**Agile Manifesto - 01**

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

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

By this, we are uncovering better ways of developing software by doing it and helping others do it, and this means learning.

Learning: as an organization (or team) on how to get better at developing software

* **Individuals and interactions over processes and tools**
  + As part of valuing individuals and interactions: we will value Face to Face communication
  + In estimation work, we will value interaction and conversations more than the actual result of what the estimate is.
* **Working software over comprehensive documentation**
* Create shared understanding by having more impactful conversations.
* Create learning and shared understanding via working software.
* Let’s not go crazy on documenting everything.
* **Customer collaboration over contract negotiation**
* We want to collaborate and delight our customers.
* However, we must change the way we write the contracts. When we write out contracts, let’s keep our customer’s success in mind.
* **Responding to change over following a plan.**
* Let’s value being ready for changes and responding to any change.

**Agile Principles – 02**

*Principles are more descriptive guideline, to help us make decisions through our agile activities.*

1. **Deliver Value Faster -** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

* Faster or early on and continuously

1. **Welcome Change -** Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

* Delighting customers and really responding to their need

1. **Deliver Working Software Frequently -** Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

* Working software frequently.

1. **Work Together Daily -** Businesspeople and developers must work together daily throughout the project.

* Collaboration.

1. **Build Projects Around Motivated Individuals -** Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.

* Solve more and give motivational space.

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

* Communication.

1. **Working Software is Key** - Working software is the primary measure of progress.

* Working software is key, what have you done, and not how much.

1. **Sustainable Development** - Agile processes promote sustainable development. Thee sponsors, developers, and users should be able to maintain a constant pace indefinitely.

* Think about a sustainable pace that team can develop in.

1. **Attention to Technical Excellence -** Continuous attention to technical excellence and good design enhances agility.

* Quality.

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

* By building less and delivering more.

1. **Self-Organizing Teams -** The best architectures, requirements, and designs emerge from self-organizing teams.

* Skills be a team.

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

* Reflect and adjust.

**Scrum Framework – 02**

**Overview**

*Lightweight framework* designed to solve complex problems that require sprinting or iterating through a solution.

* Inspection, adaption, transparency.
* Inspect and adapt.

*Sprint time-box* (time selected by the team anywhere from 1-4 weeks)

* Planning, developing, testing, and reviewing of a working unit of software is completed with this time box.
* Daily planning, standup, review and retrospective, all happen inside of the time box.

*Scrum*

* It is an agile way of developing software or any project.
* It is more of a framework than a methodology or process.
* Recommends a self-organizing and cross-functional team.
* If there is a challenge or a conflict in the team, they come together and discuss and decide on an own way to move forward as a team.
* It is unique because it has specific roles/responsibilities, ceremonies and artifacts.

**Values**

Scrum value really support team members when they are working to solve complex problems.

1. **Openness**
   * Open to living the scrum values over just doing scrum.
   * Team members are open to uncovering better ways of developing software and solving problems.
2. **Commitment**

* Commitment to the sprint goal.
* Team commits to working as a team and doing their best to bring value to the customer.
* Team commits t give their best action and effort versus to the results at the end of the sprint.

1. **Focus**

* Focus on the sprint’s worth of work.
* Focus on solving problems and removing impediments.
* Focus on the bringing value to the customers.

1. **Courage**

* Team members show up with courage working in tough problems.
* Team members show up with courage while dealing with constant change and adapting.

1. **Respect**

* Respect each other in the team and be professional.
* Respect and embrace the differences (different opinions, ways of working, solving problems culture and background).

**Rules**

Scrum value really support team members when they are working to solve complex problems.

1. **Product Owner**

* Talk to the stakeholders and users to understand their needs and vision of the product.
* Come back to the team and help the team understand the product vision.
* Also give the team some description of the product feature and what needs to be built.
* Product owner writes a simple description of the product feature from an end user’s perspective (called user stories)

1. **The Scrum Master**

* Ensures scrum is understood and used skillfully by scrum team.
* Servant team.
* Models agile mindset and scrum framework.

