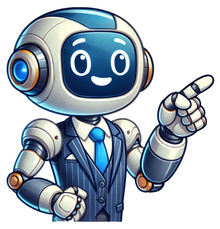
Continue



The way each team practices agile should be unique to their needs and culture. Indeed, no two teams inside Atlassian have identical agile practices. Although many of our teams organize their work in sprints, estimate in story points, and prioritize their backlogs, we're not die-hard practitioners of scrum. Or kanban. Or any other trademarked methodology. Instead, we give each team the autonomy to cherry-pick the practices that will make them most effective. And we encourage you to take a similar approach. For example, if you're on a queue-oriented team like IT, kanban provides a solid foundation for your agile practice. But nothing should stop you from sprinkling in a few scrum practices like demo sessions with stakeholders or regular retrospectives. The key to doing agile right is embracing a mindset of continuous improvement. Experiment with different practices and have open, honest discussions about them with your team. Keep the ones that work, and throw out the ones that don't. Home Free Resources Articles Posted by SCRUMstudy® on June 23, 2024 Categories: Agile Agile Frameworks Iterative Development Scrum Scrum Processes The Scrum Agile development process, as defined by SCRUMstudy, is a structured framework designed to facilitate iterative and incremental product delivery. It emphasizes adaptability and responsiveness to changing requirements, ensuring that teams can deliver high-quality products efficiently and effectively. Central to Scrum is its iterative nature, where work is organized into short, fixed-length iterations called Sprints. Each Sprint typically lasts from one to four weeks, during which cross-functional teams collaborate to deliver potentially shippable increments of the product. Daily Stand-up Meetings, Sprint Planning, Sprint Reviews, and Retrospectives are key ceremonies that promote transparency, inspection, and adaptation throughout the project lifecycle. By embracing Scrum principles, organizations can foster innovation, improve team collaboration, and swiftly respond to customer feedback, ultimately enhancing their ability to deliver value in dynamic business environments. Scrum, an agile framework for managing complex projects, emphasizes collaboration, adaptability, and iterative progress towards a well-defined goal. Scrumstudy, a leading authority in Scrum and Agile certifications, offers a comprehensive suite of resources and training programs to equip professionals with the knowledge and skills needed to implement Scrum effectively. By following structured Scrum processes, teams can enhance productivity, foster innovation, and deliver high-quality outcomes efficiently. Scrumstudy's certifications, grounded in the extensive SBOK® Guide, ensure that practitioners are well-prepared to navigate the dynamic landscape of modern project management. Scrum processes address the specific activities and flow of a Scrum project. In total there are 19 processes which are grouped into five phases. Please take a look at these processes and the activities listed under to them in order to understand the flow of a Scrum Project better. Initiate Create Project Vision—In this process, the Product Owner is identified. Based on the project's business case, the Product Owner then creates a Project Vision Statement. This Project Vision Statement provides the overall guidance, inspiration, and focus for the project. Identify Scrum Master and Business Stakeholder(s)—in this process, the Scrum Master is identified using specific Selection Criteria. Form Scrum Team—in this process, Scrum Team members are identified. Normally the Product Owner has the primary responsibility of selecting team members, but often does so in collaboration with the Scrum Master. Develop Epic(s)—in this process, the Project Vision Statement serves as the basis for developing Epic(s). User Group Meetings may be held to develop Epic(s). Create Prioritized Product Backlog —in this process, Epic(s) are refined, elaborated, and then prioritized to create a Prioritized Product Backlog for the project. The Done Criteria is also established at this point. Conduct Release Planning—in this process, the Scrum Core Team reviews the User Stories in the Prioritized Product Backlog to develop a Release Planning Schedule, which is essentially a phased deployment schedule that can be shared with the project stakeholders. Length of Sprint is also determined in this process. Plan and Estimate Create User Stories—In this process User Stories and their related User Story Acceptance Criteria are created. User Stories are usually written by the Product Owner and are designed to ensure that the customer's requirements are clearly depicted and can be fully understood by all business stakeholders. User Story Writing Exercises may be held which involves Scrum Team members creating the User Stories. User Stories are incorporated into the Prioritized Product Backlog. Approve, Estimate, and Commit User Stories—In this process the Product Owner approves User Stories for a Sprint. Then, the Scrum Master and Scrum Team estimate the effort required to develop the functionality described in each User Story, and the Scrum Team commits to deliver the customer requirements in the form of Approved, Estimated, and Committed User Stories. Create Tasks—In this process the Approved, Estimated, and Committed User Stories are broken down into specific tasks and compiled into a Task List. Often a Task Planning Meeting is held for this purpose. Estimate Tasks—In this process the Scrum Core Team, in Task Estimation Meetings, estimate the effort required to accomplish each task in the Task List. The result of this process is an Effort Estimated Task List. Create Sprint Backlog—In this process the Scrum Core Team holds Sprint Planning Meetings where the group creates a Sprint Backlog containing all tasks to be completed in the Sprint. Implement Create Deliverables—In this process, the Scrum Team works on the tasks in the Sprint Backlog to create Sprint Deliverables. A Scrumboard is often used to track the work and activities being carried out. Issues or problems being faced by the Scrum Team could be updated in an Impediment Log. Conduct Daily Standup—In this process, everyday a highly focused, Time-boxed meeting is conducted referred to as the Daily Standup meeting. This is the forum for the Scrum Team to update each other on their progress and any impediments they may be facing. Refine Prioritized Product Backlog—In this process, the Prioritized Product Backlog is continuously updated and maintained. A Prioritized Product Backlog Review Meeting may be held, in which any changes or updates to the backlog are discussed and incorporated into the Prioritized Product Backlog as appropriate. Review and Retrospect Convene Scrum of Scrums—In this process Scrum Team representatives convene for Scrum of Scrums Meetings in predetermined intervals or whenever required to collaborate and track their respective progress, impediments, and dependencies across teams. This is relevant only for large projects where multiple Scrum Teams are involved. Demonstrate and Validate Sprint—In this process, the Scrum Team demonstrates the Sprint Deliverables to the Product Owner and relevant business stakeholders in a Sprint Review Meeting. The purpose of this meeting is to secure approval and acceptance from the Product Owner for the Deliverables created in the Sprint. Retrospect Sprint—In this process, the Scrum Master and Scrum Team meet to discuss the lessons learned