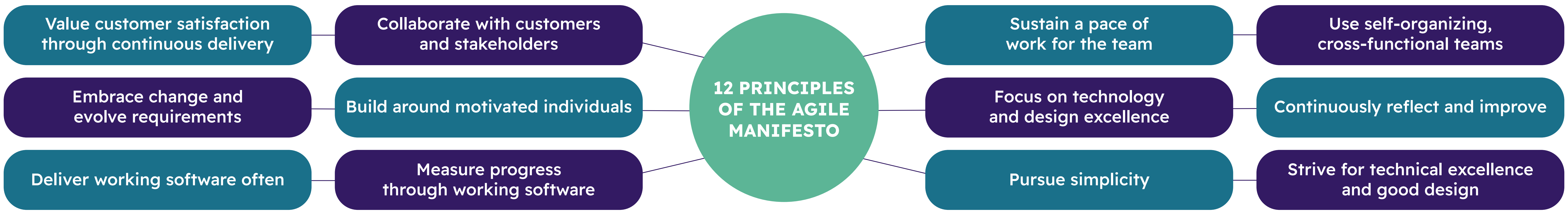
Although it was originally developed for software development, agile project management has proven effective for teams of different kinds. If you’re looking to get started with Agile, you’ve come to the right place.

THE AGILE MANIFESTO

Software and project management benefit from the Agile manifesto's ideals. A team of software engineers originally presented it in 2001 intending to develop a more malleable and adaptable project management method. Four guiding principles and twelve guiding ideals make up the Agile manifesto. These are the four pillars upon which the Agile manifesto rests:

- Prioritize people and interaction over focusing on systems and machinery
- Fully functional program over exhaustively documented
- Contract negotiation is secondary to customer collaboration
- Adapting to new circumstances instead of sticking to a set schedule

These principles stress the significance of cooperation with customers, flexibility in the face of change, and the primacy of delivering working software as a metric for success.



12 PRINCIPLES OF THE AGILE MANIFESTO

Value customer satisfaction through continuous delivery

Collaborate with customers and stakeholders

Sustain a pace of work for the team

Use self-organizing, cross-functional teams

Embrace change and evolve requirements

Build around motivated individuals

Focus on technology and design excellence

Continuously reflect and improve

Deliver working software often

Measure progress through working software

Pursue simplicity

Strive for technical excellence and good design

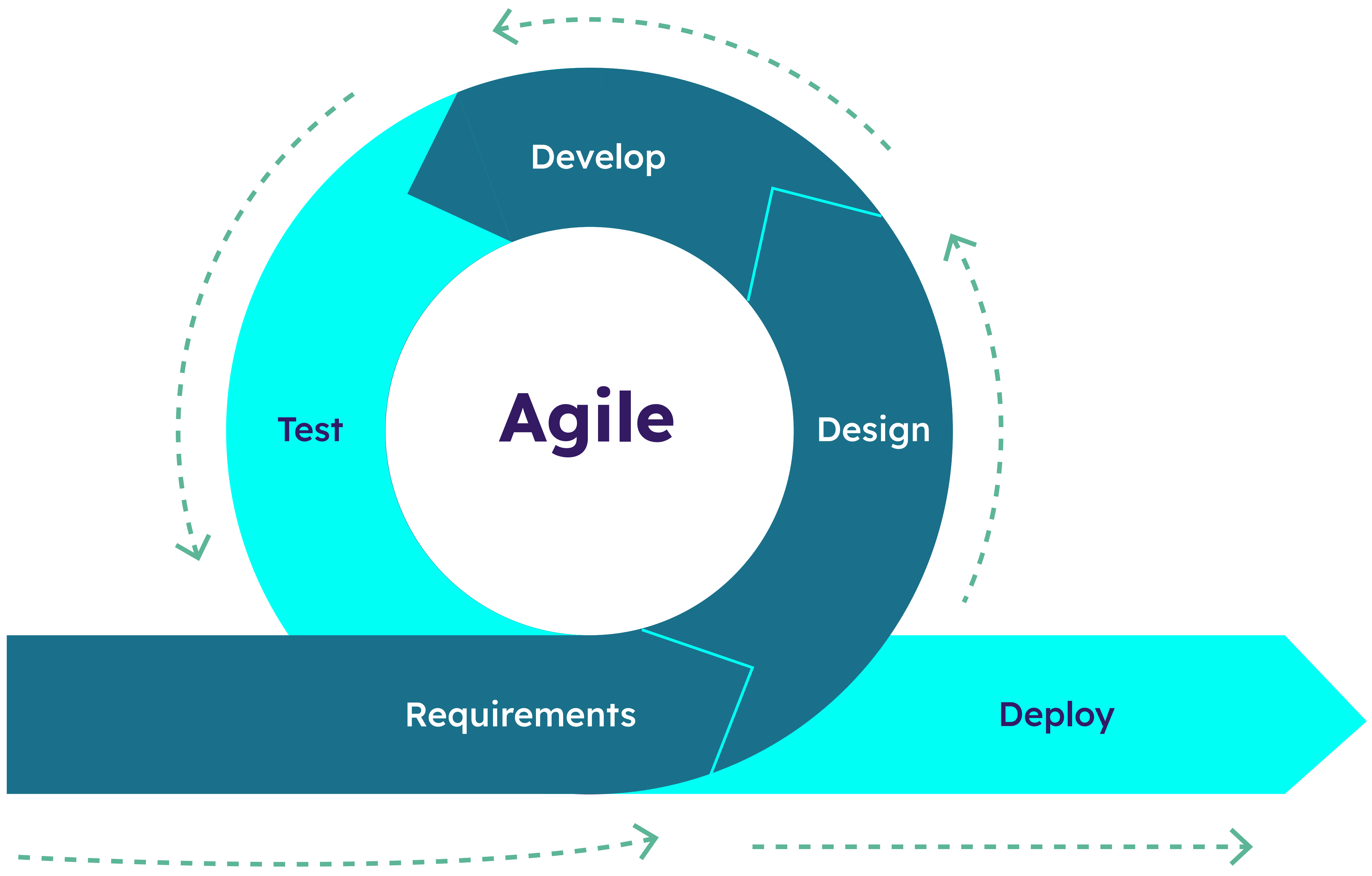
Agile’s twelve guiding principles are as follows

- Our top objective is to ensure that the customers are happy through the timely and consistent delivery of high quality software.
- It’s okay to make last-minute adjustments to your criteria. As a result of adopting an agile approach, customers gain a competitive edge.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale.
- Both business leaders and developers need to collaborate often during the project.
- Create initiatives centered on enthusiastic people. Trust them to perform the job, provide them with the required resources, and relax.
- Communicating with and within a development team through in-person meetings is the most efficient and successful technique.
- Having functional software should be the primary goal.
- Agile methods are more conducive to long-term growth. The pace set by the backers, developers, and consumers should be sustainable indefinitely.
- Maintaining a focus on technological superiority and well designed systems increases flexibility.
- There must be an emphasis on simplicity or the ability to get the most done with the least amount of effort.
- Self-organizing teams produce superior designs, requirements, and architectures.
- The team periodically examines its performance to determine where it might improve and adjusts its methods of operation.

These guidelines outline the best practices for delivering high-quality software, responding to requirements as they evolve, collaborating effectively, monitoring and improving performance, and much more. The Agile manifesto is more of a set of ideals and ideas than a rigid technique, making it flexible enough to be used for a wide variety of projects and teams. It is widely utilized in many industries, including software development, IT services, and others, to satisfy consumers and boost productivity.