1. **Scrum Team**

* Build product increment.
* Dev team is cross functional. A group of professionals who are capable of deliver a potentially releasable increment at the end of the sprint.
* Self-organizing.
* Accountable as a team.
* Three to nine people.

**Role**

Scrum value really support team members when they are working to solve complex problems.

1. **Product Owner**

* Tasks.
* Priorities.
* What we need to have done.
* Functionality that is needed.
* MVP (Minimum Viable Product).

1. **The Scrum Master**

* What did you do yesterday?
* What are you going to focus on today?
* Do you have impediments?
* What went well?
* What did go well?
* What do you want to improve?

1. **Scrum Team**

* Developer.
* Business analyst.

**Sprint / Scrum Ceremonies – 03**

* **Daily Stand-up Meeting**
* This is a quick 15-minute meeting to do inspect and adapt, which is the core oof Scrum Framework where everybody comes in and answer the principles Wh-questions.
* This is the meeting to inspect how the team is doing towards achieving the goal for the sprint and identifying and taking an action item for resolving any impediment or the roadblock.
* **Sprint Planning Meeting**
* We are planning on what goes in used two weeks where we can take in the work and we get this work done, so is shippable to the customer and the customer can use it and give us feedback.
* It is essential to create a clear concrete plan on how they are going to work and how they are going to create this potential shippable item, besides the limitations they have in this number of weeks iteration.
* **Sprint Review**
* *What is it? -*- Meeting conducted by the team to share the outcomes of the Sprint with external stakeholders.
* *What attends* – Sprint reviews are attended by the entire team, including the product owner. Ideally the customer present at the sprint review.
* *How long is it* – Sprint reviews are normally between 1 and 2 hours in length.
* **Sprint Retrospective**
* *What went well this sprint?*
* *What did not go well?*
* *What should be improve?*
* **Backlog Grooming**
* Problems to solve.
* Improve.
* Getting ready for next spring.

**Users – 04**

* **User Story**
* Simple description of a product feature.
* User stories are great invitations for conversations.
* That is written from an end user’s point of view.
* *The 3 C’s:*  Card (Who, What and Why), Conversation (discussing information), Confirmation (confirming what is understood).
* Who is the user?
* What kind of customers are they? What will we empower our customers to do? What is their goal?
* What kind of value will it bring? Why?
* **Acceptance Criteria**
* They are simple notes or conditions added to the user story.
* That tells what the user story must do to satisfy the needs of the customer – Defined by the product owner.
* Confirmation phase of the 3 C’s.
* Acceptance Criteria enriches the user story by making it testable and ensures the story is ready to for demo.
* Product Owner writes the initial Acceptance Criteria and brings them to discuss with the team.
* It should be testable with pass/fail results.
* It should be clear and concise.
* It should be created with shared understanding.
* *How many acceptance criteria per user story? -*  it depends on the team… recommended 3-7 per user story.
* **Writing Great User’s Stories**
* *INVEST:* independent, negotiable, valuable, estimable, small, testable.
* *FORMAT:* As an (user), I want (goal) so that I can (value or why).
* **Product Backlog**
* It is a prioritized list of user stories outlining the need of the business.
* It is expected that it contains will continue to grow as the wants of the business typically.
* The sprint backlog is created from the product backlog in the sprint planning ceremony.
* **Sprint Backlog**
* It contains the user stories the team is committing to complete in the upcoming sprint.
* **Working Agreements**
* It is a list of rules, expectations, and procedures that govern how a team will work together.
* It most have buy-in from each team member.
* It should be visible and accessible.
* It is a living document, meaning it, should be updated, and changed as needs change.
* **Definition of Ready**
* It means the user story contains enough details to be worked.
* It has completed the 3 C’s.
* It should be clear, concise, and actionable.
* It is a mutually agreed upon set of conditions a user story must meet for development to begin.
* **Definition of Done**
* It can help teams increase their productivity.
* Agreement document that team creates.
* This document is accessible by everybody in the team.
* It is a common understanding of how to work with each other.
* If it is done, then it has acceptance criteria, also, code review is done by another developer in the team, a small (relevant) note on that user story on how it was implemented, and story is given to product owner for final review.
* Definition of done can be its own document, but it generally found as part of the working agreement.
* Checklist items may vary depending on the project we are working on.
* Entirely finished to go and live.
* **Product Increment**
* A potentially shippable, vertical slice of a solution that is created as part of a sprint time-box.
* Traditional methodology: analysis, requirements, design, develop/build, test, release. *HORIZONTAL SLICED*
* Agile methodology: analysis, requirements, design, develop/build, test, release. *VERTICAL SLICED WITH PRODUCT INCREMENT*