throughout the Sprint. This information is documented as lessons learned which can be applied to future Sprints. Often, as a result of this discussion, there may be Agreed Actionable Improvements or Updated Scrum Guidance Body Recommendations. Release Ship Deliverables—In this process, Accepted Deliverables are delivered or transitioned to the relevant business stakeholders. A formal Working Deliverables Agreement documents the successful completion of the Sprint. Retrospect Project—In this process, which completes the project, organizational business stakeholders and Scrum Core Team members assemble to retrospect the project and identify, document, and internalize the lessons learned. Often, these lessons lead to the documentation of Agreed Actionable Improvements, to be implemented in future projects. Posted by SCRUMstudy® on January 16, 2024 Categories: Agile Agile Frameworks Iterative Development Scrum Scrum Processes The Scrum Agile development process, as defined by SCRUMstudy, is a structured framework designed to facilitate iterative and incremental product delivery. It emphasizes adaptability and responsiveness to changing requirements, ensuring that teams can deliver high-quality products efficiently and effectively. Central to Scrum is its iterative nature, where work is organized into short, fixed-length iterations called Sprints. Each Sprint typically lasts from one to four weeks, during which cross-functional teams collaborate to deliver potentially shippable increments of the product. Daily Stand-up Meetings, Sprint Planning, Sprint Reviews, and Retrospectives are key ceremonies that promote transparency, inspection, and adaptation throughout the project lifecycle. By embracing Scrum principles, organizations can foster innovation, improve team collaboration, and swiftly respond to customer feedback, ultimately enhancing their ability to deliver value in dynamic business environments. Scrum, an agile framework for managing complex projects, emphasizes collaboration, adaptability, and iterative progress towards a well-defined goal. Scrumstudy, a leading authority in Scrum and Agile certifications, offers a comprehensive suite of resources and training programs to equip professionals with the knowledge and skills needed to implement Scrum effectively. By following structured Scrum processes, teams can enhance productivity, foster innovation, and deliver high-quality outcomes efficiently. Scrumstudy's certifications, grounded in the extensive SBOK® Guide, ensure that practitioners are well-prepared to navigate the dynamic landscape of modern project management. Scrum processes address the specific activities and flow of a Scrum project. In total there are 19 processes which are grouped into five phases. Please take a look at these processes and the activities listed under to them in order to understand the flow of a Scrum Project better. Initiate Create Project Vision—In this process, the Product Owner is identified. Based on the project's business case, the Product Owner then creates a Project Vision Statement. This Project Vision Statement provides the overall guidance, inspiration, and focus for the project. Identify Scrum Master and Business Stakeholder(s)—in this process, the Scrum Master is identified using specific Selection Criteria. Form Scrum Team—in this process, Scrum Team members are identified. Normally the Product Owner has the primary responsibility of selecting team members, but often does so in collaboration with the Scrum Master. Develop Epic(s)—in this process, the Project Vision Statement serves as the basis for developing Epic(s). User Group Meetings may be held to develop Epic(s). Create Prioritized Product Backlog —in this process, Epic(s) are refined, elaborated, and then prioritized to create a Prioritized Product Backlog for the project. The Done Criteria is also established at this point. Conduct Release Planning—in this process, the Scrum Core Team reviews the User Stories in the Prioritized Product Backlog to develop a Release Planning Schedule, which is essentially a phased deployment schedule that can be shared with the project stakeholders. Length of Sprint is also determined in this process. Plan and Estimate Create User Stories—In this process User Stories and their related User Story Acceptance Criteria are created. User Stories are usually written by the Product Owner and are designed to ensure that the customer's requirements are clearly depicted and can be fully understood by all business stakeholders. User Story Writing Exercises may be held which involves Scrum Team members creating the User Stories. User Stories are incorporated into the Prioritized Product Backlog. Approve, Estimate, and Commit User Stories—In this process the Product Owner approves User Stories for a Sprint. Then, the Scrum Master and Scrum Team estimate the effort required to develop the functionality described in each User Story, and the Scrum Team commits to deliver the customer requirements in the form of Approved, Estimated, and Committed User Stories. Create Tasks—In this process the Approved, Estimated, and Committed User Stories are broken down into specific tasks and compiled into a Task List. Often a Task Planning Meeting is held for this purpose. Estimate Tasks—In this process the Scrum Core Team, in Task Estimation Meetings, estimate the effort required to accomplish each task in the Task List. The result of this process is an Effort Estimated Task List. Create Sprint Backlog—In this process the Scrum Core Team holds Sprint Planning Meetings where the group creates a Sprint Backlog containing all tasks to be completed in the Sprint. Implement Create Deliverables—In this process, the Scrum Team works on the tasks in the Sprint Backlog to create Sprint Deliverables. A Scrumboard is often used to track the work and activities being carried out. Issues or problems being faced by the Scrum Team could be updated in an Impediment Log. Conduct Daily Standup—In this process, everyday a highly focused, Time-boxed meeting is conducted referred to as the Daily Standup meeting. This is the forum for the Scrum Team to update each other on their progress and any impediments they may be facing. Refine Prioritized Product Backlog—In this process, the Prioritized Product Backlog is continuously updated and maintained. A Prioritized Product Backlog Review Meeting may be held, in which any changes or updates to the backlog are discussed and incorporated into the Prioritized Product Backlog as appropriate. Review and Retrospect Convene Scrum of Scrums—In this process Scrum Team representatives convene for Scrum of Scrums Meetings in predetermined intervals or whenever required to collaborate and track their respective progress, impediments, and dependencies across teams. This is relevant only for large projects where multiple Scrum Teams are involved. Demonstrate and Validate Sprint—In this process, the Scrum Team demonstrates the Sprint Deliverables to the Product Owner and relevant business stakeholders in a Sprint Review Meeting. The purpose of this meeting is to secure approval and acceptance from the Product Owner for the Deliverables created in the Sprint. Retrospect Sprint—In this process, the Scrum Master and Scrum Team meet to discuss the lessons learned throughout the Sprint. This information is documented as lessons learned which can be applied to future Sprints. Often, as a result of this discussion, there may be Agreed Actionable Improvements or Updated Scrum Guidance Body Recommendations. Release Ship