AGILE PROJECT MANAGEMENT

Agile project management is an iterative methodology that divides the project lifecycle into shorter cycles, known as "sprints."
An Agile project is performed in chunks, much like Agile software development. For instance, in Agile software development, an iteration is a single cycle of the task. The project team, which should include representatives of the project's various stakeholders, reviews and critiques each part or iteration. The feedback on an iteration is used to figure out what comes next in the process.



devcom.com/tech-blog/agile-advantages-for-business/

An early advantage of using Agile Project Management is the flexibility it provides in addressing problems that crop up as the project progresses. At the correct time, a necessary change to a project can save resources and help to deliver a successful project on time and under budget.

SCALING AGILITY

Agile refers to the practice of employing Agile methods like Scrum and Kanban to coordinate and complete massively complex projects and initiatives. Organizations adopting agile at scale do so because it allows them to manage the complexity and uncertainty inherent in large projects without sacrificing the speed with which they can deliver value to customers.

One of the most well-known approaches to implementing Agile at a large scale is the Scaled Agile Framework (SAFe). By offering a framework for managing and aligning different Agile teams, SAFe enables enterprises to extend Agile across the enterprise. It offers advice on handling complex projects, and it’s based on Agile and Lean product development practices.

One such approach to implementing Agile on a massive scale is Large Scale Scrum (LeSS). LeSS, based on Scrum, is a framework for companies that employ several teams to create a single product or service. It retains the pliability and adaptability of Scrum while providing a framework for coordinating and synchronizing the efforts of different teams.

Traditional Scrum	Large Scale Scrum (LeSS)
Each team works from a backlog based only on it's own areas of responsibility.	The group works from a shared backlog for the entire project.
Each team has it's own "definition of done" for work on its own parts of the product.	There is a single "definition of done" for all teams in the group.
Each team works on its own sprint, focusing on its own area of the product.	All teams work on a common sprint, ocusing on the entire product.

productplan.com/glossary/less-large-scale-scrum/

When scaling Agile, organizations must ensure that all teams communicate and work together and that their goals are aligned. They must also put in place systems to monitor progress, mitigate risks, and provide value to customers. Creating a common understanding of the process, offering training and coaching for teams, and encouraging a culture of continuous improvement are all ways to guarantee that all teams are using the same Agile practices and principles and are working toward the same goal when implementing Agile at scale.

A strong emphasis on collaboration, communication, and alignment, as well as effective mechanisms for managing dependencies and risks, tracking progress, and delivering value to customers, is essential to successfully managing and delivering large, complex projects or programs using Agile methodologies like Scrum or Kanban.

ADVANTAGES OF AGILE METHODOLOGY

The advantages of using an agile methodology for project management are numerous. There are many benefits of using Agile, but some of the most notable are:

- Faster delivery of value: It is the goal of agile approaches to expeditiously provide benefits to the client. This is accomplished by dividing the whole project into smaller, more manageable pieces (iterations or sprints) and delivering working software at the conclusion of each iteration.
- Increased flexibility and adaptability: According to the founders of the agile methodology, “responding to change over following a plan” is the most important aspect of every project. This means that teams can adjust to new requirements or obstacles, making them ideal for projects with a lot of wiggle room.
- Improved collaboration and communication: Agile techniques stress the need for face-to-face interaction amongst team members, clients, and other stakeholders. That way, everyone can be on the same page and perform their part with the same knowledge of the situation.
- Greater visibility and transparency: With agile methods, you can easily track the development of your project at any time. This is accomplished through the use of Kanban boards and other methods of work visualization and by the declaration of process policies. It’s a useful tool for figuring out what’s holding teams back from achieving their objectives.

- Continuous improvement: Agile methods foster a mindset of constant refinement. Continual process evaluation and iterative change implementation are encouraged for all teams. Also, they employ feedback loops to spot and fix problems as they crop up.
- Increased customer satisfaction: In order to satisfy their needs, agile techniques prioritize speedy service delivery. This aids in making sure that consumers’ requests are being fulfilled in a timely manner.

AGILE TEAMS

Together, the members of an Agile team use Agile practices to develop and release a product or service. Developers, testers, product owners, and stakeholders are all common members of an Agile team. Since agile teams are self-organizing, they are also responsible for choosing the most efficient means by which to perform their tasks.

The principles of Agile development, upon which Agile teams are founded, place a premium on responsiveness to change. They are made to be flexible in the face of new demands or unanticipated obstacles and to provide customers with value as soon as feasible.

The key characteristics of Agile teams are:

- Cross-functional: Developers, testers, product owners, and stakeholders are just a few of the members of an agile team. It’s now possible for the team to take care of everything associated with the project, from development to launch it.
- Self-organizing: The greatest approach to getting work done is up to the agile team itself. They are trusted with discretion and encouraged to accept responsibility for their work.
- Collaborative: Delivering a product or service requires strong collaboration between members of an agile team. Nothing beats the power of face-to-face communication and cooperation when it comes to getting everyone on the same page and working toward a single goal.
- Adaptable: Teams using the agile methodology are flexible and can respond to new or shifting demands. They are adaptable and may make necessary changes as the project evolves.

- Focused on delivering value: In an agile environment, the priority is always on the client and how quickly they can receive a return on their investment in the product or service. At the end of each cycle, sometimes known as a “sprint,” they aim to release a working version of the software.
- Continuous improvement: Agile teams are strongly committed to continuous improvement, and as a result, they are constantly seeking new methods to enhance their process and provide even more value to their customers.

AGILE ADVANCED ROADMAPS

In order to plan and visualize the long-term course of an Agile project or product, experts recommend creating an “Advanced Agile Roadmap.” A group makes them of specialists to let everyone involved with the project know what they should be aiming for and what stage they are in. Timelines, dependencies, and resource allocation are just some of the features that may be incorporated into an advanced Agile Roadmap, along with other types of information. They’re modifiable as the project develops, and more data becomes available because of this pliability and adaptability.

Tools and approaches like user stories, epics, themes, and milestones can be used to construct them. In contrast to epics, which contain numerous features and functionalities, user stories are brief, straightforward descriptions of a single feature or capability requested by a user. Milestones are significant points in the project timetable that signify significant achievements or deliverables, while themes are collections of epics that are in accordance with the project’s overarching aims.

More advanced Agile Roadmaps may also make use of visual aids like Gantt charts, Kanban boards, and burndown charts. Kanban boards depict the workflow of activities and track progress, whereas Gantt charts display the chronology of the project and the dependencies between tasks. You can use burndown charts to keep tabs on how many user stories and other tasks have been completed in a certain time frame. The best Agile Roadmaps are collaborative and iterative, allowing for constant feedback and revisions as the project develops. This paves the way for nimbleness in the face of shifting needs and unanticipated obstacles. As the project progresses, stakeholders can provide more feedback and input by reviewing the Advanced Agile Roadmap and adding their own comments and suggestions.