**Estimating – 05**

* **Concept**
* *Roughly calculating –* How long is going to take us to complete a particular user story and how long it is going to take us get from beginning to done.
* *Elicited information* – Their needs and what they want.
* *Past Experiences* – Try, fail, success.
* *Standard Documentation* – For estimating.
* **Story Point Estimate**
* Numeric value representing the effort needed to complete a user story assigned by the development (Scrum) team*.*
* **Non-Agile projects**
* Absolute estimate.
* Measured in days.
* Updated usually cause major impacts.
* **Agile projects**
* Relative estimate.
* Placed into a bucket.
* Updated usually cause minor impacts.
* **Why Estimate**
* Provides estimated duration.
* Drives out clarification questions.
* Highlights complex and high-risk tasks (that were previously undefined or accounted for).
* **Estimation Techniques**
* *BUS -* Big/Uncertain/Small.
* T-Shirt sizes.
* Fibonacci sequence

***BUS -* Big/Uncertain/Small.**

* *Categorized into groupings:*

Big, uncertain, small.

* *Each user story is compared to others and assigned to a group:*

**“**Big” stories should be broken up if possible.

“Uncertain” stories need to be groomed or broken up.

“Small” some want to keep their estimation concepts small in their projects.

***T-*Shirt Sizes**

* *Categorized into groupings:*

XS, S, M, L, XL.

* *Each user story is compared to others and assigned to a t-shirt size:*

**“**Larger” stories should be broken up if possible.

**Fibonacci Sequence**

* *Abstract value to represent size:*

1, 2, 3, 5, 8, 13, 21... (add together the two previous numbers to get the next one).

* *Each user story is compared to others and assigned “Story Points”:*

**“**Larger” stories should be broken up if possible.

* *Number of points per sprint is the team’s velocity.*

**Agile Roles – 06**

* **The Role Triangle**

**Product Owner**

* *Brings busines vision.*

Including what and why.

**Development Team**

* *Defines technical vision.*

Self-organizing.

Decides how.

**Scrum Master**

* *Servant leader.*

Facilitates scrum events.

Removes impediments.