Deliverables—In this process, Accepted Deliverables are delivered or transitioned to the relevant business stakeholders. A formal Working Deliverables Agreement documents the successful completion of the Sprint. Retrospect Project—In this process, which completes the project, organizational business stakeholders and Scrum Core Team members assemble to retrospect the project and identify, document, and internalize the lessons learned. Often, these lessons lead to the documentation of Agreed Actionable Improvements, to be implemented in future projects. Scrum, short for "scrummage," is a term that comes from rugby that means to restart play after there's been a minor rule infringement. When a scrum occurs, players are packed closely together to gain possession of the ball and work together to push it forward. While this might seem completely unrelated to software, it was used to detail an analogy between rugby teams and product development teams in a 1986 Harvard Business Review paper by Hirotaka Takeuchi and Ikujiro Nonaka. They used the term to describe all members of a team working closely together to incrementally move a project forward. Each increment or iteration consists of two- to four-week sprints, where the goal of each sprint is to build the most important features first and come out with a potentially shippable product. This was then built on by Jeff Sutherland, who further applied the concept and framework to software processes. What is a Sprint? As mentioned above, a Scrum sprint is a time-boxed iteration where a specific and prioritized set of work is completed. They are relatively short, usually lasting two to four weeks, to deliver a potentially shippable product increment at the end of each sprint. The key phases or components of a sprint include: Planning - at the beginning of a sprint, the Scrum team (including the Product Owner, Scrum Master, and Development Team) collaborates to define the scope of work to be accomplished during this iteration. Daily standups - each day, the team holds a brief standup meeting to discuss progress, challenges, and plans for the next 24 hours. This fosters communication and ensures everyone is on the same page. Review - at the end of the sprint, a review meeting takes place to showcase the completed work to stakeholders and gather feedback. Retrospective - each sprint also concludes with a retrospective where the team reflects on what went well, what could be improved, and how to enhance their processes for the next iteration. What is an Epic? An epic is a large body of work that can be broken down into smaller, more manageable tasks or user stories. Epics are essentially placeholders for substantial features or projects that are too big to be completed in a single sprint. They provide a high-level view of the work that needs to be done and help with long-term planning and prioritization. Once an epic is defined, it is broken down into smaller, more detailed user stories that can be addressed within a single sprint. This allows for better estimation, planning, and execution of the work. What is Velocity? Scrum velocity is a metric that measures the amount of work a Scrum team can complete in a single sprint. It provides insights into the team's capacity and helps with future sprint planning by providing a baseline for predicting how much work the team can take on. Velocity = Sum of Story Points Completed ÷ Number of Completed Sprints It's important to note that velocity is not a measure of productivity or efficiency. Instead, it serves as a guide for the team to forecast their capacity and make informed decisions during planning. What is a Scrum Board? A Scrum board is a visual management tool employed by Scrum teams to monitor and organize work during a sprint. Typically displayed in a shared workspace, it consists of columns representing different stages of work completion. The primary columns include "To-Do," listing tasks or user stories planned for the sprint; "In Progress," showcasing items actively being worked on, and "Done," indicating completed tasks ready for review. The boards offer a clear, real-time snapshot of the team's progress. While physical boards with sticky notes are common, digital tools are also widely used, especially in virtual or distributed team settings. These are an integral part of the Scrum framework, aiding teams in managing their workload efficiently and ensuring that everyone is aligned on the current status of tasks. Agile vs. Scrum Agile and Scrum are terms often used in the context of software development methodologies, but they represent different concepts. Agile is a broader philosophy or set of principles outlined in the Agile Manifesto, emphasizing values such as flexibility, customer collaboration, and iterative development. It's a mindset that prioritizes responding to change and delivering value continuously. On the other hand, Scrum is a specific framework within the Agile philosophy. It defines roles, events, and artifacts to guide the development process. Scrum offers a structured approach with defined roles, ceremonies (learn more about Scrum ceremonies here), and artifacts (learn more about Scrum artifacts here) to implement Agile principles effectively. Scrum in Software Development Scrum is a lightweight software development methodology that focuses on having small time-boxed sprints of new functionality that are incorporated into an integrated product baseline. Scrum places an emphasis on transparent customer interaction, feedback and adjustments rather than documentation and prediction. Instead of phases, Scrum projects are broken down into releases and sprints. At the end of each sprint you have a fully functioning system that could be released: With Scrum, the requirements for the project do not have to be codified up-front. Instead, they are prioritized and scheduled for each sprint. The requirements are composed of user stories that can be scheduled into a particular release and sprint: Scrum is often deployed in conjunction with other agile methods such as Extreme Programming (XP), because these methods are mostly complimentary. In this example, XP focuses on the engineering (continuous exploration, continuous integration, test-driven development, etc.), and Scrum focuses on the project management side (burn-down, fixed scope for sprints/iterations) as part of the product management. Because of this, project managers should choose elements of the Scrum project management methodology and other methods/tools together for the specific project. Scrum is a more defined project management methodology in terms of tools and processes, so it is often easier to adopt from day one with less initial invention and customization. Benefits of Agile Scrum Methodology Traditionally projects were delivered in a series of phases that are based on increasing levels of certainty around the system being built: However, this somewhat outdated approach has several drawbacks: It is not flexible to changes in customer requirements Time is wasted building features that nobody needs or will use The end user cannot give feedback until it's completely coded You don't know how stable the system is until the end When using the agile Scrum methodology, the key benefits are: Flexibility & Adaptability Flexibility & Adaptability Because the requirements (consisting of features, user stories and other backlog items) are deliberately kept high-level at the start, you have the flexibility to adapt the requirements from feedback from the customer and end