This is new text. Line 2 added

The Agile methodology is a project management approach that focuses on incremental and iterative steps to completing projects. This involves short-term development cycles which allows quick delivery, continuous collaboration, adapting and responding to change, early and continuous feedback.

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

Agile projects have short iterations enabling the project team to receive early and continuous feedback. When sequential development approaches are used, the customer often does not see the product until the project is nearly completed. At that point, it is often too late for the development team to effectively address any issues the customer may have. By getting frequent customer feedback as the project progresses, Agile teams can incorporate most of the changes into the development process. Early and frequent feedback helps the team focus on the features with the highest business value, or associated risk.

Agile benefits

* Ability to respond to change
* Delivery speed/time to market
* Increased project visibility
* Improved alignment towards changing Project priorities
* Project risk reduction

Agile Disadvantages

* Lesser documentation and a higher focus on codes.
* Teams may use inconsistent practises
* Frequent changes may have an impact on estimation of timelines, outcomes, and budgets of projects, scope outlining.
* difficult for new joiners to adapt to the fast workplace environment

**Scrum**

Scrum is an agile project management framework that helps teams structure and manage their work through a defined set of Scrum practices & processes.

**Scrum Roles/Team – or Accountabilities (as defined in Scrum guide)**

* Product Owner –
* Brings business to the team.
* Builds, manages, prioritizes/orders the product backlog based on business value.
* Ensuring that the Product Backlog is transparent, visible and understood.
* Bridge the gap between what the business wants and what the team understands.
* Helps the rest of the Scrum Team gain clarity about what would be “valuable.”
* Make decisions to maximize ROI
* Responsible for maximizing the product's value
* Optimizing the value of the work the Scrum team performs
* Responsible for monitoring the progress of release and monitoring the remaining work towards project goal
* PO works with the Scrum team and creates the Product goal, PO is accountable for the Product goal.
* Product goal can be changed based on the discretion of the PO.
* Sprint can be cancelled by the PO if the sprint goal becomes obsolete.
* Scrum Master –
* Facilitate Sprint events and team meetings *if required.*
* Accountable for the Scrum Team’s effectiveness and establishing Scrum as defined in the Scrum guide
* Coach teams, Product Owners, and the business to improve its Scrum processes and optimize delivery
* Helps those outside the Scrum Team understand how to interact with them.
* 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
* Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.
* Developers –
* Anyone in the team that is delivering work – designers/developers/testers etc.
* Development team is self-managing (self-organizing removed in 2020) and cross-functional

The Scrum Guide outlines the accountabilities of the Developers as:

* 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
* Holding each other accountable as professionals.

Role of Management: (Scrum Guide does not mention anything around Management)

* Help the Scrum Master with impediments and cause organizational changes that increase the productivity of the Scrum Team.
* Provide the Product Owner with business insights and information.
* set a vision and a direction for the business.

In order for a Scrum Team to be successful, they should be:

* **Cohesive** - They should work together as a tight-knit collaborative team, supportive of one another and fostering trust among team members
* **Focused** - Scrum Teams are focused on creating value and achieving the Product Goal
* **Cross-functional** - The team consists of individuals with the diverse set of skills and experience needed to accomplish their goals, members have all the skills necessary to create value each Sprint.
* **Self-managing** - They are empowered by the organization to determine how to do their work without being directed, they can internally decide who does what, when, and how.
* **Non-hierarchical** - They act as one team, with no sub-teams or organizational hierarchy
* **Accountable** - The entire team is accountable for the success in creating a valuable, useful Increment every Sprint