AGILE CONVERSATIONS

The Agile development methodology relies heavily on open communication between team members. To guarantee the project is in line with corporate goals and user needs, they necessitate constant and transparent communication between team members, stakeholders, and customers. There are several types of Agile conversations, including:

- Daily Scrum: A daily meeting in which team members report on their accomplishments, identify obstacles, and plot out the next day’s activities.
- Sprint Planning: Sprint planning is a meeting when the team discusses what tasks will be performed in the following sprint.
- Sprint Review: The meeting in which the team presents the sprint’s worth of work to stakeholders and receives their comments.
- Sprint Retrospective: A gathering when the team looks back at the previous sprint, analyzes what went well and what may be better, and formulates a strategy for fixing any problems that were discovered.
- Backlog Grooming: User stories, tasks, and issues are all discussed and prioritized in this meeting.
- Customer/User Interviews: Feedback from customers and end-users is gathered at these meetings and used to guide future development.

These Agile discussions are vital because they help the team stay on the same page with the project’s objectives, guarantee that the product is satisfying the end-users demands, and allow problems to be discovered and fixed rapidly. Discussions in the Agile methodology are open to all participants, their ideas and feedback are shared openly, and the conversation evolves based on what has been said.

PLANNING AND ESTIMATION IN AGILE

Agile planning and estimating involve calculating the time frame, manpower, and materials required for a given project or product. Flexibility and adaptability are at the heart of the Agile planning and estimation method, which places a premium on regular assessments and adjustments as the project develops.

“User stories” are a common unit of work breakdown in the Agile approach to planning and estimation. A user story is a concise, easy-to-understand description of a desired feature or function. You can use them to determine the extent of your project and single out its constituent parts.

Planning poker is commonly used in Agile for estimation purposes. All team members contribute to an estimate for a user story using the consensus-based planning poker technique. Members of the team discuss the user story and then offer their best guess in the form of a “story point” (a relative unit of measure). The group then has a discussion about the estimations and agrees on a final figure.

The team can then utilize the estimations to generate a backlog of user stories. User stories that have yet to be finished can be found in the backlog. With the backlog in hand, the team may organize their upcoming sprint’s tasks (a time-boxed period of 1-4 weeks where a specific set of work is completed).

USER STORIES AND ACCEPTANCE CRITERIA

In Agile development, user stories are used to specify the features that the end user expects. They are brief, straightforward explanations of a function or feature written from the customer's point of view. The standard format for a user story is "As a [user], I desire [feature], so that [benefit]." The acceptance criteria define the conditions under which a user narrative can be considered finished. They defined the goals and expected results of the user narrative in a specific and quantifiable way. Bullet point style is common for acceptance criteria, which are normally developed in conjunction with the client or end user.

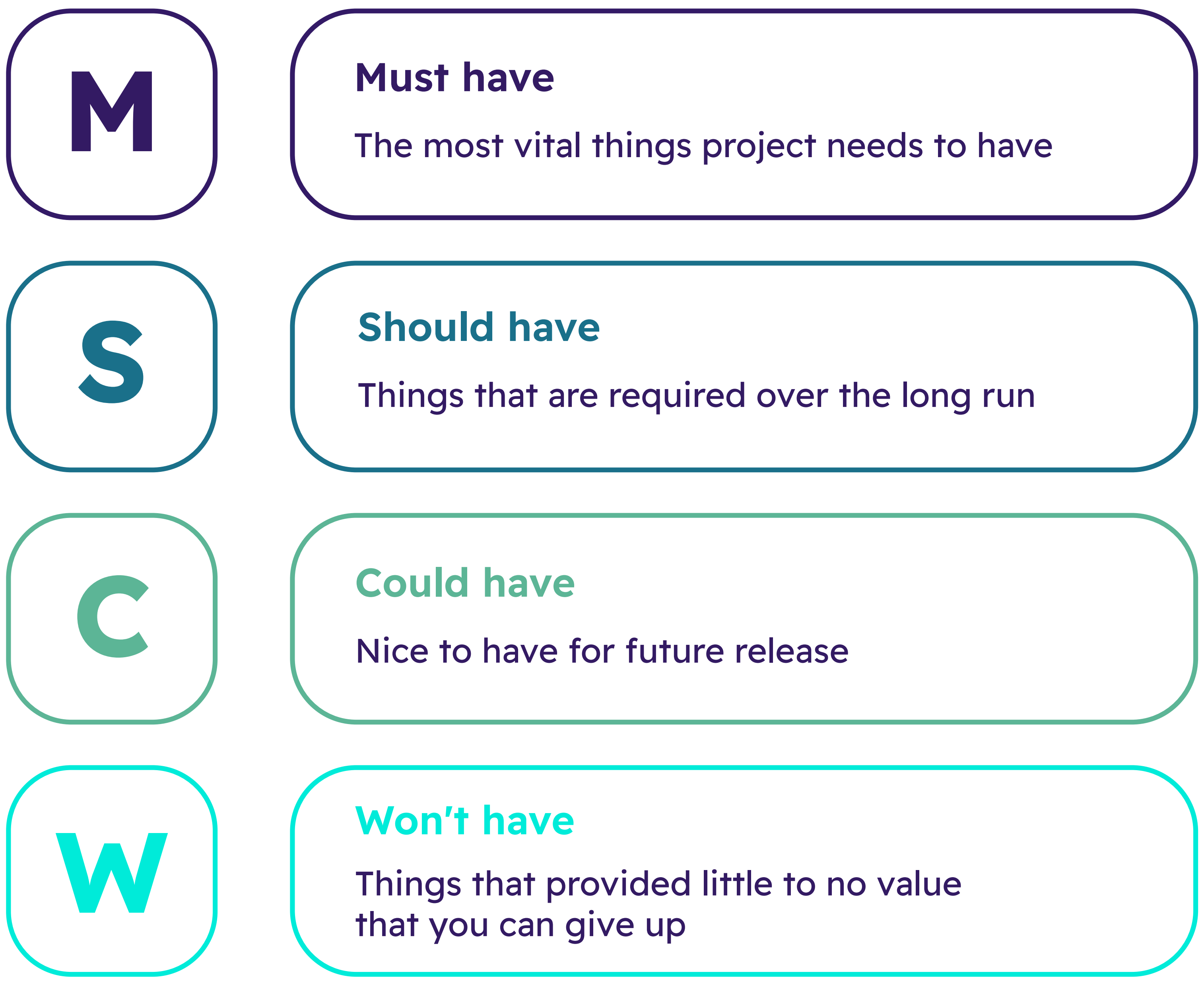
The team will be working towards the same goal and will have a better understanding of the user story if acceptance criteria are established. They help define the scope of the project and guarantee that the delivered product is suitable for its intended purpose. They also offer a quantifiable criterion for when a user story is finished, which is useful for monitoring development and prioritizing tasks.

Brief, easy-to-understand narratives told from the perspective of the product's end user are known as "user tales." The acceptance criteria for a user story are a list of requirements that must be met before the story can be considered complete. These criteria are commonly expressed in bullet point format and are developed in close collaboration with the client or end user. Together, they help to ensure that everyone on the team is on the same page and that the final product is exactly what the user wants.

