

Irfan Saeed's Agile Paradigm Shift – Agile Mindset

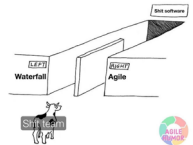


Agile is not about five Scrum ceremonies (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective, Backlog Refinement) that make up a Sprint.

Agile is not a menu of things from which you can cherry pick. This is a system. Agile teams are dedicated teams. They have a single purpose, a clear objective, a protector, a product owner who guides and shepherds the team along.

If your mindset, culture and management team still live, breath and practice SDLC, having a Scrum Master, JIRA, Confluence, User Stories Backlog, Velocity and labeling Agile will NOT make any difference unless you truly believe in following principles:

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



- Successful use of Scrum depends on people becoming more proficient in living five values: **Commitment, Focus, Openness, Respect, and Courage.**
- The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consist of on Scrum Master, one Product Owner, and Developers. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focuses on one objective at a time, the Product Goal.
- Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when and how.
- The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people.
- The Scrum Team is responsible for all product related activities from stakeholder collaboration, verification maintenance, operation, experimentation, research and development, and anything else that might be required. **They are structured and empowered by the organization to managed their own work.** Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.
- The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. Scrum defines three specific accountabilities within the Scrum Team: the Developer, the Product Owner, and the Scrum Master.

You are not Agile if you, your team and manger:

- Complains about changes and asks for change request. (violates principle #2).
- Team member who is sill idle waiting for user story or task to be assigned (violates principle # 11).

Scrum Ceremonies

- Sprint planning meeting
- Daily Scrum
- Sprint review meeting
- Sprint Retrospective

Scrum Ceremonies

- Sprint planning meeting
- Daily Scrum
- Sprint review meeting
- Sprint Retrospective

Four Values of Agile Manifesto

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

Defined Processes vs. Empirical Processes

- Empirical processes are interactive, incremental, change often, adapt, and pass through the reviews
- Industrial work relies on defined processes
- Knowledge work relies on empirical processes
- A defined process defines all steps in advance
- Empirical processes are change driven

Scrum Team (only three roles)

- Product Owner
- Scrum Master
- Development Team members

Team Characteristics

- Self-Organized
- Cross -Functional

Three Scrum Pillars (TIA)

- Transparency
- Inspection
- Adaptation

Sprint Review

- Timebox: 4 hours for 4 weeks sprints
- Attendees: Complete Scrum Team & Stakeholders
- Goal: Demo of project work and assessing feedback

Scrum Basics

- A framework for complex adaptive problems
- Lightweight
- Simple to understand
- Difficult to master

Sprint Planning Meeting

- Timebox: 8 hours (max) for 4 weeks sprints. Less for Shorter
- Attendee: Scrum Team (all)
- Goal: Capacity, Sprint Goal / Definition of Done, Sprint Backlog

Daily Scrum

- Timebox: 15 minutes
- Attendees: Complete Scrum Team
- Goal: Progress and Impediments
- What has been accomplished since last meeting? What will be done before the next meeting? What obstacle are in the way?

Sprint Retrospective

- Timebox: 3 hours for 4 weeks sprints
- Attendees: Complete Scrum Team & Stakeholders. Product Owner (Optional)
- Goal: Brainstorm and agree on what is working and what is not

Sprint Planning

Daily Scrum

Sprint Review

Sprint Retrospective

- Maximum Duration 30 Days

Activities

Pre-Sprint Activities

- Vision Statement: concise description of the goals of the project
- Product Roadmap: visual timeline of major product features to be delivered and is normally created by the Product Owner
- Stories: requirements normally written by the Product Owner and come from customer requirements
- Product Backlog: made up of stories and prioritized

Sprint Activities

- Sprint Planning meetings plan what will go into a Sprint
- The Product Owner prioritizes requirements and decides contents of the Sprint Backlog
- Stories make up Sprint Backlog
- Team breakdown stories into tasks
- Team takes 30 days or so to deliver an agreed amount of stories
- Daily Scrum of 15 minutes for team to collaborate with each other
- Sprint review team demonstrates the completed stories to customer in a Sprint Demo
- Scrum Retrospective team reviews Sprint and looks for improvement (lessons learned)
- Scrum Master makes sure the Scrum process is followed entirely and offers coaching

Scrum Events

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Sprint Details

- Timebox – 2 to 4 weeks
- Product owner can cancel the sprint
- Work ~Aprox. Capacity
- Goal: To Do, Doing, Done