**Scrum Ceremonies**

* Sprint –
* Container of all Scrum events
* Scrum divides a project into iterations (called sprints), maximum of 4 weeks
* Sprint Planning –
* Marks the start of the sprint
* Requires everyone in the Scrum Team - Often the Scrum Master acts as the facilitator for this event, though this is not required. Facilitation of Sprint Planning can be done by anyone at the event.
* In Sprint Planning, the team plans the work for the Sprint, defines why is the sprint needed (Sprint goal) identifies what can be done, and how the chosen work will be completed.
* Specific user stories are added from the PBI to the Sprint Backlog by the Developers.
* Sprint Goal is set by the entire Scrum team
* Developers break down user stories into tasks or subtasks as needed
* Should not exceed 8 hrs for a 1 month sprint, shorter for shorter sprints
* Covers 3 topics: WHY – defining the sprint goal, WHAT – what needs to be done, HOW – how the work will be done(breaking the stories into tasks)
* Daily Scrum –
* short meeting in which team members check in and plan for the day.
* check progress toward the sprint goal, adapt sprint backlog as necessary
* call out issues or blockers
* only developers are required to be present as per the scrum guide, PO and SM can attend if actively working as developers.
* SM needs to ensure that the Devs have this meeting and keep it timeboxed (15 mins)
* Developers are responsible for conducting daily Scrum.
* Scrum Guide 2020 does not require the Devs to stick to the 3 question format, they can choose any format as long as the format focusses towards sprint goal.
* Sprint Review –
* Occurs at the end of the sprint, requires everyone in the Scrum Team and also the stakeholders.
* Session to review the work completed and showcase it to stakeholders, gain feedback.
* As per scrum guide: it’s an event to inspect the increment and adapt PB immediately
* Discuss the progress toward the Product Goal
* Collaborate on what to do next, future adaptations. Consider what the next most valuable thing to do is and update the Product Backlog, identify possible items that can be selected in the next sprint.
* Should not exceed 4 hrs for a 1-month sprint
* Sprint Retrospective –
* Occurs at the end of the sprint, requires everyone in the Scrum Team.
* Team discusses about what went well and what needs to be improved for the next sprint, how improvements can be incorporated.
* Retrospectives cover topics such as the process, people, organizations, and tools.
* Root cause analysis
* Documenting lessons learnt
* Should not exceed 3 hrs for a 1-month sprint
* Scrum Guide: Purpose of Retro is to plan ways to increase quality and effectiveness.
* Product Backlog Refinement –
* Not an official scrum event.
* Refinement is a continuous activity, usually done once per sprint or per week
* We go through each story in the backlog, defining scope of the story and identifying dependencies, adding details, estimates (story pointing), and priority to items in the product backlog.
* Story Pointed user stories are then moved to Sprint Backlog.
* From test perspective, we define testing activities, estimate testing effort for all testing tasks
* Major objective of Refinement is after the session, the scope, dependencies should be clearly defined so that the item can be picked up in the sprint or the items are Ready for sprint planning.
* READY- means stories are clear to the team, team has the required info to deliver it, small enough to be completed in a sprint based on team’s Definition of Done.

PBI's should be decomposed to make them small enough to be Done during a Sprint. Product Backlog refinement is an ongoing activity where the Scrum Team collaborates to add detail, estimates, and order the Product Backlog items. One of the objectives of refinement is to break down Product Backlog items into smaller, more manageable pieces to ensure they can be completed within a single Sprint. This helps the Developers to have a clear understanding of the work and facilitates transparency, adaptability, and effective Sprint Planning.