PRIORITIZATION AND VALUE DELIVERY

Work items, such as user stories, are prioritized during the process of deciding in what order they will be performed. Prioritization is an iterative process that occurs throughout an Agile development project. In order to properly prioritize user stories, it is crucial to first grasp their relative importance. Prioritization works toward the end of providing the most benefit to the consumer or end-user as soon as possible. There are several common approaches to prioritization in Agile development, such as:

- MoSCoW: Prioritizing items based on their level of importance (Must Have, Should Have, Could Have, and Won’t Have)
- Kano model: Prioritizing items based on their level of customer satisfaction (Must-Haves, Attractive, one dimensional, and Indifferent)
- The Eisenhower matrix: Prioritizing items based on their level of urgency and importance (Urgent and Important, Not Urgent but Important, Urgent but Not Important, and Not Urgent or Important)



M **Must have** The most vital things project needs to have

S **Should have** Things that are required over the long run

C **Could have** Nice to have for future release

W **Won’t have** Things that provided little to no value that you can give up

Work items, such as user stories, are prioritized during the process of deciding in what order they will be performed. Prioritization is an iterative process that occurs throughout an Agile development project. In order to properly prioritize user stories, it is crucial to first grasp their relative importance. Prioritization works toward the end of providing the most benefit to the consumer or end-user as soon as possible.

DAILY STAND-UPS AND RETROSPECTIVES

In Agile development, daily meetings (called daily Stand-ups or daily Scrums) are held every day. The daily stand-up is a time for team members to report on their progress, identify impediments, and make plans for the following day. The meeting is meant to be a brief, focused check-in on keeping the team aligned and on track and is typically limited to 15 minutes or less.

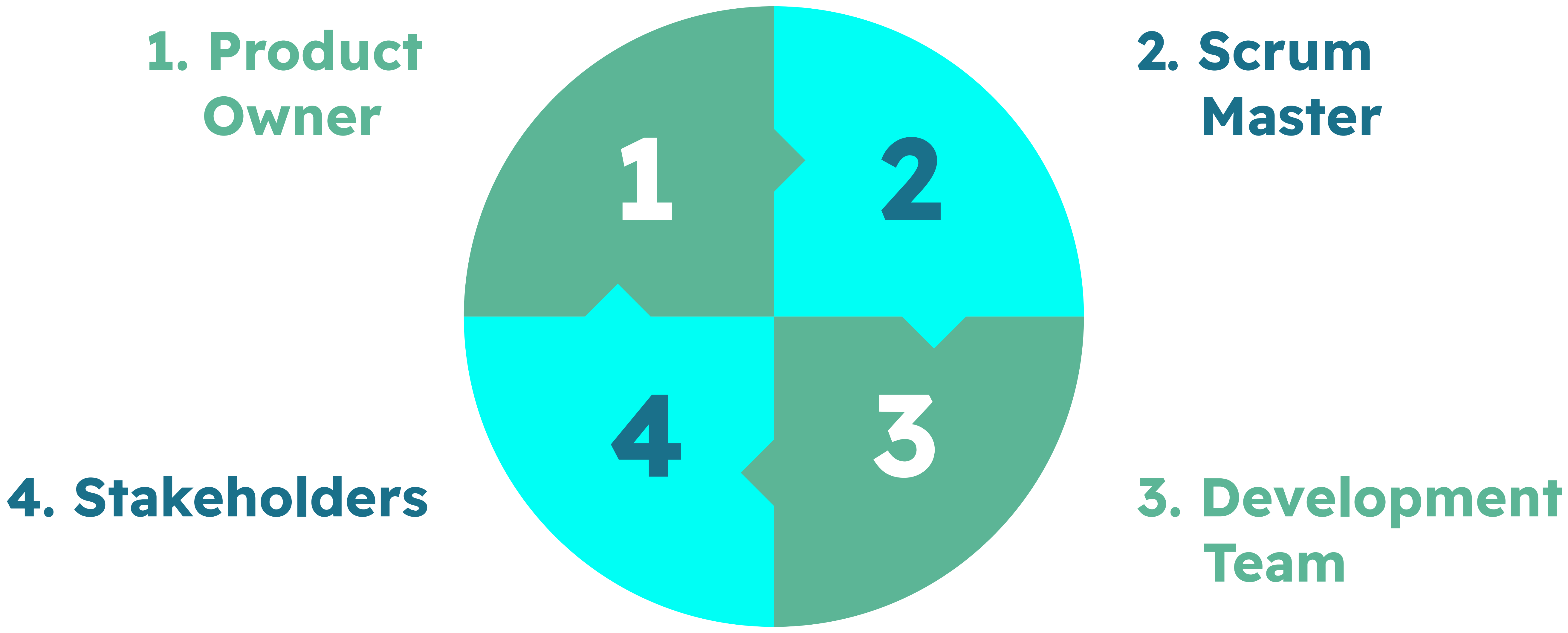
The Scrum Master is usually in charge of the daily stand-up, and it takes place at the same time and place every day. The team members respond to the following questions during the meeting:

- What did you accomplish yesterday?
- What will you work on today?
- Are there any blockers or issues that need to be addressed?

At the close of each sprint, iteration, or project, Agile teams gather for debriefing sessions, known as retrospectives.
The goal of the retrospective is to reflect on the previous sprint, identify successes and opportunities for growth, and formulate a strategy for tackling challenges. The retrospective is the group's chance to assess its current workflow and think about how it could be enhanced.
The Scrum Master usually takes charge of retrospectives, but everyone on the team is expected to contribute. A typical meeting will begin with a discussion of the sprint's successes and failures. The group then determines what led to the occurrence of the issue and devises a strategy to fix it. The team decides what they want to accomplish at the end of each sprint or iteration.

AGILE ROLES AND RESPONSIBILITIES

In an agile development team, everyone has a specific role to play.



Some typical positions and their associated duties in Agile teams are as follows:

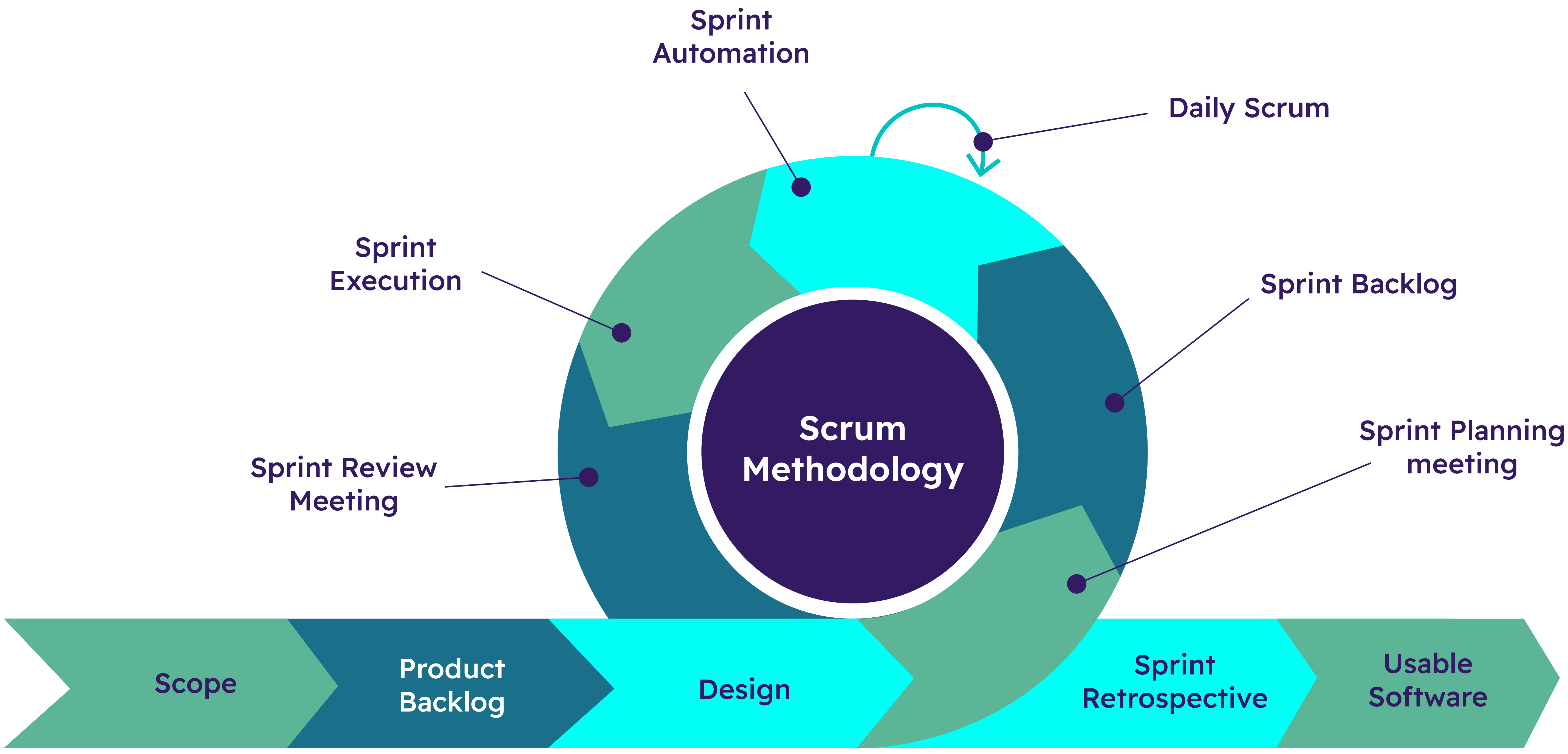
- Product Owner: They act as the customer and stakeholder advocate, guides the team's prioritization efforts, and guarantees the end product is satisfactory.
- Scrum Master: The Scrum Master's job is to help the team work more efficiently by adhering to Agile values and methods. The Scrum Master's other responsibilities include those of coach, mentor, and facilitator.
- Development Team: The delivery of the final product is the responsibility of the Development Team. They need to write up the user stories, create all the code, and test the product.
- Stakeholders: Customers, final users, and everyone else who is affected in any way by the project or product are all examples of stakeholders.

Participating in activities such as prioritizing user stories and accepting the finished product, they offer comments and suggestions throughout the project.

It’s important to note that in Agile development, the roles and responsibilities are not fixed and can change over time. Members of Agile teams are also expected to be able to switch gears quickly and easily, as well as take on new duties as they arise.

SCRUM FRAMEWORK

Scrum is an Agile development framework that emphasizes teamwork, adaptability, and rapid response to change. Scrum is a framework for managing software development projects that adheres to Agile principles and is meant to be both lightweight and straightforward.



inapps.net/everything-you-need-to-know-about-scrum-methodology-in-2022/

The Scrum framework consists of several components:

- Scrum Team: Delivering the final product falls within the purview of the Scrum team, a multidisciplinary group. The team consists of a Product Owner, a Scrum Master, and a development team.
- Product Backlog: To keep track of everything that needs to be done for the project, a backlog of tasks is created. The Product Owner is the person accountable for the product backlog and how the items in it are prioritized.

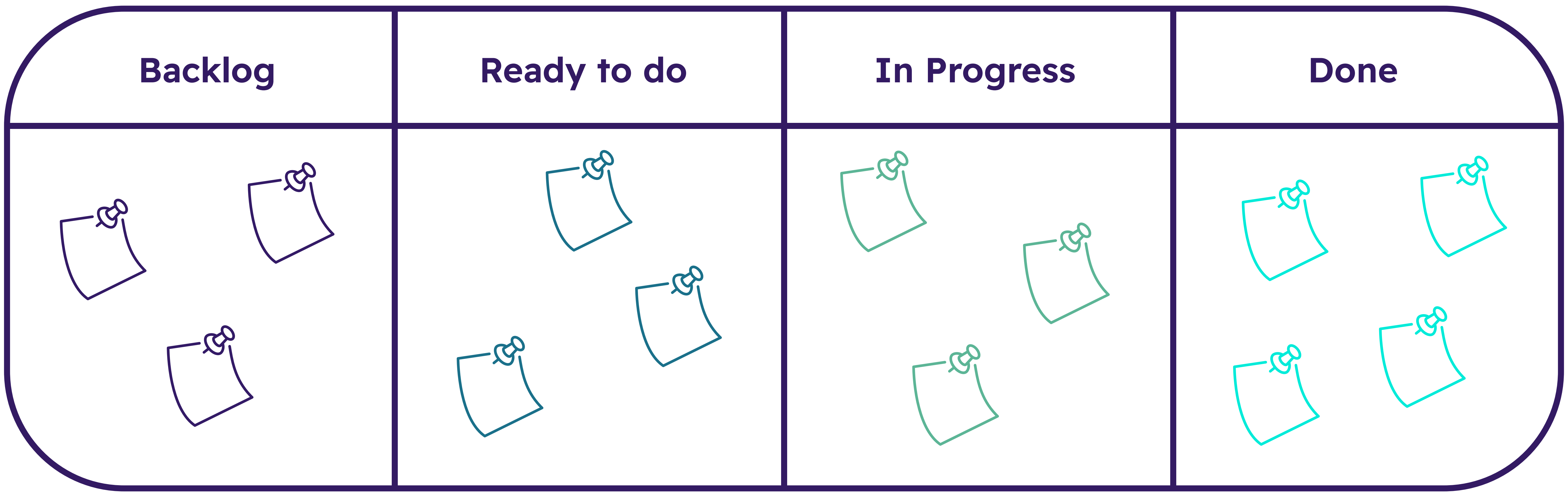
- Sprint: Sprints typically last from one to four weeks and are designed to ensure that a defined set of tasks is accomplished. In order to finish the items on the product backlog, the team works together to get them done and then produces a potentially release-ready product increment at the very end of the sprint.
- Sprint Planning: At the start of each sprint, a planning meeting is held to determine what tasks will be tackled during that sprint. A few things are chosen at random from the product backlog, and the team makes a public commitment to fulfill them within the sprint.
- Daily Scrum: Every day, the team gathers for a short stand-up meeting to discuss the previous day’s accomplishments and make plans for the next. The Scrum Master is in charge of running the meeting and aiding the team in finding and fixing any problems.
- Sprint Review: At the end of each sprint, a review meeting is convened to assess the progress made and hear opinions from those who matter most during the sprint.
- Sprint Retrospective: At the very end of each sprint, the team gathers for a retrospective meeting to examine the highlights and lowlights of the sprint and devise a strategy for moving forward.

KANBAN METHOD

Kanban is an alternative to Scrum and other Agile approaches using a pull system to manage work. The Kanban technique is a tool for improving the flow of work in an organization, and it is based on the concepts of just-in-time production.
The Kanban method consists of several components:

- Kanban Board: This is a visual tool used to track tasks that have been assigned, are currently being worked on, and have been finished. Columns for “To Do,” “In Progress,” and “Done” are standard.
- Work Items: Work items are the discrete actions or tasks that make up the whole of the work that must be done. On the Kanban board, they take the form of cards.
- Work in Progress (WIP) Limits: WIP limit is the maximum number of active tasks. Reducing the number of active projects will improve productivity.
- Pull-based System: Since the Kanban approach is pull based, work is moved into the “In Progress” column as soon as there is room. This ensures that tasks are accomplished in a timely way and that the team is not overworked.
- Continuous Improvement: The Kanban technique promotes continuous progress through regular process reviews and course corrections.