users 'on the fly'. This comes from the ability to redefine and replan after each release and sprint, giving the team plenty of room to respond to changes. Creativity & Innovation Scrum allows integrated teams of developers, designers, testers, and business/functional experts to all work together rather than working in individual 'silos'. This cross-discipline collaboration fosters creativity and innovation, as well as more out-of-the-box solutions to problems that pop up. Lower Risks & Costs A study found that the majority of features delivered in traditional waterfall projects were never used or needed by customers, due to scope creep and feature bloat. With Scrum, you are ruthlessly pruning the release backlog after each sprint so these unnecessary features (and their associated costs) are quickly weeded out. The short-time boxes reduce the likelihood of risks materializing during the sprint and thus address increment delivery. Quality Improvement With engineering approaches such as continuous exploration, continuous integration, continuous delivery, test-driven development, DevOps, and relentless test automation, Scrum yields much higher-quality code, products and process. All of these benefit the final product that you deliver to stakeholders, which is one of the major reasons why Scrum is the most widely-used methodology when it comes to software development. Organizational Synergy Scrum lets team members from a variety of disciplines harness their different skills in a cross-cutting product development ethos. This ensures that the finished product is optimized by and uses the best skills in the organization, not just what's written on someone's job title. Employee Satisfaction With the flexibility to adjust scope and still deliver results, the problem of delivering a fixed scope, fixed-schedule project with a team goes away. Instead of long nights frantically replanning and retesting functionality, the team has a sustainable operating pace that makes employees proud of their work and not fighting to deliver against impossible deadlines. Customer Satisfaction One of the challenges of software development for customers is that they cannot often appreciate the functionality of a system until they see it. By delivering working functionality in small increments, customers can see what they are getting earlier in the process and be confident that it is what they actually need or expect. Who Can Benefit from Scrum? Scrum is useful for a wide variety (the majority, in fact) of software development businesses and projects where you need the ability to get real-time feedback from users and customers. Customers in diverse industries such as finance, banking, insurance, life sciences, transportation, logistics, manufacturing, education, and industrial systems adopt Scrum as their primary software development lifecycle. Even the industries most associated with waterfall approaches (aerospace, defense, and automotive) are now adopting Scrum. The ability to update software in satellites and cars in real-time means that iterative approaches are now more feasible. However, Scrum approaches aren't without challenges. For instance, with Scrum you can develop and release new production-ready code in two-weekly sprints — but you still need to produce reports, documentation, and paperwork to allow that functionality to be used. This has previously hampered Scrum adoption in some industries. To account for and improve upon this, the new trend in agile Scrum projects is to simply define the required regulatory documentation and paperwork as part of the product. Developers then ensure that the automated lifecycle management systems in place can generate the documentation automatically from the project teams as part of each sprint. We call this "Agile Compliance." Roles in Agile Scrum There can be many different roles in a Scrum project, but there are three core roles (see this guide for more information on Scrum team size and how these roles come together). These three primary roles are product owner, scrum master, and scrum team: Product Owner This is the person responsible for managing the backlog of work and overseeing releases and sprints. They act as the project manager for the Scrum team, although the agile community generally doesn't use that term. The product owner is responsible for representing the voice of the customer (and other external stakeholders), ensuring that all development work reflects the needs of the end user. According to the official Scrum Guide, the primary responsibility of the product owner is to manage the product backlog. This means ordering and prioritizing work, optimizing the value of the work, and ensuring all team members understand what work needs to be tackled next. Scrum Master This role is responsible for helping the team to follow the Scrum process, and focuses almost exclusively on the process itself. They are not involved in the decision-making, but act as a lodestar to guide the team through the project with their experience and expertise. In contrast with traditional project managers, they are not necessarily involved in agile release planning. That's done by the product owner and the Scrum team. A Scrum Master doesn't act as a pseudo-project manager, because a Scrum team is meant to be self-organizing. The Scrum Master is the glue that holds the project together by facilitating, though not participating, in the daily Scrum meeting (one of the Scrum ceremonies we discuss in more detail here). Scrum Masters also help the team maintain their burndown chart and set up retrospectives, sprint reviews, and agile sprint planning sessions. Scrum (Development) Team According to the Scrum Alliance, a typical Scrum team (called a development team in the Scrum 2020 revision) consists of five to nine people and generally includes the typical functional roles required to complete the project. In software development, that usually means architects, testers, developers, and designers — but could also include security, performance, hardware, and other experts. The team acts collectively to determine how to achieve their goals, with the specific features they work on determined by the priority established by the product owner. The way they work is guided by the Scrum process, as monitored by the Scrum Master. Everything else is up to the team to manage, with the Scrum Master providing as much support as needed to allow that to happen. For example, each team member can take a user story from the prioritized product backlog, then decide individually how to execute that work. This level of autonomy is a cornerstone of Scrum. It encourages strong bonds between team members and helps create a positive working environment. While the idea of a team exists in waterfall projects as well, in that environment the team is functionally managed by the project manager, rather than being self-managed. Other Roles All of these people are committed to the scrum project, and are augmented by various other specialist roles such as testers, test automation engineers, cybersecurity experts, user experience designers, business analysts, functional experts, and performance/load engineers. In addition, there are some ancillary roles that are not core to the Scrum project team, but are heavily involved in the project outcome. These include customers, executive team members, and other stakeholders. All of these roles participate throughout the process, helping organize and guide artifacts, which we discuss more here. Why Use SpiraTeam for Scrum