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

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

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

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

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

Scrum Team

- ◆ The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.
- ◆ The Product Owner is also accountable for effective Product Backlog management, which includes:
 - ◆ Developing and explicitly communicating the Product Goal;
 - ◆ Creating and clearly communicating Product Backlog items;
 - ◆ Ordering Product Backlog items; and,
 - ◆ Ensuring that the Product Backlog is transparent, visible and understood.
- ◆ The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.
- ◆ For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.
- ◆ The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Product Owner

- ◆ The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.
- ◆ The Scrum Master is accountable for the Scrum Team's effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.
- ◆ Scrum Masters are true leaders who serve the Scrum Team and the larger organization.
- ◆ The Scrum Master serves the Scrum Team in several ways, including:
 - ◆ Coaching the team members in self-management and cross-functionality;
 - ◆ Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
 - ◆ Causing the removal of impediments to the Scrum Team's progress; and,
 - ◆ Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.
- ◆ The Scrum Master serves the Product Owner in several ways, including:
 - ◆ Helping find techniques for effective Product Goal definition and Product Backlog management;
 - ◆ Helping the Scrum Team understand the need for clear and concise Product Backlog items;
 - ◆ Helping establish empirical product planning for a complex environment; and,
 - ◆ Facilitating stakeholder collaboration as requested or needed.
- ◆ The Scrum Master serves the organization in several ways, including:
 - ◆ Leading, training, and coaching the organization in its Scrum adoption;
 - ◆ Planning and advising Scrum implementations within the organization;
 - ◆ Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
 - ◆ Removing barriers between stakeholders and Scrum Teams.

Scrum Master

- ◆ Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.
- ◆ The specific skills needed by the Developers are often broad and will vary with the domain of work. However, the Developers are always accountable for:
 - ◆ Creating a plan for the Sprint, the Sprint Backlog;
 - ◆ Instilling quality by adhering to a Definition of Done;
 - ◆ Adapting their plan each day toward the Sprint Goal; and,
 - ◆ Holding each other accountable as professionals.

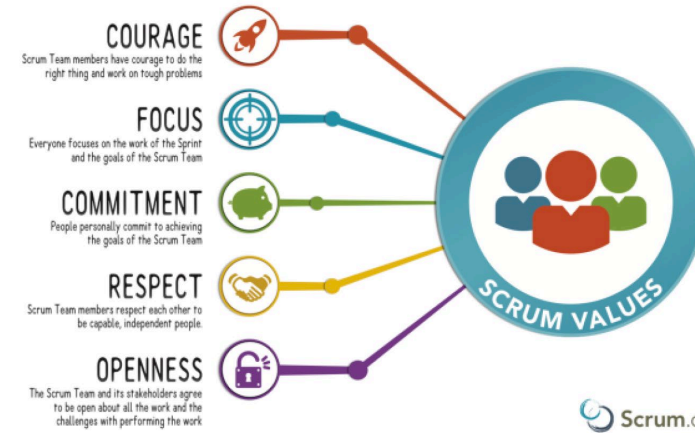
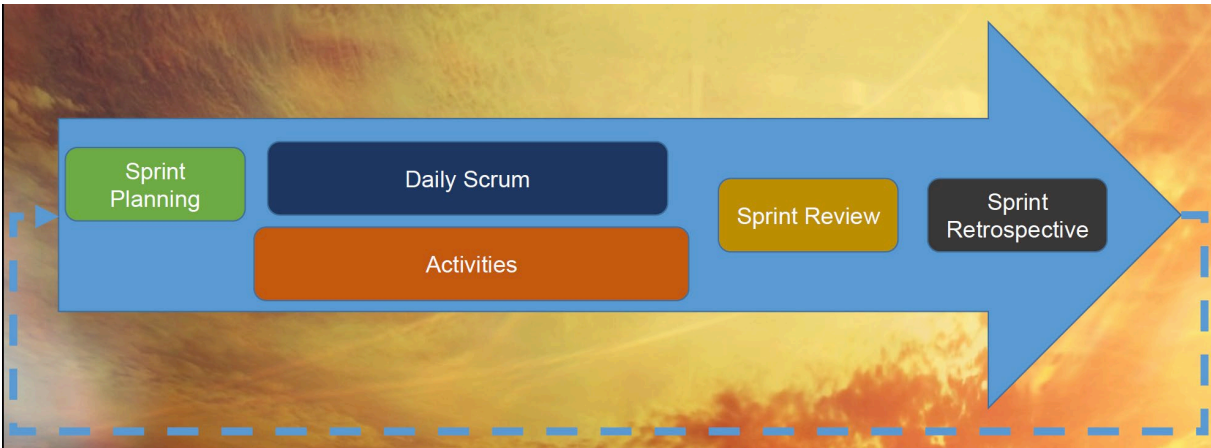
Developers