The Kanban technique is adaptable and has found use in domains as diverse as software development, manufacturing, and service provision. It's great for teams that want to streamline their processes and increase the value they provide to their clients.



LEAN METHODOLOGY

The purpose of the Lean methodology is to streamline the delivery of value to customers while simultaneously cutting down on unnecessary overhead costs. Its origins lie in manufacturing, but it has since found widespread use in fields as diverse as software development, healthcare, and the provision of consumer services.

The Lean methodology is based on the following principles:

- Define value: Find out what the consumer or end-user wants and work toward giving it to them.
- Map the value stream: Learn how value is being delivered to the client and where there may be room for enhancement.
- Create flow: If you want to maximize the value delivered to your customers, you need to get rid of any obstacles in its path.
- Establish pull: Adopt a pull-based model where tasks are added to the process only when they are required.
- Pursue perfection: Maintain a focus on continuous process improvement and waste reduction.

Lean methodology has several tools and practices that can be used to implement the principles, such as:

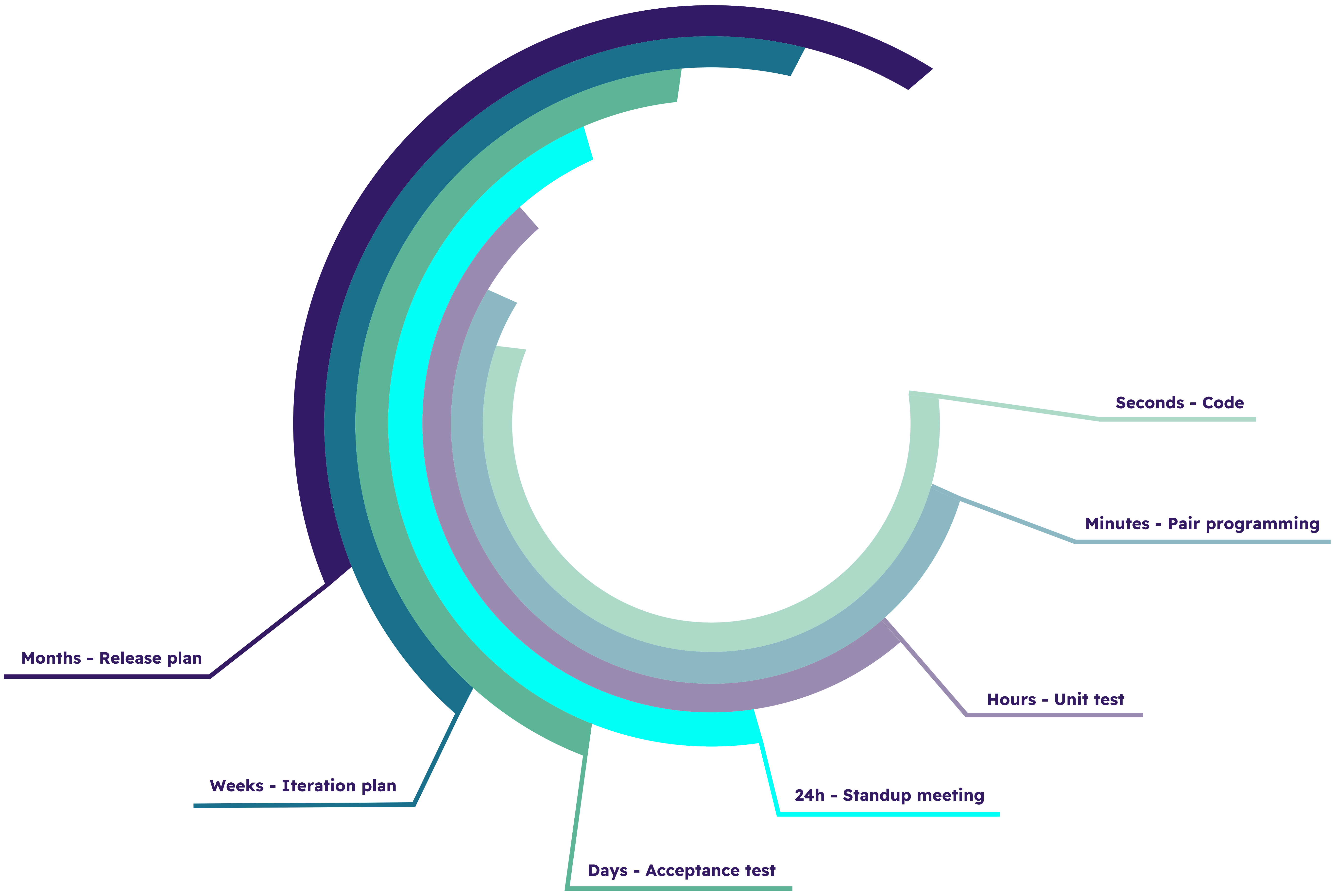
- Value Stream Mapping: A method of analyzing value streams in order to spot problems and develop solutions.
- Kanban boards: A device used to track the completion of tasks and enforce quotas.
- 5S: A method implemented to enhance workplace security, efficiency, and quality through standardization and order.
- Kaizen: Method used to make incremental improvements to an existing process.

- Standard Work: A method that creates a stable routine that can be repeated over and over again; this makes it much simpler to spot where adjustments need to be made.

The customer and the provision of value to the customer are given prominent status in Lean methodology. Overproduction, flaws, waiting, superfluous motion, over-processing, excess inventory, and unused talent are all examples of waste that lean production seeks to eradicate.

EXTREME PROGRAMMING (XP)

Customer satisfaction, timely feedback, and ongoing development are the three pillars of the Extreme Programming (XP) methodology for creating software. Due to the high level of dedication it necessitates from both the development team and the customer, XP is one of the most intense and disciplined types of Agile development.



extremeprogramming.org/

XP is based on 12 practices, divided into four categories:

- Planning: One to two-week iterations are the norm for XP teams when planning their work. User stories are defined, prioritized, and the team makes an iteration plan.
- Design: Teams using XP create the system iteratively, prioritizing ease of use and adaptability. The system's design is constantly being refined thanks to the practice of refactoring.

- Coding: The code produced by XP groups is concise and well-organized, making it simple to both comprehend and update. Automated testing is used to check the code and provide immediate feedback.
- Testing: By utilizing automated testing, XP teams can guarantee that their code is error-free and receive valuable feedback quickly. They also test at various points during development to catch bugs before they affect the end product.

CRYSTAL METHOD

This is a collection of Agile practices that can be modified to meet the requirements of each given endeavor. The goal in creating the Crystal techniques was to have an Agile development framework that was both light and adaptable.