Projects? When your company requires a system to manage your Scrum or XP projects, there are a lot of choices in the marketplace. However, if you want the most complete solution that can handle your user stories, backlog items, and acceptance tests there is only one solution: SpiraTeam. Backlog Planning The list of user stories, requirements, defects, and features is called the Product Backlog. SpiraTeam lets you organize the backlog by component, package, or priority. You can order and categorize them by intuitive drag-and-drop editing, called backlog grooming or refinement. Sprint Planning The requirements, user stories, tasks, test cases, and outstanding bugs or issues can be scheduled against the different iterations / sprints in the release. The system will calculate the available effort as items are added. SpiraTeam is a complete Scrum project management system in one package that manages your project's requirements, releases, sprints, tasks, and bugs. Designed specifically to support agile methodologies such as Scrum, it allows teams to manage all their information in one environment. SpiraTeam Scrum Extreme Programming Epic Requirement Epic Epic Requirement Backlog Item User Story Task Task Task Release Release Release Sprint Sprint Iteration Test Case Acceptance Test Acceptance Test Incident Defect Defect Support for Scrum Tasks In addition to a powerful planning board for managing user stories, epics, and backlog items, SpiraTeam includes an easy-to-use set of boards for managing tasks within each sprint. SpiraTeam lets you view the tasks associated with each of the user stories within the sprint and quickly see the completion status at a glance. Reporting Dashboards SpiraTeam also provides reporting dashboards of key project quality and progress indicators —requirements test coverage, task progress, project velocity, as well as top risk and issues. This is all kept in one consolidated view that is tailor-made for Scrum and Extreme Programming projects, as well as supporting any legacy/hybrid waterfall projects. Not convinced? The top reasons that our customers say they choose SpiraTeam over other Scrum project management solutions are: Highly intuitive web application that provides a complete picture of a project's status and health, yet only requires a web browser. Available as both a SaaS/hosted service and an on-premise deployment, SpiraTeam is both affordable and easy to deploy for your Scrum projects. In addition, we provide superb technical support that ensures that any questions or concerns are dealt with in a timely and professional manner. How do I Get Started? To learn more about SpiraTeam and how it can improve your Scrum project management methodology, please: Take a tour of the available features Read some of the testimonials from satisfied customers Sign up for a free 30-day trial version to try it out for yourself Why Use Rapise For Scrum Projects? One of the key tenets of Scrum is that fully integrated, tested, and releasable code is available at all times. When you couple this requirement with the accelerated timeframes possible with Scrum, manual testing is not going to cut it. You need a test automation solution that can be integrated fully into your development process and can be adapted to your changing needs: Rapise is the most powerful and flexible automated testing tool on the market. With support for testing desktop, web, and mobile applications, Rapise can be used by testers, developers, and business users to rapidly and easily create automated acceptance tests that integrate with your user stories and other backlog items in SpiraTeam. Self-Healing Tests One of the obstacles to implementing test automation on an agile Scrum or XP project is that the application's user interface may be changing frequently, with new pushes to production daily. Therefore, it is critical that tests created one day, continue to work seamlessly the next. Rapise comes with a built-in machine learning engine that records multiple different ways to locate each UI object, and measures that against user feedback to determine the probabilistic likelihood of a match. Put simply, this means that even when you change the UI, Rapise can still execute the test and determine if there is a failure or not. How do I Get Started? To learn more about Rapise and how it can improve your Scrum and XP software testing please: Take a tour of the available features Read some of the testimonials from satisfied customers Sign up for a free 30-day trial version to try it out for yourself Agile is a mindset and philosophy that describes a set of principles in the Agile Manifesto, On the other hand, Scrum is a framework that prescribes roles, events, artifacts, and rules/guidelines to implement that mindset. In other words, Agile is the mindset and Scrum is the framework that prescribe a process for implementing the agile philosophy. Agile Umbrella Agile is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto that is a way of thinking that enables teams and businesses to innovate, quickly respond to changing demand, while mitigating risk. Organizations can be agile using many of the available frameworks available such as Scrum, Kanban, Lean, Extreme Programming (XP) and etc. The Scrum Umbrella What is Agile? The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability which refer to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The primary goal of being Agile is empowered the development team the ability to create and respond to change in order to succeed in an uncertain and turbulent environment. What is Scrum? Scrum and agile are not the same but scrum is one of the agile processes. They are based in iterative development. The requirements and solutions of agile obtained by the association between the cross-functional and self-organization teams, and when implemented properly can help teams address complex problems by incrementally delivering products of the highest value while mitigating risk. Scrum involves prompt inspection and adaptation, teamwork is enhanced by leadership philosophy, accountability and self-organization, best practices of engineering which helps in the delivery of prompt high-quality software. How Does Scrum Work? A Scrum process is distinguished from other agile processes by specific concepts and practices, divided into the three categories of Roles (product owner, scrum master, development team and other stakeholder), events, artifacts, and rules. To start a Scrum process, a product owner creates a prioritized wish list called a product backlog. During sprint planning the backlog is sized for complexity and business value (priority). The product owner (client) and the development team determine which backlog items get added to the sprint. The team has a certain amount of time (called a sprint, usually two to four weeks) to complete its work, but it meets each day to assess its progress (daily Scrum). Along the way, the Scrum Master keeps the team focused on its goal. At the end of the sprint the team reviews their progress, shows the client the working product, and reviews what went well or what they need to improve on for the next sprint. Then the cycle repeats. The Agile Scrum Framework Note that: Scrum encourages us to hold five key events during a Sprint, they intend to help the team work efficiently and closely together, as well as to improve our knowledge and become more effective in the future. These five events are: Sprint Planning Daily Scrum Sprint Review Sprint Retrospective The Sprint About Visual