- **Burn down Chart:** Shows the amount of work which remaining in the current backlog. Time is displayed on the horizontal axis and work remaining in the backlog is shown on the vertical axis. Over time, items are pulled from the backlog, a plot line reveals work remaining. Burn down charts can be used in Sprint Backlogs and Product Backlogs.
- **Burn up Chart:** Shows the amount of work completed. Time is displayed on the horizontal axis and work remaining in the backlog is shown on the vertical axis. Over time, as the project progresses and items are pulled from the backlog, a plot line showing the completed work will rise. **Utilizing Burn Up Chart:** Track the work that has been completed. As work is done the line moves upward. Provides additional insight into the project status.
- **Daily Scrum:** Time boxed event of 15 minutes for the Development Team to discuss what they did since the last Daily Scrum, what they'll work on today, and to identify and impediments that are preventing progress.
- **Definition of Done (DoD):** A common understanding of the Scrum Team's expectations that the Increment must create to be releasable into production. Everyone must agree upon the DoD.. **Definition of "Done" Done":** The shared understanding of what it means for a piece of work to be considered complete
- **Development Team:** The developer role within a Scrum Team. These are the people completing the work within an iteration and are accountable for creating a releasable increment of product each Sprint.
- **Empiricism:** Scrum is an empirical process framework, meaning work and decisions are based on observation, experience and experimentation. Scrum Empiricism has three pillars: inspection, transparency, and adaptation.
- **Forecast of functionality:** Development Team's selection of items from the Product Backlog they deems possible to complete in a Sprint.
- **Increment:** The end result of a Sprint is a piece of working software added to previously created Increments. The sum of all project increments equate to the project's product.

- **Product Backlog:** Prioritized, ordered list of the user stories to be completed by the Development Team to create, maintain and sustain a product. The Product Owner manages and maintains the Product Backlog for a Scrum Project. An ordered list of everything (aka stories) that might be needed in the final product.
- **Product Backlog Refinement:** The Product Owner and the Development Teams add granularity, detail, and prioritization to the the stories within the Product Backlog.
- **Product Owner:** Scrum role that is accountable for maximizing the product's value by managing and expressing business and functional expectations for a product to the Development Team. This role is the Value Optimizer.
- **Scrum:** A project management framework that defines the rules and roles required in complex product development. Scrum consists of three roles: Scrum Master, Product Owner, and the Development Team.
- **Scrum Board:** A poster or collection of posters to visualize communication for and by the Scrum Team. Sometimes called an information radiator.
- **Scrum Master:** Scrum role that guides and coaches the Scrum Team and the organization to proper understanding and implementation of Scrum.
- **Scrum Team:** Self organizing team consisting of a Product Owner, Development Team and Scrum Master.
- **Self organization:** Management principle where teams organize their work. Self organization happens within the Scrum framework. Self organizations means the development team will determine how to accomplish work, rather than being directed by a project manager or management.
- **Sprint:** An iteration within Scrum. Sprints are a time boxed event typically lasting four weeks or less.
- **Sprint Backlog :** Sprint Backlog: The selected stories from the Product Backlog become the Sprint Backlog. The Sprint Goal is based on the Sprint Backlog. Selected items (stories) from the Product Backlog to be delivered through a Sprint, along with the Sprint Goal and plans for delivering the items and realizing the Sprint Goal.
- **Sprint Goal:** Defines the purpose of a Sprint, often a business problem that is addressed. It's the theme of the current sprint and what the Development Team aims to accomplish.