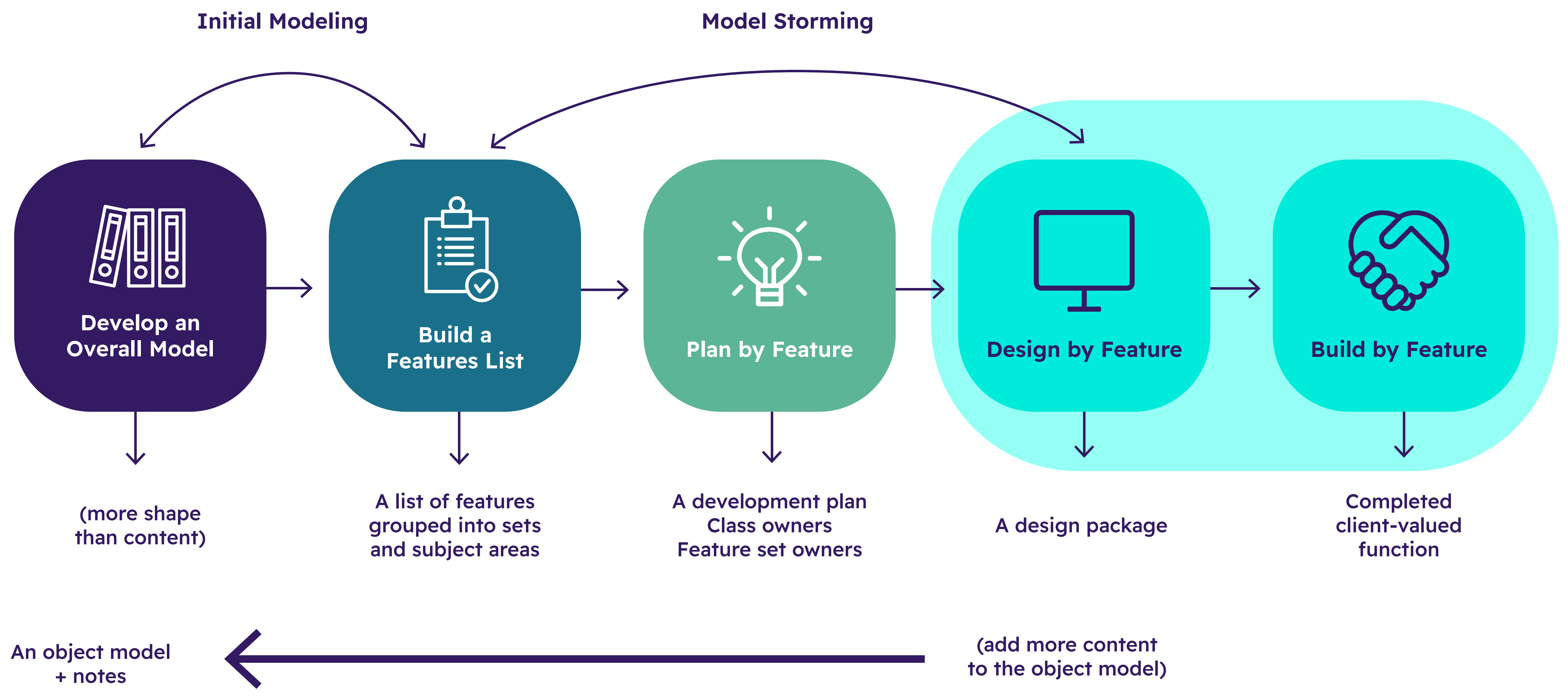
The Crystal methodologies are based on the following principles:

- Human Needs: A key focus of the Crystal methods is on the needs of the project's stakeholders, from the development team through the client to the end-user.
- Flexibility: Crystal's techniques are adaptable, so they may be modified to meet the requirements of any given task, program, or company.
- Communication: In the Crystal methods, open and honest dialogue is regarded as crucial between the team and the client.
- Technical Excellence: The Crystal approaches advocate for creating well-written code and employing effective methods and resources.
- Continuous Improvement: The Crystal techniques promote perpetual progress by emphasizing frequent retrospectives and the identification and removal of the trash.

Several distinct approaches, with varying foci and methodology, make up the Crystal family of approaches. To give one example, Crystal Clear prioritizes agile, low-risk, small-team initiatives. On the other side, Crystal Yellow is best suited for low- to medium-risk, high-performance endeavors. Crystal Orange is utilized in high-risk, high-performance endeavors.

FEATURE-DRIVEN DEVELOPMENT (FDD)

The goal of the Agile process, known as Feature-Driven Development (FDD), is to make incremental enhancements to the system in a series of short iterations. Jeff De Luca created FDD, and it's built on these five best practices.



Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
A list of features grouped into sets and subject areas
A development plan Class owners Feature set owners
A design package
Completed client-valued function
An object model + notes
(add more content to the object model)

agilemodeling.com/

- Develop an overall model: The first step in FDD is to build a system-level model that summarizes the system's architecture and describes its individual parts.
- Build a features list: To begin developing a product, FDD groups first compile a list of features, rank them in importance, and then plot out how to achieve them.
- Plan by feature: Each feature's development is planned out in great detail by FDD teams, down to the individual jobs, the resources needed, and the delivery timetable.
- Design by feature: In FDD, teams meticulously plan out the design of each feature in light of the model as a whole and the other elements that will be created.

- Build by feature: Teams using FDD create each feature over the course of a number of iterations that last only a few weeks at most.

FDD also includes several other practices, such as:

- Model reviews, feature reviews, and design reviews should be held on a regular basis to keep the team focused and on task.
- The emphasis on reusability and reworking to enhance the design of the system over time.
- The process of using automated tests to check and fix code while receiving immediate feedback.
- FDD’s rigid framework can be an asset for large, complicated projects involving numerous parties and a great deal of flux. It is especially helpful for businesses that want to implement incremental changes to the system in a measured and managed fashion.

FDD’s key benefit is that it encourages team members to hone a common understanding of the system, strengthening the team’s cohesiveness and increasing the likelihood that the end product will satisfy the customer’s requirements. FDD’s emphasis on iterative, incremental development cycles also aids in mitigating risk, raising product quality, and delighting customers.

LEAN STARTUP METHODOLOGY

Based on Lean production and Agile development tenets, the Lean Startup technique is a framework for creating successful new products and enterprises. The Lean Startup methodology was developed to speed up the process of validating business concepts and creating goods that people actually want.

The Lean Startup methodology is based on several key principles:

- Build-Measure-Learn: Building a minimum viable product (MVP), testing its efficacy, and iteratively refining it are essential tenets of the Lean Startup technique.
- Validated Learning: Following the tenets of the Lean Startup movement, business owners are urged to get feedback from actual consumers to confirm their products' viability.
- Continuous Innovation: By promoting iterative experimentation, rapid iteration, and low-volume testing, the Lean Startup technique fosters a culture of constant innovation.
- Customer Development: To create a product that sells, it's crucial to understand your target market's wants and needs, as the Lean Startup technique emphasises.
- Entrepreneurial Management: Using data-driven decision making and cultivating an entrepreneurial mindset are two key tenets of the Lean Startup approach to product development.

In the early stages of a product's development, when the idea is still unproven, and there is a great deal of uncertainty, the Lean Startup process is typically employed. It can also be employed by well-established businesses that value innovation and the rapid introduction of new items.

AGILE AND TRADITIONAL PROJECT MANAGEMENT COMPARISON

Agile and traditional project management are two alternative approaches to the same problem. The principles of agile project management center on iterative and incremental development with a strong emphasis on responsiveness to change and input from the project's end users. Scrum and Kanban are two examples of agile approaches that place emphasis on responding to change by allowing for course corrections as they occur throughout a project, ultimately benefiting the customer. An Agile team is one that is self-managing and interdisciplinary, with clear roles, responsibilities, and procedures specified by the technique itself.