Paradigm Visual Paradigm help organizations stay competitive and responsive to change faster and better in today's fast changing environment. Our award-winning products are trusted by over 320,000 users in companies ranging from small business, consultants, to blue chip organizations, universities and government units across the globe. It enables organizations to improve business and IT agility and foster innovation through popular open standards and process frameworks.Visual Paradigm, a killer Agile feature in 2018, introduced Scrum Process Canvas for automating the way a Scrum team to create, manage and deploy software application that empowers the team to continuously improve their performance at unprecedented speed and scale. Manage the Entire Scrum Process in One Page Automate the Scrum Framework in a fun and enjoyable dashboard with eye-catching updated status. Manage Backlog, Multiple Sprints of different Scrum Roles with a single-page visually executable canvas Allow instantly access, review and generate scrum artifacts and related documents to be archived in the Shared Cabinet Automate the Scrum events and related activities with self-explanatory instructions, samples and required document templates. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Toyota Connected: How the Ideas of the Agile Product Operating Model Shape Portfolio Management Strategy In this episode of the Scrum.org Community Podcast, Dave West and PST Lucas Smith, Director of Agile Program & Services, Toyota Connected, explore how Toyota Connected structures its funding models, defines product life cycles, and manages dependencies in complex product ecosystems. Parmenion Reboots Their Agile Initiatives with an Increase in Backlog Quality of More Than 30% While Parmenion already was using Agile, they realized they needed to work in a way that would maximize value delivery and decided to work with PST Simon Reindl to help them achieve a growth mindset, embrace the Scrum Values and become more outcome oriented. Strobbo Adopts Professional Scrum to Accelerate Go-to-Market Strobbo, an HR platform transitioned to Professional Scrum, focusing on self-management, cross-functionality, and goal-oriented planning. Scrum events were overhauled, and new techniques were introduced to improve engagement and transparency. Scrum at Gillette - Launching a New Razor with Scrum David Ingram, VP of Global R&D at Gillette chats with Dave West, CEO at Scrum.org about how they developed a new exfoliating razor using Scrum, why they chose to use Scrum to develop it and how they adapted to Scrum. They discuss how Gillette mitigated risks and made some culture shifts in the process, while making focus a priority. Pentacomp's Use of the Nexus Framework This case study talks about its work with the Ministry of Health in Poland and how they used the Nexus framework to effectively deliver its systems that have been critical to providing services to patients during and post Covid-19 pandemic. NEC's Agile Journey with Scrum and Kanban Steve Lizotte, VP of Engineering at NEC explores the importance of collaboration, continuous improvement, and maintaining a customer-centric approach while sharing insights on separating program teams from product teams, finding the right balance between custom solutions and standardized offerings, and more. Intralinks Scrum Reboot This Time With The Values In their first attempt at attaining agility, Intralinks took a well-intentioned "mechanical" implementation of Scrum, but failed to deliver against their goal of greater agility. So, they took on a "Scrum Reboot" and succeeded by augmenting the mechanics of Scrum with the fundamental idea of inspection and adaptation and the Scrum Values. Yes, we really are the Home of Scrum! Scrum co-creator Ken Schwaber founded Scrum.org in 2009 to be a different kind of organization with people dedicated to a mission of helping people and teams solve complex problems. This mission drives everything we do. We've seen individuals and organizations use the power of Scrum to transform the way they work and deliver value. We want to help others enjoy the same success. As a mission-based organization focused on helping people and teams, we provide many different ways to learn about Scrum and a community to help people learn Go beyond the mechanical aspects of Scrum and transform your mindset and focus while embracing the Scrum Values Continuously learn both in and out of the classroom with different types of learning that range from activity-based training to self-learning at your own pace Professional Scrum Trainers Learn from over 350 highly experienced Scrum practitioners who all bring their real-life experiences to the classroom Get hands-on, interactive, globally consistent training in all classes and opportunities to further grow your knowledge Globally Recognized Certifications Prove your knowledge and get certified with industry respected certification tests that do not require renewal From 27,375 Trustpilot reviews Attained a Professional Scrum certification Taken a Professional Scrum course When it comes to training, are you looking for more than standard corporate training with a trainer simply presenting to the class? Scrum.org training courses embody the principles of Professional Scrum while providing an interactive learning experience. Learners continuously tell us how much they love our courses! Professional Scrum Trainers build our courses together, while bringing their own experiences to the classes they teach. They all use the same teaching materials ensuring consistent learning among your team members. Scrum.org provides a variety of courses including the Scrum fundamentals, role-based learning, and courses that go beyond the framework with training about complementary practices. Enables all members of the Scrum Team to learn Scrum while doing it, experiencing what it is like to deliver products using the Scrum framework An interactive, activity-based course where students gain a strong understanding of Professional Scrum and the role of the Scrum Master. Students gain an understanding of the critical role that the Product Owner plays on the Scrum Team including a number of Product Ownership practices that they can use. Students learn techniques and methods to effectively manage a Product Backlog that clearly expresses the future vision of the product and brings transparency to what is most valuable for the Scrum Team to work on to delight stakeholders and customers. This one-day training course offers a hands-on learning experience to participants, enabling them to enhance value creation by applying discovery and validation skills in product development. Develop proficiency in facilitation skills, so that you can help your Scrum Team better solve problems, create a shared understanding and foster transparency. Learn how to become a better facilitator to improve Scrum Team, stakeholder and customer interactions. Participants learn about the EBM framework through a series of exercises to help leaders guide their teams toward continuously improving customer outcomes, organizational capabilities, and business results. Teaches Scrum practitioners how to apply Kanban practices to their work without changing Scrum, bringing greater transparency and flow. Learn modern UX techniques and practices that effectively enable Scrum Teams to best work with customers and their feedback to deliver higher value. You may think that you can just go to the class and automatically get certified. You may also think that