- **Sprint Planning:** A meeting that is time boxed to eight hours to start a Sprint. This meeting allows the Scrum Team to inspect the work from the Product Backlog that's most valuable to be done in the next Sprint. The Sprint Backlog is selected during Sprint Planning.
- **Sprint Retrospective:** A meeting held towards the end of a sprint. This meeting is a time boxed event of 3 hours and serves for the Scrum Team to inspect the past Sprint and plan for improvements to be enacted during the next Sprint. This is a learning lessons opportunity.
- **Sprint Review:** A time boxed meeting of four hours held at the end of the development work of a Sprint. The Scrum Team will demonstrate what they've created for the stakeholders. Stakeholders will inspect the Increment and assess the impact of the work performed on overall progress and update the Product backlog with any changes.
- **Stakeholder:** These are people external to the Scrum Team with a vested interest in the product that that the project is created. Stakeholders are represented by the Product Owner.
- **Technical Debt:** The overhead of maintaining the product caused by poor design decisions, code refactoring, and the total cost of ownership.
- **Velocity:** An indication of the average amount of Product Backlog the Development Team completed during an iteration.
- **Increment :** The set of all the Product Backlog items completed so far in the project (up to the end of a certain Sprint)
- **Monitoring Progress towards a Goal :** The performance measurement and forecast for the whole project
- **Monitoring Sprint Progress :** The performance measurement and forecasts for a single Sprint
- **Increment Details:** The product increment is the outcome of an iteration. The product increment is a chunk of the project work. The development team and the product owner must agree what done means for an increment. Sum of all completed Product Backlog items at the end of a Sprint. Each Increment must be "Done". Must be releasable. Product Owner may/may not release a certain Increment.

- **Definition of Done:** Shared understanding of what it means for a piece of work to be “Done”. Definition of “Done” must be discussed and agreed upon at the beginning of the project so that future Increments would be releasable. Over time, the team will improve their definition of "Done" to include more stringent criteria.
- **Definition of Done:** Multiple Scrum Teams on a single project: Might not be possible to use the same definition of “Done” for all teams, because they might be working on items of different natures. Each Scrum Team will define its own definition of “Done” and delivers its items based on that definition. Integration of definitions of “Done” should be capable of creating a potentially releasable Increment at the project level.
- **Monitoring Project Progress:** Product Owner responsible to monitor the progress of the project. Should be done at least once per Sprint Review. Product Owner determines the amount of remaining work and compares it to the remaining work of the previous Sprints. Forecasts the completion date of the project. All stakeholders should have access to this information.
- **Utilizing Burndown Chart:** Track the work that remains to be done on a project. Measures the team progress in completing the project work.
- **Understanding Team Velocity:** Velocity is the measure of a team's capacity for work per iteration. Measured in the same unit that the team estimates the work. Velocity very early and then stabilizes. Velocity tends to plateau.

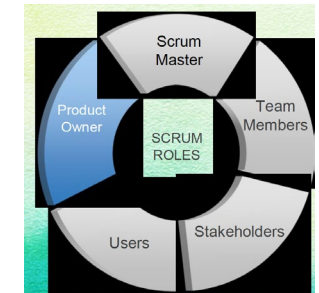
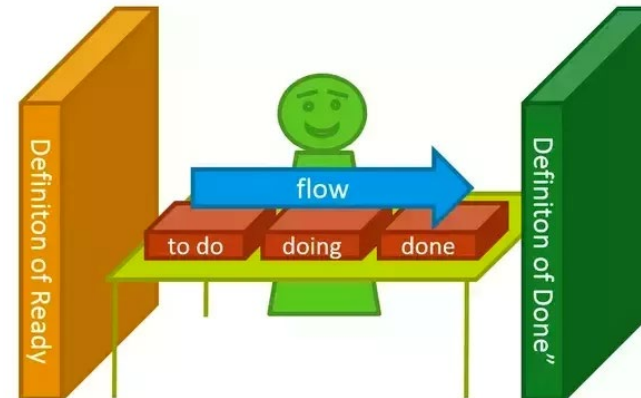
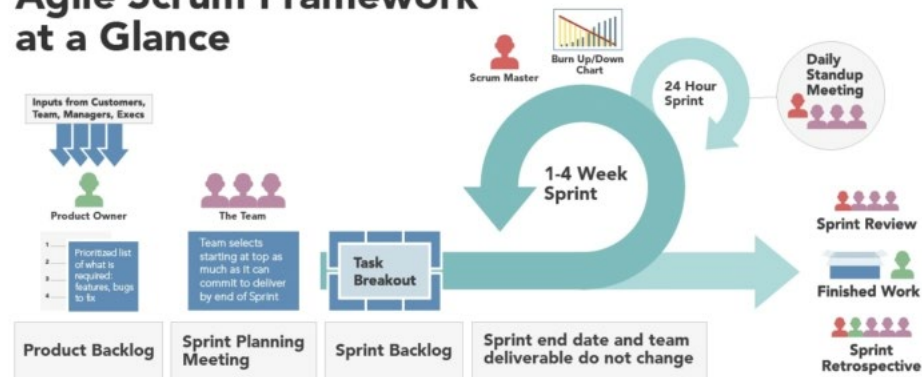