* **Subject Matter Expert**
* Senior User.
* Subject Matter Expert.
* They understand the technology and processes.
* Provides their need and the need of the other users.
* Participates in Sprint reviews (and other feedback demos).
* Provides continuous feedback on product deliverables.
* Gains and communicates feedback from other users.
* Provides perspective and context for requirements.
* Completes requested testing in a timely manner.
* Ensures user story results are used properly.
* **Business Users**
* Newer to the team.
* Provides their need (differentiates wants and needs).
* Provide perspective and context for requirements.
* Attends feedback demos they are invited to.
* Attends, watches, or reads training on deliverables.
* Utilizes deliverables and gives feedback.
* Completes requested testing in a timely manner.
* **Project Sponsor**
* The ones that say if the project will provide value or solution to the business.
* Approves project and budget.
* Provides the big picture view.
* Helps shape the project need and outcomes.
* Ensures project stays aligned to company objective.
* Participates in Sprint Reviews (and other feedback demos).
* Recognizes the team for quality work.
* Encourages cross-team / cross-department collaboration.
* Provides perspective on product impact and use.
* **Business Leaders**
* Any type of business management or leadership roles on the business side.
* Enables access to necessary business resources.
* Provides the need of management.
* Ensures user story results are creatin value for the business.
* Escalates resolution of identified impediments (as needed).
* Participates n Sprint Reviews (and other feedback demos).
* Encourages cross-team / cross-department collaboration.
* Acknowledges team successes and provides support.
* **Technology Leaders**
* Managers of technology, IT, supervisor, or executives.
* Work with development team and scrum master.
* Hires team members with the right skillset.
* Gives up control – empowers, trust, and supports the team.
* Enables access to necessary technology resources.
* Escalates resolution to identified impediments (as needed).
* Participates in Sprint reviews (and other feedback demos).
* Encourages cross-team / cross-department collaboration.
* Helps team successes and resolves team conflicts.
* Ensures solution is aligned with organizational standards.
* **Agile Detractors – Leadership ☹**
* Demands mid-sprint changes.
* Hijacks ceremonies to discuss “high-priority” topics.
* Not available for demo. Then request changes at the Sprint Review.
* Discourages communication with other teams.
* Removes or changes available resources mid-project.
* Does not Value the agile mindset. People over processes and communication over documentation.
* **Agile Coach**
* He / She is an experienced Agile practitioner.
* Trains, guides, and supports.
* They understand the ceremonies, the ways, the roles, and responsibilities.

**Tools – 07**

* **Team Velocity**
* It is how many points of stories can the team complete in each Sprint.
* **Burndown Chart**
* It is a visual representation of work being completed, throughout the various Sprints (story points in sprints, total and completed).
* Slightly behind.
* Ahead of schedule.
* Behind schedule.
* Back on track.
* **Burn-up Chart**
* It is a visual representation to show that work was added there and that was not that the team did not get done with the work (story points in sprints, total and completed).
* Work removed.
* Work added.
* Adjusted estimates.
* Work added.
* On track.

**Kanban – 08**

Provides early and consistent value and collaborative and self-managing teams.

* **Basic Principles**
* Start with what you do know.
* Agree to purse incremental change.
* Respect current process, roles, responsibilities, titles.
* Encourage acts of leadership at all levels.
* **Cards**
* To Do.
* Doing.
* Done.
* **General Practices**
* Visualize the workflow.
* Limit work in progress (WIP).
* Manage flow.
* Make policies explicit.
* Implement feedback loops.
* Collaborate for improvement, evolve experimentally.
* **Key Metrics**
* Lead time: Full time. From the time that task was created and put into that to do column. It goes from task create, to start work and task live.
* Cycle time: Not full time. From the time that task was put into that doing column until the time it was completes. It goes from start work to task live.
* **Digital Boards**
* Trello.
* JIRA.
* **Swimlanes**
* Extremely priority
* Medium priority.
* Low priority.
* **Steps to Get Started**
* Visualize your work.
* Limit work in progress (WIP)
* Adapt, monitor, and improve.

**Scrumban – 09**

* **Overview**
* Visualize workflow with a Kanban board.
* Utilizes daily scrum.
* Work is pulled, rather than assigned.
* Strict work in progress (WIP) limits.
* Project team roles not clearly specified.
* Specialized team members.
* **Sprint**
* To timebox or not to timebox, that is the question. So, it could be continuous flow or timebox.
* **Continuous Flow**
* Fluid movement of work – no timeboxes.
* Pull items into *Doing* based on team capacity.
* Plan at a certain number of *To Do* items.
* Lead and cycle time are the key metrics.
* **Timebox**
* Utilizes sprints (1-4 weeks).
* Iteration planning, reviews, and retrospectives.
* Uses *ready* status to organize work items.
* Uses lead and cycle time or burndown charts.
* **Triage and Feature Freeze**
* Used as the project deadline is approaching.
* Team identifies most important features not yet done.
* Remainder of work items are frozen.
* Fucuses the team’s energy on finishing critical tasks.