you must take a class to get certified. Both of these assumptions are actually incorrect. There is value in validating the knowledge you gain in the class, and we also understand that some people prefer to self-learn and leverage their experience. Our certifications prove what you know, while our classes teach you. It is recommended to take a class, but not required. Professional Scrum certifications are designed for a variety of experience levels. They are globally recognized and respected because they test a true understanding of Scrum and how to apply it. There are no renewals required on our certifications. Once you earn it, you keep it! There are three levels of Scrum Master certifications to validate and certify your knowledge and understanding of Scrum and the Scrum Master role. There are three levels of Product Owner certifications to validate and certify your knowledge and understanding of the Product Owner role. The Professional Scrum Developer certification validates and certifies your knowledge and understanding of the Developers accountabilities in Scrum. PSF Skills certification is evidence that you have demonstrated an understanding on how to harness facilitation principles, skills and techniques, and how effective facilitation can help teams move forward toward their desired outcomes. The Professional Agile Leadership I certification validates and certifies an understanding about how leaders can best support their teams in an agile environment. The Professional Agile Leadership - Evidence Based Management certification validates and certifies an understanding about how leaders can best support their teams in an agile environment. The Scaled Professional Scrum certification validates and certifies an understanding of scaling fundamentals to enable multiple Scrum Teams working together. The Professional Scrum with Kanban certification validates and certifies an understanding of how to use Scrum with Kanban to improve value creation and delivery. If you are looking to learn beyond the classroom, we have a variety of resources you can explore and use. Continue expanding your knowledge with blogs, videos, webcasts, case studies, whitepapers, and more. View all resources As human beings, we often learn a lot from the experiences of others. There are a variety of ways to learn from other Professional Scrum practitioners including a forum, events, meetups, blogs, podcasts, and more. Join our community! Hiren Doshi The Three-Horizon Approach to Sprint Planning Most teams plan a Sprint with one question in mind: "What can we deliver this Sprint?" But here's a different take. What if, within every Sprint, we planned across all three horizons of innovation? Want to learn more? See all blogs Heard enough from us? Learn from our students! We collect reviews through a third party, Trustpilot, and also survey students 6 months after they take a class to check in on the value they gained and learnings they applied. Quick Summary: Scrum is an Agile framework that breaks complex projects into short, iterative cycles (sprints), enabling teams to deliver value faster and adapt to change. In Scrum development, a cross-functional team (product owner, scrum master, developers) works from a prioritized product backlog through a repeatable lifecycle of planning, daily stand-ups, development, review, and retrospective. This article covers how Scrum works, its benefits, challenges, real-world case studies of Scrum success, and tips to help your Scrum team thrive. Whether you're a developer, project manager, or business stakeholder, you'll learn how Scrum can improve project outcomes and what it takes to implement it successfully. What is Scrum Development in Agile? Scrum is a lightweight Agile development framework focused on iterative progress, team collaboration, and continuous improvement. In Agile terms, Scrum provides a structured way to apply the Agile philosophy (which values adaptability and customer feedback) into practice. Agile is the broad mindset of iterative, value-driven development, while Scrum is a specific methodology (framework) within Agile that defines how teams should organize work in short cycles and continuously refine their approach. Think of Agile as the guiding principles and Scrum as one popular way to execute those principles. Scrum was co-created by Jeff Sutherland and Ken Schwaber in the 1990s as a response to traditional waterfall project failures. It gets its name from rugby – much like a rugby scrum brings players together to strategize, Scrum in development brings team members together to tackle problems. A Scrum team self-organizes and rapidly iterates on product versions, learning and adjusting as they go. This empirical approach (learning by doing and reflecting) is at Scrum's core. The Scrum Guide (by Schwaber and Sutherland) formalizes the framework: it defines specific roles, events, and artifacts that Scrum teams use to manage work. The framework itself is simple and intentionally semi-prescriptive – it provides just enough structure (roles, rules, and events) to remove ambiguity in the development process, while leaving flexibility for teams to adapt. How Scrum F

Design 51% of respondents said their company’s existing culture/structure made it difficult to adopt Agile and Scrum. Rigid hierarchies or command-and-control mindsets clash with Scrum’s collaborative, self-organizing nature. For Scrum to take root, organizations often need to shift to a more trust-based, learning-oriented culture. 2. Transitioning from Waterfall 44% struggled with moving from a traditional Waterfall approach to Scrum. Old habits like expecting fixed scope, or teams being organized in silos (analysis, dev, test separately) can impede Scrum, which demands cross-functional teams and flexibility. The inertia of “how we’ve always done it” can be hard to overcome. 3. Undefined Success Metrics 41% noted a lack of clear metrics to identify and measure success in Scrum. Unlike Waterfall which might use milestone completion, Scrum teams and management may be unsure how to measure progress (beyond velocity) or impact. Establishing meaningful KPIs (like customer satisfaction, cycle time, defect rates) is crucial to demonstrate Scrum’s value. 4. Alignment with Other Teams/Projects 40% found it challenging to align Scrum teams with other projects in a portfolio. If the rest of the organization isn’t agile, Scrum teams can face external dependencies or conflicting timelines. Coordination mechanisms (like Scrum of Scrums or scaled agile frameworks) are needed when multiple teams must collaborate. 5. Lack of Trust in Teams 38% cited lack of trust as an issue. Management or customers may not yet trust the team’s process or estimates, possibly leading to micromanagement or pressure that disrupts Scrum. Conversely, team members need to trust each other to speak openly (e.g. in retrospectives) and to take ownership. 