Scrum.org
© 1999-2015 Scrum.org All Rights Reserved



Story Size	Story point equivalent - Configuration	Story point equivalent - Customization
XS	1	2
S	2	3
M	3	5
L	5	8
XL	8	13
XXL	13	21
XXXL	21	34

Agile Scrum Framework at a Glance



- 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4. Business people and developers must work together daily throughout the project.
- 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 7. Working software is the primary measure of progress.
- 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 9. Continuous attention to technical excellence and good design enhances agility.
- 10. Simplicity—the art of maximizing the amount of work not done—is essential.
- 11. The best architectures, requirements, and designs emerge from self-organizing teams.
- 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

12 Principles Behind the Agile Manifesto. I do think one of the beautiful things about these principles is you need to think of them in a holistic way. You can't just cherry pick a few of them. We can get into why that can lead to bad outcomes—and some companies are doing that today, and they think they're doing agile, but they get in trouble. Agile is not a menu of things from which you can cherry pick. This is a system. Agile teams are dedicated teams. They have a single purpose, a clear objective, a protector, a product owner who guides and shepherds the team along. These guys and gals work together on the single objective.

At the core, you need to be putting the customer first. You need to be clear on who the customer is, what problem you're trying to solve, what matters to the customer, and prioritize. Always come back to who the customer is. In some cases, the customer can be the internal customer. But often, you need to make sure that it's the external customer. It's about making work fun again. Imagine that. Imagine getting your team members to enjoy what they're doing and feel like they're accomplishing their mission.

In typical organizations, the distance between the customer and the people doing the coding is eight layers of translation. That can only lead to wrong prioritization, compromise, and, in the end, your likelihood of delighting the customer and doing something that's "aha" is reduced.

That's principle number one and incredibly important.

The second principle that I would add is around how to focus on people interactions versus process. So, how do you make sure that your team members don't just take a project plan on what we need to do and toss it over the wall to [another] team, but actually collaborate?

It's scary for most organizations to let go. We have built organizations that are hierarchical, inspired from the military. Everything needs to flow up, all the way to the top, to people that have been promoted—based on past behavior and successes—to people that supposedly know more. Every time you go up and down this chain, you have translation layers, and you lose some of the nuances. Now we're saying, no, let's flip it around. We're going to let the people who are closest to the problem, closest to the customer, make the trade-off within the scope that we've agreed is the scope that they can operate in. That's what makes it agile. That's what makes it speedy. That's what makes it flexible.

I also see many organizations, where they'll get the input that what they're building is not right, and they continue to invest in it just because they've already invested x amount of dollars. They just feel like, "I need to bring it to the end. I need to bring it to the finish line even if it's not valuable." If it's not working, kill it, sunset it. Focus on the right things that are relevant for your business.

At the end of the sprint, it's kind of a belly flop. It's nothing special. That is a big problem that we're seeing more and more in companies. If you don't take a system view, and you don't think about all these components together, you're not going to get the expected outcome.

<https://kanbanize.com/agile/project-management/principles>

<https://kanbanize.com/agile/project-management/workflow>

<https://kanbanize.com/agile/project-management/planning>

<https://kanbanize.com/agile/project-management/agile-implementation>

<https://kanbanize.com/agile/project-management/agile-metrics>

<https://kanbanize.com/agile/project-management/tasks-epics-initiatives-themes>

<https://kanbanize.com/agile/project-management/agile-retrospective>

<https://kanbanize.com/agile/project-management/epics>

<https://kanbanize.com/agile/project-management/user-stories>

<https://kanbanize.com/agile/project-management/estimation>

<https://kanbanize.com/agile/project-management/reporting>

<https://kanbanize.com/agile/project-management/ceremonies>

<https://kanbanize.com/agile/project-management/benefits-of-agile>

<https://kanbanize.com/agile/project-management/agile-best-practices>

<https://kanbanize.com/agile/project-management/tools>

<https://kanbanize.com/agile/project-management/agile-vs-waterfall>