Waterfall, or traditional project management, follows a strict timetable and prioritizes planning, control, and predictability. The project is broken down into stages, and its success is determined by how quickly its various subtasks are completed. Managers and workers are well-delineated in conventional project management structures. The project manager is the one who sets the parameters for the various functions and procedures.

The way in which changes are dealt with is where Agile differs most from the traditional method of project management. Because of their malleability and adaptability, agile techniques welcome and even promote change as the project progresses. However, traditional techniques tend to be rigid once the project plan is finalized, making it difficult to make adjustments.

CHALLENGES OF IMPLEMENTING AGILE

Getting everyone on the development team and throughout the company to adopt a new way of thinking is one of the biggest obstacles to successfully implementing Agile. Common obstacles to implementing Agile include:

- Resistance to change: Some team members may be reluctant to adopt an agile mentality because of the drastic shift it needs. Particularly difficult for established businesses accustomed to the more traditional project management approaches.
- Lack of understanding of Agile: When companies lack an in-depth understanding of Agile’s guiding concepts and practices, successful adoption can be challenging. This is often the most difficult part of adopting Agile for a company.
- Difficulty with estimation: Estimating time and materials accurately can be difficult for teams new to Agile methodology.
- Difficulty with prioritization: Teams used to working with a defined scope and timetable may find it difficult to shift to agile, encouraging them to prioritize work based on the value it provides to customers.
- Difficulty with collaboration: Some companies may have trouble adopting the Agile methodology because of the high level of collaboration required among team members.
- Difficulty with communication: Some businesses may have trouble with the part of Agile that necessitates regular and good communication among team members.

- Difficulty with testing and quality assurance: It might be difficult for teams not used to working in an Agile environment to adopt Agile because of the emphasis placed on automated testing and continuous integration.
- Difficulty with tracking progress: Some businesses may have trouble adopting the Agile methodology because of the new requirements for monitoring and evaluating operations.

AGILE METRICS AND MEASUREMENT

Agile metrics and measurement are used to keep tabs on development and evaluate the project's success. The development team's performance, product quality, and customer happiness may all be measured with agile metrics. Some common Agile metrics include:

- Velocity: This statistic measures how much work can be finished in an iteration. In most cases, it is rated in terms of tale points or hours.
- Burndown/Burn-up charts: The team's progress towards the iteration's goals are plotted on these charts. Burn-up charts illustrate what still needs to be done, while burndown charts display what has already been done.
- Lead time: The time to finish a user story, from its initial identification through its final deployment, is tracked using this statistic.
- Cycle time: From feature discovery to final release, this metric tracks how long it takes to implement a new addition.
- Defect density: In this metric, the number of errors per line of code is calculated.
- Code coverage: Code coverage indicates the amount of code that has been tested automatically.
- Customer satisfaction: The level of client contentment with the product is quantified by this indicator. A number of different surveys and other approaches exist for gauging this.
- Net Promoter Score (NPS): This metric measures the percentage of customers who say they would suggest your business to others.

AGILE CHANGE MANAGEMENT

The term “Agile change management” refers to the method used to control and apply alterations in an Agile project. Changes must be identified, evaluated, and implemented according to Agile principles and values.

Agile change management’s primary objective is to guide the implementation of changes in a way that benefits the project and the team as a whole. Some of the most important parts of an agile approach to change management are:

- Identify the change: Finding the need for change is the first step in an Agile approach to managing transformation. Researching customers, evaluating statistics, or going over team member comments are all possible ways to accomplish this.
- Assess the impact: Analyzing the effect that the modification will have on the project is the next phase. The costs, hazards, and dependencies of the transition must be calculated.
- Plan the change: Once the change’s effects have been evaluated, implementation strategies can be developed. Planning for a transition properly involves things like figuring out what tools will be needed and really doing that planning.
- Implement the change: Putting the modification into action is the subsequent stage. Included in this process are informing the team, giving them any necessary training or support, and keeping tabs on how the transition is going.

- Monitor and evaluate: Agile change management culminates in a period of observation and assessment. This entails keeping an eye on how the adjustment is affecting the project, gauging progress, and soliciting input from team members.

CONCLUSION AND NEXT STEP

The Agile methodology is gaining popularity because it can adapt to the various shifts in the evolution of technology. Teams now have more leeway to break down comprehensive requirements into more manageable chunks; this allows for more frequent product releases.

Ensuring your team is well-trained and understands the fundamentals of the Agile model is crucial to successfully deploying the Agile methodology. All participants in an Agile project must maintain constant lines of communication with one another. Each team member will be expected to make decisions and contribute to the ongoing evolution of the product.

Naturally, without effective project management, the Agile technique will fail. A manager of a large software project needs to be able to coordinate the efforts of several teams and integrate their contributions into a cohesive whole so that the outcome is of the highest possible quality.

REFERENCES

Ali, A. (2023, January 20). What happens in 5 Ceremonies in Scrum? | Scrum in 15 minutes | 5 Ceremonies of Scrum | Scrum in PMP. https://www.linkedin.com/posts/ameralipmp_what-happens-in-5-ceremonies-in-scrum-activity-7022118570117480448-XNjB

Arun, R. (2023, January 27). What is Agile: Understanding Agile Methodologies and Principles. Simplilearn.com. https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-agile

Atlassian. (n.d.-a). What is Agile? https://www.atlassian.com/agile
Atlassian. (n.d.-b). What is Agile? https://www.atlassian.com/agile
Bika, N. (2022, February 2). Scrum Master job description. Recruiting Resources: How to Recruit and Hire Better. https://resources.workable.com/scrum-master-job-description

Keita, B. (2023a, January 12). What is Agile Project Management? A Comprehensive Guide. Invensis Learning Blog. https://www.invensislearning.com/blog/what-is-agile-project-management/

Keita, B. (2023b, January 12). What is Agile Project Management? A Comprehensive Guide. Invensis Learning Blog. https://www.invensislearning.com/blog/what-is-agile-project-management/

Linders, B. (2013, September 5). Agile Retrospectives, Can You Skip Them? InfoQ. https://www.infoq.com/news/2013/09/skip-agile-retrospectives/

Okereke, B. S. (2023, January 25). The Value of Agile Project Management in Today's Rapidly Changing Business Environment. https://www.linkedin.com/pulse/value-agile-project-management-todays-rapidly-changing-sfc-

OpenXcell (CMMI Level 3). (2022, December 21). 15 Scrum Best Practices that will help you set up your agile workflow. https://www.linkedin.com/pulse/15-scrum-best-practices-help-you-set-up-your-agile-workflow-

Principles behind the Agile Manifesto. (n.d.). https://agilemanifesto.org/principles.html

Shastri, S. (2023a, January 18). What is a Sprint? https://www.linkedin.com/pulse/what-sprint-sudhanshu-shastri

Shastri, S. (2023b, January 18). What is a Sprint? https://www.linkedin.com/pulse/what-sprint-sudhanshu-shastri

Stellman, A. J. G. (2016, October 21). 12 Agile principles. O'Reilly Media. https://www.oreilly.com/content/12-agile-principles/
Vasiliauskas, V. (2022, June 28). What is Agile Methodology: Types, List & Hierarchy (2022). Teamhood. https://teamhood.com/agile/hierarchy-of-agile-methodologies/

Business
Explained