6. Low Enthusiasm or Buy-In 34% pointed to lack of enthusiasm from team members or Product Owners. If the team is just going through the motions without believing in Scrum, the events can become perfunctory. Or a Product Owner who isn’t fully engaged can derail the process by not providing timely input or decisions. Genuine buy-in at all levels is needed for Scrum to flourish. Read: SDLC(Software Development Lifecycle) Success Factors and Best Practices for Scrum Teams There are several success factors and best practices that can dramatically improve a Scrum team’s effectiveness and outcomes: Executive Support and Agile Mindset Clear Roles and Training Cross-Functional, Empowered Team Focus on Customer Value Effective Scrum Ceremonies Sustainable Pace and Quality Practices Transparency and Communication Continuous Improvement and Learning 1. Executive Support and Agile Mindset For Scrum teams to reach their potential, management must not only permit Agile ways of working but actively support them. This means trusting teams to self-organize, providing necessary resources (e.g. training, tools), and removing systemic impediments. When leadership champions Agile values and gives teams psychological safety to experiment and sometimes fail, Scrum can truly thrive. An organization aligned on Agile mindset will integrate Scrum teams into its strategy rather than treating them as isolated experiments. 2. Clear Roles and Training Ensure that everyone understands their role and responsibilities in Scrum. A well-trained Product Owner who can write effective user stories and prioritize decisively, and a skilled Scrum Master who can facilitate and coach, make a huge difference. Investing in certification or workshops can be worthwhile. It’s also a best practice to educate stakeholders on how Scrum works so that, for example, clients know to expect demos every sprint and provide feedback, rather than waiting until the end. Clarity on roles prevents confusion and overlap, and allows the team to work in unison. 3. Cross-Functional, Empowered Team Scrum teams should have all the skills needed to deliver increments (dev, QA, design, etc.) and team members should collaborate rather than work in silos. Encourage teamwork by co-locating team members (or using good virtual collaboration tools for distributed teams), and by fostering respect for each person’s contributions. Empower the team to make technical decisions and to estimate their own work – this autonomy drives accountability and creativity. Keeping the team size manageable (5-9 people) also helps maintain effective communication. 4. Focus on Customer Value The Product Owner and team should keep the focus on delivering customer-centric value in each sprint. Techniques like defining clear acceptance criteria for user stories, and regularly re-prioritizing the backlog based on user feedback or market changes, ensure the team is always working on the most impactful things. A best practice is to frequently remind the team of the “why” behind each backlog item (the user context) so that everyone is aligned with the end-user’s needs. 5. Effective Scrum Ceremonies Make the Scrum events productive and engaging. For example, in Sprint Planning, make sure the team discusses each selected story enough to understand it and feels confident it can be done. In Daily Scrums, keep it time-boxed and focused on coordination (not status reporting to a manager). For Sprint Reviews, allow key stakeholders to attend and give feedback; this makes stakeholders feel involved and gives the team valuable perspectives. And crucially, hold Retrospectives consistently and act on their outcomes. The retrospective is where the team identifies process improvements – perhaps they need a better code review practice or to change how they estimate. Implementing improvement ideas in the next sprint shows the team that the retrospective matters, leading to continuous improvement (remember the earlier quote about retrospectives being the linchpin of Agile success). 6. Sustainable Pace and Quality Practices Scrum teams perform best when maintaining a sustainable work pace. Burning out the team with overtime is counterproductive; instead, velocity should stabilize over time at a pace the team can comfortably sustain. Encourage technical excellence by integrating good engineering practices: automated testing, code reviews, continuous integration, and so on. “Definition of Done” should be clearly defined (e.g. code completed and tested, integrated, documented) to avoid quality shortcuts. When quality is baked in, the increments truly become potentially shippable, and the team doesn’t accumulate crippling technical debt. 7. Transparency and Communication Encourage an environment where team members freely communicate issues and ideas. If something is blocking progress, the Scrum Master should find out quickly (often via the daily scrum) and seek to resolve it. Make use of information radiators (task boards, burndown charts visible to all) to radiate progress. High transparency builds trust with stakeholders as well – they see the team’s progress and challenges in real-time. As one Scrum principle states, “Scrum’s roles, artifacts, events, and rules exist to create minimal disruption and maximum transparency” – this transparency enables empiricism, where decisions are based on observed reality. 8. Continuous Improvement and Learning The best Scrum teams are always learning – about their product domain, about better ways to collaborate, and about improving their Agile techniques. They may do brief knowledge-sharing sessions, attend Agile community meetups, or rotate roles to build empathy (e.g. a developer pairs with the PO to learn more about stakeholder management). Some teams use metrics from their process (like analyzing why a sprint goal wasn’t met) not to blame, but to learn and adapt. Over time, this relentless improvement mindset makes the team highly adaptable and resilient. They view challenges as things to retro on and solve, not reasons to abandon Scrum. Conclusion Scrum development has proven itself as a powerful approach for delivering complex projects with agility and quality. By breaking work into sprints, empowering cross-functional teams, and embracing continuous feedback, Scrum helps organizations innovate faster and respond to change more effectively than traditional methods. The journey to Scrum maturity isn’t without hurdles – teams must foster a new mindset, align stakeholders, and learn through retrospection – but the payoff is a more resilient and productive development process. Whether you're new to Scrum or looking to optimize your Agile process, Decipher Zone offers specialized Agile product development services tailored to your unique needs. Talk to an Agile Expert at Decipher Zone → Remember, at the heart of Scrum’s success are people: engaged team members, supportive leadership, and satisfied customers working together in a transparent, adaptive cycle. Read: Trends, Methodologies, and Outsourcing Strategies for 2025 If you’re looking to improve your team’s agility or considering Scrum for your next project, keep the best practices and lessons from this guide in mind. Start with the basics (roles, events, and artifacts), be patient through the learning curve, and never skip the retrospective! With time, your team can iterate into a well-oiled Scrum machine that delivers “twice the work in half the time.” Finally, success with Scrum often comes easier with the right expertise by your side. Decipher Zone has extensive experience in Agile product development and building dedicated Scrum teams that hit the ground running. Whether you’re initiating a new Scrum team or scaling Agile across programs, our experts can guide you through the process and tailor Scrum to your unique needs. Ready to unleash the potential of Scrum in your organization? Embrace the Agile mindset, and don’t hesitate to reach out for guidance or support. By taking the plunge into Scrum development, you’re investing in a process of continuous improvement – one that can transform not just your project outcomes, but also the culture and capability of your team for the better.