Acceptance Criteria – defined by PO

**Scrum Artefacts**

* Product Backlog –
* The team’s “To Do” list.
* Primary list of work that needs to be done and includes the product goal.
* Contains list of features, requirements, enhancements, and fixes
* Acts as the input for the sprint backlog.
* Owned, managed and prioritized by Product Owner, can be done by Devs but PO remains accountable
* Product Goal: describes a future state of the product which can serve as the target for the Scrum teams to plan against.
* Sprint Backlog –
* The Sprint Backlog is a plan created by and for the Developers.
* List of items, user stories, or changes selected for implementation in the current sprint cycle.
* The Sprint Backlog is created by the Developers during the Sprint Planning Event.
* Sprint backlog includes the WHY – Sprint goal, Which – list of ‘Ready’ items from Product backlog, and HOW – how the work will be completed based on tasks and sub tasks.
* Shows the plan for the current sprint.
* Before each Sprint, the team chooses which items it will work on from the Product Backlog.
* There should be only one Sprint Backlog; there should be no separate backlog for enhancements, defects or other tasks
* The Sprint Backlog can have one or two Sprint Retrospective improvement items that the team would like to work on during the Sprint
* Sprint backlog can change during sprint (details added as more info is gained) but Sprint goal should not change. It evolves as new information emerges or when there are changes in the team's understanding of the work required.
* Increment –
* The Increment is the latest version of the product that conforms to the Definition of Done.
* If a developer adds something to the product that makes it no longer conform to the DoD, that version of the product is not the Increment. At that point, the Increment is the last version of the product that did conform to the DoD.
* Each increment must be usable and should deliver some value, even though small.
* Definition of Done –
* The Definition of Done describes the quality standards for the Increment to be considered “Done” and in a usable state for it to be effectively inspected during the Sprint Review. Knowing that the Increment conforms to the DOD helps stakeholders understand the completeness of the work they are reviewing.
* Definition of Done: All of the characteristics and standards an Increment needs to meet in order to be marked as complete and released. The Scrum Guide says the Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product. Once the Definition of Done is met, the Increment is Done and can be delivered.
* Increases transparency.
* Creates a common understanding of what completed work means.
* Guides the Developers in knowing how many Product Backlog Items they can select for the Sprint Backlog.

DOR (Definition of Ready): INVEST: **I**ndependent **N**egotiable **V**aluable **E**stimable **S**mall **T**estable

DOR term does not exist in Scrum guide

* Sprint burndown chart –
* Sprint burndowns are a graphical way of showing how much work is remaining in the sprint
* Not an official scrum artifact

Multi Scrum:

* Scrum team does not need to have same start or end date
* Teams should be able to produce an integrated increment which can be discussed during Sprint review.
* Should have 1 Product backlog
* Should have same DOD, they must mutually define and comply with same DOD
* Should have 1 PO

Scrum is based on empiricism, self-management, and continual improvement. The competency includes the focus areas of:

**Empiricism**, **Scrum** **Values**, **Scrum** **Team**, **Events**, **Artifacts**, **Done**, and **Scaling**.

Empiricism: Empiricism is the philosophy that all knowledge originates in experience and observations. It’s a cornerstone of the scientific method and underlies much of modern science and medicine. In the context of Scrum, empiricism refers to the idea that solving complex problems, or doing complex work, can only be done using an exploratory process rather than relying on predetermined plans.

Lean thinking reduces waste and focuses on the essentials. As you will see Scrum removes waste (impediments) and prioritises value (essentials).

Scrum utilizes three pillars to support this approach: transparency, inspection and adaptation.

* the team must be **transparent** about their progress and the product being developed
* they must **inspect** this progress and the product regularly
* and they must be able to **adapt** their work and processes based on what they observe

Scrum Values: ***Focus, Openness, Respect, Commitment and Courage***

A Spike (not in official Scrum guide) is a time-boxed activity that allows the Scrum Team to investigate and learn more about a specific area of work. It can be used to reduce uncertainty and gain knowledge before starting the development of a Product Backlog Item (PBI). Spikes are often used when the team needs to explore a new technology, research a solution, or conduct experiments to gather information. The outcome of a Spike is the knowledge gained, which can then be used to make informed decisions and estimate the effort required for the PBI.

Product Backlog management includes:

* Clearly expressing Product Backlog items;
* Ordering the items in the Product Backlog to best achieve goals and missions;
* Optimizing the value of the work the Development Team performs;
* Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
* Ensuring the Development Team understands items in the Product Backlog to the level needed.

Technical debt is essentially the trade-off made when choosing a quick, potentially suboptimal solution now instead of applying a better approach that would take longer. This can result in a less transparent view of the product's current state, as hidden issues or poor design may not be immediately apparent. Additionally, technical debt can hinder a team's productivity over time because resources may have to be diverted to address the suboptimal solutions or design choices made earlier. There's no fixed timeframe for addressing technical debt in Scrum—it's up to the team to decide when and how to tackle it, considering their goals and standards for the product. Lastly, contrary to enhancing the development process, technical debt often introduces uncertainty and can slow things down.

Velocity refers to the amount of Product Backlog items that a Developers can complete within a Sprint. It represents the work that is turned into a potentially releasable Increment during the Sprint.